



PUSAT KARIR SMK BERBASIS TEKNOLOGI INFORMASI

Prof. Dr. Ekohariadi, M.Pd.
Rina Harimurti, S.Pd., M.T.
Yeni Anistyasari, S.Pd., M.Kom.
Asmunin, S.Kom., M.Kom.

SMK
BISA-HEBAT
SIAP KERJA • SANTUN • MANDIRI • KREATIF





PEDOMAN TEKNIS

PUSAT KARIR SMK 4.0 BERBASIS TEKNOLOGI INFORMASI

Oleh:

Prof. Dr. Ekohariadi, M.Pd
Rina Harimurti, S.Pd., M.T
Yeni Anistyasari, S.Kom., M.Kom
Asmunin, S.Kom., M.Kom

TIM UNIVERSITAS NEGERI SURABAYA

2020

KATA PENGANTAR

Alhamdulillahirobbil'alamin, segala puji syukur kepada Allah SWT yang telah memberikan berkah sehingga pedoman teknik berjudul Pusat Karir SMK 4.0 Berbasis Teknologi Informasi berhasil berhasil disusun. Buku Pedoman Teknis ini merupakan bagian dari penelitian NSPK sebagai kerjasama Kementerian Pendidikan dan Kebudayaan dengan Universitas Negeri Surabaya dalam rangka meningkatkan mutu lulusan.

Kajian ini didasari keadaan bahwa saat ini adalah era industri 4.0 yang ditandai dengan penggunaan kecerdasan buatan, internet of things (IoT), dan memperhatikan sisi humanis. Di sisi lain, Di bidang ekonomi, pada 2015 Indonesia bersama negara-negara ASEAN lain membentuk ASEAN Economic Community atau Masyarakat Ekonomi ASEAN (MEA) dengan empat pilar (a) a single market and production base, (b) a highly competitive economic region, (c) a region of equitable economic development, and (d) a region fully integrated into the global economy. MEA akan mengubah ASEAN menjadi wilayah dengan pergerakan barang, jasa, investasi, tenaga kerja terampil, dan aliran modal yang lebih bebas. Tenaga kerja Indonesia bersaing secara bebas dengan tenaga kerja negara ASEAN lainnya.

Pedoman Teknis ini berisi teknis pengembangan aplikasi pusat karir SMK berbasis web menggunakan teknologi *open-source*. Pedoman teknis ini terdiri dari lima bagian. Bagian satu berisi tentang revolusi industri 4.0 dan Bursa Kerja Khusus. Bagian dua berisi terknolgi pengembangan aplikasi. Bagian tiga berisi analisis kebutuhan dan desain perangkat lunak. Bagian empat berisi detail implementasi perangkat lunak, dan bagian lima berisi simpulan dan kemungkinan pengembangan perangkat lunak.

Akhir kata, semoga pedoman teknik ini bermanfaat.

Ketua Tim Penyusun

Prof. Dr. Ekohariadi, M.Pd

DAFTAR ISI

KATA PENGANTAR	iii
DAFTAR ISI	iv
DAFTAR GAMBAR.....	vii
DAFTAR TABEL.....	xii
BAB 1 BURSA KERJA KHUSUS DI ERA INDUSTRI 4.0	1
1.1 Pengantar	1
1.2 Revolusi Industri 4.0	1
1.3 Bursa Kerja Khusus di Era Industri 4.0	4
BAB 2 STUDI TERKAIT PENGEMBANGAN PERANGKAT LUNAK	8
2.1 Tahap Pengembangan Perangkat Lunak.....	8
2.2 Web Development.....	14
2.3 PHP.....	16
2.4 Laravel	20
2.5 Composer	24
2.6 UUID	28
2.7 MySQL	30
2.8 HTML dan CSS	32
2.9 Bootstrap.....	32
2.10 JavaScript	34
BAB 3 ANALISIS KEBUTUHAN DAN DESAIN SISTEM.....	36
3.1 Analisis kebutuhan.....	36
3.2 Desain Proses	39
3.2.1 Proses Cetak CV.....	39
3.2.2 Fitur Dasar Siswa	40
3.2.3 Tracer Study.....	41
3.2.4 Edit data IDUKA.....	42

3.2.5	IDUKA Tambah Lowongan Pekerjaan.....	43
3.3	Desain database	44
3.3.1	Entity Relationship Diagram	44
3.3.2	Deskripsi Struktur Tabel.....	45
3.4	Desain Antarmuka.....	77
3.4.1	Antarmuka Halaman Utama	78
3.4.2	Antarmuka Login	79
3.4.3	Dashboard aplikasi	80
3.4.4	Kelola Data	81
BAB 4	IMPLEMENTASI PUSAT KARIR.....	83
4.1	Persiapan Lingkungan Pengembangan	83
4.2	Editor dan IDE	84
4.3	Tool Database	87
4.4	Implementasi Aplikasi.....	89
4.4.1	Struktur Direktori	89
4.4.2	Laravel Routing.....	93
4.4.3	Konfigurasi Database	98
4.4.4	Proses Login atau Autentikasi	99
4.4.5	Laman Utama Aplikasi	102
4.4.6	Kelola Data Perusahaan	103
4.4.7	Kelola Data Siswa	111
4.4.8	Kelola Tracer Study.....	120
4.4.9	Kelola Lowongan Pekerjaan	130
4.4.10	Kelola Pendaftaran Perusahaan	146
4.4.11	Kelola Agenda Kegiatan	149
4.4.12	Kelola Pengguna.....	153
4.4.13	Kelola FAQ.....	158
4.4.14	Kelola Panggilan Tes.....	161
4.4.15	Kelola Profil Website	165

4.4.16	Kelola Tips Karir	168
4.4.17	Kelola Berkas BKK.....	172
4.4.18	Kelola Pendaftar Siswa/Alumni.....	175
BAB 5 LEVEL AKSES DAN ANTARMUKA GRAFIS		180
5.1	Level Akses	180
5.2	Fitur Aplikasi sesuai Level Akses	181
5.3	Hasil dan Pembahasan.....	184
BAB 6 PENUTUP		250
6.1	Kesimpulan.....	250
6.2	Saran	251
DAFTAR PUSTAKA		252

DAFTAR GAMBAR

Gambar 1.1 Sejarah Revolusi Industri.....	2
Gambar 1.2 Industri 4.0.....	3
Gambar 2.1 Waterfall Model.....	9
Gambar 2.2 Model Prototype.....	11
Gambar 2.3 RAD Model	12
Gambar 2.4 Spirall Model	13
Gambar 2.5 Agile Model	14
Gambar 2.6 Arsitektur Aplikasi Web.....	15
Gambar 2.7 Frontend dan Backend	16
Gambar 2.8 Teknologi Keahlian Pengembangan Web	17
Gambar 2.9 Website Resmi PHP	18
Gambar 2.10 Server Side Programming.....	18
Gambar 2.11 Survey Popularitas Bahasa Pemrograman Web.....	19
Gambar 2.12 Kode HTML	20
Gambar 2.13 Kode PHP didalam kode HTML.....	21
Gambar 2.14 Konsep MVC (Model View Controller).....	21
Gambar 2.15 Website Resmi Laravel	23
Gambar 2.16 Website Resmi Composer	25
Gambar 2.17 Packagist	26
Gambar 2.18 Detail Package Carbon	27
Gambar 2.19 Menggunakan Package UUID di Controller.....	29
Gambar 2.20 Generate kode UUID	30
Gambar 2.21 Website Resmi MySQL	31
Gambar 2.22 Website Resmi Bootstrap.....	33
Gambar 2.23 Grid System pada Bootstrap	33
Gambar 3.1 Use Case Pusat Karir SMK.....	38
Gambar 3.2 Alur Proses Profil dan CV Siswa	39
Gambar 3.3 Fitur Akun Siswa/Alumni	40
Gambar 3.4 Alur Siswa/Alumni Mengisi Tracer Study.....	41
Gambar 3.5 Edit data IDUKA	42
Gambar 3.6 Alur Tambah Lowongan Pekerjaan	43
Gambar 3.7 ERD Pusat Karir SMK	44
Gambar 3.8 Relasi di Eloquent	45
Gambar 3.9 Lanjutan Tabel Pusat Karir SMK	46

Gambar 3.11 Tampilan Utama Pusat Karir SMK.....	78
Gambar 3.12 Tampilan Form Login.....	79
Gambar 3.13 Dashboard Aplikasi.....	80
Gambar 3.14 Tampilan Kelola Data	81
Gambar 4.1 Website XAMPP	83
Gambar 4.2 Editor Visual Studio Code.....	85
Gambar 4.3 Editor Notepad++	86
Gambar 4.4 Tool MySQL PHPMyAdmin	87
Gambar 4.5 Unduh phpMyAdmin	88
Gambar 4.6 Tampilan HeidiSQL	89
Gambar 4.7 Tahapan Pengembangan Pusat Karir SMK	90
Gambar 4.8 Testing Laravel	92
Gambar 4.9 Struktur direktori dan file di Laravel 6.x	93
Gambar 4.10 Routing dalam Model MVC	94
Gambar 4.11 Routing Aplikasi	95
Gambar 4.12 Routing Groups untuk Iduka	96
Gambar 4.13 Routing Group untuk Siswa.....	97
Gambar 4.14 Routing Group untuk Admin	98
Gambar 4.15 Pengaturan database di Laravel.....	99
Gambar 5.1 Level Akses Aplikasi	180
Gambar 5.2 Fitur Level Akses Admin	181
Gambar 5.3 Fitur Level Akses BKK	182
Gambar 5.4 Fitur Level Akses IDUKA.....	183
Gambar 5.5 Fitur Level Akses Siswa	183
Gambar 5.6 Laman Utama Pusat Karir SMK	184
Gambar 5.7 Form Login Pusat Karir SMK	185
Gambar 5.8 Pencarian Logongan Kerja.....	186
Gambar 5.9 Laman Lowongan Kerja	186
Gambar 5.10 Detail Lowongan Kerja	187
Gambar 5.11 Laman Perusahaan	188
Gambar 5.12 Laman Tips dan Artikel.....	189
Gambar 5.13 Laman Pendaftaran Akun Siswa.....	190
Gambar 5.14 Laman Pendaftaran Perusahaan	191
Gambar 5.15 Dashboad Siswa dan Alumni	191
Gambar 5.16 Laman Konfigurasi Profil	192
Gambar 5.17 Informasi Personal	193
Gambar 5.18 Kontak dan Media Sosial	194

Gambar 5.19 Informasi Kesehatan.....	194
Gambar 5.20 Tambah Riwayat Pendidikan Format	195
Gambar 5.21 Data Riwayat Pendidikan Formal	195
Gambar 5.22 Tambah Keahlian atau Skill.....	196
Gambar 5.23 Data Keahlian atau Skill	196
Gambar 5.24 Progress Kelengkapan Profil.....	197
Gambar 5.25 Cetak CV	197
Gambar 5.26 Laman Berkas Portofolio Siswa	198
Gambar 5.27 Laman Pantau Lamaranku.....	198
Gambar 5.28 Laman Panggilan Tes.....	199
Gambar 5.29 Bukti telah mengisi Tracer Study.....	200
Gambar 5.30 Cetak data isian Tracer Study	200
Gambar 5.31Cetak CV.....	201
Gambar 5.32 Dashboad Login BKK	201
Gambar 5.33 Tautan Aplikasi Pusat Karir SMK.....	203
Gambar 5.34 Laman Kelola Perusahaan.....	204
Gambar 5.35 Detail Data Perusahaan	204
Gambar 5.36 Tambah Data Perusahaan	205
Gambar 5.37 Edit Data Perusahaan	206
Gambar 5.38 Hapus Data Perusahaan	207
Gambar 5.39 Kelola Data Lowongan Pekerjaan	208
Gambar 5.40 Tambah Lowongan Pekerjaan.....	209
Gambar 5.41 Detail Lowongan Pekerjaan	210
Gambar 5.42 Kelola Deskripsi, Syarat dan Keahlian.....	211
Gambar 5.43 Edit Lowongan Kerja	212
Gambar 5.44 Tambah Dekripsi, Syarat dan Keahlian.....	212
Gambar 5.45 Tambah Deskripsi Pekerjaan.....	213
Gambar 5.46 Tambah Syarat/Kualifikasi Pekerjaan	213
Gambar 5.47 Tambah Keahlian	214
Gambar 5.48 Data Pelamar	214
Gambar 5.49 Tambah Panggilan Tes	215
Gambar 5.50 Seleksi pada Panggilan Tes	216
Gambar 5.51 Hapus Panggilan Tes	216
Gambar 5.52 Hapus Lowongan Kerja	217
Gambar 5.53 Publish Lowongan Kerja.....	217
Gambar 5.54 Data Pendaftar.....	218
Gambar 5.55 Pendaftar Perusahaan	218

Gambar 5.56 Data Siswa.....	219
Gambar 5.57 Tambah Data Siswa	221
Gambar 5.58 Import Data Siswa	222
Gambar 5.59 Data Portofolio Siswa.....	222
Gambar 5.60 Nilai Akademik Siswa	223
Gambar 5.61 Lihat Isian Data Tracer Study.....	224
Gambar 5.62 Detail data Siswa	225
Gambar 5.63 Edit Data Siswa	225
Gambar 5.64 Hapus Data Siswa	226
Gambar 5.65 Mengisi Tracer Study.....	227
Gambar 5.66 Mengisi Data Tracer Bekerja.....	228
Gambar 5.67 Mengisi data Tracer kuliah	229
Gambar 5.68 Mengisi Data Tracer Berwirausaha	230
Gambar 5.69 Kepuasan terhadap Sekolah.....	231
Gambar 5.70 Data Isian Tracer Study	232
Gambar 5.71 Data Agenda Kegiatan	232
Gambar 5.72 Detail Data Agenda.....	233
Gambar 5.73 Tambah Agenda Kegiatan	234
Gambar 5.74 Edit Agenda Kegiatan.....	235
Gambar 5.75 Hapus Agenda Kegiatan	236
Gambar 5.76 Tips dan Artikel	236
Gambar 5.77 Data Social Media	237
Gambar 5.78 Profil Aplikasi	238
Gambar 5.79 Data FAQ	238
Gambar 5.80 Berkas BKK	239
Gambar 5.81 Tambah Berkas BKK.....	240
Gambar 5.82 Data Pengguna	241
Gambar 5.83 Reset Password	242
Gambar 5.84 Edit Data Pengguna.....	242
Gambar 5.85 Hapus Data Pengguna.....	243
Gambar 5.86 Data BKK.....	244
Gambar 5.87 Tambah BKK.....	244
Gambar 5.88 Edit BKK	245
Gambar 5.89 Hapus data BKK.....	245
Gambar 5.90 Kategori Perusahaan.....	246
Gambar 5.91 Jumlah Karyawan	247
Gambar 5.92 Data Jurusan	247

Gambar 5.93 Data Waktu Bekerja.....	248
Gambar 5.94 Dashboard Admin Pusat Karir	249

DAFTAR TABEL

Tabel 3.1 Struktur tabel ms_agama	47
Tabel 3.2 Struktur tabel ms_agenda.....	47
Tabel 3.3 Struktur tabel ms_aktivitas_alumni.....	48
Tabel 3.4 Struktur tabel ms_bkk.....	49
Tabel 3.5 Struktur tabel ms_faq.....	50
Tabel 3.6 Struktur tabel ms_file.....	50
Tabel 3.7 Struktur tabel ms_file_siswa.....	51
Tabel 3.8 Struktur tabel ms_gaji	52
Tabel 3.9 Struktur tabel ms_goldarah	52
Tabel 3.10 Struktur tabel ms_hak_akses	53
Tabel 3.11 Struktur tabel ms_jenjang_pendidikan	53
Tabel 3.12 Struktur tabel ms_jml_karyawan	54
Tabel 3.13 Struktur tabel ms_jurusan	54
Tabel 3.14 Struktur tabel ms_karir	55
Tabel 3.15 Struktur tabel ms_kategori_perusahaan.....	55
Tabel 3.16 Struktur tabel ms_kelas.....	56
Tabel 3.17 Struktur tabel ms_kota	56
Tabel 3.18 Struktur tabel ms_level_pekerjaan	56
Tabel 3.19 Struktur tabel ms_lowongan.....	57
Tabel 3.20 Struktur tabel ms_lowongan_deskripsi	58
Tabel 3.21 Struktur tabel ms_lowongan_keahlian.....	59
Tabel 3.22 Struktur tabel ms_lowongan_syarat	60
Tabel 3.23 Struktur tabel ms_matpel	60
Tabel 3.24 Struktur tabel ms_omset	60
Tabel 3.25 Struktur tabel ms_panggilan_tes	61
Tabel 3.26 Struktur tabel ms_pendaftar	62
Tabel 3.27 Struktur tabel ms_peroleh_info.....	63
Tabel 3.28 Struktur tabel ms_perusahaan	64
Tabel 3.29 Struktur tabel ms_profil	65
Tabel 3.30 Struktur tabel ms_relevansi_pekerjaan	66
Tabel 3.31 Struktur tabel ms_siswa	66
Tabel 3.32 Struktur tabel ms_sosmed	68
Tabel 3.33 Struktur tabel ms_status_kontrak.....	68
Tabel 3.34 Struktur tabel ms_tips_karir	69

Tabel 3.35 Struktur tabel ms_waktu_bekerja	69
Tabel 3.36 Struktur tabel tr_bidang_keahlian	70
Tabel 3.37 Struktur tabel tr_lamaran.....	71
Tabel 3.38 Struktur tabel tr_nilai_akademik	71
Tabel 3.39 Struktur tabel tr_panggilan_tes	72
Tabel 3.40 Struktur tabel tr_pendaftar_perusahaan	73
Tabel 3.41 Struktur tabel tr_riwayat_pendidikan	73
Tabel 3.42 Struktur tabel tr_tracer_study.....	74
Tabel 3.43 Struktur tabel users	77
Tabel 4.1 Method Controller Perusahaan	104
Tabel 4.2 Method Controller Siswa	112
Tabel 4.3 Method Controller Lowongan	130
Tabel 4.4 Method Controller Pendaftar Iduka	146
Tabel 4.5 Method Controller Agenda	149
Tabel 4.6 Method Controller Pengguna.....	153
Tabel 4.7 Method Controller FAQ.....	159
Tabel 4.8 Method Controller Panggilan Tes	162
Tabel 4.9 Method Controller Profil.....	166
Tabel 4.10 Method Controller Tips Karis	169
Tabel 4.11 Method Controller Berkas BKK.....	172
Tabel 4.12 Method Controller Pendaftar Siswa	175

BAB 1

BURSA KERJA KHUSUS DI ERA INDUSTRI 4.0

1.1 Pengantar

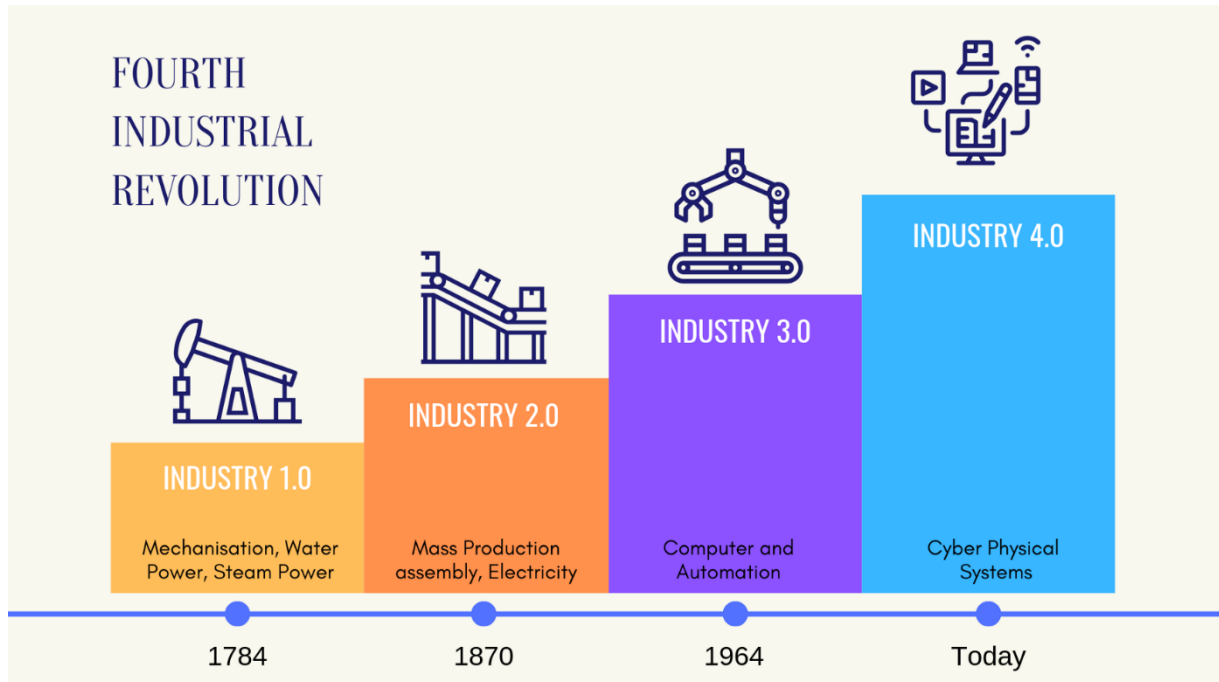
Biro Pusat Statistik (BPS) mencatat jumlah pengangguran pada Agustus 2020 adalah sebanyak 9,77 juta. Jumlah ini naik 2,67 juta orang dari periode yang sama tahun 2019. Dari jumlah pengangguran tersebut, yang paling tinggi adalah lulusan SMK (Sekolah Menengah Kejuruan). Tingkat pengangguran terbuka (TPT) dari lulusan SMK sebesar 13,55% dan menjadi yang tertinggi sementara yang paling rendah adalah lulusan Sekolah Dasar (SD) yaitu 3,61%. Sedangkan sisanya, Sekolah Menengah Pertama (SMP) sebesar 6,46%, Sekolah Menengah Atas (SMA) sebesar 8,08%, dan lulusan perguruan tinggi jenjang Sarjana sebesar 7,35%.

Dengan angka pengangguran yang tembus 9,77 juta orang, maka TPT menjadi 7,07% atau naik dibandingkan Agustus 2019 yang sebesar 5,23%. Jika dilihat menurut tempat tinggal, pengangguran terbuka di perkotaan meningkat 8,98% dan perdesaan naik 4,71%. Menurut jenis kelamin, TPT laki-laki sebesar 7,46% atau lebih tinggi dibandingkan TPT perempuan yang sebesar 6,46%. Dibandingkan Agustus 2019, TPT laki-laki naik 2,13% dan perempuan naik sebesar 1,24%. Menurut kelompok umur, penduduk kelompok usia muda dari 12-24 tahun merupakan yang paling tinggi yaitu mencapai 20,46%. Sementara kelompok usia tua yaitu 60 tahun ke atas merupakan yang paling rendah yaitu 1,70%. Sementara TPT kelompok usia 25-59 tahun meningkat 5,04%.

1.2 Revolusi Industri 4.0

Revolusi Industri 4.0 merupakan perubahan besar di sektor industri yang menggabungkan teknologi otomatisasi dengan teknologi siber. Revolusi Industri 4.0 dikenal juga dengan istilah *cyber physical system*. Era ini ditandai dengan berkembangnya teknologi Internet of Things (IoT), big data, kecerdasan buatan, cloud computing, blockchain, machine learning, data science, dan lain-lain. Saat ini, Indonesia tengah bersiap menghadapi revolusi industri 4.0 dan mulai berbenah menanggapi adanya perubahan industri dengan meluncurkan *roadmap* 'Making Indonesia 4.0'. Roadmap ini memberikan arah yang jelas bagi pergerakan industri nasional di masa depan, termasuk fokus pada pengembangan sektor prioritas yang akan menjadi kekuatan Indonesia menuju Industri 4.0.

Konsep revolusi industri 4.0 pertama kali diperkenalkan oleh ekonom asal Jerman Prof. Klaus Schwab melalui bukunya *The Fourth Industrial Revolution* yang menyatakan bahwa revolusi industri 4.0 secara fundamental dapat mengubah cara kita hidup, cara bekerja, dan cara berhubungan satu dengan yang lain.



Gambar 1.1 Sejarah Revolusi Industri

Berdasarkan Gambar 1.1, empat evolusi industri dari dahulu hingga kini adalah sebagai berikut.

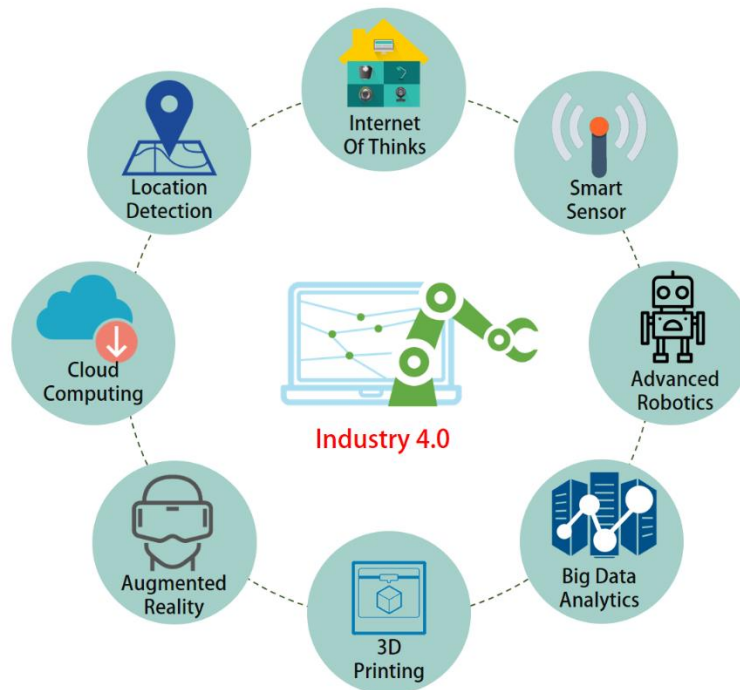
1. Revolusi industri yang pertama terjadi pada akhir abad ke-18. Hal ini ditandai dengan ditemukannya alat tenun mekanis pertama pada tahun 1784. Kala itu, industri diperkenalkan dengan fasilitas produksi mekanis yang menggunakan tenaga air dan uap. Peralatan kerja yang awalnya bergantung pada tenaga manusia dan hewan akhirnya digantikan dengan mesin tersebut. Akibatnya, meski jumlah produksi meningkat, banyak orang yang menganggur.
2. Revolusi industri 2.0 terjadi di awal abad ke-20. Kala itu ada pengenalan produksi massal berdasarkan pembagian kerja. Produksi massal ini dimungkinkan dengan adanya listrik dan jalur perakitan. Lini produksi pertama melibatkan rumah potong hewan di Cincinnati, Amerika Serikat, pada 1870.
3. Awal tahun 1970 ditengarai sebagai perdana kemunculan revolusi industri 3.0 yang dimulai dengan penggunaan elektronik dan teknologi informasi guna otomatisasi

produksi. Debut revolusi industri generasi ketiga ditandai dengan kemunculan pengontrol logika terprogram pertama (PLC), yakni modem 084–969. Sistem otomatisasi berbasis komputer ini membuat mesin industri tidak lagi dikendalikan manusia. Biaya produksi dapat ditekan oleh karena penerapan hal ini.

4. awal 2018 hingga sekaranglah zaman revolusi industri 4.0. Industri 4.0 adalah industri yang menggabungkan teknologi otomatisasi dengan teknologi cyber. Ini merupakan tren otomatisasi dan pertukaran data dalam teknologi manufaktur. Pada era ini, industri mulai menyentuh dunia *virtual*, berbentuk konektivitas manusia, mesin dan data, semua sudah ada di mana-mana. Istilah ini dikenal dengan nama *Internet of Things* (IoT).

Sejalan dengan perkembangan revolusi industri 4.0 tersebut, perusahaan membutuhkan pekerja dengan keterampilan baru, yang mungkin tidak ada sebelumnya. Beberapa bidang pekerjaan akan mengalami peluang untuk berkembang pesat, sementara bidang pekerjaan yang lain mungkin akan menurun bahkan hilang. Teknologi menjadi solusi lebih baik dibandingkan kemampuan manusia yang terbatas. Pada industri 4.0 terjadi konektivitas secara nyata antara manusia, mesin, dan data, artinya terjadi otomatisasi dan digitalisasi.

INDUSTRY 4.0 FRAMEWORK - THE DIGITAL TECHNOLOGIES



Gambar 1.2 Industri 4.0

Revolusi industri 4.0 tidak datang tanpa membawa masalah baru. Salah satu masalah yang mungkin ditimbulkan oleh revolusi ini yakni terciptanya pengangguran yang dipengaruhi oleh melebarinya ketimpangan ekonomi. Digitalisasi dapat menggeser peran konvensional di dalam pasar. Sopir transportasi konvensional seperti sopir ojek pangkalan, angkot, dan taksi berpeluang masuk jurang pengangguran akibat kemunculan transportasi daring yang dinilai jauh lebih murah dan nyaman di mata masyarakat saat ini. Penggunaan robot dalam mendukung otonomisasi di ranah industri manufaktur dan jasa akan semakin tidak terelakkan. Hal ini didorong keinginan perusahaan untuk memangkas biaya yang ditimbulkan sumber daya manusia.

Perkembangan teknologi yang pesat cepat atau lambat akan berpengaruh pada permintaan tenaga kerja di masa depan. Ke depan, permintaan tenaga kerja bergeser. Industri akan cenderung memilih tenaga kerja terampil menengah dan tinggi (*middle and highly-skilled labor*) ketimbang tenaga kerja kurang terampil (*less-skilled labor*) karena perannya dalam mengerjakan pekerjaan repetisi dapat digantikan dengan otonomisasi robot.

1.3 Bursa Kerja Khusus di Era Industri 4.0

Pusat Pengembangan Karir Siswa/Bursa Kerja Khusus (BKK) adalah Lembaga yang dibentuk di SMK, Politeknik/Universitas, dan Lembaga pelatihan yang menjadi mitra lembaga pemerintah sebagai unit pelaksana yang melakukan kegiatan/memberikan layanan bagi peserta didik dan lulusan berupa informasi pasar kerja, pendaftaran pencari kerja, penyuluhan dan bimbingan jabatan, penyaluran dan penempatan kerja. Selain penempatan lulusan, BKK juga melakukan pemasaran tamatan (*job matching*) dengan menjadi penghubung antara pencari kerja lulusan SMK dengan penyedia lapangan kerja.

BKK SMK merupakan salah satu komponen penting dalam mengukur keberhasilan pendidikan di SMK, karena BKK menjadi lembaga yang berperan mengoptimalkan penyaluran tamatan SMK dan sumber informasi untuk pencari kerja. Pemberdayaan BKK SMK merupakan salah satu fungsi dalam manajemen sekolah yaitu sebagai bagian pembinaan terhadap proses pelaksanaan kegiatan BKK SMK yang telah direncanakan dalam upaya mencapai tujuan pendidikan SMK. BKK SMK merupakan salah satu komponen pelaksanaan pendidikan sistem ganda, karena tidak mungkin bisa dilaksanakan proses pembelajaran yang mengarah kepada kompetensi jika tidak ada pasangan industri/usaha kerja, sebagai lingkungan kerja dimana siswa belajar keahlian dan profesional serta etos kerja sesuai dengan tuntutan dunia kerja.

Departemen tenaga kerja menentukan tugas dan fungsi dari Bursa Kerja Khusus di satuan pendidikan menengah yaitu:

1. Memberikan layanan informasi ketenagakerjaan pada pelajar dan alumni yang akan memasuki dunia kerja.
2. Membina dan mengembangkan hubungan kerjasama dengan lembaga pemerintah dan swasta termasuk dunia kerja dan alumni yang telah bekerja dalam pengadaan informasi tentang tentang latihan kerja dan penyalurannya sebagai tenaga kerja.
3. Mengadakan kegiatan yang berkaitan dengan rekrutmen dan seleksi calon tenaga kerja atas permintaan Depnaker atau lembaga pemerintah lain atau swasta atas bimbingan dari departemen tenaga kerja.
4. Membina hubungan dengan alumni yang telah bekerja dan berhasil dalam bidang usaha untuk membantu memberikan peluang menyalurkan, menempatkan alumni baru dari almamaternya yang memerlukan pekerjaan.
5. Membantu mengembangkan dan menyempurnakan program pendidikan dan memperhatikan tuntutan lapangan kerja serta meningkatkan peran tenaga pengajar dalam pembinaan karir siswa alumni.

Bursa kerja khusus mempunyai peran penting dalam layanan antar kerja diantaranya dengan memberikan informasi ketenagakerjaan, membina dan mengembangkan hubungan kerjasama dengan lembaga pemerintah dan swasta, melaksanakan kegiatan yang berkaitan dengan rekrutmen dan seleksi, membina hubungan dengan alumni yang sudah bekerja serta membantu usaha pengembangan dan penyempurnaan program pendidikan yang membawa manfaat yang sesuai dengan tujuan yang telah ditentukan.

BKK memberikan informasi mengenai ketenagakerjaan mulai dari menerima, menampung, mengidentifikasi dan mendata jenis-jenis informasi yang didapat dari dunia kerja kemudian menyampaikan informasi tersebut kepada siswa dan alumni sekolah yang bersangkutan. Selain itu, BKK juga memiliki andil dalam pelaksanaan pengembangan karir siswa dan lulusannya, dalam hal ini BKK bekerjasama dengan bimbingan konseling yang ada di Sekolah.

BKK merupakan lembaga yang dibentuk di SMK sebagai unit pelaksana yang memiliki beberapa tujuan yaitu (1) Sebagai wadah dalam mempertemukan tamatan dengan pencari kerja, (2) Memberikan layanan kepada tamatan sesuai dengan tugas dan fungsi masing-masing seksi yang ada dalam BKK, (3) Sebagai wadah dalam pelatihan tamatan yang sesuai dengan permintaan pencari kerja, (4) Sebagai wadah untuk menanamkan jiwa wirausaha bagi tamatan melalui pelatihan. Ruang lingkup kegiatan BKK antara lain:

1. Penyusunan database siswa lulusan SMK pencari kerja dan perusahaan pencari tenaga kerja dan penelusuran tamatan siswa SMK.

2. Menjaring informasi tentang pasar kerja melalui iklan di media massa, internet, kunjungan ke dunia usaha (industri) maupun kerjasama dengan lembaga penyalur tenaga kerja dan Depnakertrans.
3. Membuat leaflet informasi dan pemasaran lulusan SMK yang dikirim ke dunia usaha/industri yang terkait Depnakertrans.
4. Penyaluran calon tenaga kerja lulusan SMK ke dunia usaha dan industri.
5. Melakukan proses tindak lanjut hasil pengiriman dan penempatan tenaga kerja melalui kegiatan penjajakan dan verifikasi.
6. Mengadakan program pelatihan ketrampilan tambahan/khusus bagi siswa dan lulusan SMK disesuaikan dengan bidang keahlian yang diperlukan.
7. Mengadakan program bimbingan menghadapi tahapan proses penerimaan siswa dalam suatu pekerjaan (wawancara, psikotest).
8. Memberikan informasi kepada para alumni ataupun para lulusan SMK lain yang membutuhkan informasi tentang lowongan kerja.

Tolok ukur keberhasilan BKK dapat dilihat dari keberhasilan atau tercapainya tujuan-tujuan dari pelaksanaan BKK. Berikut ini adalah beberapa hal yang dapat dijadikan sebagai indikator keberhasilan BKK yaitu:

1. Pemenuhan kelengkapan perizinan dan legalitas. Keberhasilan dari program ini dapat diindikasikan dengan adanya SK dari kepala sekolah dan terbitnya surat ijin dari departemen tenaga kerja.
2. Kelengkapan fasilitas BKK, bertujuan melengkapi fasilitas BKK untuk memperlancar kegiatannya seperti kelengkapan ruangan, meja, kursi, alat tulis, komputer, dan lain-lain.
3. Pendaftaran alumni lulusan SMK, dibuktikan dengan adanya data tentang nama dan alamat yang lengkap dari lulusan yang digunakan untuk proses perekrutan calon tenaga kerja ketika ada lowongan atau kesempatan kerja yang ditawarkan melalui BKK.
4. Kunjungan dan penawaran kerjasama ke DUDI. Diindikasikan dengan banyaknya jumlah industri yang mau bekerjasama dengan BKK SMK dari seluruh industri yang dikunjungi.
5. Pengiriman atau penyaluran lulusan ke dunia kerja. Indikasi keberhasilan program ini dapat diketahui dengan banyaknya lulusan atau alumni SMK tersebut yang dapat tersalur ke dunia kerja melalui BKK.

Penelusuran lulusan (*tracer study*) merupakan salah satu kegiatan sekolah dan BKK yang sangat penting untuk selalu dilakukan secara rutin. Kegiatan ini bermanfaat bagi

sekolah, alumni dan siswa yang sedang aktif. Melalui kegiatan penelusuran lulusan SMK diharapkan dapat:

1. Untuk memperbaiki pengajaran dan pembelajaran di sekolah.
2. Untuk mengumpulkan informasi yang penting sehingga bisa digunakan untuk memperbaiki program
3. Memperoleh data riil mengenai kondisi lulusan maksimal 10 bulan setelah lulus
4. Mengklasifikasi kondisi lulusan sudah bekerja, bekerja mandiri, melanjutkan belajar, dan belum bekerja.
5. Menginformasikan lapangan pekerjaan di industri melalui bursa kerja
6. Memberikan motivasi lulusan dalam memperoleh kesempatan kerja dan melanjutkan belajar/kuliah.

BAB 2

STUDI TERKAIT PENGEMBANGAN PERANGKAT LUNAK

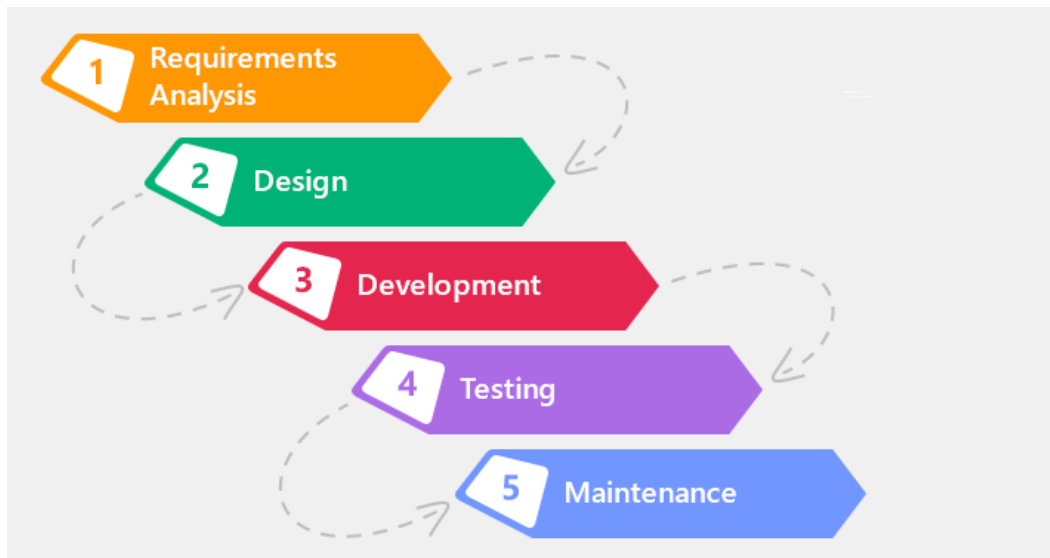
2.1 Tahap Pengembangan Perangkat Lunak

SDLC atau *Software Development Life Cycle* adalah siklus hidup pengembangan perangkat lunak. SDLC merupakan tahapan-tahapan pekerjaan yang dilakukan oleh analis sistem dan programmer dalam membangun sistem informasi dan metode dalam mengembangkan sistem tersebut. Sistem yang dibangun dengan menggunakan SDLC akan memudahkan dalam mengidentifikasi masalah dan merancang sistem sesuai kebutuhan dalam menyelesaikan permasalahan tersebut. Semua software developer dalam perusahaan pastinya menggunakan SDLC mengembangkan software atau aplikasi. Dalam ilmu rekayasa perangkat lunak, SDLC adalah suatu proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan perangkat lunak.

Semua *software developer* dalam perusahaan pastinya menggunakan SDLC mengembangkan software atau aplikasi. Setidaknya, terdapat 6 model dalam pengembangan perangkat lunak yakni (1) Waterfall, (2) Prototype, (3) Spiral, (4) RAD (Rapid Application Development), (5) Agile, dan (6) Scrum. Tidak ada model terbaik, karena setiap model memiliki karakteristik dan spesifikasi yang spesifik sehingga tidak cocok untuk semua kasus. Setiap model memiliki kelebihan dan kekurangan. Metode pengembangan perangkat lunak Pusat Karir SMK adalah *Waterfall Model*.

Model *Waterfall* adalah model pengembangan perangkat lunak yang menekankan fase-fase yang berurutan dan sistematis, dimulai dari spesifikasi kebutuhan pengguna melalui proses perencanaan (*planning*), pemodelan (*modelling*), pembangunan (*construction*), dan penyebaran (*deployment*) yang berujung pada dukungan terus menerus untuk sebuah perangkat lunak yang utuh. Model ini dapat digunakan pada saat kebutuhan untuk sebuah masalah telah dipahami dengan baik, dan pekerjaan dapat mengalir secara berurutan (sekuen) dari proses komunikasi hingga penyebaran (*deployment*). Pada setiap tahapan diumpamakan seperti bak air yang mengalir dari tempat tinggi ke tempat yang lebih rendah, artinya sebuah proses baru hanya dapat dilanjutkan setelah satu tahap sebelumnya telah selesai dengan sempurna.

Model *waterfall* terdiri dari 5 tahapan yakni *Requirement Analysis*, *Design*, *Development*, *Testing*, dan *Maintenance*.



Gambar 2.1 Waterfall Model

Penjelasan detail dari masing-masing tahapan berdasarkan gambar 2.1 adalah sebagai berikut:

1. Requirement Analysis

Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh perangkat lunak yang akan dibangun. Fase ini harus dikerjakan secara lengkap untuk bisa menghasilkan desain yang lengkap. Tujuan tahapan ini adalah memahami *software* yang dibutuhkan pengguna dan batasan *software*. Informasi ini biasanya dapat diperoleh melalui wawancara, survey atau diskusi.

2. Design

Pada proses desain, dilakukan penerjemahan syarat kebutuhan ke sebuah perancangan desain perangkat lunak yang dapat diperkirakan sebelum dibuatnya proses pengkodean (*coding*). Proses ini berfokus pada struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail algoritma procedural. Desain sistem yang dibuat antara lain:

- Flowchart
- Swimline Diagram
- Data Flow Diagram (DFD)
- CDM (*Conseptual Data Model*) dan PDM (*Physical Data Model*)
- Desain antarmuka (Input dan Output)

3. Implementation

Pada tahap ini terjadi proses menerjemahkan perancangan desain ke bentuk yang dapat dimengerti oleh mesin, dengan menggunakan kode-kode bahasa pemrograman. Kode program yang dihasilkan masih berupa modul-modul (unit) kecil yang nantinya akan digabungkan pada tahap berikutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas melalui mekanisme pengujian unit (*unit testing*).

4. Integration & Testing

Di tahap ini dilakukan penggabungan modul-modul yang sudah dibuat dan dilakukan pengujian ini dilakukan untuk mengetahui apakah *software* yang dibuat telah sesuai dengan desainnya dan fungsi pada *software* terdapat kesalahan atau tidak. Pada tahap ini, klien atau pengguna menguji apakah sistem tersebut telah sesuai dengan yang disetujui.

5. Operation & Maintenance

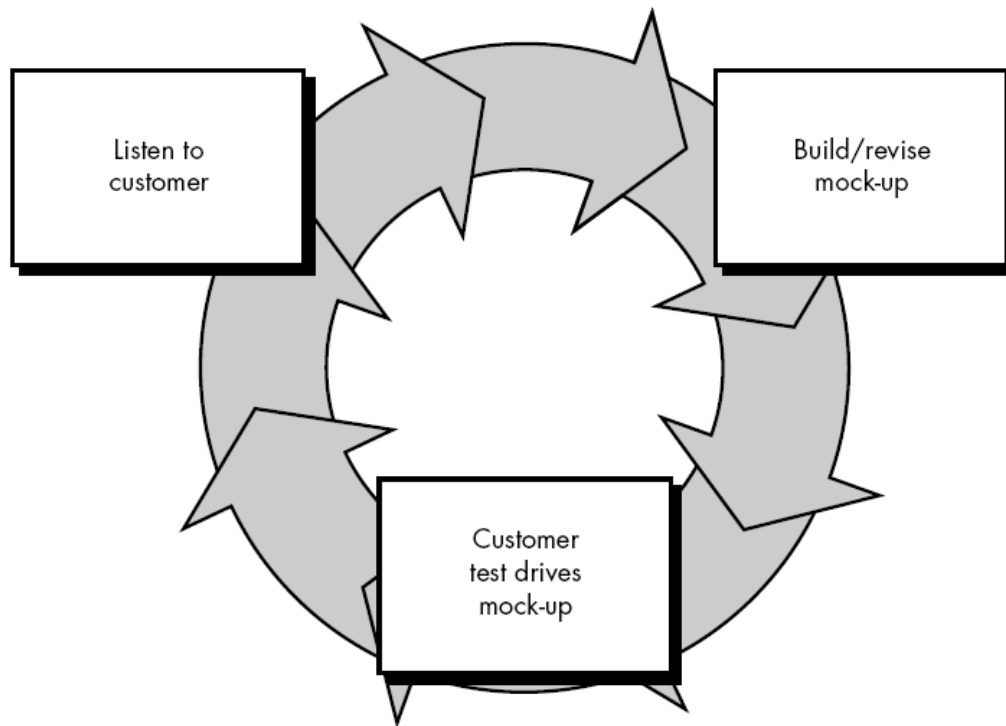
Merupakan tahap terakhir dalam model *waterfall*. *Software* yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru. Pada tahap ini, dilakukan instalasi dan proses perbaikan sistem sesuai yang disetujui.

Ciri-ciri dari model *waterfall* dalam pengembangan perangkat lunak antara lain: (1) Proyek dibagi-bagi dalam beberapa fase yang saling berurutan, (2) Penekanan pada perencanaan, jadwal, deadline dan implementasi keseluruhan sistem, (3) Kontrol yang ketat dalam siklus hidup proyek dengan menggunakan bantuan dokumentasi tertulis.

Kelebihan model *waterfall* antara lain (1) Relatif mudah dimengerti dan mudah digunakan, (2) *Requirement* dari sistem bersifat stabil, (3) Baik dalam manajemen control, (4) Bekerja dengan baik ketika kualitas lebih diutamakan dibandingkan dengan biaya dan jadwal/deadline. Sedangkan kelemahannya antara lain: (1) Semua kebutuhan sistem harus diketahui di awal, (2) Integrasi sekaligus di akhir sistem, (3) *Testing* hanya dilakukan pada setiap akhir fase, bahkan tidak jarang *testing* hanya dilakukan di akhir-akhir proyek, (4) Membutuhkan waktu yang cukup lama meski kadang proyeknya tidak terlalu besar, (5) Perubahan *requirement* dapat merubah keseluruhan proses yang telah dilaksanakan.

Model *prototype* merupakan model pengembangan *software* yang memungkinkan pengguna/user memiliki gambaran awal tentang program yang akan dikembangkan serta melakukan pengujian awal. Model *prototype* juga memberi fasilitas bagi pengembang dan user untuk saling terkait dan berinteraksi. Dalam model ini, *prototype* dari perangkat lunak yang dihasilkan kemudian dipresentasikan kepada pelanggan, dan pelanggan tersebut

diberikan kesempatan untuk memberikan masukan sehingga perangkat lunak yang dihasilkan nantinya betul-betul sesuai dengan keinginan dan kebutuhan pelanggan.

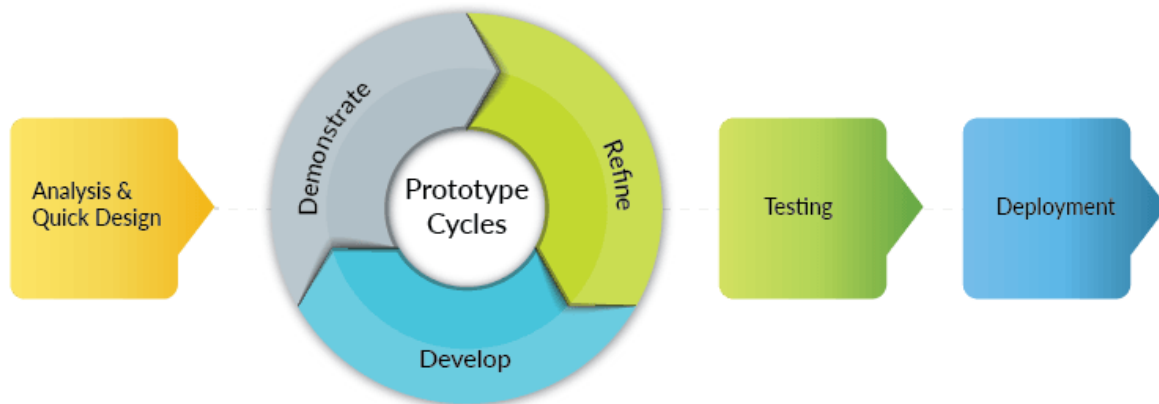


Gambar 2.2 Model Prototype

Kelebihan model prototype yaitu (1) Pelanggan berpartisipasi aktif dalam pengembangan sistem, sehingga hasil produk pengembangan akan semakin mudah disesuaikan dengan keinginan dan kebutuhan pelanggan, (2) Penentuan kebutuhan lebih mudah diwujudkan, (3) Mempersingkat waktu pengembangan produk perangkat lunak, (4) Adanya komunikasi yang baik antara pengembang dan pelanggan, (5) Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan, (6) Penerapan menjadi lebih mudah karena pelanggan mengetahui apa yang diharapkannya, (7) Perbaikan kesalahan relative cepat. Sedangkan kelemahannya yaitu (1) Proses analisis dan perancangan terlalu singkat, (2) Biasanya kurang fleksibel dalam menghadapi perubahan, (3) Pengembang kadang-kadang membuat kompromi implementasi dengan menggunakan sistem operasi yang tidak relevan dan algoritma yang tidak efisien, (4) Mengurangi kreatifitas perancangan, (5) Cakupan pengembangannya sistem dapat lebih luas.

Model pengembangan yang lain adalah RAD (*Rapid Application Development*) yaitu sebuah model proses perkembangan perangkat lunak sekuensial linier yang menekankan siklus perkembangan yang sangat pendek (kira-kira 60 sampai 90 hari). Model RAD ini merupakan sebuah adaptasi “kecepatan tinggi” dari model sekuensial linier dimana

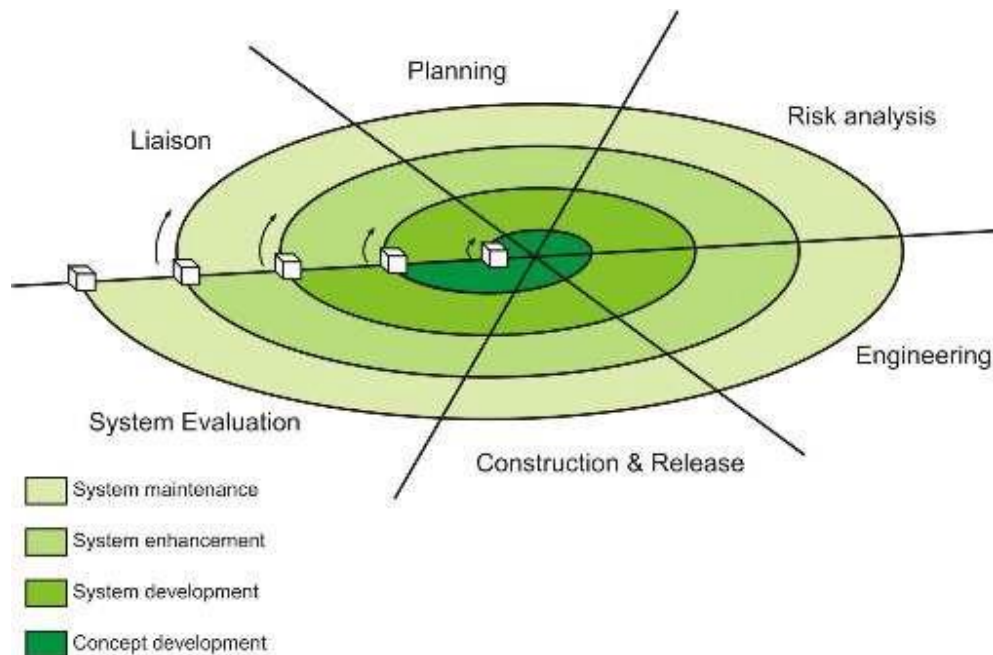
perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen. Model ini bertujuan mempersingkat waktu yang biasanya diperlukan dalam siklus hidup pengembangan konvensional.



Gambar 2.3 RAD Model

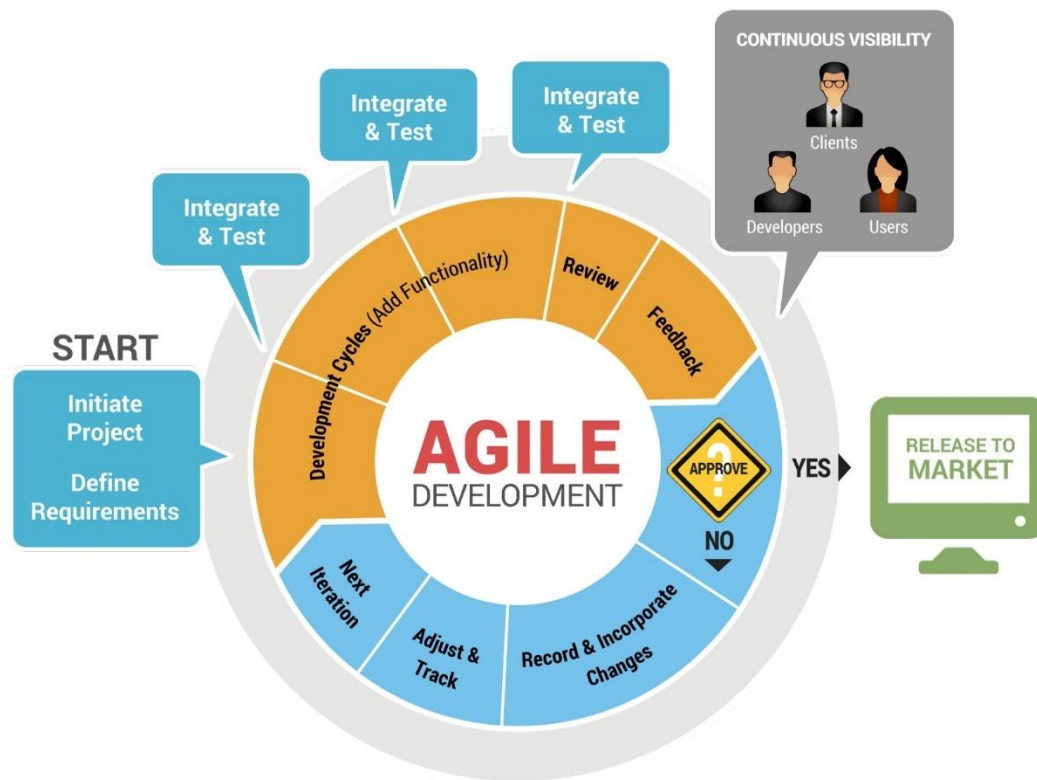
Kelebihan model RAD antara lain: (1) Pengembangan yang cepat, (2) adanya prototype, (3) pengurangan penulisan kode yang kompleks dikarenakan reuse dari kode yang sudah ada. Kelemahan model RAD yaitu (1) tidak relevan untuk proyek skala besar, (2) memerlukan komitmen yang kuat antara pengembangan dengan konsumen, (3) tidak semua aplikasi sesuai untuk RAD, bila system tidak dapat dimodulkan dengan teratur, pembangunan komponen penting pada RAD akan menjadi sangat bermasalah, (4) Membutuhkan Tenaga kerja yang banyak untuk menyelesaikan sebuah proyek dalam skala besar, (5) Jika ada perubahan di tengah-tengah pengerjaan maka harus membuat kontrak baru antara pengembang dan pelanggan.

Model *Spirall* merupakan model pengembangan perangkat lunak yang evolusioner yang memadukan sifat iteratif model *prototype* dan aspek sistematis dari mode sekuensial. Version Release meningkat setiap iterasi terjadi. Kelebihan model *Spirall* yaitu (1) cocok untuk proyek skala besar, (2) manajemen kesalahan baik, (3) menggunakan prototype sebagai mekanisme pengurangan resiko dan pada setiap keadaan didalam evolusi produk. Kelemahannya yaitu (1) waktu pengembangan cukup lama, (2) dibutuhkan ahli dalam penanganan resiko, (3) kesulitan dalam meyakinkan konsumen.



Gambar 2.4 Spirall Model

Model *Agile* merupakan model pengembangan jangka pendek yang memerlukan adaptasi cepat dan pengembangan terhadap perubahan dalam bentuk apapun. Dalam agile terdapat beberapa poin penting diantaranya sebagai berikut: (1) Interaksi antar personal lebih penting daripada proses dan alat, (2) Software yang berfungsi lebih penting daripada dokumentasi yang lengkap, (3) Kolaborasi dengan klien lebih penting daripada negosiasi kontrak, (4) Sikap tanggap lebih penting daripada mengikuti rencana/plan. Kelebihan model Agile yaitu (1) Functional dapat dibuat dengan cepat dan dilakukan testing, (2) Perubahan dengan cepat ditangani. Sedangkan kekurangannya yaitu (1) Analisis, desain, dan pengembangan sulit diprediksi, dan (2) Dapat memunculkan permasalahan dari arsitektur maupun desain.



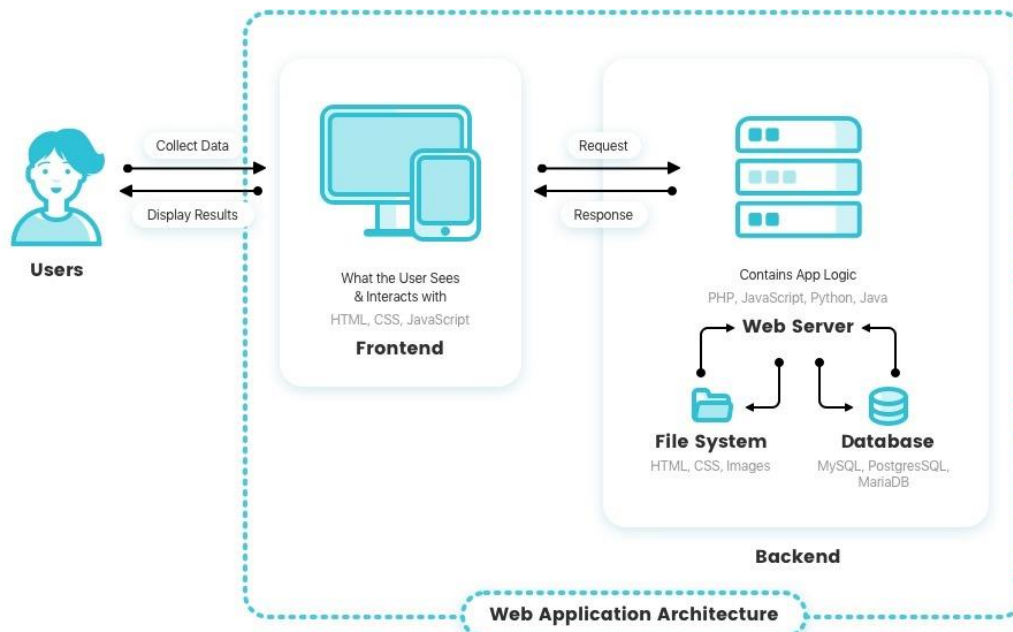
Gambar 2.5 Agile Model

2.2 Web Development

Aplikasi berbasis web adalah sebuah aplikasi yang dapat diakses menggunakan web browser atau penjelajah web melalui jaringan internet atau intranet. Berdasarkan penjelasan di atas, maka dapat diterjemahkan secara bebas bahwa aplikasi berbasis web merupakan sebuah aplikasi yang diakses melalui web browser dengan menggunakan jaringan sebagai media transmisi. Aplikasi web juga merupakan sebuah perangkat lunak atau software yang di kodekan dengan bahasa pemrograman seperti html, javascript, css, ruby, python, php, dan bahasa pemrograman lainnya.

Aplikasi web menjadi populer karena kemudahan tersedianya aplikasi klien untuk mengaksesnya, penjelajah web, yang kadang disebut sebagai suatu thin client (klien tipis). Kemampuan untuk memperbarui dan memelihara aplikasi web tanpa harus mendistribusikan dan menginstalasi perangkat lunak pada kemungkinan ribuan komputer klien merupakan alasan kunci popularitasnya. Aplikasi web yang umum misalnya webmail, toko ritel, lelang online, wiki, papan diskusi, weblog.

Arsitektur aplikasi web terdiri dari *Front-End* dan *Back-End*. *Front-End* adalah tampilan yang dilihat oleh pengguna, dan pengguna dapat melakukan interaksi. Sedangkan *Back-End*, merupakan *logic* dari aplikasi dan koneksi ke database. Arsitektur aplikasi web dapat dilihat pada gambar 2.2 berikut.

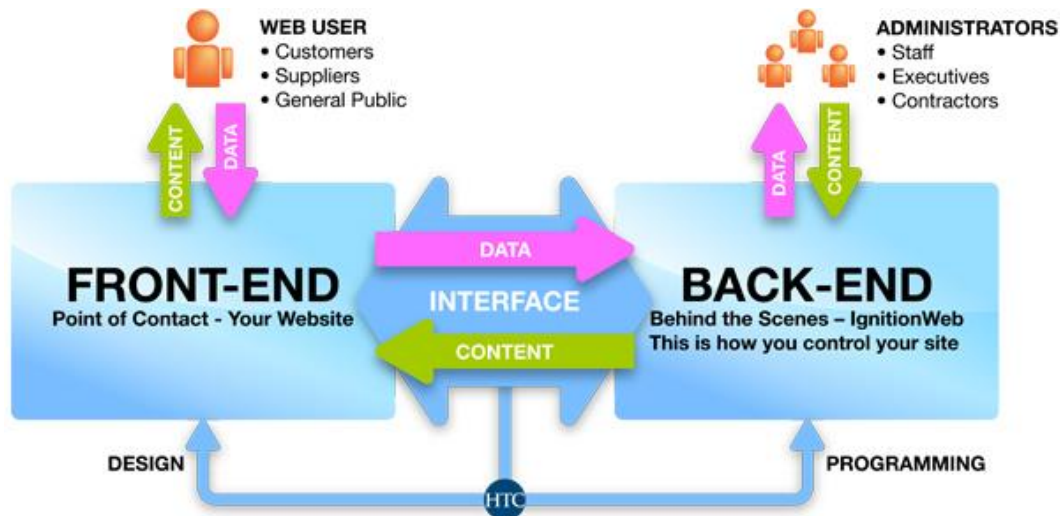


Gambar 2.6 Arsitektur Aplikasi Web

Berdasarkan Gambar 2.6, *Front-End* dibangun menggunakan teknologi HTML, CSS dan JavaScript, sedangkan teknologi untuk backend adalah PHP, JavaScript, Java, Python. Pada aplikasi web, pengguna melakukan permintaan (*request*) melalui tampilan *frontend*. Request tersebut kemudian diteruskan ke *Back-End* untuk diproses, ambil data dari database, dan hasil dari pemrosesan dikembalikan ke *Front-End* dalam bentuk *response* dalam format HTML murni sehingga dapat dilihat kembali pengguna.

Aplikasi web memiliki keunggulan dan kekurangan jika dibandingkan dengan aplikasi berbasis desktop. Kelebihan aplikasi web antara lain: (1) Tidak memerlukan sebuah lisensi ketika menggunakan aplikasi berbasis web dikarenakan lisensi merupakan sebuah tanggung jawab dari penyedia layanan aplikasi, (2) Tidak memerlukan spesifikasi yang tinggi untuk melakukan dan menggunakan aplikasi web, (3) Dapat dijalankan dimanapun dan kapanpun tanpa harus melakukan instalasi, (4) Dapat digunakan di berbagai jenis sistem operasi, (5) Dapat diakses melalui banyak media seperti komputer, laptop, smartphone. Sedangkan kelemahan aplikasi web yaitu (1) Membutuhkan keamanan yang tinggi, hal ini dikarenakan pada jenis aplikasi berbasis web dijalankan secara terpusat,

sehingga apabila sistem server mengalami down maka sistem aplikasi disisi client tidak dapat beroperasi dengan normal, (2) Dibutuhkan sebuah jaringan dan atau koneksi internet untuk melakukan atau menggunakan aplikasi ini.



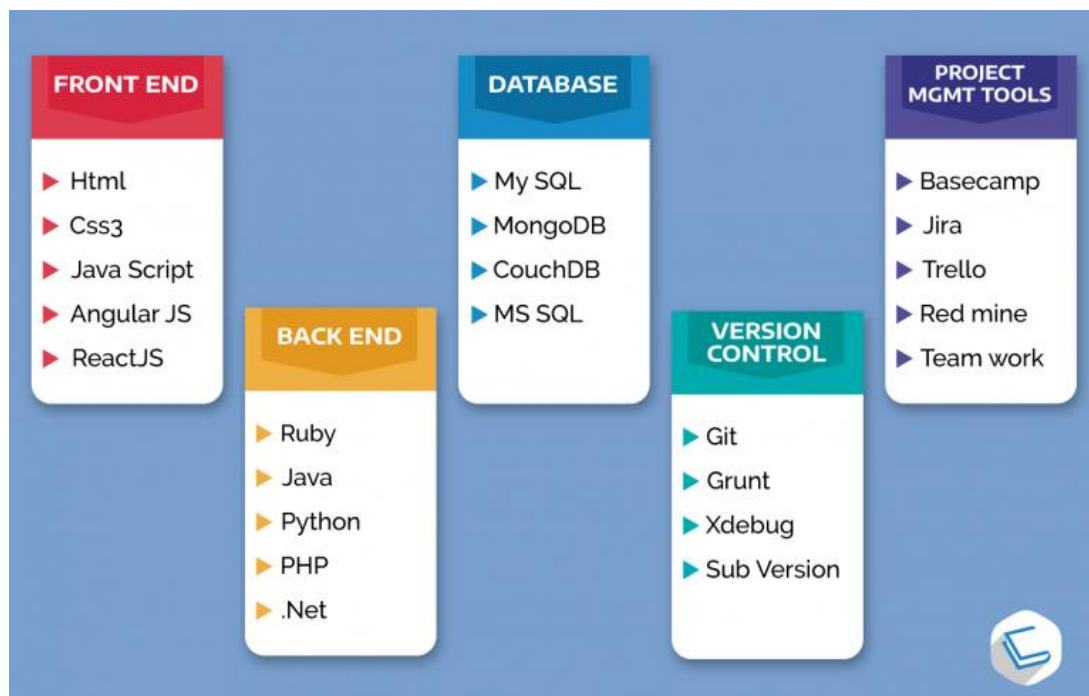
Gambar 2.7 Frontend dan Backend

Seiring perkembangan teknologi, web telah berkembang menjadi lebih kompleks. Ini membuat developer juga menjadi lebih terspesialisasi. Artinya, seorang *developer* tidak harus melakukan keseluruhan proses membangun web lagi, tetapi fokus pada bagian tertentu saja, apakah itu salah satu bahasa pemrograman, framework, atau bagian teknis lainnya. *Front-end developer* menggunakan HTML, CSS, dan JavaScript dalam membangun website, sedangkan *Back-End developer* menggunakan PHP, Python, Ruby, dan lainnya untuk pengembangan *Back-End* suatu website. Full-Stack developer bekerja pada bagian front-end dan back-end sehingga menguasai HTML, CSS, JavaScript, dan satu atau lebih bahasa pemrograman untuk back-end. Untuk menjadi *full-stack developer*, diperlukan keahlian dalam teknologi *front-end*, *back-end*, *database*, *version control* dan *project managemen tools*.

2.3 PHP

Pengertian PHP, PHP Adalah bahasa *scripting server-side*, Bahasa pemrograman yang digunakan untuk mengembangkan situs web statis atau situs web dinamis atau aplikasi Web. PHP singkatan dari *Hypertext Pre-processor*, yang sebelumnya disebut *Personal Home*

Pages. Script sendiri merupakan sekumpulan instruksi pemrograman yang ditafsirkan pada saat runtime. Sedangkan Bahasa scripting adalah bahasa yang menafsirkan skrip saat runtime. Dan biasanya tertanam ke dalam lingkungan perangkat lunak lain. Karena php merupakan scripting server-side maka jenis bahasa pemrograman ini nantinya script/program tersebut akan dijalankan/diproses oleh server. Berbeda dengan javascript yang *client-side*.

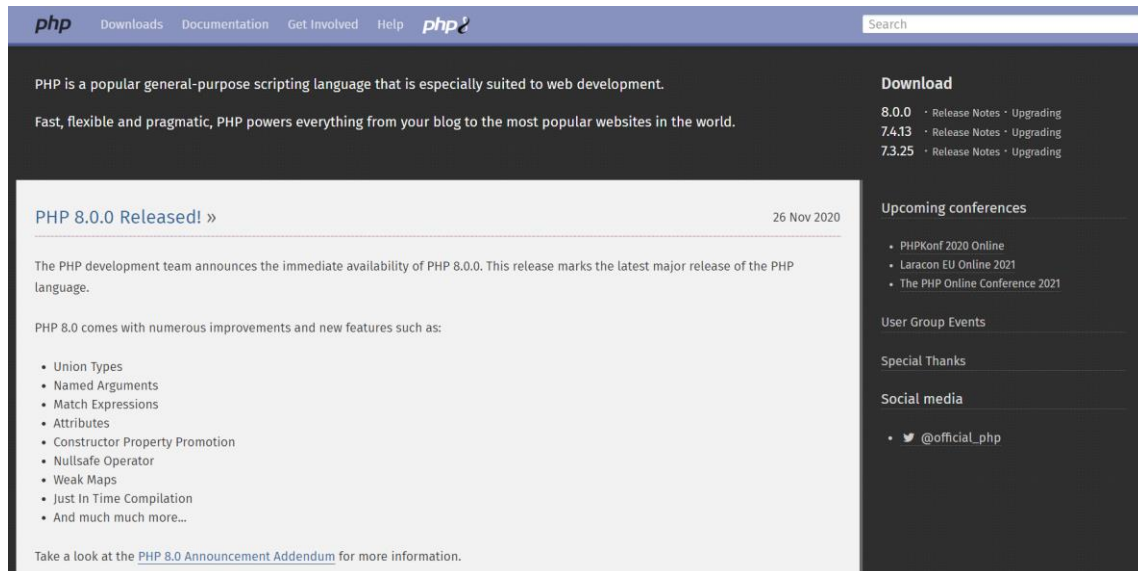


Gambar 2.8 Teknologi Keahlian Pengembangan Web

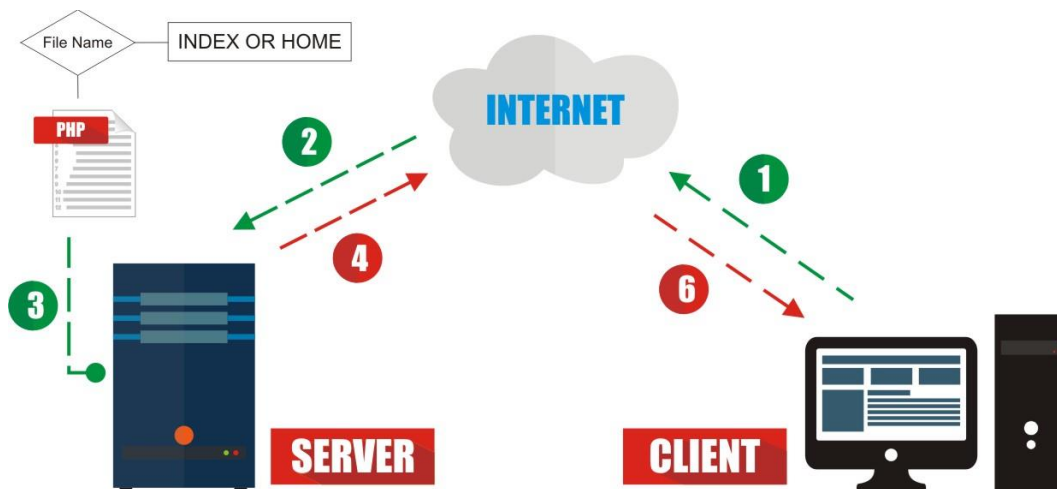
Perangkat Lunak PHP dapat diunduh melalui website resminya di <https://www.php.net/>. Terdapat tiga jenis versi PHP yang dapat diunduh yaitu versi 7.3.25, versi 7.4.13 dan versi paling baru 8.0.0. PHP tersedia untuk semua platform sistem operasi, jadi selama proses unduh harus disesuaikan dengan sistem operasi yang digunakan. Tampilan utama website resmi PHP dapat dilihat pada gambar 2.9.

Menurut Wikipedia, *PHP:Hypertext Preprocessor* adalah bahasa skrip yang ditanamkan atau disisipkan ke dalam HTML yang digunakan untuk membangun website dinamis. PHP bersifat *server side programming*, artinya semua kode PHP dijalankan di server, dan hasil dari pemrosesan dikirim ke *client* dalam format HTML murni. Cara kerja teknologi *server side programming* dapat dilihat pada gambar 2.10, yaitu:

1. Pengunjung melakukan permintaan (*request*) melalui komputer *client*
2. Permintaan tersebut kemudian diteruskan ke *server*
3. *Server* memproses permintaan menggunakan kode PHP
4. *Server* mengirim hasil pemrosesan ke *client* dalam bentuk HTML murni



Gambar 2.9 Website Resmi PHP



Gambar 2.10 Server Side Programming

PHP dapat digunakan dengan gratis (*free*) dan bersifat *Open Source*. PHP dirilis dalam lisensi PHP License, sedikit berbeda dengan lisensi GNU General Public License (GPL) yang

biasa digunakan untuk proyek Open Source. Kemudahan dan kepopuleran PHP sudah menjadi standar bagi programmer web di seluruh dunia. Menurut wikipedia pada februari 2014, sekitar 82% dari web server di dunia menggunakan PHP. PHP juga menjadi dasar dari aplikasi CMS (Content Management System) populer seperti Joomla, Drupal, dan WordPress.

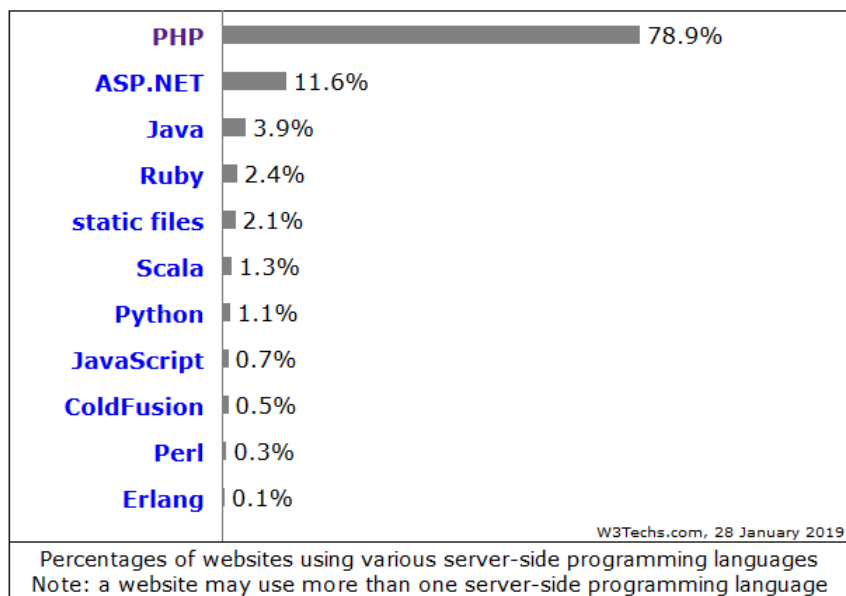
Dikutip dari situs w3techs.com, (diakses pada 28 Januari 2019), berikut adalah market share penggunaan bahasa pemrograman server-side untuk mayoritas website di seluruh dunia:

Usage of server-side programming languages for websites

This diagram shows the percentages of websites using various server-side programming languages. See [technologies overview](#) for explanations on the methodologies used in the surveys. Our reports are updated daily.

How to read the diagram:

PHP is used by 78.9% of all the websites whose server-side programming language we know.



Gambar 2.11 Survey Popularitas Bahasa Pemrograman Web

Untuk membuat halaman web, sebenarnya PHP bukanlah bahasa pemrograman yang wajib digunakan. Kita bisa saja membuat website hanya menggunakan HTML saja. Web yang dihasilkan dengan HTML (dan CSS) ini dikenal dengan website statis, dimana konten dan halaman web bersifat tetap. Sebagai perbandingan, website dinamis yang bisa dibuat menggunakan PHP adalah situs web yang bisa menyesuaikan tampilan konten tergantung situasi. Website dinamis juga bisa menyimpan data ke dalam database, membuat halaman yang berubah-ubah sesuai input dari user, memproses form, dan lain-lain.

Untuk pembuatan web, kode PHP biasanya di sisipkan ke dalam dokumen HTML. Karena fitur inilah PHP disebut juga sebagai Scripting Language atau bahasa pemrograman script. Sebagai contoh penggunaan PHP, misalkan kita ingin membuat list dari nomor 1 sampai nomor 10. Dengan menggunakan HTML murni, kita bisa membuatnya secara manual seperti kode berikut ini:

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Contoh list dengan HTML</title>
5    </head>
6    <body>
7      <h2>Daftar Absensi Mahasiswa</h2>
8      <ol>
9        <li>Nama Mahasiswa ke-1</li>
10       <li>Nama Mahasiswa ke-2</li>
11       <li>Nama Mahasiswa ke-3</li>
12       <li>Nama Mahasiswa ke-4</li>
13       <li>Nama Mahasiswa ke-5</li>
14       <li>Nama Mahasiswa ke-6</li>
15       <li>Nama Mahasiswa ke-7</li>
16       <li>Nama Mahasiswa ke-8</li>
17       <li>Nama Mahasiswa ke-9</li>
18       <li>Nama Mahasiswa ke-10</li>
19     </ol>
20   </body>
21 </html>
```

Gambar 2.12 Kode HTML

Halaman HTML tersebut dapat dibuat dengan mudah dengan cara men-copy-paste tag sebanyak 10 kali dan mengubah sedikit angka-angka no urut di belakangnya. Namun jika yang kita inginkan adalah menambahkan list tersebut menjadi 100 atau 1000 list, cara copy-paste tersebut menjadi tidak efektif. Jika menggunakan PHP, kita tinggal membuat perulangan for sebanyak 1000 kali dengan perintah yang lebih singkat seperti Gambar 2.13 dan Gambar 2.14.

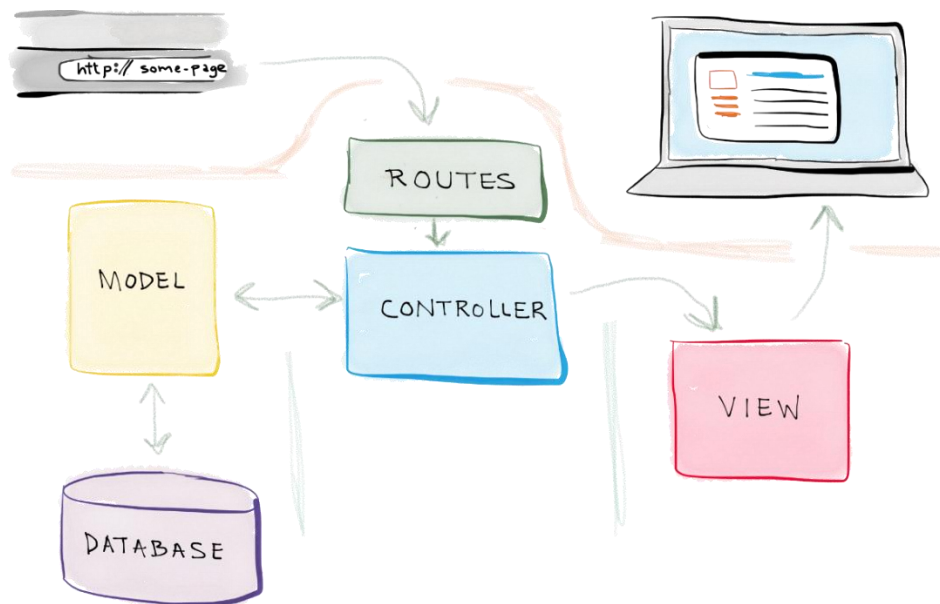
2.4 Laravel

Laravel Framework adalah salah satu dari framework berbasis bahasa pemrograman PHP yang sangat populer. Pada dasarnya, Laravel digunakan untuk pengembangan aplikasi berbasis web, akan tetapi bisa juga digunakan untuk pengembangan REST API service. Pada aplikasi berbasis website umumnya terdapat tiga pembagian tugas besar, yaitu pengolahan tampilan, pengolahan data, dan pengolahan bisnis proses/logika kerja yang

dikenal dengan MVC (Model View Controller). Laravel mengadopsi pola pengembangan Model-View-Controller (MVC) yang bertujuan untuk memenuhi single responsibility principle (SRP), yaitu prinsip pemisahan komponen berdasarkan tugasnya masing-masing.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Contoh list dengan PHP</title>
5   </head>
6   <body>
7     <h2>Daftar Absensi Mahasiswa</h2>
8     <ol>
9       <?php
10        for ($i= 1; $i <= 1000; $i++)
11        {
12          echo "<li>Nama Mahasiswa ke-$i</li>";
13        }
14      ?>
15    </ol>
16  </body>
17 </html>
```

Gambar 2.13 Kode PHP didalam kode HTML



Gambar 2.14 Konsep MVC (Model View Controller)

Dengan mengadopsi pola pengembangan MVC, Laravel dapat membagi ketiga tugas yang ada pada aplikasi dengan pembagian tugas komponen Model bertugas untuk menangani pengolahan data, Komponen View bertugas untuk menangani pengolahan tampilan kepada pengguna, dan Komponen Controller bertugas untuk menangani

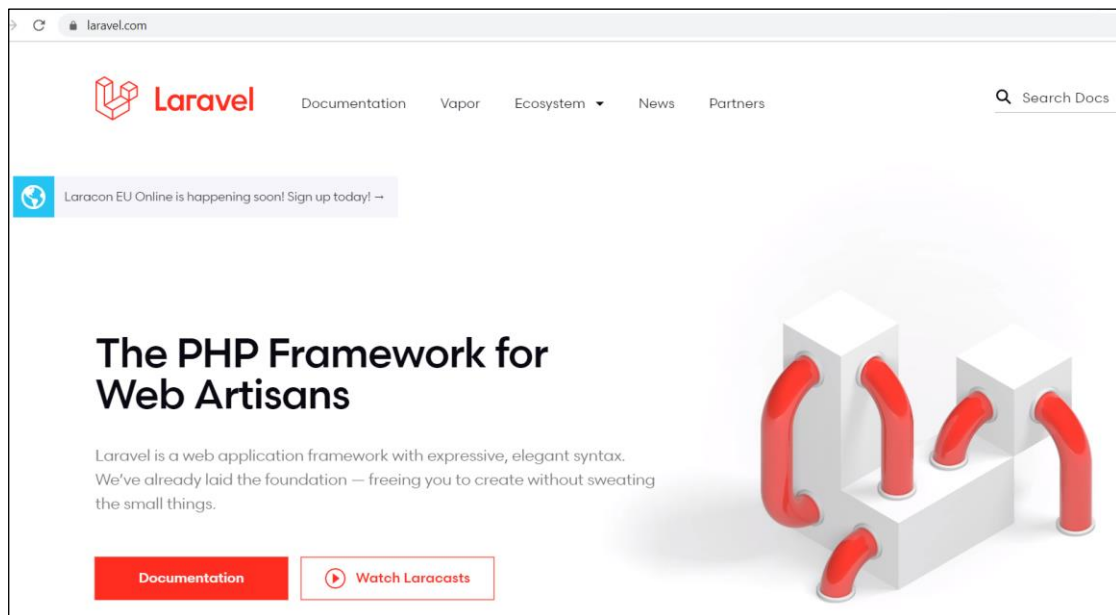
pengerjaan bisnis proses. dalam praktiknya, pola pengembangan MVC pada Laravel didukung oleh satu komponen yang tidak kalah penting, yaitu Router yang bertugas untuk mengarahkan request yang dikirimkan melalui browser.

Menurut sejarahnya, Laravel pertama kali diluncurkan sudah sejak bulan Juni 2011. Tidak butuh waktu yang lama framework yang dikembangkan oleh Taylor Otwell ini bisa menjadi framework terpopuler di dunia, tepatnya bulan Desember 2013. Laravel menjadi framework yang mendapatkan bintang paling banyak pada social media untuk para developer yaitu Github. Bahkan sekarang ini Laravel juga menjadi salah satu framework PHP yang populer di Indonesia. Keunggulan laraval yaitu (1) mudah dan dokumentasi lengkap, (2) bersifat open-source, (3) fitur lengkap, (4) menggunakan arsitektur MVC (Model View Controller), (5) komunitas yang besar. Beberapa alasan menggunakan Laravel yaitu (1) template yang ringan, (2) memiliki banyak library object-oriented, (3) mendukung MVC, (4) tersedia tool Artisan, (5) modul independent, dan (6) menggunakan unit testing.

Framework Laravel mendukung webserver seperti apache dan nginx. Pastikan sudah menginstall XAMPP atau menginstall PHP manual, untuk sistem operasi Ubuntu bisa dilakukan instalasi secara individu Apache, PHP, MySQL. Berikut persyaratan yang konfigurasi PHP yang harus disiapkan:

- a. PHP $\geq$ 7.2.5
- b. BCMath PHP Extension
- c. Ctype PHP Extension
- d. Fileinfo PHP extension
- e. JSON PHP Extension
- f. Mbstring PHP Extension
- g. OpenSSL PHP Extension
- h. PDO PHP Extension
- i. Tokenizer PHP Extension
- j. XML PHP Extension

Laravel menggunakan Composer untuk mengelola dependensinya, oleh karena itu harus terpasang pada komputer.



Gambar 2.15 Website Resmi Laravel

Ada beberapa cara yang digunakan untuk melakukan instalasi Laravel, salah satunya adalah menggunakan Composer. Instalasi menggunakan composer adalah cara yang paling banyak digunakan oleh *developer* PHP. Perintah composer untuk instalasi Laravel adalah:

`composer create-project --prefer-dist laravel/laravel blog`

penjelsan dari perintah adalah:

- composer adalah perintah untuk menjalankan Composer
- create-project** adalah perintah untuk membuat *project* dari *library* yang sudah ada
- prefer-dist** adalah *flag* untuk mengutamakan peng-*install*-an melalui folder *dist* jika ada
- laravel/laravel** adalah nama dari *package* yang akan digunakan sebagai sumber kode program
- blog** adalah nama dari *project* yang akan buat.

Untuk meng-install Laravel dengan versi yang lebih spesifik, bisa dilakukan dengan perintah dibawah ini.

```
composer create-project --prefer-dist laravel/laravel blog "5.8.*"
```

Contoh diatas adalah perintah untuk meng-install Laravel versi 5.8.*. Tanda * pada versi diatas menandakan bahwa menggunakan versi patch paling terakhir pada versi 5.8. Untuk pengembangan secara local dan ingin menggunakan *PHP's build-in development server*

sebagai server di aplikasi, maka bisa menggunakan perintah `Artisan serve`. Perintah tersebut akan menjalankan server web melalui <http://localhost:8000>.

Laravel memiliki banyak sekali fitur yang sangat membantu dalam membangun sebuah app dengan cepat, seperti facade, collection, migration, eloquent ORM, blade dan lain sebagainya. Laravel memiliki Eloquent ORM yang memungkinkan berinteraksi dengan database tanpa harus menulis SQL. ORM (*Object-Relational Mapping*) itu sendiri tidak terikat dengan Laravel dan Eloquent. ORM juga tidak hanya ada pada Laravel. ORM merupakan sebuah teknik di pemrograman yang memungkinkan berinteraksi dengan database menggunakan paradigma *object-oriented*. Mendapatkan data semua pengguna di tabel dengan nama `users`, biasanya menggunakan perintah SQL: `SELECT * FROM `users``. Akan lebih mudah bila menggunakan ORM di Laravel, yaitu Eloquent menjadi: `App\User::all()`. intinya, tidak perlu lagi menggunakan SQL, tapi gunakan *method-method* yang tersedia di *library* ORM yang digunakan. secara bawaan, *Eloquent model* terdapat di dalam folder `app/`.

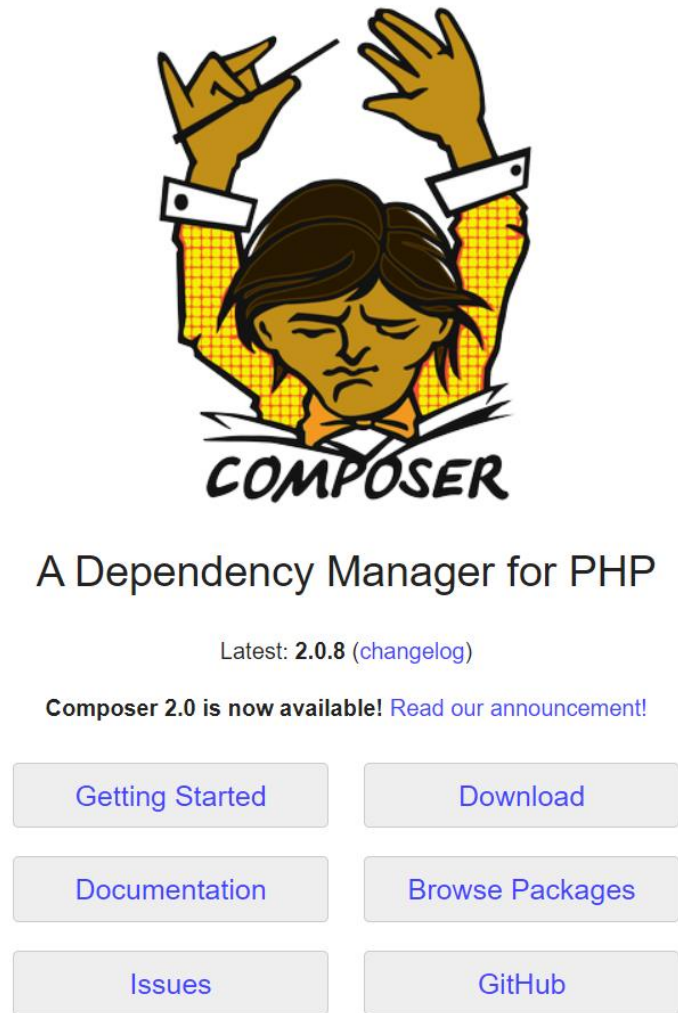
2.5 Composer

Composer adalah alat untuk manajemen ketergantungan (*dependency management*) di PHP. *Dependency management* merupakan sebuah *tools* yang memecahkan masalah seperti:

- a. terdapat *project* yang membutuhkan *library*, misalnya *library* untuk Export data ke Excel yaitu PHPExcel dan DomPDF untuk export data ke PDF.
- b. *library* tersebut juga membutuhkan *library* lainnya. Dalam kasus ini, PHPExcel membutuhkan *library ext-xml* dan DomPDF membutuhkan *phenx/php-font-lib*. Kasus seperti ini disebut ketergantungan (*dependency*) Jadi, bisa disebut PHPExcel memiliki ketergantungan pada *ext-xml*.
- c. Sulit dan ribet untuk *download* dan *install* semua *library* tersebut beserta dependensinya secara manual.

Cara kerja Composer adalah tulis *library* apa saja yang dibutuhkan, kemudian Composer yang akan mencari versi *library* yang sesuai dan melakukan *download* dan instalasi secara otomatis. Dengan composer, tidak perlu repot *download library* PHP manual satu-persatu. Begitupun dengan proses *update library*, tidak perlu *update* satu-persatu, cukup ubah satu file (*composer.json*), jalankan perintah `composer update` untuk mengupdate semua *library*. Ketika akan menginstal sebuah *library*, maka Composer otomatis mengetahui apakah *library* tersebut membutuhkan *library* lain atau tidak. Jika

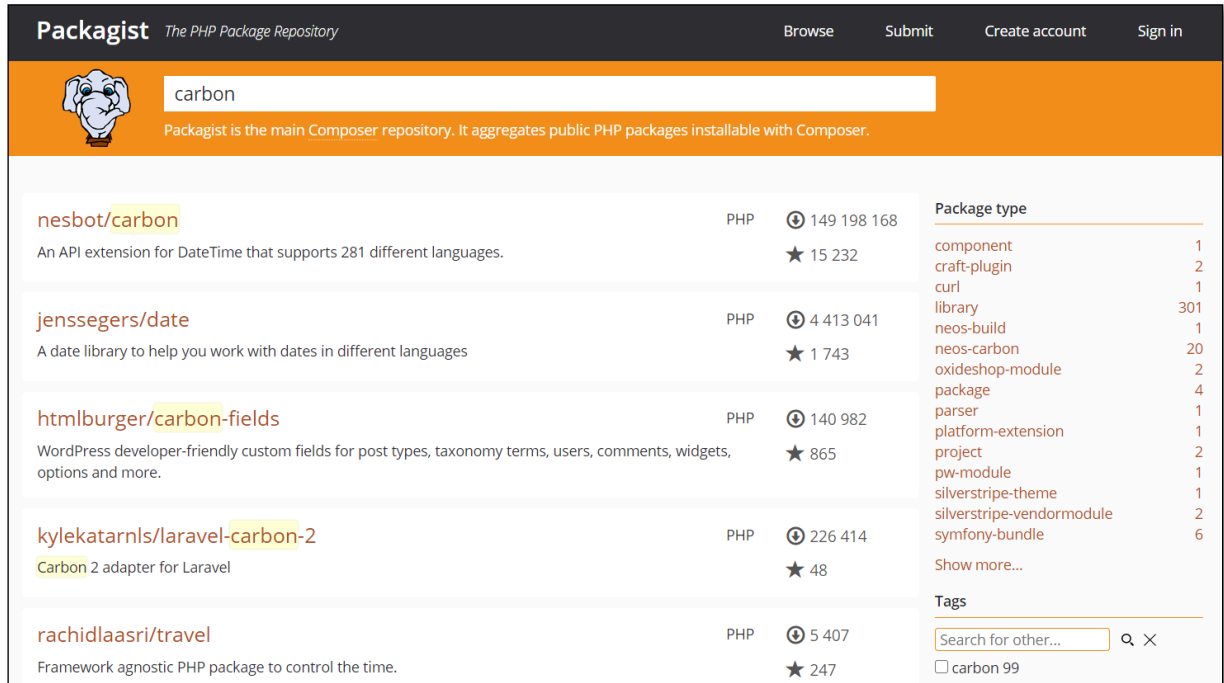
membutuhkan *library* lain maka Composer akan secara otomatis melakukan *download* juga. Bukan hanya itu, Composer juga akan mengecek apakah versi PHP dan ekstensi PHP pada server sudah sesuai dengan yang dibutuhkan oleh *library* tersebut. Kemudian yang terakhir juga bisa menyesuaikan versi pada *library* yang akan dipasang.



Gambar 2.16 Website Resmi Composer

Cara menginstall composer di sistem berbasis Unix (Linux dan OSX) dan Windows cukup berbeda. Untuk pengguna sistem operasi Windows, caranya: buka browser, download installer di <https://getcomposer.org/Composer-Setup.exe> kemudian lakukan instalasi dengan mengikuti tahapan proses yang tersedia. Setelah menginstall composer, buka terminal/command prompt di Windows (Tekan Window+R ketik cmd), lalu jalankan perintah: `composer -V`. Jika muncul tampilan versi Composer, artinya instalasi telah berhasil. Sebagai contoh kasus, dibutuhkan *library Carbon* yang digunakan untuk memanipulasi objek *DateTime*. Untuk mendapatkan package dengan composer, buka alamat website

<https://packagist.org/>, kemudian ketikkan “Carbon” dikotak pencarian. Klik pada hasil pertama pencarian yang muncul yaitu “nestbot/carbon” sehingga ditampilkan laman detail package Carbon.



Gambar 2.17 Packagist

Beberapa informasi yang ditampilkan pada laman detail antara lain:

- Deskripsi package
- Total install
- Website dokumentasi dan hosting source code
- Data pembuat Package
- Update Package terakhir
- dependency

Pada laman detail *Package* tersedia panduan untuk melakukan instalasi baik melalui Composer maupun tidak. Untuk melakukan instalasi *package*, buat sebuah folder misalnya di D:\htdocs\coba-carbon. Pada folder ini, buatlah file baru bernama composer.json dengan isi sebagai berikut:

```
{
    "require": {
        "nesbot/carbon": "^2.16"
    }
}
```

}

Packagist The PHP Package Repository

Browse Submit Create account Sign in

Search packages...

nesbot/carbon

composer require nesbot/carbon

An API extension for DateTime that supports 281 different languages.

Maintainers

Details

[github.com/briannesbitt/Carbon](#)
[Homepage](#)
[Source](#)
[Issues](#)

Fund package maintenance!
[Open Collective](#)
[Tidelift](#)

Installs: 149 319 145
Dependents: 3 120
Suggesters: 44
Security: 0
Stars: 14 885
Watchers: 243
Forks: 1 151
Open Issues: 5

2.42.0 2020-11-28 14:25 UTC

requires	requires (dev)	suggests
- php: ^7.1.8 ^8.0 - ext-json: * - symfony/polyfill-mbstring: ^1.0 - symfony/translation: ^3.4 ^4.0 ^5.0	- doctrine/orm: ^2.7 - friendsofphp/php-cs-fixer: ^2.14 ^3.0 - kylekatarnis/multi-tester: ^2.0 - phpmd/phpmd: ^2.9 - phpstan/extension-installer: ^1.0	None

3.x-dev
dev-master / 2.x-dev

2.42.0
2.41.5
2.41.4
2.41.3
2.41.2
2.41.1

Gambar 2.18 Detail Package Carbon

Kemudian untuk melakukan instalasi ketikkan perintah: Composer install sehingga proses instalasi berjalan. Tunggu sampai proses instalasi selesai. Jika berhasil unduh Carbon, akan ada beberapa file dan folder baru di folder coba-carbon didalam direktori vendor. Penjelasan file dan folder tersebut adalah:

- composer.json, file yang buat, berisi dependensi library dari project.
- composer.lock, file ini mencatat versi package yang saat ini terinstal.
- vendor, folder ini berisi package yang telah kita install. Setiap package yang ditulis di bagian require akan di download ke folder ini.
- vendor/autoload.php, berfungsi memanggil autoloader dari composer.

Salah satu keunggulan dari composer adalah autoloader. Fitur ini berfungsi untuk memanggil class yang sesuai ketika membutuhkan class dari suatu library. Jadi, jika banyak

library yang di-install, maka hanya cukup menulis require untuk file vendor/autoload.php. Dan, autoload ini cukup cerdas dengan hanya me-load class yang di butuhkan.

2.6 UUID

UUID (*Universally Unique Identifier*) adalah kumpulan dari 32 karakter yang dibuat secara acak sehingga kecil kemungkinan menghasilkan nilai yang sama. Umumnya, pengembang aplikasi menggunakan *auto_increment* untuk dijadikan *primary key* pada saat merancang database, yang secara otomatis berisi nilai angka urut 1, 2 3, dan seterusnya. Penggunaan *auto_increment* dipilih karena mudah diterapkan dan memiliki performa yang baik. Beberapa masalah muncul ketika menggunakan *auto_increment* diantaranya:

1. Karena berisi angka, maka pengguna aplikasi dapat dengan mudah melihat data lain dengan hanya mengganti nilai angkanya. Lebih fatal lagi, data dapat ditampilkan dalam jumlah banyak melalui mekanisme perulangan (*looping*).
2. Untuk aplikasi yang tersimpan terdistribusi (banyak server), penggunaan *auto_increment* sering menyebabkan konflik karena mendapatkan nilai yang sama.
3. Untuk transaksi yang banyak dan cepat, *auto_increment* kurang efektif digunakan.

Permasalahan yang terjadi ketika menggunakan *auto_increment* seperti diatas dapat diselesaikan dengan menggunakan UUID.

UUID terdiri dari 36 karakter, kombinasi huruf dan angka disertai tanda dash (-). Contoh UUID adalah f5b0e56c-0ee2-45f4-8a61-d5c91be1641d, yang secara otomatis dibangkitkan (*generate*) dengan menggunakan fungsi khusus yang tersedia pada DBMS (*Database Management System*) atau melalui *UUID Generator*. Penggunaan UUID tidak harus menghilangkan *primary key* yang bersifat *auto_increment*, akan tetapi dapat digunakan secara bersamaan. Tipe data untuk kolom (field) yang diisi dengan UUID adalah **char** atau **varchar** dengan panjang (*length*) 36.

Ada banyak *package* untuk membuat UUID, dan yang paling sering digunakan untuk PHP adalah **ramsey/uuid** yang dikembangkan oleh Ben Ramsey. proses instalasi dapat dilakukan dengan mudah menggunakan **composer**, kemudian dilakukan *import package* tersebut kemudian menggunakan *method* yang tersedia untuk melakukan *generate* UUID. Proses instalasi **ramsey/uuid** menggunakan composer adalah sebagai berikut:

```
composer require ramsey/uuid
```

setelah terpasang, langkah selanjutnya adalah menggunakan *package* ramsey/uuid dalam program untuk melakukan *generate* UUID seperti potongan kode berikut.

```
use Ramsey\Uuid\Uuid;
```

```

use Ramsey\Uuid\Exception\UnsatisfiedDependencyException;

try {
    echo $this->attributes['uuid'] = Uuid::uuid4()->toString();
} catch (UnsatisfiedDependencyException $e) {
    echo $e->getMessage();
}

```

Potongan kode diatas akan melakukan *generate* UUID dan menampilkannya di *browser*. Penggunaan UUID dalam pengembangan aplikasi pusat karir SMK juga menggunakan **ramsey/uuid**. Setelah dilakukan proses instalasi ramsey/uuid, maka sebelum digunakan harus dipanggil terlebih dahulu.

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\MasPeriodeModel;
7  use App\Http\Requests;
8  use App\Http\Controllers\Controller;
9  use Illuminate\Contracts\Auth\Guard;
10 use DB;
11 use Session;
12 use Validator;
13 use Input;
14 use Ramsey\Uuid\Uuid;
15 use Crypt;
16 use Response;
17 //use Excel;
18 use Spreadsheet_Excel_Reader;
19

```

Gambar 2.19 Menggunakan Package UUID di Controller

Kemudian UUID tersebut digunakan yakni dengan melakukan membangkitkan kode UUID. Potongan kode programnya seperti terlihat pada gambar 2.20 berikut.

```

}
else{
    $idbkk = Session::get('idbkk');
}

$uuid4 = Uuid::uuid4();
$insert = array(
    'nama'=>$nama,
    'nisp'=>$nisp,
    'idjurusan'=>$idjurusan,
    'tgl_lahir'=>$tgl_lahir,
    'tempat_lahir'=>$tempat_lahir,
    'alamat'=>$alamat,
    'telp'=> $telp,
    'hp'=> $hp,
    'email'=> $email,
    'nik' => $nik,
    'jk' => $jk,
    'idagama' => $idagama,
    'hobi' => $hobi,
    'alamat_tinggal' => $alamat_tinggal,
    'idkota' => $idkota,
    'idkelas' => $idkelas,
    'klaster' => $klaster,
    'idbkk' => $idbkk,
    'created_at'=>date ('Y-m-d H:i:s'),
    'userid_created'=>Session::get('userid'),
    'id'=>$uuid4
);

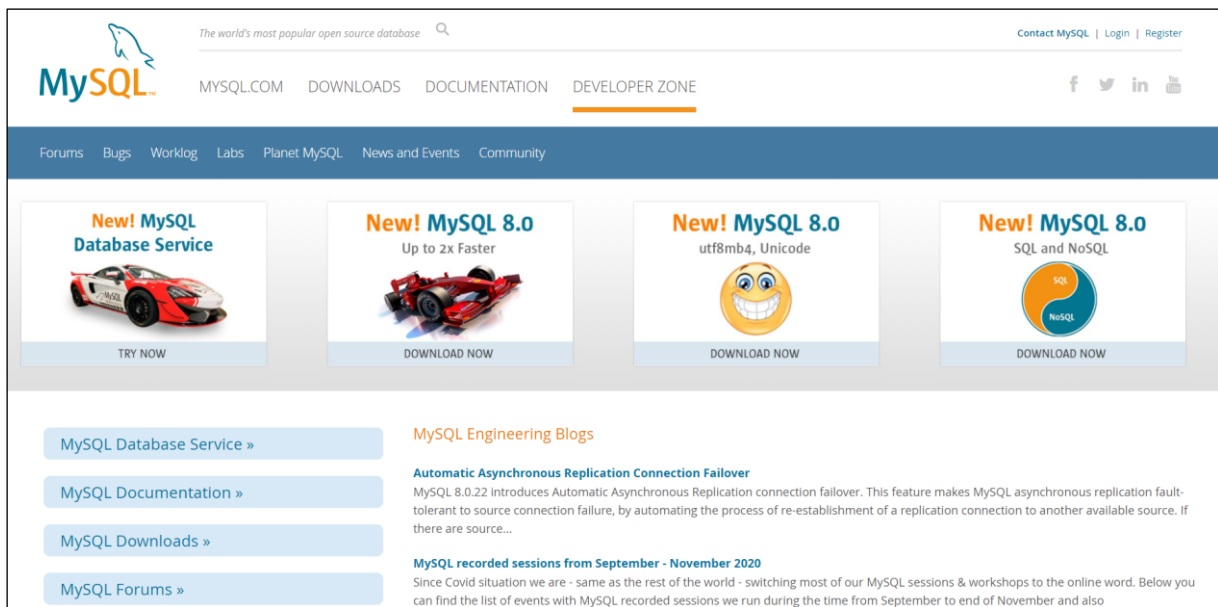
```

Gambar 2.20 Generate kode UUID

2.7 MySQL

MySQL merupakan DBMS (*Database Management System*) untuk pengelolaan database relasional yang bersifat *open-source*. MySQL berfungsi untuk mempermudah manajemen database. Pada tanggal 16 Januari 2008 Sun Microsystems, Inc mengumumkan aksi korporasi akuisisi terhadap MySQL AB sehingga menjadikan Sun sebagai salah satu perusahaan dengan produk platform open source terbesar seperti Java, OpenSolaris dan akhirnya MySQL. Saat ini hak kepemilikan aplikasi MySQL telah dibeli oleh perusahaan Oracle Corporation. Akan tetapi, Oracle membagikan teknologi MySQL dengan gratis melalui lisensi GPLv2.

MySQL dapat diunduh melalui website resmi di <https://www.mysql.com>. Versi terbaru adalah 8.0.22 dengan ukuran file 405.2 MB. MySQL tersedia untuk banyak sistem operasi, oleh karena itu ketika melakukan unduh MySQL harus dipilih terlebih dahulu sistem operasi yang digunakan. Setelah diunduh, berikutnya adalah melakukan instalasi MySQL yang dapat dilakukan dengan mudah. Selain MySQL, melalui website resmi (gambar 2.21) juga dapat diunduh perangkat lunak lain untuk meningkatkan kegunaan dari MySQL.



Gambar 2.21 Website Resmi MySQL

Perangkat lunak lain yang berfungsi sebagai pelengkap MySQL dan dapat diunduh antara lain:

- MySQL Workbench
- MySQL Shell
- MySQL Cluster
- MySQL Router
- MySQL for Visual Studi
- Connector C++, Java, .NET, Node.js, ODBC dan Python

Jika dibandingkan dengan DBMS yang lain, kelebihan MySQL adalah (1) Dapat terintegrasi dengan bahasa pemrograman lain, (2) Resource RAM yang kecil, (3) Support Multi User, (4) Open Source, (5) Struktur tabel yang fleksibel, dan 6. Keamanan terjamin. Kekurangan MySQL yaitu (1) Kurang cocok untuk aplikasi game dan mobile, (2) Kurang bisa diandalkan dalam pengelolaan database dengan kapasitas besar, dan (3) Technical Support kurang begitu bagus.

Perintah SQL yang dapat dijalankan di MySQL dapat dibagi menjadi empat (4) yaitu DDL (*Data Definiton Language*), DML (*Data Manipulation Language*), DCL (*Data Control Language*), dan TCL (*Transaction Control Language*). Perintah yang termasuk DDL yaitu create, alter, drop, truncate, comment, dan rename. Perintah yang termasuk DML yaitu select, insert, update, dlete, merge, call, explain plan, lock table. Perintah yang termasuk DCL yaitu grant dan revoke, sedangkan perintah yang termasuk TCL adalah commit, rollback, savepoint, dan set transaction.

2.8 HTML dan CSS

HTML (*Hypertext Mark Up Language*) merupakan bahasa *markup* (penanda) untuk membuat website untuk menampilkan informasi di Internet. Informasi yang ditampilkan dapat berupa teks, audio, video, dan gambar. Fungsi dasar dari HTML adalah untuk membuat struktur dasar sebuah website, dengan cara menandai (*Tag*) bagian mana (elemen) yang harus dicetak tebal, miring, berwarna dan lain-lain.

Merupakan bahasa markup dan bukan bahasa pemrograman karena tidak memiliki variabel, function, struktur kontrol, pengkondisian. Markup bisa juga disebut *Tag* yang berfungsi untuk memberitahu browser, bagaimana sebuah teks akan ditampilkan. Sebagai contoh tag HTML untuk membuat teks untuk dijadikan sebagai heading, title, paragraph. Penulisan tag diawali dengan < dan diakhiri dengan >, kebanyakan tag terdiri dari satu pasang, yakni tag pembuka dan tag penutup, namun ada juga tag yang tidak memiliki pasangan.

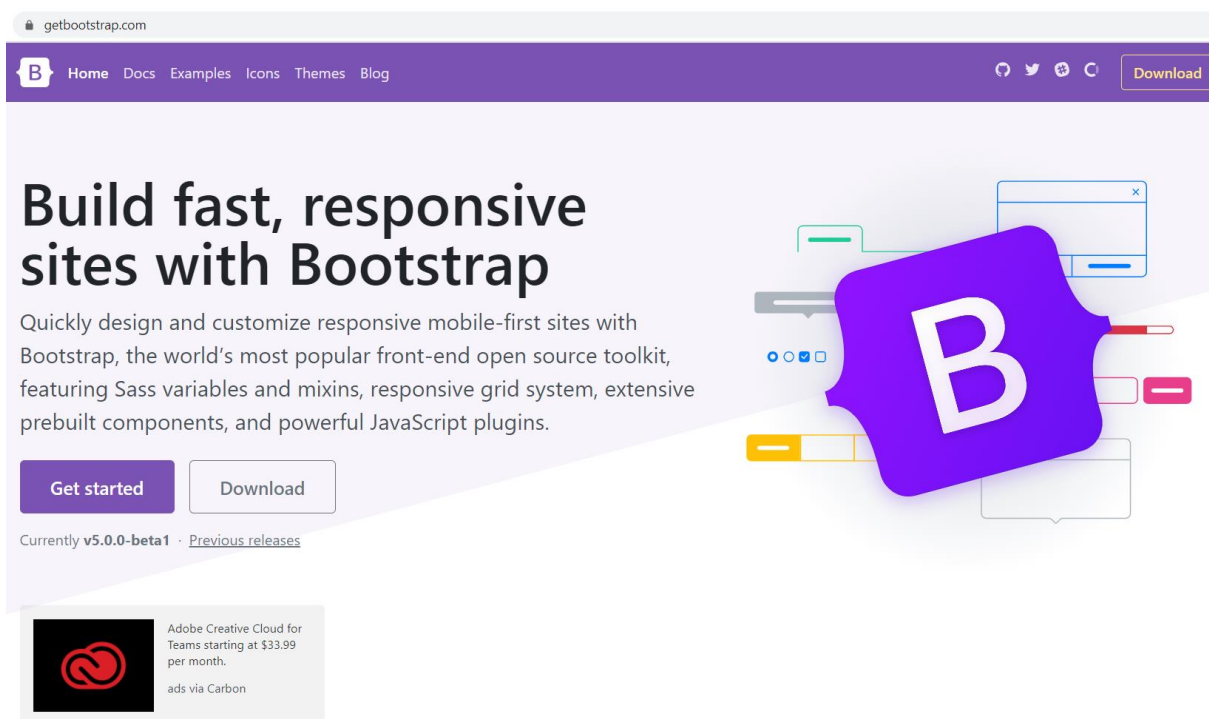
Oleh karena itu, kedua struktur ini harus bekerja bersama-sama untuk melakukan tindakan tertentu. Dengan CSS, Anda bisa menghadirkan tampilan yang seragam di banyak halaman website. Semua ini berkat keberadaan fungsinya. Anda pun tak perlu lagi menghabiskan waktu berjam-jam lamanya hanya untuk mengubah, misalnya, lebar desain. Jika menggunakan CSS, yang perlu dilakukan adalah merujuk atau mengarahkan halaman web ke file CSS. Nantinya semua perubahan dapat dilakukan di satu file saja. Fungsi CSS tidak terbatas hanya di gaya teks saja, tetapi juga dapat digunakan untuk membuat aspek lainnya di halaman web, seperti tampilan tabel dan gambar.

Sedangkan file *bootstrap.js* merupakan bagian inti dari Bootstrap. Di dalamnya terdapat file JavaScript yang bertanggung jawab atas interaktivitas website. Developer menggunakan jQuery untuk menghemat waktu karena tidak perlu menulis sintaks JavaScript berulang kali. jQuery sendiri adalah open source yang populer dan juga library JavaScript lintas platform yang memperbolehkan user-nya untuk menambah berbagai fungsionalitas ke website.

2.9 Bootstrap

Merupakan salah satu kerangka kerja (*framework*) paling populer kalangan *web developer*, yang digunakan untuk mempermudah dan mempercepat pembuatan halaman website. Disebut mempercepat karena bootstrap sudah menyediakan css dan javascript yang siap pakai dan mudah dikembangkan. Bootstrap telah menyediakan kumpulan komponen dasar yang siap pakai untuk pengembangan *front-end*. Bootstrap dapat diakses

melalui alamat URL <https://getbootstrap.com/>, dan versi terbaru pada saat panduan teknis ini dibuat adalah versi 5.0.0-beta1 (gambar 2.22).



Gambar 2.22 Website Resmi Bootstrap

Dalam pengembangan *front-end*, bootstrap menggunakan konsep *Grid System* yang terdiri dari 12 grid. Grid berfungsi untuk mengatur ukuran lebar dari masing-masing komponen dan terkait fitur *responsive* tata letak tampilan. Konsep *grid system* pada bootstrap dapat dilihat pada gambar 2.23 berikut.

Span 12											
Span 6						Span 6					
span 4				span 4				span 4			
Span 4				Span 8							
Span 3			Span 3			Span 3			Span 3		
Span 2		Span 2		Span 2		Span 2		Span 2		Span 2	
Span 1	Span 1	Span 1	Span 1	Span 1	Span 1	Span 1	Span 1	Span 1	Span 1	Span 1	Span 1

Gambar 2.23 Grid System pada Bootstrap

2.10 JavaScript

JavaScript (JS) adalah bahasa pemrograman yang ringan, *interpreted*, bersifat dinamis, dan dapat bersifat ke arah *object oriented programming* atau *functional programming*, yang dirintis dan didesain oleh Brendan Eich. Pada perkembangannya JavaScript mulai distandarisasi oleh organisasi bernama ECMA (European Computer Manufacturers Association) yang setiap standarnya bisa disebut dengan penamaan EcmaScript dan tahun rilis standarnya, misalnya ES2015, ES2016, ES2017, dan seterusnya. Karena telah distandarisasi oleh ECMA, maka implementasi JavaScript di dalam teknologi web bisa merata dan diperluas ke implementasi di luar browser. Misalnya dengan munculnya *runtime* JavaScript yang terkenal dengan nama Node JS yang membuat JavaScript bisa berjalan di lingkungan server, desktop, IoT, blockchain, machine learning, dan masih banyak lagi.

JavaScript adalah bahasa pemrograman web yang bersifat *Client Side Programming Language*. *Client Side Programming Language* adalah tipe bahasa pemrograman yang pemrosesannya dilakukan oleh *client* atau seluruh prosesnya berjalan pada sisi pengguna bukan server. Karena diolah pada sisi client, JavaScript sangat bergantung pada pengaturan dan kemampuan browser ketika melakukan sebuah proses (*compiling* atau *rendering* pada DOM). Bahkan pengguna dapat sepenuhnya tidak mengizinkan JavaScript berjalan pada browser dengan menonaktifkan dukungan JavaScript pada browser. Contoh Javascript Engine yang populer adalah V8 Engine dari Google, ia dipakai oleh Node.js dan browser Chrome. ada juga SpiderMonkey, javascript engine yang dipakai oleh Mozilla termasuk di Firefox dan produk-produk Mozilla lainnya.

JavaScript saat ini merupakan salah satu bahasa pemrograman yang banyak dipakai dalam pembuatan aplikasi web dan aplikasi lainnya. Beberapa orang banyak yang menyebutkan JavaScript adalah *lingua franca of the web, multi purpose versatile programming language*. Beberapa framework dan library JS pun sudah banyak bermunculan dengan tujuan fungsinya masing-masing, misalkan React, Angular, Vue, Svelte untuk membuat halaman web bagian *front end*, Express, Adonis, Nest, Hapi untuk membuat *backend service* aplikasi web dan mobile, TensorFlowJS, BrainJS, StdlibJS untuk machine learning, Electron JS dan NW.js yang terkenal untuk pembuatan aplikasi desktop, Phaser dan Construct untuk game engine, React Native, Ionic, NativeScript, PWA untuk pembuatan aplikasi mobile. JavaScript saat ini biasanya menggunakan standar penulisan berdasarkan ES2015 atau ES6.

BAB 3

ANALISIS KEBUTUHAN DAN DESAIN SISTEM

Pada bab ini, dibahas desain sistem aplikasi pusat karir SMK yang meliputi desain proses, desain database, desain antarmuka, desain level akses, dan desain teknologi.

3.1 Analisis kebutuhan

Kebutuhan perangkat lunak adalah kondisi, kriteria, syarat atau kemampuan yang harus dimiliki oleh perangkat lunak untuk memenuhi apa yang disyaratkan atau diinginkan pemakai. Terdapat tiga jenis kebutuhan perangkat lunak yaitu kebutuhan fungsional (*functional requirement*), kebutuhan antarmuka (*interface requirement*), dan kebutuhan unjuk kerja (*performace requirement*). Penjelasan dari masing-masing kebutuhan perangkat lunak adalah sebagai berikut.

1. Kebutuhan Fungsional, disebut juga kebutuhan operasional yaitu kebutuhan yang berkaitan dengan fungsi atau proses transformasi yang harus mampu dikerjakan oleh perangkat lunak. Contohnya, (1) perangkat lunak harus mampu menyimpan data siswa, (2) perangkat lunak harus mampu membuat laporan siswa yang diterima kerja, (3) perangkat lunak harus mampu menampilkan informasi rekapitulasi data.
2. Kebutuhan Antarmuka, kebutuhan yang menghubungkan perangkat lunak dengan elemen perangkat keras, perangkat lunak, atau basis data. Contoh: akses ke basis data menggunakan MySQLi Driver.
3. Kebutuhan Unjuk Kerja, kebutuhan yang menetapkan karakteristik unjuk kerja yang harus dimiliki oleh perangkat lunak, misalnya: kecepatan, ketepatan, frekuensi.

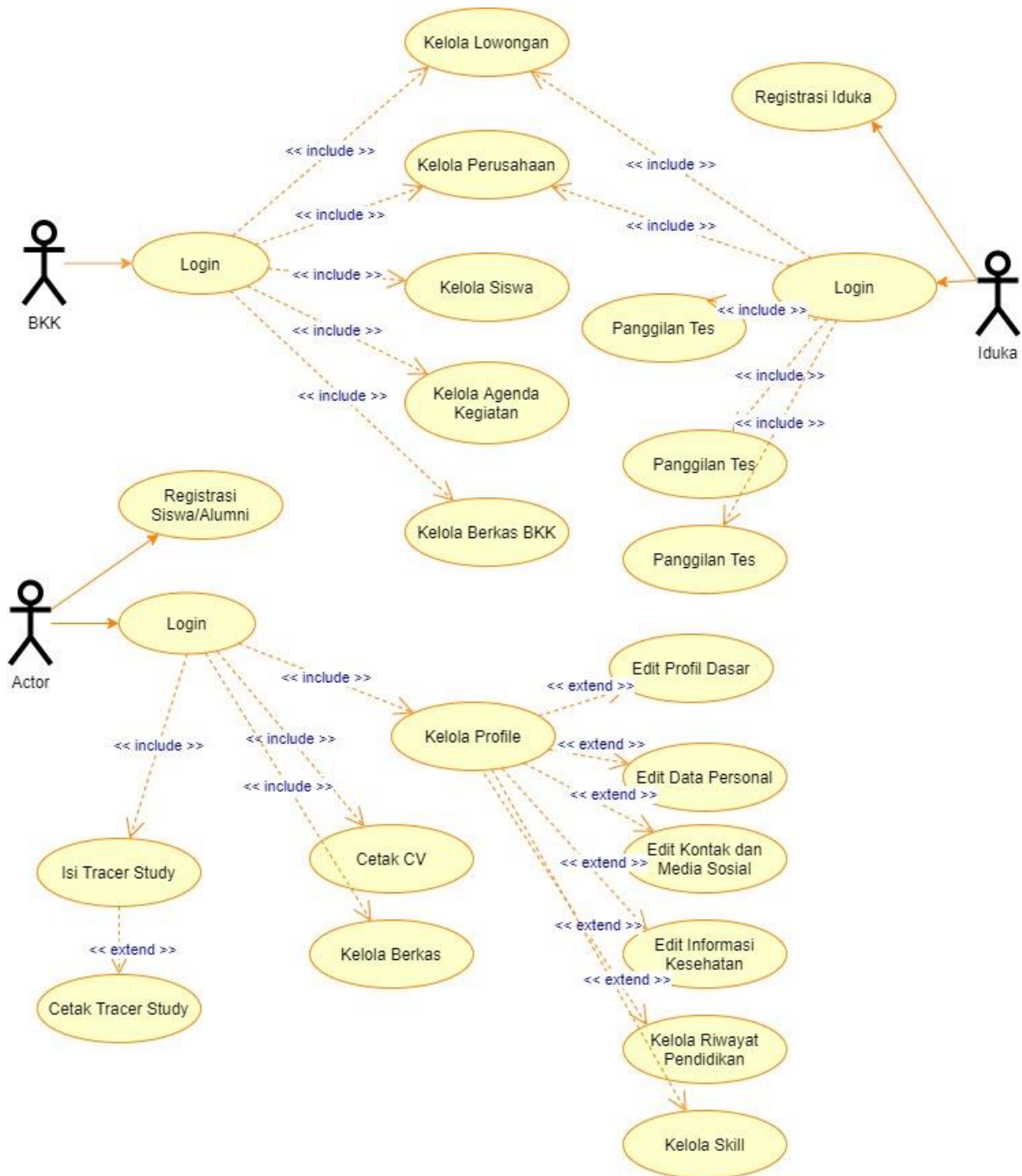
Analisis Kebutuhan Perangkat Lunak merupakan aktifitas awal dari siklus hidup pengembangan Perangkat Lunak. Analisis kebutuhan merupakan Proses mempelajari kebutuhan pemakai untuk mendapatkan definisi kebutuhan sistem atau perangkat lunak. Tujuan analisis kebutuhan adalah:

1. Memahami masalah secara menyeluruh (komprehensif) yang ada pada perangkat lunak yang akan dikembangkan seperti ruang lingkup produk perangkat lunak (*product space*) dan pemakai yang akan menggunakannya.
2. Mendefinisikan apa yang harus dikerjakan oleh perangkat lunak untuk memenuhi keinginan pelanggan

Secara teknis pelaksanaan pekerjaan analisis kebutuhan perangkat lunak pada dasarnya terdiri dari urutan aktivitas:

1. **Mempelajari dan memahami persoalan**, mempelajari masalah yang ada pada perangkat lunak yang dikembangkan, sehingga dapat ditentukan: siapa pemakai, dimana perangkat lunak digunakan, pekerjaan apa saja yang dibantu oleh perangkat lunak, apa saja cakupan dari pekerjaan tersebut, apa yang menjadi kendala dilihat dari sisi teknologi yang digunakan. Cara yang digunakan adalah wawancara dengan pemakai, observasi atau pengamatan lapangan, kuesioner, mempelajari referensi atau dokumen-dokumen yang digunakan, seperti dokumen hasil analisa dan perancangan perangkat lunak.
2. **Mengidentifikasi kebutuhan pemakai**, tahap identifikasi kebutuhan pemakai (user requirement) pada prakteknya menjadi satu pelaksanaannya dengan pemahaman masalah. Hanya saja substansi yang ditanyakan ada sedikit perbedaan, yaitu (1) fungsi apa yang diinginkan pada perangkat lunak, (2) data atau informasi apa saja yang akan diproses, (3) kelakuan sistem apa yang diharapkan, (4) antarmuka apa yang tersedia (*software interfaces*, *hardware interfaces*, *user interfaces*, dan *communication interfaces*). Untuk menangkap kebutuhan dari pemakai dengan baik, terutama kesamaan persepsi. seorang analis membutuhkan: (1) komunikasi dan brainstorming yang intensif dengan pelanggan, (2) pembuatan prototype perangkat lunak atau screenshot, (3) data atau dokumen yang lengkap.
3. **Mendefinisikan kebutuhan perangkat lunak**, kebutuhan pemakai yang belum terstruktur tersebut akan dianalisis, diklasifikasikan, dan diterjemahkan menjadi kebutuhan fungsional, antarmuka dan unjuk kerja perangkat lunak. Contoh: “data yang dimasukkan oleh bagian penjualan bisa langsung dijurnal” setelah dianalisis, diklasifikasikan dan diterjemahkan, mungkin akan menghasilkan pendefinisian kebutuhan fungsional, antarmuka, dan unjuk kerja. Kemudian kebutuhan tersebut akan dimodelkan atau digambarkan dengan teknik analisis dan alat bantu tertentu.
4. **Membuat dokumen spesifikasi kebutuhan perangkat lunak**, Semua kebutuhan yang telah didefinisikan selanjutnya dibuat dokumentasinya yaitu Spesifikasi Kebutuhan Perangkat Lunak (SKPL) atau *Software Requirement Specification* (SRS).
5. **Mengkaji ulang (review) kebutuhan**, mengkaji ulang (validasi) kebutuhan apakah SKPL sudah konsisten, lengkap, dan sesuai dengan yang diinginkan oleh pemakai.

Gambar 3.1 berikut adalah use case dari pusat karir SMK yang dibuat berdasarkan analisis kebutuhan yang dilakukan di BKK SMK Negeri dikota Surabaya dan Bojonegoro. Terdapat tiga actor yang terlibat yakni perusahaan, siswa/alumni, dan BKK.



Gambar 3.1 Use Case Pusat Karir SMK

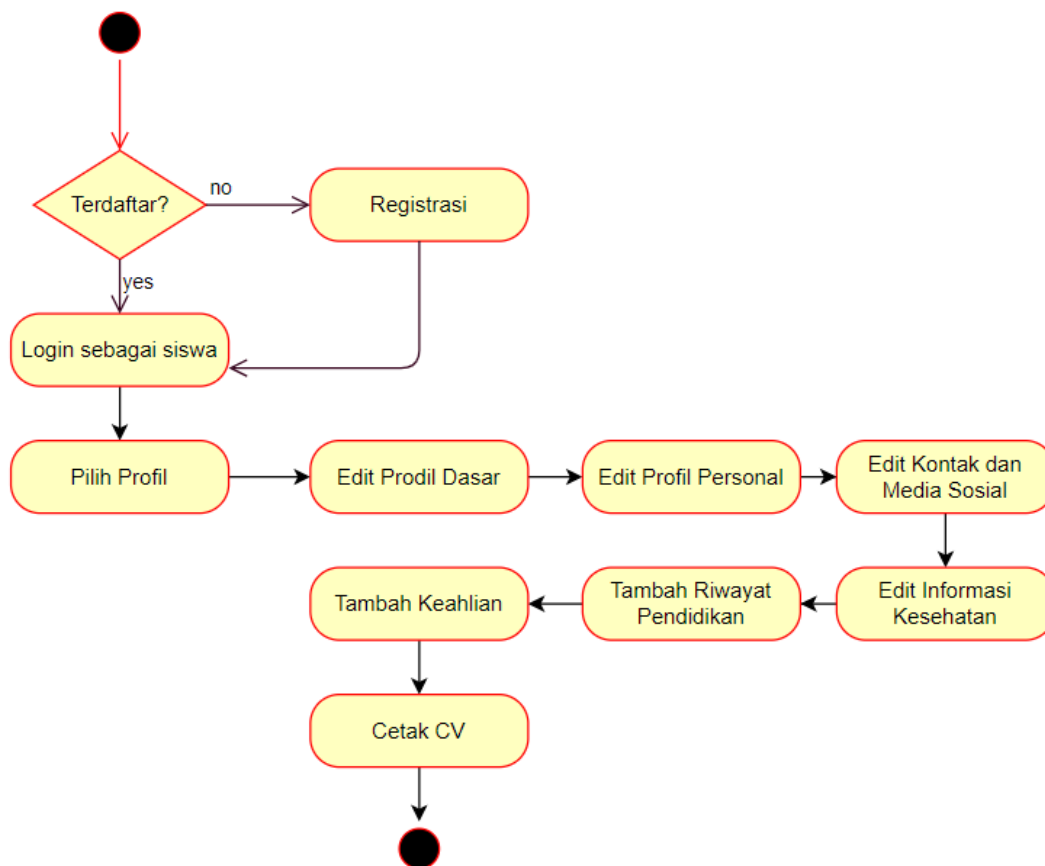
Berdasarkan use case pada gambar 3.1 dapat dijelaskan bahwa BKK dapat melakukan kelola data lowongan kerja, kelola perusahaan, kelola sisw/alumni, kelola agenda kegiatan, dan kelola berkas BKK. Sedangkan actor Iduka dapat melakukan registrasi Iduka,

kelola lowongan, kelola perusahaan, panggilan tes. Actor siswa dapat melakukan login, kelola profil, cetak CV, kelola berkas portofolio, tracer study, dan kelola keahlian.

3.2 Desain Proses

3.2.1 Proses Cetak CV

Salah satu fitur dalam aplikasi pusat karir SMK adalah CV (Curriculum Vitae) secara online. Siswa yang telah terdaftar dapat melengkapi data yang terdiri profil dasar, profil personal, kontak dan media social, informasi kesehatan, riwayat pendidikan, dan keahlian. Siswa yang belum terdaftar dapat melakukan registrasi terlebih dahulu. Alur prosesnya dapat dilihat pada gambar 3.2

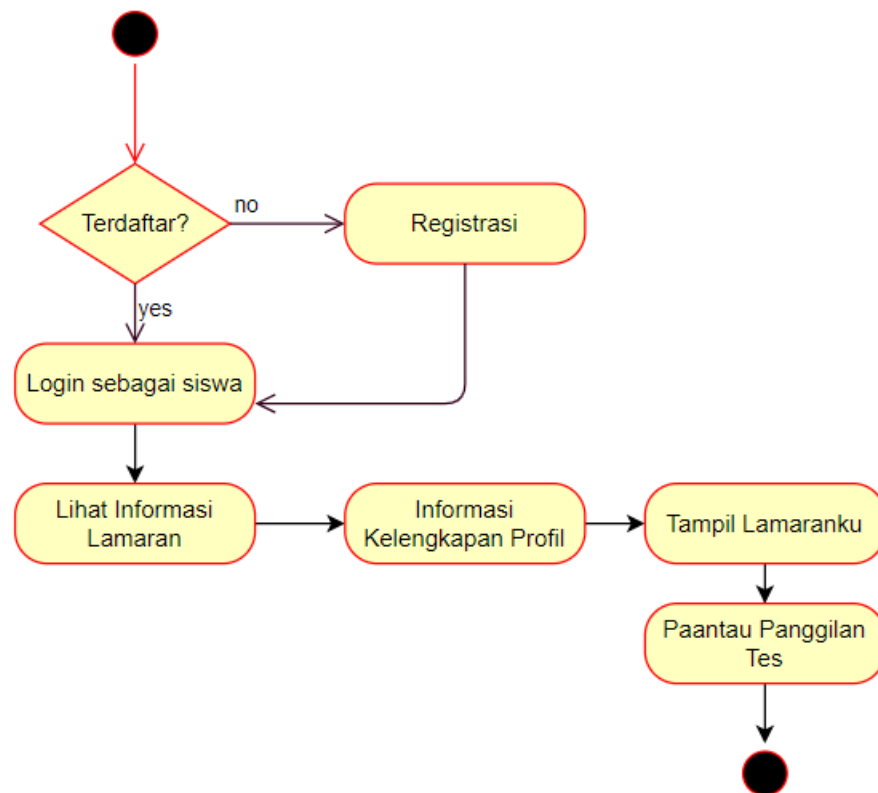


Gambar 3.2 Alur Proses Profil dan CV Siswa

Cetak CV baru dapat dilakukan jika data profil siswa sudah lengkap semua. Setiap data siswa dilengkapi, maka nilai prosentase kelengkapan profil akan bertambah. Tombol cetak CV akan aktif jika nilai kelengkapan profil siswa telah 100%.

3.2.2 Fitur Dasar Siswa

Siswa atau alumni yang telah berhasil login dapat melihat dan memantau lowongan pekerjaan yang telah di-apply, informasi kelengkapan profil, tampil lamaranku, pantau panggilan tes, dan logout.



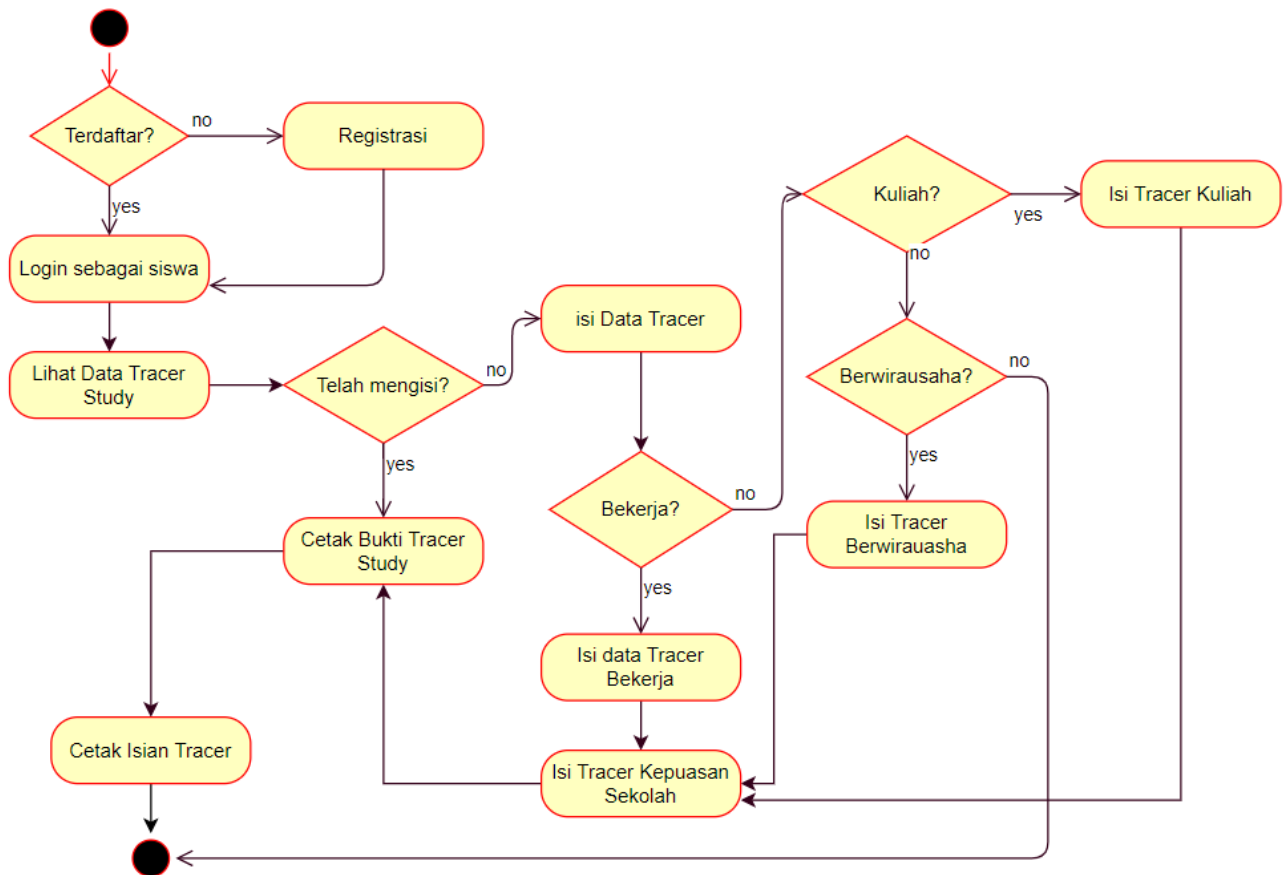
Gambar 3.3 Fitur Akun Siswa/Alumni

Penjelsan dari fitur dasar akun siswa/alumni yaitu:

- Lihat informasi lamaran, pada bagian ini siswa dapat menampilkan informasi jumlah lamaran kerja yang telah di-apply.
- Informasi kelengkapan profil, menampilkan informasi prosentase kelengkapan profil untuk menentukan dapat dicetak atau belum.
- Tampil lamaranku, untuk menampilkan detail lamaran kerja yang telah dilakukan oleh siswa/alumni
- Pantau panggilan tes, untuk menampilkan informasi terkait panggilan tes.

3.2.3 Tracer Study

Yaitu kegiatan pelacakan lulusan yang dilakukan oleh BKK. Alur aplikasi untuk pengguna siswa dalam kegiatan tracer study dapat dilihat pada gambar



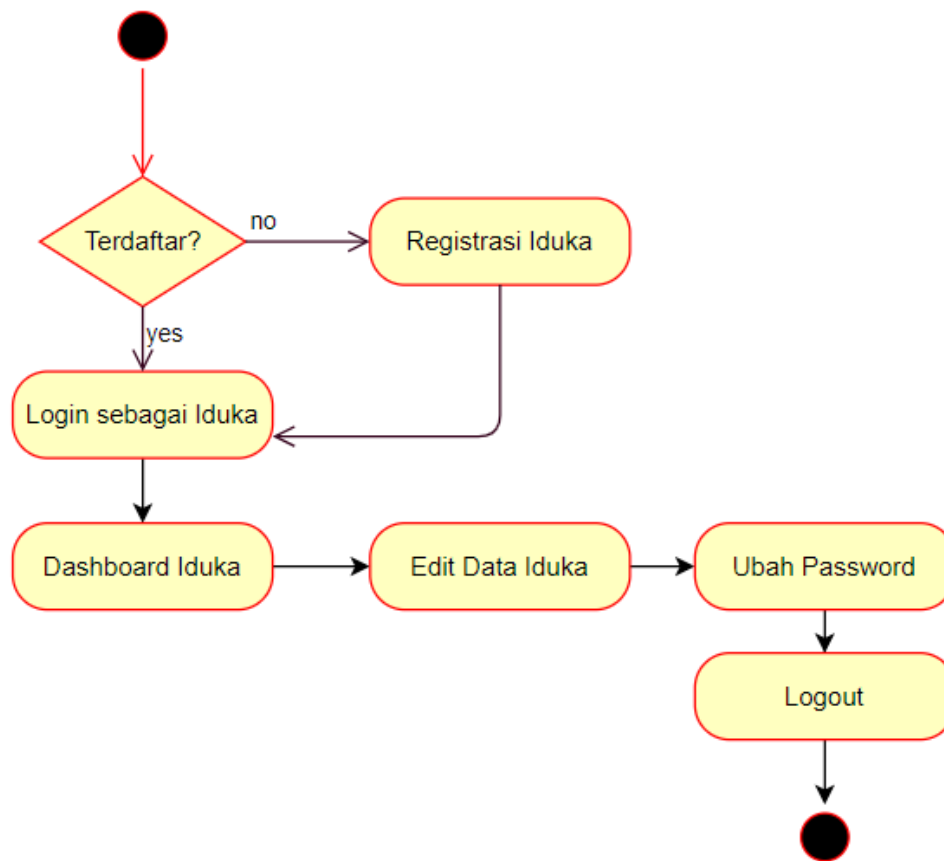
Gambar 3.4 Alur Siswa/Alumni Mengisi Tracer Study

Alur pengisian tracer study oleh siswa/alumni adalah sebagai berikut:

1. Siswa/alumni login ke aplikasi
2. Tampilkan data tracer study
3. Jika belum mengisi, maka lakukan pengisian data tracer study yang meliputi data tracer bekerja, data tracer kuliah, data tracer berwirausaha, dan data kepuasan sekolah.
4. Cetak bukti pengisian tracer study
5. Cetak data hasil pengisian tracer study

3.2.4 Edit data IDUKA

Perusahaan atau IDUKA jika belum terdaftar pada aplikasi dapat melakukan registrasi dengan mengisi data kelengkapan yang tersedia. Jika data diisi dengan lengkap dan dilakukan penyimpanan, maka perusahaan akan mendapat akun yang dapat digunakan untuk login di aplikasi. Alur IDUKA melakukan edit data dan ubah password dapat dilihat pada gambar 3.5.

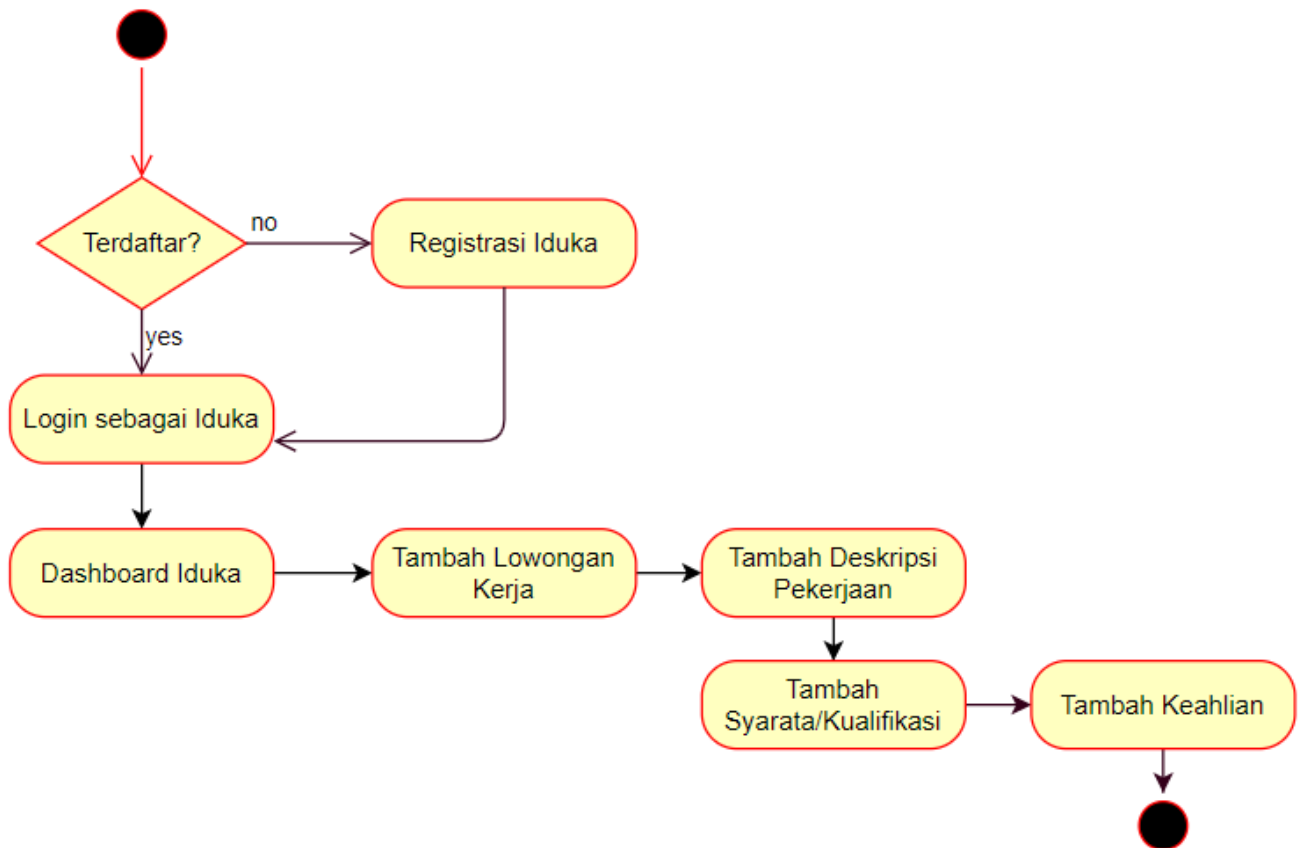


Gambar 3.5 Edit data IDUKA

IDUKA yang telah melakukan registrasi dan mendapat akun melakukan login ke aplikasi. Jika berhasil, maka ditampilkan *dashboard*. Melalui tautan yang ada, IDUKA dapat melakukan perubahan data jika terdapat kesalahan pada data IDUKA atau kurang lengkap. Selain itu, IDUKA juga dapat melakukan perubahan *password*.

3.2.5 IDUKA Tambah Lowongan Pekerjaan

Salah satu hak akses yang dapat dilakukan oleh IDUKA adalah membuka atau *publish* lowongan pekerjaan. Alur menambah data lowongan pekerjaan dapat dilihat pada gambar 3.6 berikut.



Gambar 3.6 Alur Tambah Lowongan Pekerjaan

Penjelasan gambar 3.6 adalah:

1. Jika IDUKA belum terdaftar, maka melakukan pendaftaran terlebih dahulu
2. IDUKA yang telah terdaftar melakukan login
3. Ditampilkan dashboard aplikasi untuk level akses IDUKA
4. Melalui tautan lowongan, IDUKA menambah data lowongan pekerjaan baru. Data lowongan diisi dengan lengkap.
5. Pada data lowongan yang telah tersimpan, IDUKA dapat menambah data: Deskripsi Pekerjaan, Syarat/Kualifikasi, dan Keahlian.

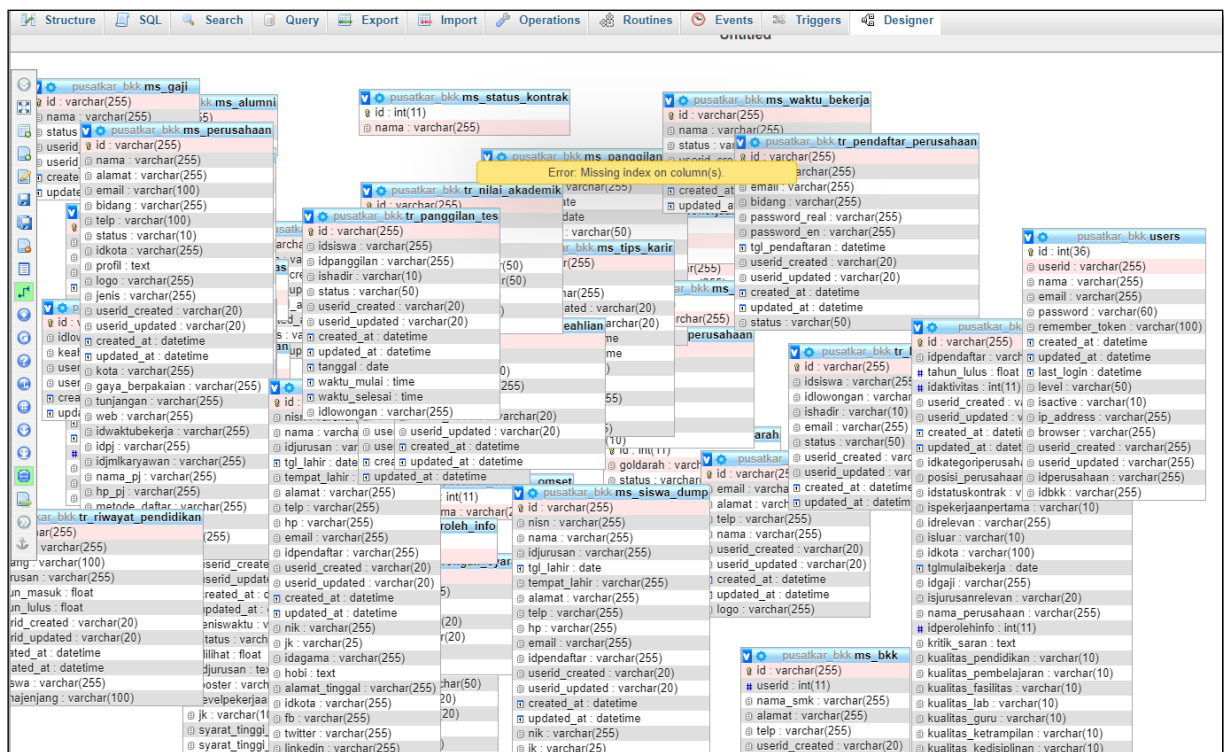
6. Jika terdapat kesalahan pada data yang diisikan, IDUKA dapat melakukan perubahan dan hapus data.
7. Selesai menambah data lowongan pekerjaan, IDUKA melakukan logout dari aplikasi.

3.3 Desain database

Desain database berisi *Entity Relationship Diagram* (ERD) dan penjelasan dari masing-masing tabel.

3.3.1 Entity Relationship Diagram

Relasi antar tabel yang digunakan dalam pengembangan aplikasi pusat karir SMK lebih banyak dilakukan pada level SQL atau Query. Relasi antar tabel memanfaatkan *eloquent relationships* yang disediakan oleh Laravel. Melalui eloquent, telah terdapat fitur untuk melakukan relasi *One to One*, *One to Many*, *One to Many (Inverse)/Belong To*, *Has One Thought*, dan *Has Many Thought*. daftar tabel yang digunakan di pusat karir SMK terlihat seperti gambar 3.7 berikut.



Gambar 3.7 ERD Pusat Karir SMK

Terlihat bahwa tidak ada relasi secara visual antara satu tabel dengan tabel yang lain. semua relasi antar tabel dilakukan melalui kode program dilaravel melalui *eloquent relationship*. Harus diakui terdapat kelemahan ketika tidak ada relasi antar tabel khususnya masalah *constraint* dan validasi. Untuk menutupi kelemahan tersebut, semua validasi dilakukan melalui aplikasi, termasuk *form validation*. Gambar 3.8 berikut adalah contoh penerapan join yang dilakukan pada level kode program.

```

28 public function index()
29 {
30     if(Session::get('level')=='A')
31     {
32         $siswa = DB::table('ms_siswa')->select('ms_siswa.*','ms_jurusan.nama as jurusan')
33         |->leftjoin('ms_jurusan','ms_jurusan.id','=', 'ms_siswa.idjurusan')->orderBy('created_at','desc')->get();
34     }
35     else if(Session::get('level')=='BKK')
36     {
37         $siswa = DB::table('ms_siswa')->select('ms_siswa.*','ms_jurusan.nama as jurusan')
38         |->leftjoin('ms_jurusan','ms_jurusan.id','=', 'ms_siswa.idjurusan')
39         |->where('ms_siswa.idbkk',Session::get('idbkk'))
40         |->orderBy('created_at','desc')->get();
41     }
42
43     return view ('admin.data_siswa.index',compact('siswa'));
44 }
45

```

Gambar 3.8 Relasi di Eloquent

Berdasarkan potongan kode program di gambar 3.8, terlihat bahwa terjadi relasi *left join* antara tabel *ms_siswa* dengan *ms_jurusan* berdasarkan *idjurusan*. Hasil query kemudian diurutkan berdasarkan *crated_at* secara *descending*. untuk akun level BKK, selain terjadi relasi *left join*, dan data diurutkan, juga terdapat tambahan kondisi *Where* untuk melakukan *filter* data agar yang ditampilkan adalah siswa pada BKK tersebut. Setelah query dijalankan dan data diperoleh kemudian ditampilkan pada file *data_siswa_index* yang terdapat pada direktori *admin*.

3.3.2 Deskripsi Struktur Tabel

Jumlah tabel yang digunakan dalam pengembangan aplikasi pusat karir SMK adalah 45 tabel, yang tersimpan dalam DBMS (Database Management System) MySQL. Alasan memilih MySQL karena terdapat versi *Community* yang digunakan secara gratis, cepat, dan merupakan DBMS yang paling populer untuk aplikasi web. Secara fungsi, tabel yang digunakan terbagi menjadi dua yakni tabel master (referensi) dan tabel transaksi. Tabel master digunakan untuk referensi data yang diperlukan dalam pengisian data, sedangkan tabel transaksi digunakan untuk menyimpan data transaksi. Tampilan tabel yang digunakan tersaji pada gambar 3.9 berikut ini.

Table	Action	Rows	Type	Collation	Size
☐ ms_agama	★ Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_agenda	★ Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_aktivitas_alumni	★ Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_alumni	★ Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_bkk	★ Browse Structure Search Insert Empty Drop	9	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_faq	★ Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_file	★ Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_file_siswa	★ Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_gaji	★ Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_goldarah	★ Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_hak_akses	★ Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_jenjang_pendidikan	★ Browse Structure Search Insert Empty Drop	10	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_jml_karyawan	★ Browse Structure Search Insert Empty Drop	8	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_jurusan	★ Browse Structure Search Insert Empty Drop	405	InnoDB	latin1_swedish_ci	64.0 KiB
☐ ms_karir	★ Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_kategori_perusahaan	★ Browse Structure Search Insert Empty Drop	53	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_kelas	★ Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	16.0 KiB

☐ ms_kota	★ Browse Structure Search Insert Empty Drop	~91,509	InnoDB	utf8_general_ci	6.5 MiB
☐ ms_level_pekerjaan	★ Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_lowongan	★ Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_lowongan_deskripsi	★ Browse Structure Search Insert Empty Drop	13	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_lowongan_keahlian	★ Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_lowongan_syarat	★ Browse Structure Search Insert Empty Drop	21	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_matpel	★ Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_omset	★ Browse Structure Search Insert Empty Drop	8	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_panggilan_tes	★ Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_pendaftar	★ Browse Structure Search Insert Empty Drop	9	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_peroleh_info	★ Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_perusahaan	★ Browse Structure Search Insert Empty Drop	11	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_profil	★ Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_relevansi_pekerjaan	★ Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_siswa	★ Browse Structure Search Insert Empty Drop	9	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_siswa_dump	★ Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_sosmed	★ Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_status_kontrak	★ Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_tips_karir	★ Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16.0 KiB
☐ ms_waktu_bekerja	★ Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16.0 KiB
☐ tr_bidang_keahlian	★ Browse Structure Search Insert Empty Drop	15	InnoDB	latin1_swedish_ci	16.0 KiB
☐ tr_lamaran	★ Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16.0 KiB

☐ tr_nilai_akademik	★ Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16.0 KiB
☐ tr_panggilan_tes	★ Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	32.0 KiB
☐ tr_pendaftar_perusahaan	★ Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16.0 KiB
☐ tr_riwayat_pendidikan	★ Browse Structure Search Insert Empty Drop	9	InnoDB	latin1_swedish_ci	16.0 KiB
☐ tr_tracer_study	★ Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16.0 KiB
☐ users	★ Browse Structure Search Insert Empty Drop	22	InnoDB	latin1_swedish_ci	32.0 KiB
45 tables	Sum	~92,213	InnoDB	latin1_swedish_ci	7.3 MiB

Gambar 3.9 Lanjutan Tabel Pusat Karir SMK

Tabel yang diawali dengan **ms_** artinya tabel master, sedangkan yang diawali dengan **tr_** adalah tabel transaksi. Penjelasan struktur tabel secara detail, *primary key*, *foreign key*, dan relasi dari semua tabel yang digunakan adalah sebagai berikut.

- Tabel **ms_agama**

Merupakan data tabel master agama, digunakan untuk menyimpan data agama yang ada di Indonesia. Tabel ini digunakan sebagai referensi ketika pengguna mengisi data agama. struktur tabelnya tersaji pada tabel 3.1 berikut.

Tabel 3.1 Struktur tabel ms_agama

No	Nama Field	Tipe data (Length)	Keterangan
1	id	int(11)	Not Null, Primary key, auto_increment
2	agama	varchar(40)	Null
3	status	char(1)	Null

Tabel **ms_agama** tidak memiliki *foreign key* karena menjadi referensi pada tabel-tabel yang lain.

- Tabel **ms_agenda**

Tabel ini digunakan untuk menyimpan data agenda kegiatan pada aplikasi pusat karir SMK. Tabel ini dapat digunakan pada level pusat karir maupun pada level BKK (SMK). Struktur tabelnya tersaji pada tabel 3.2 berikut.

Tabel 3.2 Struktur tabel ms_agenda

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	agenda	text	Null
3	tempat	varchar(100)	Null
4	lokasi	varchar(100)	Null
5	tgl_mulai	date	Null
6	tgl_selesai	date	Null
7	deskripsi	text	Null

No	Nama Field	Tipe data (Length)	Keterangan
8	biaya	varchar(30)	Null
9	status	char(1)	Null
10	waktu_mulai	time	Null
11	waktu_selesai	time	Null
12	dilihat	int(11)	Null
13	penyelenggara	varchar(100)	Null
14	idbkk	varchar(50)	Null, <i>foreign key</i> dari tabel BKK
15	ispublic	char(1)	Null
16	create_at	dateTime	Not Null
17	update_at	dateTime	Not Null

Tabel *ms_agenda* memiliki *primary key* dengan tipe data *varchar*, tidak seperti pada umumnya yang bertipe *integer* dan *auto_increment*. Tipe data ini dipilih karena data yang disimpan pada *primary key* adalah UUID. Terdapat *foreign key* dari tabel BKK artinya BKK dapat membuat agenda yang bersifat local dengan field *ispublic* bernilai 0.

- Tabel *ms_aktivitas_alumni*

Merupakan tabel master yang digunakan untuk menyimpan data aktivitas alumni, yakni bekerja, wirausaha, studi lanjut dan lainnya. Struktur tabelnya tersaji pada tabel 3.3 berikut ini.

Tabel 3.3 Struktur tabel *ms_aktivitas_alumni*

No	Nama Field	Tipe data (Length)	Keterangan
1	id	int(11)	Primary key, Not Null, dan Auto_increment
2	Nama	Varchar(100)	Null

Tabel ini digunakan sebagai referensi pada saat pengisian form pendataan dan registrasi data alumni.

- Tabel ms_bkk

Digunakan untuk menyimpan data BKK (Bursa Kerja Khusus) yang terdapat pada tiap SMK, walaupun tidak semua SMK memiliki BKK. satu data BKK mewakili satu SMK. Struktur tabel ms_bkk tersaji pada tabel 3.4 berikut.

Tabel 3.4 Struktur tabel ms_bkk

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	userid	int	Null
3	nama_smk	varchar(100)	Null
4	alamat	varchar(100)	Null
5	telp	varchar(30)	Null
6	userid_created	varchar(20)	Null
7	userid_updated	varchar(20)	Null
8	created_at	datetime	Null
9	updated_at	datetime	Null

- Tabel ms_faq

Tabel ini digunakan untuk menyimpan data FAQ (*Frequently Asked Questions*) yang merupakan salah satu fitur pada aplikasi pusat karir SMK. Struktur tabel ms_faq tersaji pada tabel 3.5.

Tabel ini tidak memiliki foreign key, artinya yang dapat melakukan CRUD (*Create, Read, Update, Delete*) hanya dari pusat karir SMK. BKK tidak dapat melakukan CRUD pada tabel ini.

- Tabel ms_file

Digunakan untuk menyimpan data berkas pusat karir SMK dan BKK dalam bentuk *softcopy*. Berkas *softcopy* menjadi backup dan arsip dari berkas *hardcopy*, sehingga memudahkan mendapatkan kembali dengan mudah dan cepat. Struktur tabel ms_file tersaji pada tabel 3.6 berikut.

Tabel 3.5 Struktur tabel ms_faq

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	pertanyaan	text	Null
3	jawaban	text	Null
4	status	char(1)	Null
5	userid_created	varchar(10)	Null
6	userid_updated	varchar(10)	Null
7	created_at	datetime	Not Null
8	updated_at	datetime	Not Null

Tabel 3.6 Struktur tabel ms_file

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	namafile	varchar(100)	Not Null
3	ukuran	varchar(10)	Not Null
4	status	char(1)	Not Null, 0 = deleted, 1 = aktif
5	userid_created	varchar(10)	Null
6	userid_updated	varchar(100)	Null
7	created_at	datetime	Not Null
8	updated_at	datetime	Not Null
9	nama	varchar(100)	Null
10	download	int(11)	Not Null
11	idbkk	varchar(50)	Null, Foreign Key dari tabel bkk
12	ektensi	varchar(50)	Not Null
13	dilihat	int(11)	Not Null
14	ispublic	char(1)	Not Null

Yang dapat melakukan CRUD pada tabel ms_file adalah pusat karir SMK dan BKK. jika kolom ispublic bernilai 1, maka yang melakukan CRUD adalah pusat karir SMK, dan jika kolom ispublic bernilai 0, maka berkas tersebut dimiliki oleh BKK.

- Tabel ms_file_siswa

Tabel ini digunakan untuk menyimpan data portofolio siswa dan alumni. Portofolio bias dalam bentuk file, video, foto, dan audio. Yang dapat melakukan CRUD pada tabel ini adalah siswa dan alumni. Struktur tabel ms_file_siswa tersaji pada tabel 3.7 berikut.

Tabel 3.7 Struktur tabel ms_file_siswa

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Namafile	Varchar(100)	Not Null
3	Ukuran	Varchar(10)	Not Null
4	Status	Char(1)	Not Null, 0=deleted, 1=aktif
5	Userid_created	Varchar(10)	Null
6	Userid_updated	Varchar(10)	Null
7	Created_at	Datetime	Not Null
8	Updated_at	Datetime	Not Null
9	Nama	Varchar(100)	Null
10	Download	Int(11)	Not Null
11	Ekstensi	Varchar(10)	Not Null
12	Dilihat	Int(11)	Not Null
13	Idsiswa	Varchar(50)	Not Null, Foreign Key tabel siswa
14	Tahun	Char(4)	Null
15	Instansi	Varchar(50)	Null
16	Tempat	Varchar(50)	Null

Isi tabel tb_file_siswa adalah semua portofolio yang dimiliki oleh siswa, bisa berupa

sertifikat LSP, sertifikat juara perlombaan, pengalaman project yang pernah dilakukan, dan lain-lain. Data ini nantinya digunakan oleh sekolah dan iduka sebagai referensi dalam melihat kemampuan dan prestasi siswa.

- Tabel ms_gaji

Tabel ini digunakan untuk menyimpan data master gaji yang digunakan dalam info lowongan kerja. Struktur tabel ms_gaji tersaji pada tabel 3.8 berikut.

Tabel 3.8 Struktur tabel ms_gaji

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Nama	Varchar(100)	Not Null
3	Status	Char(1)	Not Null
4	Userid_created	Varchar(10)	Null
5	Userid_updated	Varchar(10)	Null
6	Created_at	Datetime	Not Null
7	Updated_at	Datetime	Not Null

Data pada tabel ini digunakan pada form pilihan gaji pada saat CRUD lowongan kerja, dan apply lowongan.

- Tabel ms_goldarah

Digunakan untuk menyimpan data master golongan darah yang digunakan sebagai referensi pada saat siswa atau alumni mengisi data pribadi terkait kesehatan. Struktur tabel ms_goldarah tersaji pada gambar 3.9 berikut.

Tabel 3.9 Struktur tabel ms_goldarah

No	Nama Field	Tipe data (Length)	Keterangan
1	id	Int(2)	Primary Key, Auto_increment
2	Goldarah	Varchar(10)	Not Null
3	Status	Char(1)	Null

- Tabel ms_hak_akses

Tabel ini digunakan untuk menyimpan data level akses pada pengguna aplikasi pusat karir SMK. Terdapat empat (4) level akses yaitu admin, BKK, siswa/alumni dan iduka. Struktur tabel ms_hak_akses tersaji pada tabel 3.10 berikut.

Tabel 3.10 Struktur tabel ms_hak_akses

No	Nama Field	Tipe data (Length)	Keterangan
1	id	int(2)	Primary key, Auto_increment
2	Level	Varchar(40)	Null
3	Nama	Varchar(50)	Null
4	Keterangan	Varchar(50)	Null

Data pada tabel ini digunakan untuk mengelompokkan level akses pengguna. pengguna yang menjadi kelompok level akses tertentu, akan memiliki hak akses yang sama.

- Tabel ms_jenang_pendidikan

Tabel ini digunakan untuk menyimpan data jenjang pendidikan, mulai SD sampai S3. Data pada tabel ini digunakan sebagai referensi yang terkait dengan pendidikan. Struktur tabel ms_jenang_pendidikan tersaji pada tabel 3.11 berikut.

Tabel 3.11 Struktur tabel ms_jenang_pendidikan

No	Nama Field	Tipe data (Length)	Keterangan
1	id	Int(2)	Primary Key, Auto_Increment
2	Jenang	Varchar(30)	Not Null
3	Status	Char(1)	Null

- Tabel ms_jml_karyawan

Tabel ini digunakan untuk menyimpan data jumlah karyawan pada suatu IDUKA. Tabel ini digunakan sebagai referensi dalam membuat lowongan pekerjaan. Struktur tabel ms_jml_karyawan tersaji pada tabel 3.12 berikut.

Tabel 3.12 Struktur tabel ms_jml_karyawan

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Nama	Varachar(50)	Not Null
3	Status	Char(1)	Null
4	Userid_created	Varchar(20)	Null
5	Userid_updated	Varchar(20)	Null
6	Created_at	Datetime	Not Null
7	Updated_at	Datetime	Not Null

- Tabel ms_jurusan

Tabel ini digunakan untuk menyimpan data jurusan sesuai dengan spektrum SMK. Data pada tabel ini digunakan untuk menentukan jurusan siswa atau alumni. Struktur tabel ms_jurusan tersaji pada tabel xx berikut.

Tabel 3.13 Struktur tabel ms_jurusan

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Nama	Varchar(50)	Not Null
3	Userid_created	Varchar(20)	Null
4	Userid_updated	Varchar(20)	Null
5	Created_at	Datetime	Not Null
6	Updated_at	Datetime	Not Null
7	Status	Char(1)	Null

- Tabel ms_karir

Tabel ini digunakan untuk menyimpan data jenis karir siswa dan alumni SMK. Data pada tabel ini dijadikan referensi pada pengisian tracer study. Struktur tabel ms_karir tersejadi pada tabel 3.14 berikut.

Tabel 3.14 Struktur tabel ms_karir

No	Nama Field	Tipe data (Length)	Keterangan
1	id	Int(11)	Primary key, Auto_increment
2	Nama	Varchar(50)	Not Null

- Tabel ms_kategori_perusahaan

Tabel ini digunakan untuk menyimpan data kategori perusahaan. Data pada tabel ini digunakan pada CRUD data IDUKA (Industri, Dunia Usaha dan Dunia Kerja) untuk menentukan kategori dan bidang usahanya. Struktur tabel ms_kategori_perusahaan tersaji pada tabel 3.15 berikut.

Tabel 3.15 Struktur tabel ms_kategori_perusahaan

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Nama	Varchar(50)	Not Null
3	Userid_created	Varchar(20)	Null
4	Userid_updated	Varchar(20)	Null
5	Created_at	Datetime	Not Null
6	Updated_at	Datetime	Not Null
7	Status	Char(1)	Not Null

- Tabel ms_kelas

Tabel ini digunakan untuk menyimpan data kelas di SMK. Tabel ini digunakan untuk sebagai referensi untuk melengkapi data siswa/alumni. Struktur tabel ms_kelas tersaji pada tabel 3.16 berikut.

- Tabel ms_kota

Tabel ini digunakan untuk menyimpan data kota diseluruh Indonesia. Data pada tabel ini digunakan untuk referensi ketika mengisi data kota, sehingga tidak perlu mengisi secara manual tapi langsung memilih dari pilihan yang tersedia. Struktur tabel ms_kota tersaji pada tabel 3.17 berikut.

Tabel 3.16 Struktur tabel ms_kelas

No	Nama Field	Tipe data (Length)	Keterangan
1	id	Int(11)	Primary Key, Auto_Increment
2	Kelas	Varchar(50)	Not Null
3	Alias	Varchar(50)	Null

Tabel 3.17 Struktur tabel ms_kota

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(13)	Primary key dan Not Null
2	Nama	Varchar(60)	Not Null
3	Status	Varchar(20)	Null

- Tabel ms_level_pekerjaan

Tabel ini digunakan untuk menyimpan data level pekerjaan. Data pada tabel ini digunakan sebagai referensi pada saat membuat informasi lowongan pekerjaan. Struktur tabel ms_level_pekerjaan tersaji pada tabel 3.18 berikut.

Tabel 3.18 Struktur tabel ms_level_pekerjaan

No	Nama Field	Tipe data (Length)	Keterangan
1	id	Int(6)	Primary, Auto_Increment
2	Nama	Varchar(50)	Not Null
3	Status	Char(1)	Null

- Tabel ms_lowongan

Tabel ini digunakan untuk menyimpan data lowongan pekerjaan. Pengguna yang dapat melakukan CRUD pada tabel ini adalah pusat karir, BKK, dan iduka. Struktur tabel ms_lowongan tersaji pada tabel 3.19 berikut.

Tabel 3.19 Struktur tabel ms_lowongan

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Lowongan	Text	Not Null
3	Idperusahaan	Varchar(50)	Not Null
4	Idkota	Varchar(50)	Not Null
5	Idgaji	Varchar(50)	Not Null
6	Waktukerja	Varchar(50)	Not Null
7	Tgl_mulai	Date	Not Null
8	Tgl_selesai	Data	Null
9	Userid_created	Varchar(20)	Null
10	Userid_updated	Varchar(20)	Null
11	Created_at	Datetime	Not Null
12	Updated_at	Datetime	Not Null
13	Jeniswaktu	Varchar(50)	Not Null
14	Status	Char(1)	Null
15	Dilihat	Int(6)	Not Null
16	Idjurusan	Varchar(50)	Not Null
17	Poster	Varchar(50)	Null
18	Levelpekerjaan	Int(6)	Not Null
19	Jk	Char(1)	Not Null
20	Syarat_tinggi_laki	Int(2)	Not Null
21	Syarat_tinggi_pr	Int(2)	Not Null
22	Isbutawarna	Char(1)	Not Null
23	Isacamata	Char(1)	Not Null
24	Batas_usia	Int(6)	Null

No	Nama Field	Tipe data (Length)	Keterangan
25	Alamat_penempatan	Varchar(100)	Not Null
26	Tipelowongan	Varchar(50)	Null
27	Isnilai	Char(1)	Not Null
28	Isdokumen	Varchar(10)	Not Null
29	Idbkk	Varchar(50)	Null
30	Ispublic	Char(1)	Null
31	Isgaji	Char(1)	Null

Tabel ms_lowongan berelasi dengan tabel referensi gaji, waktu kerja, jurusan, level pekerjaan, kota, dan BKK. Terdapat kolom *ispublic* yang digunakan untuk menentukan lowongan yang ditampilkan dapat diakses oleh semua atau terbatas hanya pada BKK tertentu saja.

- Tabel ms_lowongan_deskripsi

Tabel ini digunakan untuk menyimpan data deskripsi pekerjaan. Data pada tabel ini digunakan pada lowongan pekerjaan. Struktur tabel ms_lowongan_deskripsi tersaji pada tabel 3.20 berikut.

Tabel 3.20 Struktur tabel ms_lowongan_deskripsi

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Idlowongan	Varchar(50)	Not Null
3	Deskripsi	Text	Null
4	Userid_created	Int(11)	Not Null
5	Userid_updated	Int(11)	Null
6	Created_at	Datetime	Not Null
7	Updated_at	Datetime	Not Null

Terlihat bahwa terdapat relasi dengan tabel lowongan pekerjaan berdasarkan kolom idlowongan. Hal ini terjadi karena deskripsi lowongan pekerjaan menggunakan data pada tabel ini.

- Tabel ms_lowongan_keahlian

Tabel ini digunakan untuk menyimpan data spesifikasi keahlian yang ditampilkan pada lowongan pekerjaan. Struktur tabel ms_lowongan_keahlian tersaji pada tabel 3.21 berikut.

Tabel 3.21 Struktur tabel ms_lowongan_keahlian

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Idlowongan	Varchar(50)	Not Null
3	Keahlian	Text	Null
4	Userid_created	Int(11)	Not Null
5	Userid_updated	Int(11)	Null
6	Created_at	Datetime	Not Null
7	Updated_at	Datetime	Not Null

Terdapat relasi dengan tabel lowongan pekerjaan. Hal ini terjadi karena ketika terdapat lowongan pekerjaan, data spesifikasi keahlian diambil dari tabel ini.

- Tabel ms_lowongan_syarat

Tabel ini digunakan untuk menyimpan data persyaratan pada lowongan pekerjaan. Struktur tabel ms_lowongan_syarat tersaji pada tabel 3.22.

Terdapat relasi dengan tabel lowongan pekerjaan. Hal terjadi karena ketika terdapat lowongan pekerjaan, data persyaratan lowongan diambil dari tabel ini. Data pada tabel ini juga digunakan sebagai filter untuk *apply* lowongan pekerjaan. Ketika siswa atau alumni *apply* lowongan pekerjaan, dan data yang diisikan tidak memenuhi persyaratan maka secara otomatis akan ditolak.

Tabel 3.22 Struktur tabel ms_lowongan_syarat

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Idlowongan	Varchar(50)	Not Null
3	Syarat	Text	Null
4	Userid_created	Int(11)	Not Null
5	Userid_updated	Int(11)	Null
6	Created_at	Datetime	Not Null
7	Updated_at	Datetime	Null

- Tabel ms_matpel

Tabel ini digunakan untuk menyimpan data mata pelajaran utama untuk melengkapi portofolio siswa/alumni dari sisi akademik. Data nilai akademik ini sangat membantu bagi pengelola BKK maupun Iduka untuk mengetahui unjuk kerja siswa sehingga ketika terdapat permintaan oleh iduka, maka pengelola BKK dapat dengan mudah menentukan siswa yang direferensikan. Struktur tabel ms_matpel tersaji pada tabel 3.23 berikut.

Tabel 3.23 Struktur tabel ms_matpel

No	Nama Field	Tipe data (Length)	Keterangan
1	id	Int(2)	Primary key, Auto_Increment
2	Nama	Varchar(50)	Not Null

- Tabel ms_omset

Tabel ini digunakan untuk menyimpan data pendapatan atau gaji alumni yang telah bekerja. Data ini digunakan pada proses *tracer study* oleh sekolah terhadap alumni yang telah bekerja. Struktur tabel ms_omset tersaji pada tabel 3.24 berikut.

Tabel 3.24 Struktur tabel ms_omset

No	Nama Field	Tipe data (Length)	Keterangan
1	id	Int(2)	Primary key, Auto_Increment
2	Nama	Varchar(50)	Not Null

- Tabel ms_panggilan_tes

Tabel ini digunakan untuk menyimpan data informasi panggilan tes bagi siswa atau alumni yang telah *apply* lowongan pekerjaan. Pengguna yang dapat melakukan CRUD pada tabel ini adalah Iduka dan BKK. Struktur tabel ms_panggilan_tes tersaji pada tabel 3.25 berikut.

Tabel 3.25 Struktur tabel ms_panggilan_tes

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Idlowongan	Varchar(50)	Not Null
3	Tgl_mulai	Date	Not Null
4	Tgl_selesai	Date	Null
5	Waktu_mulai	Time	Not Null
6	Waktu_selesai	Time	Null
7	Userid_created	Int(11)	Not Null
8	Userid_updated	Int(11)	Null
9	Created_at	Datetime	Not Null
10	Updated_at	Datetime	Null
11	Keterangan	Varachar(100)	Null
12	Nama	Varachar(50)	Not Null

Panggilan tes merupakan tahapan dalam rekrutmen tenaga kerja yang dilakukan oleh Iduka. terdapat relasi dengan tabel lowongan pekerjaan karena panggilan tes mengacu pada siswa dan alumni yang telah *apply* pada lowongan pekerjaan tertentu.

Tabel 3.26 Struktur tabel ms_pendaftar

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Nama	Varchar(50)	Not Null
3	Nis	Varchar(20)	Not Null
4	Jk	Char(1)	Not Null
5	Alamat	Varchar(50)	Not Null
6	Tgllahir	Date	Null
7	Tmplahir	Varchar(50)	Null
8	Email	Varachar(50)	Null
9	Telpon	Varchar(20)	Null
10	Hp	Varchar(20)	Null
11	Tgl_pendaftaran	Datetime	Not Null
12	Status	Char(1)	Not Null
13	Userid_created	Int(11)	Not Null
14	Userid_updated	Int(11)	Null
15	Created_at	Datetime	Not Null
16	Update_at	Datetime	Null
17	Islumni	Char(1)	Not Null
18	Nama_depan	Varchar(50)	Not Null
19	Nama_belakang	Varachar(50)	Null
20	Idkelas	Int(6)	Not Null
21	Password	Varchar(50)	Not Null
22	Idjurusan	Varchar(50)	Not Null
23	Klaster	Varchar(50)	Null
24	Idbkk	Varchar(50)	Not Null
25	Ismanual	Char(1)	Not Null

- Tabel ms_pendaftar

Tabel ini digunakan untuk menyimpan data siswa atau alumni yang mendaftar melalui aplikasi pusat karir SMK. Data siswa dapat diisikan oleh sekolah (BKK) dan siswa secara mandiri melakukan pendaftaran. Struktur tabel ms_pendaftar tersaji pada tabel 3.26.

Kolom islumni digunakan sebagai penanda bahwa yang melakukan pendaftaran adalah alumni atau siswa. Sedangkan kolom ismanual digunakan sebagai penanda bahwa data diisi secara manual atau melalui pendaftaran melalui aplikasi.

- Tabel ms_peroleh_info

Tabel ini digunakan untuk menyimpan data sumber informasi bagi siswa atau alumni ketika melamar pekerjaan. Struktur tabel ms_peroleh_info tersaji pada tabel 3.27 berikut.

Tabel 3.27 Struktur tabel ms_peroleh_info

No	Nama Field	Tipe data (Length)	Keterangan
1	id	Int(2)	Primary key, Auto_increment
2	Nama	Varchar(50)	Not Null

- Tabel ms_perusahaan

Tabel ini digunakan untuk menyimpan data IDUKA, baik dilakukan secara manual oleh pusat karir dan BKK maupun secara online yang dilakukan oleh IDUKA. Data pada tabel ini digunakan untuk proses rekrutmen yang dimulai dari *publish* lowongan kerja, *apply* lowongan, panggilan tes dan penentuan diterima tidaknya pelamar. Struktur tabel ms_perusahaan tersaji pada tabel 3.27.

Data perusahaan yang telah ada kerjasama dengan BKK, maka data akan diisikan oleh BKK. Profil data perusahaan tersebut dapat dilihat oleh siswa atau alumni pada saat *publish* lowongan pekerjaan.

Tabel 3.28 Struktur tabel ms_perusahaan

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Nama	Varchar(50)	Not Null
3	Alamat	Varchar(50)	Not Null
4	Email	Varchar(50)	Not Null
5	Bidang	Varchar(50)	Not Null
6	Telpon	Varchar(20)	Null
7	Status	Char(1)	Not Null
8	Idkota	Varchar(10)	Not Null
9	Profil	Varchar(100)	Null
10	Logo	Varchar(50)	Null
11	Jenis	Varchar(50)	Null
12	Userid_created	Int(11)	Not Null
13	Userid_updated	Int(11)	Null
14	Created_at	Datetime	Not Null
15	Updated_at	Datetime	Null
16	Kota	Varchar(50)	Not Null
17	Gaya_berpakaian	Varchar(100)	Not Null
18	Tunjangan	Varchar(50)	Null
19	Website	Varchar(50)	Null
20	Idwaktubekerja	Int(2)	Not Null
21	Idpj	Int(6)	Not Null
22	Idjmlkaryawan	Int(2)	Not Null
23	Nama_pj	Varchar(50)	Not Null
24	Hp_pj	Varchar(20)	Not Null
25	Metode_daftar	Varchar(50)	Not Null

No	Nama Field	Tipe data (Length)	Keterangan
26	Dilihat	Int(6)	Not Null
27	Idpendaftar	Varchar(50)	Not Null
28	Idbkk	Varchar(50)	Null

- Tabel ms_profil

Tabel ini digunakan untuk menyimpan data profil pengelola website pusat karir SMK yang dapat dilihat pada laman kontak. Struktur tabel ms_profil tersaji pada tabel 3.29 berikut.

Tabel 3.29 Struktur tabel ms_profil

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Email	Varchar(50)	Not Null
3	Alamat	Varchar(50)	Not Null
4	Telpon	Varchar(20)	Null
5	Nama	Varchar(50)	Not Null
6	Userid_created	Int(11)	Not Null
7	Userid_updated	Int(11)	Null
8	Created_at	Datetime	Not Null
9	Updated_at	Datetime	Null
10	Logo	Varchar(50)	Not Null

Dengan adanya tabel ini, proses perubahan identitas aplikasi dapat dilakukan dengan mudah tanpa harus membua kode program.

- Tabel ms_relevansi_pekerjaan

Tabel ini digunakan untuk menyimpan relevansi pekerjaan yang digunakan dalam prose *tracer study*. Struktur tabel ms_relevansi_pekerjaan tersaji pada gambar 3.30 berikut.

Tabel 3.30 Struktur tabel ms_relevansi_pekerjaan

No	Nama Field	Tipe data (Length)	Keterangan
1	id	Int(2)	Primary Key, Auto_Increment
2	Nama	Varchar(50)	Not Null

- Tabel ms_siswa

Tabel ini digunakan untuk menyimpan data siswa secara lengkap dan detail. Keluaran dari data ini digunakan dalam pembuatan CV (Curriculum Vitae) dan keperluan yang lain. Struktur tabel ms_siswa tersaji pada tabel 3.31 berikut.

Tabel 3.31 Struktur tabel ms_siswa

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Nisn	Varchar(20)	Null
3	Nama	Varchar(50)	Not Null
4	Idjurusan	Varchar(50)	Null
5	Tgl_lahir	Date	Not Null
6	Tmp_lahir	Varchar(50)	Null
7	Alamat	Varchar(100)	Not Null
8	Telpon	Varchar(20)	Null
9	Hp	Varchar(20)	Null
10	Email	Varchar(50)	Not Null
11	Idpendaftar	Varchar(50)	Not Null
12	Userid_created	Varchar(20)	Not Null
13	Userid_updated	Varchar(20)	Null
14	Created_at	Datetime	Not Null
15	Updated_at	Datetime	Null
17	Nik	Varchar(20)	Null

No	Nama Field	Tipe data (Length)	Keterangan
18	Jk	Char(1)	Not Null
19	Idagama	Int(2)	Not Null
20	Hobi	Varchar(100)	Null
21	Alamat_tinggal	Varchar(100)	Null
22	Idkota	Varchar(20)	Null
23	Fb	Varchar(50)	Null
24	Twitter	Varchar(50)	Null
25	Ig	Varchar(50)	Null
26	Linkedin	Varchar(50)	Null
27	Idkelas	Int(2)	Null
28	Goldarah	Int(2)	Null
29	Tinggi	Float	Not Null
30	Beratbadan	Float	Not Null
31	Isbutawarna	Char(1)	Not Null
32	Istindik	Char(1)	Not Null
33	Isdifabel	Char(1)	Not Null
34	Istato	Char(1)	Not Null
35	Iskacamata	Char(1)	Not Null
36	Riwayat_penyakit	Text	Null
37	Foto	Varchar(50)	Null
38	Idbkk	Varchar(50)	Not Null
39	Klaster	Varchar(50)	Null

Tabel ini digunakan untuk banyak keperluan. Semua transaksi yang terkait dengan data siswa, maka pasti menggunakan data pada tabel ini. Sesuai fitur utama aplikasi, data ini berfungsi untuk menyimpan data CV siswa, sehingga siswa dapat mencetak

ketika data telah lengkap. Untuk proses tracer study, apply lowongan kerja, dan proses di BKK banyak menggunakan data siswa.

- Tabel ms_sosmed

Digunakan untuk menyimpan data media sosial yang dimiliki oleh pengelola aplikasi pusat karir SMK. Struktur table ms_sosmed tersaji pada tabel 3.32 berikut.

Tabel 3.32 Struktur tabel ms_sosmed

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Nama	Varchar(50)	Not Null
3	Link	Varchar(50)	Not Null
4	Status	Char(1)	Null
5	Userid_created	Varchar(20)	Not Null
6	Userid_updated	Varchar(20)	Null
7	Created_at	Datetime	Not Null
8	Update_at	Datetime	Null

- Tabel ms_status_kontrak

Digunakan untuk menyimpan data status karyawan dalam perusahaan. Data pada tabel ini digunakan dalam proses lowongan kerja. Struktur tabel ms_status_kontrak tersaji pada tabel 3.33 berikut.

Tabel 3.33 Struktur tabel ms_status_kontrak

No	Nama Field	Tipe data (Length)	Keterangan
1	id	Int(6)	Primary Key, Auto_Increment
2	Nama	Varchar(50)	Not Null

- Tabel ms_tips_karir

Tabel ini digunakan untuk menyimpan data tips/artikel yang ditampilkan pada laman utama aplikasi. Data pada tabel ini tidak berelasi dengan tabel manapun, karena

bersifat publik dan isinya berupa informasi atau berita. Struktur tabel *ms_tips_karir* tersaji pada tabel 3.34 berikut.

Tabel 3.34 Struktur tabel ms_tips_karir

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Judul	Text	Not Null
3	Tipe	Varchar(30)	Not Null
4	Userid_created	Varchar(20)	Not Null
5	Userid_update	Varchar(20)	Null
6	Status	Char(1)	Not Null
7	Dilihat	Int(6)	Not Null
8	Penulis	Varchar(50)	Null
9	Tips	Varchar(50)	Not Null
10	Idbkk	Varchar(50)	Null
11	Created_at	Datetime	Not Null
12	Updated_at	Datetime	Null

- Tabel *ms_waktu_bekerja*

Tabel ini digunakan untuk menyimpan data waktu pekerja pada suatu perusahaan atau IDUKA. Data pada tabel ini digunakan pada lowongan kerja, sehingga pelamat data mengetahui hari kerja pada perusahaan tersebut. Struktur tabel *ms_waktu_bekerja* tersaji pada tabel 3.35 .

- Tabel *tr_bidang_keahlian*

Tabel ini digunakan untuk menyimpan data keahlian yang dimiliki oleh siswa beserta level keahliannya. Dari data ini, dapat diketahui keahlian dari siswa atau alumni disertai dengan level keahliannya. Struktur tabel *tr_bidang_keahlian* tersaji pada tabel 3.36.

Tabel 3.35 Struktur tabel ms_waktu_bekerja

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Nama	Varchar(100)	Not Null
3	Status	Char(1)	Not Null
4	Userid_created	Varchar(20)	Not Null
5	Userid_updated	Varchar(20)	Null
6	Created_at	Datetime	Not Null
7	Updated_at	Datetime	Not Null

Tabel 3.36 Struktur tabel tr_bidang_keahlian

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Keahlian	Text	Null
3	Level	Varchar(30)	Null
4	Idsiswa	Varchar(50)	Null
5	Deskripsi	Text	Null
6	Userid_created	Varchar(20)	Not Null
7	Userid_updated	Varchar(20)	Null
8	Created_at	Datetime	Not Null
9	Updated_at	Datetime	Null

- Tabel tr_lamaran

Tabel ini digunakan untuk menyimpan data lamaran yang dilakukan oleh siswa atau alumni pada suatu lowongan pekerjaan tertentu. Struktur tabel tr_lamaran tersaji pada tabel 3.37 berikut.

Tabel 3.37 Struktur tabel tr_lamaran

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Idsiswa	Varchar(50)	Not Null
3	Idlowongan	Varchar(50)	Not Null
4	Ishadir	Char(1)	Null
5	Email	Varchar(50)	Not Null
6	Userid_created	Varchar(20)	Not Null
7	Userid_updated	Varchar(20)	Null
8	Created_at	Datetime	Not Null
9	Updated_at	Datetime	Null

- Tabel tr_nilai_akademik

Tabel ini digunakan untuk menyimpan data nilai akademik siswa atau alumni. Banyak lduka yang mensyaratkan nilai akademik untuk suatu lowongan tertentu. Struktur tabel tr_nilai_akademik tersaji pada tabel 3.38 berikut.

Tabel 3.38 Struktur tabel tr_nilai_akademik

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Idsiswa	Varchar(50)	Not Null, Foreign Key
3	Idmapel	Varchar(50)	Not Null, Foreign Key
4	Nilai	Float	Not Null
5	Userid_created	Varchar(20)	Not Null
6	Userid_updated	Varchar(20)	Null
7	Created_at	Datetime	Not Null
8	Updated_at	Datetime	Null
9	Tahun	Varchar(4)	Null

Tabel ini berelasi dengan tabel siswa dan mata pelajaran, artinya data yang diisikan pada tabel ini menjadi milik siswa dan mengacu pada mata pelajaran tertentu.

- Tabel tr_panggilan_tes

Tabel ini digunakan untuk menyimpan data panggilan tes oleh Iduka pada siswa atau alumni yang telah *apply* lowongan pekerjaan. Panggilan tes menjadi bagian dari proses rekrutmen yang dilakukan oleh Iduka. Struktur tabel tr_panggilan_tes tersaji pada tabel 3.39 berikut.

Tabel 3.39 Struktur tabel tr_panggilan_tes

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Idpanggilan	Varchar(50)	Not Null
3	Ishadir	Char(1)	Not Null
4	Status	Varchar(30)	Null
5	Userid_created	Varchar(20)	Not Null
6	Userid_updated	Varchar(20)	Null
7	Created_at	Datetime	Not Null
8	Updated_at	Datetime	Null
9	Tanggal	Date	Not Null
10	Waktu_mulai	Time	Not Null
11	Waktu_selesai	Time	Null
12	Idlowongan	Varchar(50)	Not Null

IDUKA dapat melakukan panggilan lebih dari satu dalam satu lowongan sampai dipilih calon tenaga kerja yang lulus pada semua tahapan panggilan tes. Pada tiap panggilan tes, dilakukan seleksi sehingga dapat diketahui siapa yang lolos dan gagal.

- Tabel tr_pendaftar_perusahaan

Tabel ini digunakan digunakan untuk menyimpan data Iduka yang melakukan registrasi secara mandiri melalui aplikasi pusat karir SMK. Selain mendaftar secara

mandiri, data iduka dapat diisi secara manual oleh admin pusat karir atau BKK. Struktur database tr_pendaftar_perusahaan tersaji pada tabel 3.40 berikut.

Tabel 3.40 Struktur tabel tr_pendaftar_perusahaan

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Nama	Varachar(50)	Not Null
3	Email	Varchar(50)	Not Null
4	Bidang	Varchar(50)	Not Null
5	Password_real	Varachar(100)	Not Null
6	Password_en	Varchar(50)	Not Null
7	Tglpendaftaran	Date	Not Null
8	Userid_created	Varchar(20)	Not Null
9	Userid_updated	Varchar(20)	Null
10	Created_at	Datetime	Not Null
11	Update_at	Datetime	Null
12	Status	Char(1)	Not Null

- Tabel tr_riwayat_pendidikan

Tabel ini digunakan untuk menyimpan riwayat pendidikan siswa atau alumni. Data ini digunakan pada pengisian data CV terkait pendidikan yang telah ditempuh. Struktur tabel tr_riwayat_pendidikan tersaji pada tabel 3.41 berikut.

Tabel 3.41 Struktur tabel tr_riwayat_pendidikan

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Instansi	Varchar(50)	Not Null
3	Jenjang	Varchar(30)	Not Null
4	Idjurusan	Int(6)	Not Null, Foreign Key

No	Nama Field	Tipe data (Length)	Keterangan
5	Tahun_masuk	Varchar(4)	Not Null
6	Tahun_lulus	Varchar(4)	Null
7	Userid_created	Varchar(20)	Not Null
8	Userid_updated	Varchar(20)	Null
9	Created_at	Datetime	Not Null
10	Update_at	Datetime	Null
11	Idsiswa	Varchar(50)	Not Null
12	Namajenjang	Varchar(50)	Null

Tabel ini berelasi dengan tabel jurusan dan tabel siswa. Data pada tabel ini digunakan oleh siswa dengan data jurusan diambil dari tabel jurusan.

- Tabel tr_tracer_study

Tabel ini digunakan untuk menyimpan data tracer study alumni yang dilakukan oleh sekolah atau BKK. Data awal *tracer study* dikumpulkan dari data siswa, dan tracer study pertama dilakukan pada siswa kelas XII. Struktur tabel tr_tracer_study tersaji pada tabel 3.42 berikut.

Tabel 3.42 Struktur tabel tr_tracer_study

No	Nama Field	Tipe data (Length)	Keterangan
1	id	varchar(50)	Primary key dan Not Null
2	Idpendaftar	Varchar(50)	Not Null
3	Tahun_lulus	Char(4)	Null
4	Idaktivitas	Int(11)	Not Null
5	Userid_created	Varchar(20)	Not Null
6	Userid_updated	Varchar(20)	Null
7	Created_at	Datetime	Not Null

No	Nama Field	Tipe data (Length)	Keterangan
8	Updated_at	Datetime	Null
9	Idkategoriperusahaan	Varchar(50)	Not Null
10	Posisi_perusahaan	Varchar(50)	Null
11	Idstatuskontrak	Int(6)	Not Null
12	Ispekerjaanpertama	Char(1)	Not Null
13	Idrelevan	Int(2)	Not Null
14	Isluar	Char(1)	Not Null
15	Idkota	Varchar(20)	Null
16	Tglmulaikerja	Date	Not Null
17	Idgaji	Int(2)	Not Null
18	Isjuranrelevan	Char(1)	Not Null
19	Nama_perusahaan	Varchar(50)	Not Null
20	Idperolehinfo	Int(2)	Not Null
21	Kritik_saran	Text	Null
22	Kualitas_pendidikan	Varchar(10)	Null
23	Kualitas_pembelajaran	Varchar(10)	Null
24	Kualitas_fasilitas	Varchar(10)	Null
25	Kualitas_lab	Varchar(10)	Null
26	Kualitas_guru	Varchar(10)	Null
27	Kualitas_keterampilan	Varchar(10)	Not Null
28	Kualitas_kedisiplinan	Varchar(10)	Not Null
29	Kualitas_kebersihan	Varachar(10)	Not Null
30	Kualitas_bkk	Varchar(10)	Not Null
31	Kualitas_sistem	Varchar(10)	Not Null
32	Nama_pt	Varchar(100)	Not Null

No	Nama Field	Tipe data (Length)	Keterangan
33	Idjenjang	Int(6)	Null
34	Idjurusan	Int(6)	Null
35	Idkotapt	Varachar(20)	Null
36	Tahun_masuk_pt	Char(4)	Not Null
37	Islinear	Char(1)	Not Null
38	Deskripsi_pt	Text	Null
39	Nama_usaha	Varchar(100)	Null
40	Idbidangusaha	Varchar(50)	Null
41	Idkotausaha	Varchar(20)	Null
42	Tglmulaiusaha	Date	Null
43	Islinerusaha	Char(1)	Null
44	Idomsetusaha	Int(2)	Null
45	Idrelevanusaha	Int(2)	Null
46	Deskripsi_usaha	Text	Null
47	Id_karirlainnya	Int(11)	Null
48	Nama_instansi	Varchar(100)	Null
49	Tglmulaikarir	Date	Null
50	Tglselesaikarir	Date	Null

Tabel ini memiliki paling banyak kolom yakni 50. Tabel ini menyimpan data tracer study yang terdiri bekerja, berusaha dan melanjutkan ke jejang yang lebih tinggi. Selain itu, siswa dan alumni juga diwajibkan mengisi data kepuasan selama sekolah.

- Tabel users

Tabel ini digunakan untuk menyimpan data semua pengguna pada aplikasi pusat karir SMK dari semua level. Tabel ini menyimpan data penting yaitu username dan password dari semua pengguna. Struktur tabel users tersaji pada tabel 3.43 berikut.

Tabel 3.43 Struktur tabel users

No	Nama Field	Tipe data (Length)	Keterangan
1	id	Int(11)	Primary Key, Auto_Increment
2	Userid	Varchar(50)	Not Null
3	Nama	Varchar(50)	Not Null
4	Email	Varachar(50)	Not Null
5	Password	Varchar(50)	Not Null
6	Remember_token	Varchar(50)	Not Null
7	Created_at	Datetime	Not Null
8	Updated_at	Datetime	Null
9	Last_login	Datetime	Not Null
10	Level	Varchar(50)	Not Null
11	Isaktif	Char(1)	Not Null
12	Ip_address	Varchar(50)	Not Null
13	Browser	Varchar(50)	Not Null
14	Userid_created	Varchar(20)	Not Null
15	Userid_updated	Varchar(20)	Null
16	Idperusahaan	Varchar(50)	Null
17	Idbkk	Varchar(50)	Null

Data password yang disimpan pada tabel ini dalam keadaan terenkripsi sehingga tidak dapat secara langsung. Tabel ini digunakan dalam proses login, oleh karena itu beberapa informasi seperti alamat IP, login terakhir dan browser yang digunakan juga disimpan dalam tabel.

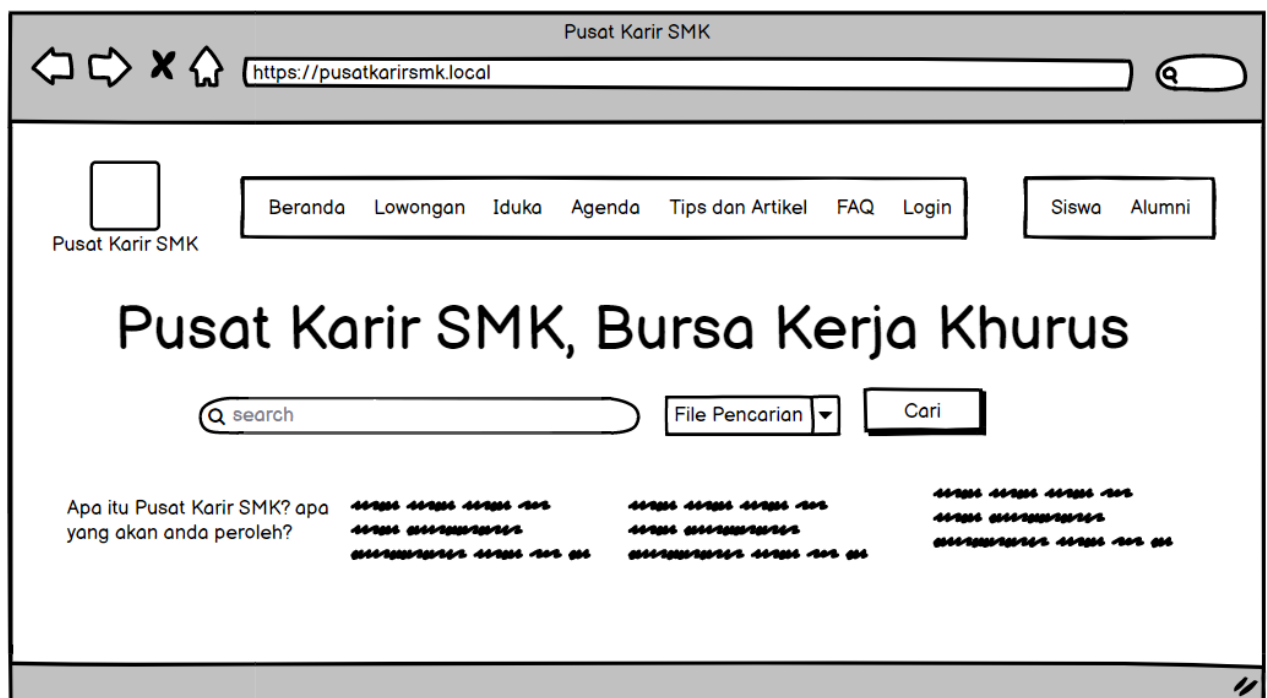
3.4 Desain Antarmuka

Salah satu hasil dari analisis kebutuhan perangkat lunak adalah kebutuhan antar muka (*interface requirement*) yang berfungsi sebagai penghubung antara pengguna aplikasi

dengan aplikasi dan database. Selain tampilan antar muka, pada desain antarmuka ini juga dijelaskan interaksi dan hubungan antara satu tampilan dengan tampilan lain, baik melalui tautan atau tombol. Karena hasil desain antarmuka digunakan oleh pengguna, maka harus mudah dipahami dan mudah digunakan. Dalam pengembangan aplikasi ini, desain antarmuka yang dilakukan sangat terbantu dengan *template bootstrap* yang telah menyediakan paket tampilan siap pakai dengan sedikit perubahan. Karena aplikasi ini lebih banyak berhubungan tampilan data, entri data, hapus data maka tidak diperlukan tampilan yang terlalu rumit.

3.4.1 Antarmuka Halaman Utama

Desain tampilan dibuat dengan menggunakan Balsamiq Mockup (<https://balsamiq.com/>) yang dapat digunakan secara *cloud* maupun di-*install* di komputer. Pembuatan *wireframe* atau desain antarmuka dapat dilakukan dengan mudah karena telah tersedia *toolbox* siap pakai. Cara menggunakannya adalah dengan *drag and drop*.



Gambar 3.10 Tampilan Utama Pusat Karir SMK

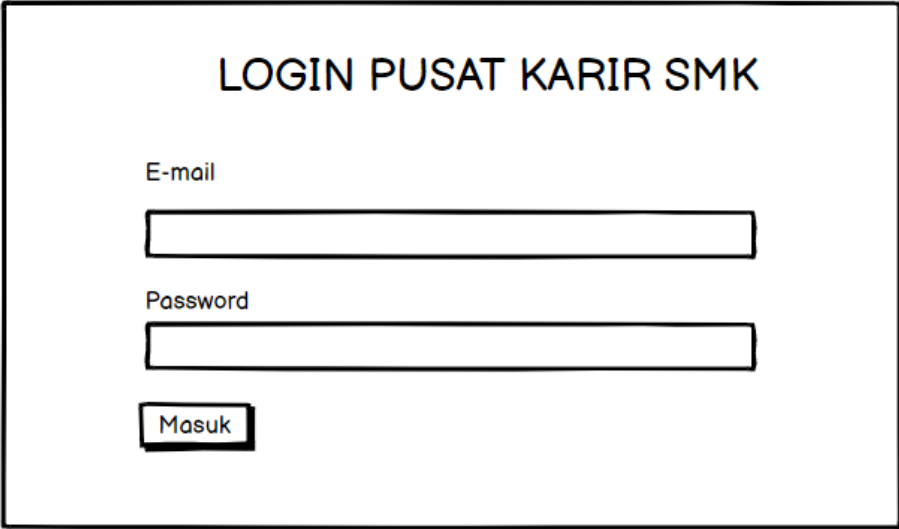
Gambar 3.11 adalah tampilan desain untuk halaman utama aplikasi pusat karir SMK. Desain tidak ditampilkan secara detail, yang penting dapat memberikan gambar besar tampilan aplikasi. Informasi yang ditampilkan pada halaman utama yaitu:

- Logo pusat karir SMK
- Tautan utama aplikasi yaitu Beranda, Lowongan, Iduka, Agenda, Tips dan Artikel, FAQ, dan Login.
- Tautan ke portal siswa dan alumni
- Pencarian data dengan kriteria tertentu

Informasi lain dapat ditambahkan agar tampilan utama menjadi lebih menarik pengguna sehingga pengguna mendapatkan informasi lebih.

3.4.2 Antarmuka Login

Desain tampilan ini digunakan untuk laman login aplikasi. Terdapat dua *textbox* yang digunakan oleh pengguna untuk memasukkan *username* dan *password* pengguna. *Username* yang digunakan adalah alamat email sedangkan *password* sesuai yang telah ditentukan.



LOGIN PUSAT KARIR SMK

E-mail

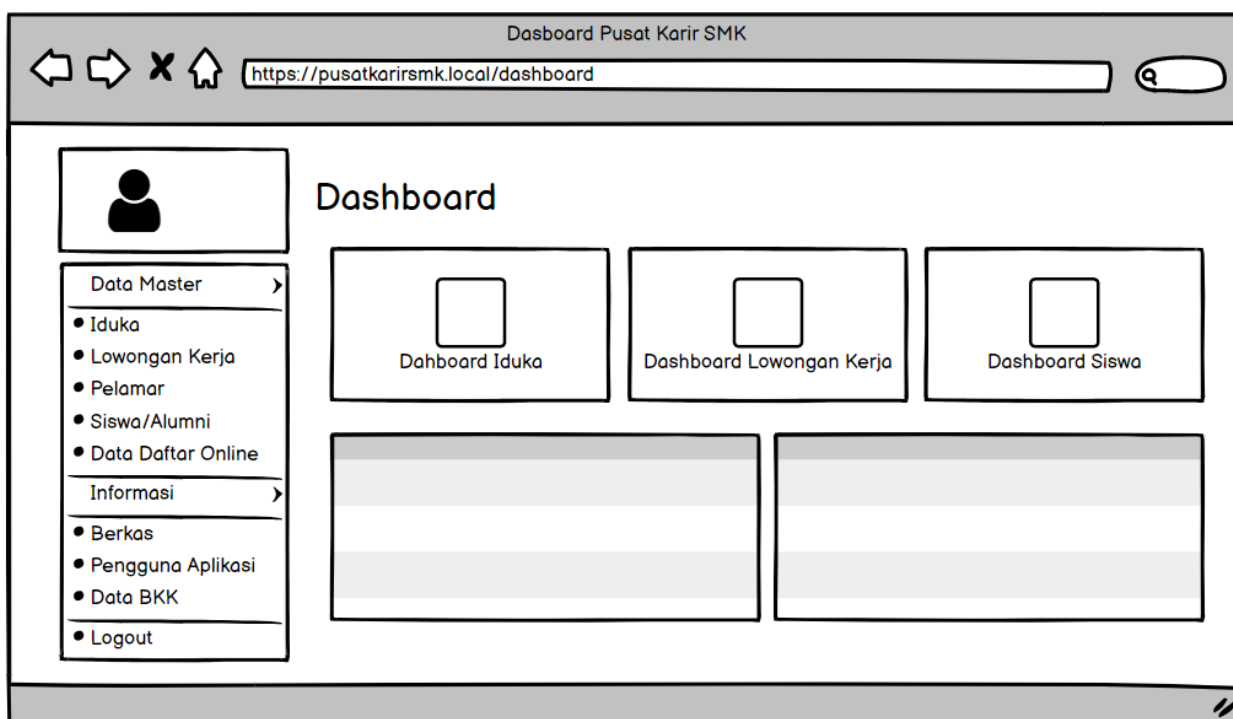
Password

Gambar 3.11 Tampilan Form Login

Tombol login digunakan untuk melakukan proses login sesuai *username* dan *password* yang telah dimasukkan. Jika *username* atau *password* tidak cocok dengan yang tersimpan di database, maka ditampilkan pesan *error login*. Jika login berhasil, maka ditampilkan laman *dashboard* sesuai dengan level aksesnya. Tampilan login digambar 3.12 berlaku untuk semua pengguna, baik siswa, alumni, IDUKA, BKK, dan pusat karir SMK.

3.4.3 Dashboard aplikasi

Desain tampilan pada gambar xx digunakan untuk laman dashboard pengguna yang telah login untuk level akses pusat karir dan BKK. Foto dengan ikon *people* menunjukkan pengguna yang sedang login. Menu atau tautan yang ditampilkan berbeda antara level akses karena memiliki hak akses yang berbeda-beda. Menu yang ada tanda “>” artinya memiliki submenu yang dapat ditampilkan dan tidak ditampilkan. Tampilan berbentuk kotak adalah data yang ditampilkan pada *dashboard* sedangkan yang berbentuk persegi panjang digunakan untuk data dalam bentuk tabel. Tampilan *dashboard* aplikasi dapat dilihat pada gambar 3.13.



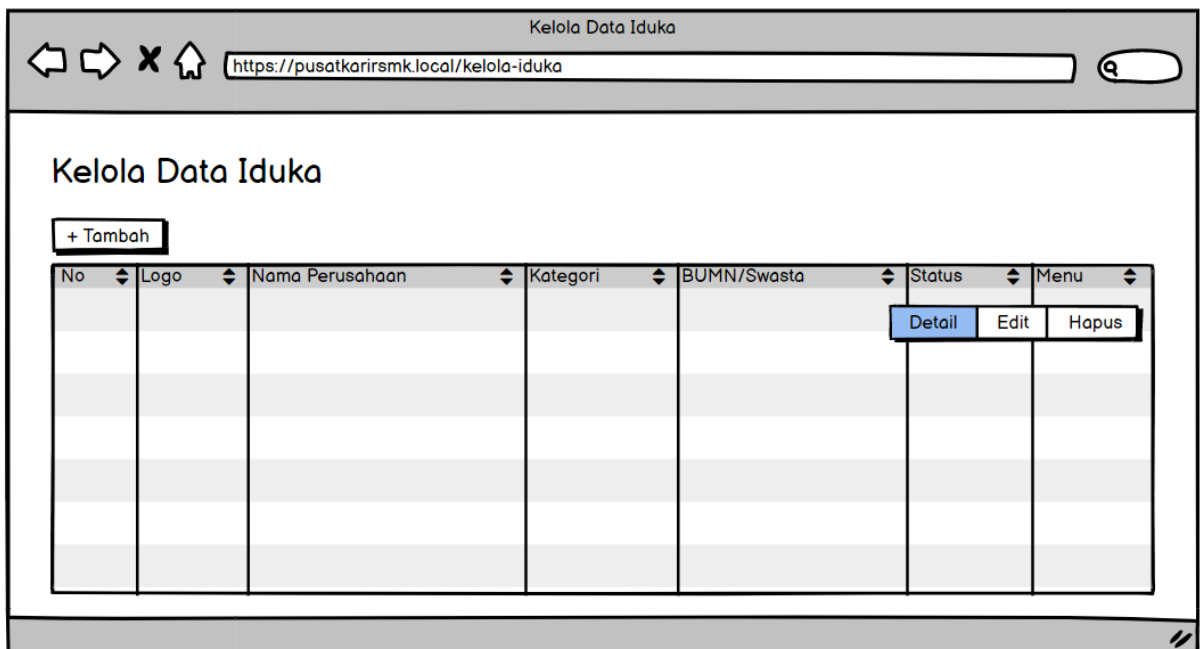
Gambar 3.12 Dashboard Aplikasi

Data yang ditampilkan pada *dashboard* merupakan data rekapitulasi dan transaksi bertujuan membantu pengguna untuk membuat keputusan secara cepat. Yang terpenting dalam *dashboard* adalah kemudahan membaca data dengan tampilan yang menarik. Beberapa data yang ditampilkan pada *dashboard* dapat diklik dan pengguna dibawa ke tujuan data. Beberapa hal yang harus diperhatikan pada dashboard antara lain:

- Data yang ditampilkan pada *dashboard* harus sesuai dengan yang dibutuhkan oleh pengguna.
- Visual hirarki, artinya pada bagian atas ditampilkan informasi secara keseluruhan dan semakin kebawah semakin detail.
- Jangan terlalu berlebihan
- Gunakan visualisasi yang tepat
- Utamakan kemudahan membaca data dibanding tampilan yang indah.

3.4.4 Kelola Data

Kelola data banyak digunakan dalam aplikasi pusat karir. Tujuan dari Kelola data adalah untuk CRUD (Create, Read, Update, Delete). Tampilan dan alur kelola data hamper sama satu dengan yang lain, hanya beda pada data yang yang dioleh.



Gambar 3.13 Tampilan Kelola Data

Laman utama dari kelola data adalah tampil data dengan fitur yang menyertainya antara lain: filter data, pengurutan, paginasi. Pada laman data juga terdapat tombol untuk operasional kelola data yaitu:

- Tambah, digunakan untuk menampilkan laman tambah data.
- Detail, digunakan untuk menampilkan laman detail data sesuai dengan data yang dipilih.

- Edit, digunakan untuk menampilkan laman edit data sesuai dengan data yang dipilih.
- Hapus, digunakan untuk hapus data.
- Cetak, digunakan untuk cetak data dalam bentuk print form.
- Export Excel, digunakan untuk export data menjadi format Excel
- Export PDF, digunakan untuk cetak data dalam format PDF
- Export CSV, digunakan untuk export data menjadi format CSV.

Tidak semua tombol tersebut disediakan oleh pengembang perangkat lunak, disesuaikan dengan kebutuhan. Tombol yang pasti ada adalah Tambah, Detail, Edit dan Hapus.

BAB 4

IMPLEMENTASI PUSAT KARIR

Pada bab ini, dibahas implementasi atau penulisan kode program dalam membangun aplikasi pusat karir SMK. Tidak semua kode program dimasukkan pada pembahasan ini, hanya yang penting dan potongan kode program saja.

4.1 Persiapan Lingkungan Pengembangan

Langkah pertama dalam pengembangan perangkat lunak adalah menyiapkan lingkungan pengembangan. Aplikasi pusat karir dikembangkan menggunakan teknologi *open-source* yakni Apache, PHP, dan MySQL. Persiapan yang dilakukan adalah menyiapkan komputer agar dapat digunakan untuk pengembangan aplikasi. Langkah tercepat dalam menyiapkan komputer agar mendukung Apache, PHP dan MySQL adalah menggunakan XAMPP. Dengan menggunakan XAMPP, proses instalasi dapat dilakukan dengan mudah dan cepat.



Gambar 4.1 Website XAMPP

Xampp adalah perangkat lunak bebas yang mendukung banyak sistem operasi. Xampp merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (localhost) yang terdiri atas program Apache HTTP Server, MySQL database dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama Xampp merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas. Xampp merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis.

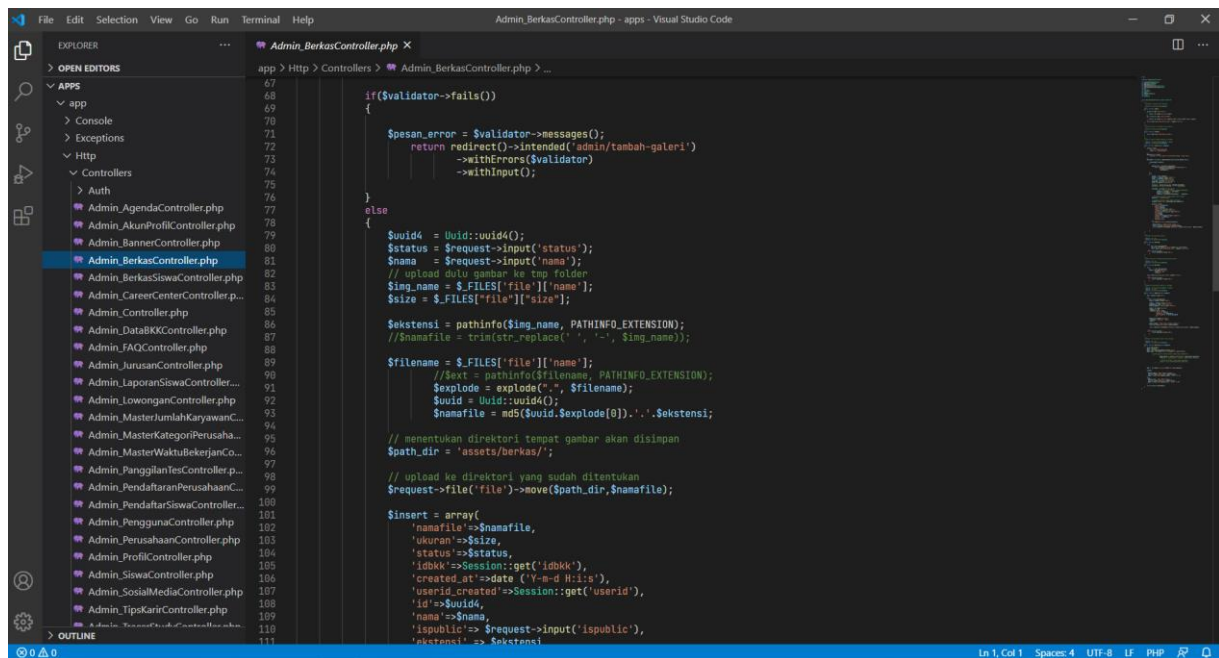
4.2 Editor dan IDE

Banyak editor dan IDE (*Integrated Development Environment*) yang digunakan dalam pengembangan aplikasi web. Beberapa cukup populer yaitu SublimeText, Notepad++, Atom, Vim, Bracket, PHP Storm, dan Visual Studio Code. Tiap editor dan IDE memiliki kelebihan dan kelemahan masing-masing, dan juga ada yang berbayar dan tidak berbayar. Tidak ada editor dan IDE terbaik, yang ada adalah yang paling sesuai dengan kebutuhan. Beberapa alasan dalam memilih editor yaitu:

- **Simple, cepat dan ringan**, membutuhkan waktu loading yang cukup berat apalagi untuk komputer dengan spec sederhana.
- **Bracket Matching**, digunakan untuk menampilkan pasangan untuk sintak coding, fungsi ini sangat berguna bagi programmer saat membaca coding sebab coding terkadang bercabang-cabang dan kita perlu mengetahui kapan blok sintax dimulai dan dimana berakhirnya.
- **Syntax Highlighting**, digunakan untuk menampilkan warna pada coding sehingga memudahkan pembacaan perintah-perintah bahasa pemrograman.
- **Syntax Folding**, digunakan untuk menyembunyikan blok coding tertentu sehingga akan meringkas tampilan coding dan programmer tidak perlu melihat coding secara keseluruhan, cukup blok yang dibutuhkan.
- **AutoComplete**, digunakan untuk mempermudah programmer dalam menulis perintah-perintah bahasa pemrograman.
- **Function List**, digunakan untuk menampilkan daftar function dalam suatu file, sehingga memudahkan pencarian function tertentu.

Dalam pengembangan aplikasi pusat karir SMK, digunakan editor VSCode (Visual Studio Code) dan Notepad++. Kedua editor tersebut dapat digunakan secara gratis, dan

langsung dapat diunduh dari Internet. Sampai panduan teknis ini dibuat, versi terbaru dari VSCode adalah 1.51. Tampilan editor visual studio dapat dilihat pada gambar 4.2 berikut.



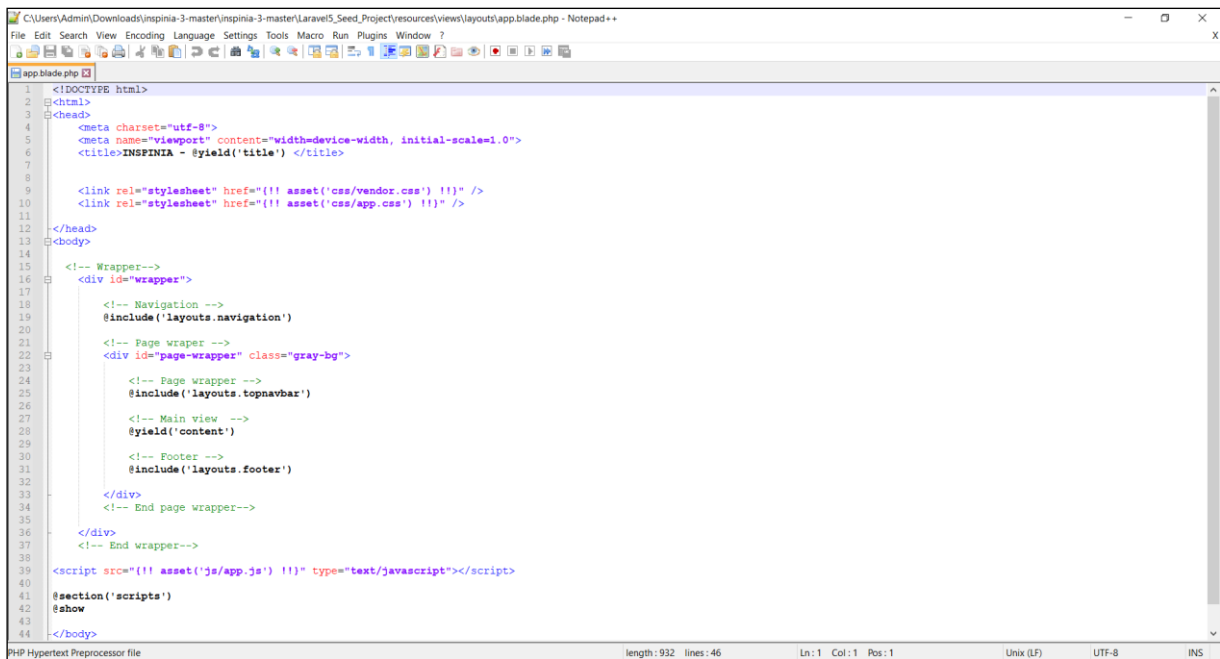
Gambar 4.2 Editor Visual Studio Code

Beberapa alasan menggunakan VSCode antara lain:

- Gratis, Berbeda dengan Sublime Text yang sebenarnya berbayar walaupun bisa digunakan secara gratis, VS Code dapat digunakan secara gratis seperti atom. VSCode dapat diunduh melalui <https://code.visualstudio.com>.
- Terintegrasi dengan Emmet, adalah plugin untuk memudahkan kita dalam membuat markup html, di VS Code secara default sudah terpasang emmet tanpa melakukan instalasi lagi.
- Terintegrasi dengan Git, sama halnya dengan Atom, secara default VS Code sudah terintegrasi dengan Git, dan icon "branch" berada disidebar sebelah kiri. VS Code akan mencatat semua perubahan yang kita lakukan jika project kita menggunakan git.
- Terdapat terminal, secara default terminal atau prompt sudah ada didalam VS Code, berbeda dengan Atom yang harus di install extensionnya lagi. Terminal sangat berguna jika dalam proses penulisan kode program memerlukan akses ke Terminal.
- Fitur AutoComplete, Ini adalah fitur yang sangat berguna buat pemula, karena fitur ini dapat mengenerate secara otomatis kode yang akan ditulis.
- Cross Platform, VS Code bisa berjalan lintas sistem operasi, baik Windows, Linux, maupun OSX.

- g. Extensions yang berlimpah, VSCode memiliki banyak extension, sehingga dapat menambahkan fitur tambahan dari VS Code yang ada dalam repository mereka. Extension dapat diakses melalui sidebar sebelah kiri.

Selain menggunakan VSCode, dalam pengembangan aplikasi ini juga menggunakan editor Notepad++. Penggunaan Notepad++ hanya digunakan untuk revisi kecil pada kode program. Notepad++ merupakan salah satu editor yang banyak digunakan dalam penulisan kode program. Fitur yang ditawarkan sudah sangat mencukupi untuk kebutuhan standar dalam pengembangan aplikasi berbasis web. Tampilan Notepad++ dapat dilihat pada gambar 4.3 berikut.

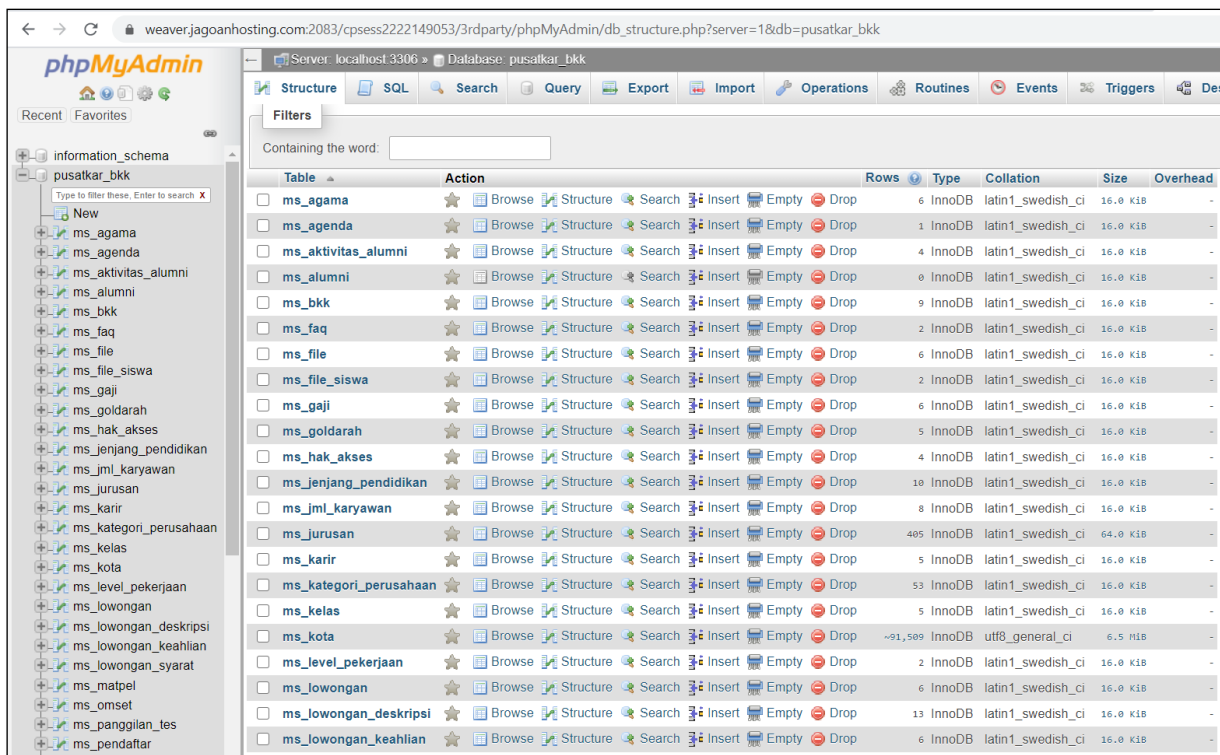


Gambar 4.3 Editor Notepad++

Notepad++ hanya tersedia untuk sistem operasi Windows dan dapat diunduh secara gratis pada alamat <https://notepad-plus-plus.org/downloads/v7.9.1/>. Ukuran file tidak terlalu besar yakni kurang dari 4 MB. Proses instalasi Notepad++ dapat dilakukan cepat, bahkan pada komputer dengan spesifikasi minimum. Setelah terpasang, maka Notepad++ dapat langsung digunakan menuliskan kode program. Untuk menggunakan fitur lebih, dapat dilakukan seting dan instalasi plugin.

4.3 Tool Database

Banyak alat bantu (tools) untuk pengelolaan database yaitu SQL Yog, Navicat, DataGrip, HeidiSQL, phpMyAdmin, dan lain-lain. alat bantu database tersebut ada yang berbayar dan gratis dengan segala kelebihan dan kekurangan masing-masing. Dalam pengembangan aplikasi pusat karir ini, tools utama dalam pengelolaan database adalah phpMyAdmin, selain itu juga menggunakan HeidiSQL. Kedua tools tersebut bersifat gratis dan mendukung DBMS MySQL. Selain gratis, alasan lain menggunakan PHPMyAdmin adalah berbasis web sehingga dapat dibuka dimana saja. Apalagi phpMyAdmin juga merupakan tools default yang disediakan oleh penyedia hosting yang digunakan oleh pusat karir SMK.



Gambar 4.4 Tool MySQL PHPMyAdmin

phpMyAdmin dapat diunduh secara gratis di <https://www.phpmyadmin.net/> kemudian pilih tautan download. Unduh versi terbaru dan sesuai dengan pilihan bahasa. PHPMyAdmin yang mendukung banyak bahasa memiliki ukuran file yang lebih besar. Secara default, bahasa yang digunakan adalah *English*. Beberapa fitur unggulan phpMyAdmin sehingga menjadi pilihan antara lain:

- phpMyAdmin memiliki interface yang user-friendly dan intuitive yang mudah dipelajari. Dengan begitu, orang-orang dapat mempelajari dan membiasakan diri dengan fitur-fiturnya dengan mudah dan cepat.
- phpMyAdmin memperbolehkan user memanfaatkan kebanyakan fungsi MySQL, termasuk mengelola database, menjalankan queries MySQL, mengeksekusi statement MySQL, mengelola user dan permission dan masih banyak lagi.
- Fitur Import dan Export data dari dan ke berbagai format.
- phpMyAdmin bisa membuat graphics database dalam bentuk PDF, bersamaan dengan beberapa queries yang kompleks dibantu dengan query-by-example.
- Membuat, menghapus, mengelola user

Released 2020-10-15, see [release notes](#) for details.

Current version compatible with PHP 7.1 and newer and MySQL 5.5 and newer.

File	Size	Verification
phpMyAdmin-5.0.4-all-languages.zip	13.7 MB	[PGP] [SHA256]
phpMyAdmin-5.0.4-all-languages.tar.gz	12.3 MB	[PGP] [SHA256]
phpMyAdmin-5.0.4-all-languages.tar.xz	7.7 MB	[PGP] [SHA256]
phpMyAdmin-5.0.4-english.tar.gz	7.7 MB	[PGP] [SHA256]
phpMyAdmin-5.0.4-english.tar.xz	5.7 MB	[PGP] [SHA256]
phpMyAdmin-5.0.4-english.zip	9.2 MB	[PGP] [SHA256]
phpMyAdmin-5.0.4-source.tar.xz	14.4 MB	[PGP] [SHA256]

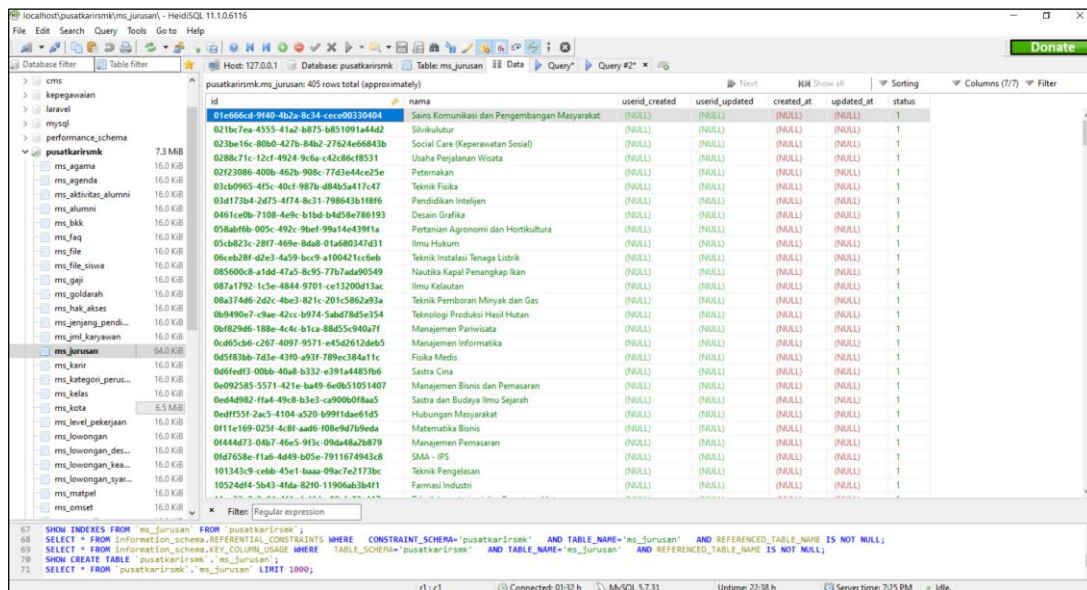
Development Versions

phpMyAdmin [Git](#) repository is located at <https://github.com/phpmyadmin/phpmyadmin.git> and you can browse them online using [GitHub](#). Note that phpMyAdmin uses Composer to manage library dependencies, when using git development versions you must manually run Composer. Please see [the documentation](#) for details.

More information about using Git is available on [development page](#).

Gambar 4.5 Unduh phpMyAdmin

Karena berbasis web, maka PHPMyAdmin menjadi kurang responsif dan tidak punya fitur *AutoCompletes*. Berdasarkan kekurangan tersebut, digunakan juga HeidiSQL. Untuk perintah query yang memerlukan fitur *AutoComplete* dan pengelolaan data sering menggunakan HeidiSQL dibandingkan phpMyAdmin.



Gambar 4.6 Tampilan HeidiSQL

4.4 Implementasi Aplikasi

4.4.1 Struktur Direktori

Aplikasi pusat karir SMK dibangun menggunakan framework Laravel versi 6.x. Alasan menggunakan Laravel 6.x adalah dukungan LTS (Long Term Support), fitur pada Laravel, dan masalah kompatibilitas dengan PHP. Laravel 6.x merupakan salah satu versi yang mendapat dukungan LTS, artinya mendapat dukungan jangka Panjang. Berdasarkan informasi resmi dari website Laravel (<https://laravel.com/docs/6.x/releases>), akan mendapat dukungan sampai minggu ketiga bulan september 2022. Tahapan dalam pengembangan aplikasi pusat karir SMK terlihat pada gambar 4.7.

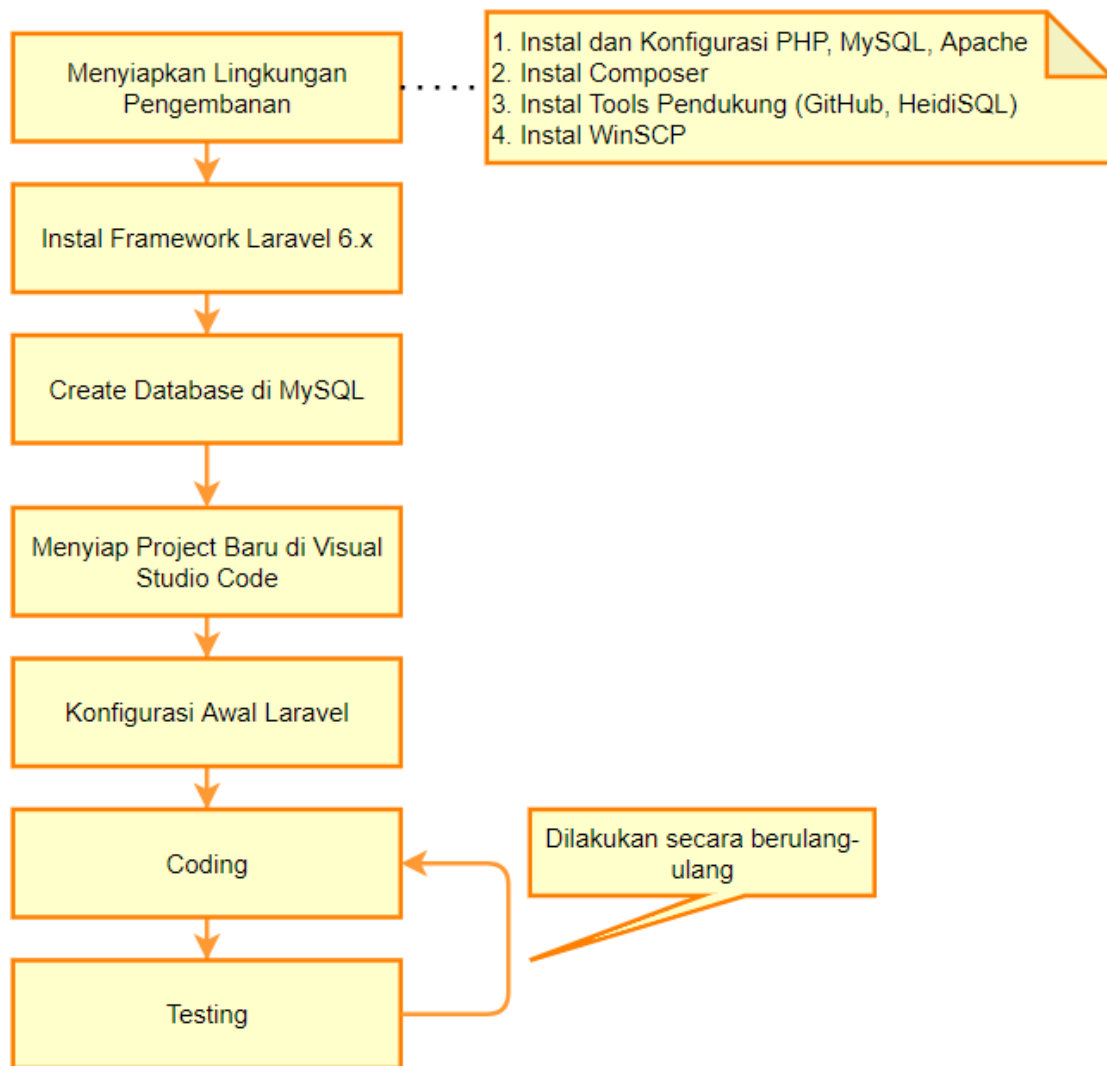
Penjelasan detail dari masing-masing tahapan gambar 4.7 adalah sebagai berikut:

1. Melakukan persialan lingkungan pengembangan

Pada tahap ini, dilakukan instalasi PHP, MySQL, Apache dan perangkat lunak lain yang dibutuhkan dalam pengembangan. Perangkat lunak lain yang di-install antara lain: Tools untuk GitHub, HeidiSQL, WinSCP, Composer, dan lain-lain.

2. Instalasi Framework Laravel 6.x

Karena instalasi Laravel menggunakan tools *Composer*, maka harus dipastikan tools tersebut telah terpasang dengan baik dan dapat digunakan. Perintah yang dituliskan di Terminal atau *Command Prompt* adalah: `composer create-project --prefer-dist laravel/laravel pusatkarirsmk "6.*"`



Gambar 4.7 Tahapan Pengembangan Pusat Karir SMK

Tunggu sampai proses instalasi Laravel selesai. Kecepatan instalasi sangat bergantung pada kecepatan koneksi Internet. Kalau sebelumnya sudah pernah melakukan instalasi Laravel, maka proses menjadi lebih cepat karena instalasi dilakukan melalui *cache*. setelah instalasi selesai, maka dilakukan pengujian melalui browser.

3. Membuat database di MySQL

Salah satu tahapan dalam konfigurasi dasar Laravel adalah terkait database yang terdiri dari: server database, port database, username, password, dan nama database. Nama database bisa diketahui ketika telah dibuat databasenya. Pada tahap ini, dibuat database di DBMS MySQL dengan menggunakan tools phpMyAdmin.

4. Menyiapkan project baru di Editor Visual Studio Code

Pada tahap ini, dibuat project di Visual Studi Code dengan menggunakan folder instalasi Laravel yakni pusatkarirsmk. Pada tahap ini juga dilakukan instalasi plug-In VSCode yang dapat membantu mempercepat pengembangan aplikasi.

5. Konfigurasi awal Laravel

Sebelum memulai penulisan kode program di VSCode, terlebih dahulu dilakukan konfigurasi pada framerok Laravel yaitu: *application key*, konfigurasi *Environment* (file *.env*), pengaturan direktori *public*, pengaturan *cache*, dan pengaturan *timezone&local* pada file *config/app.php*.

6. Testing Laravel

Pada tahap ini dilakukan pengujian terhadap framework Laravel yang telah dipasang dan dikonfigurasi. Untuk melakukan pengujian, dapat dilakukan dengan mengetikkan alamat URL <http://localhost/pusatkarirsmk/public> atau dengan mengetikkan: `php artisan serve`, kemudian buka pada URL <http://127.0.0.1:8000>. Tampilan hasil kedua URL yang digunakan terlihat pada gambar 4.8 berikut.



Gambar 4.8 Testing Laravel

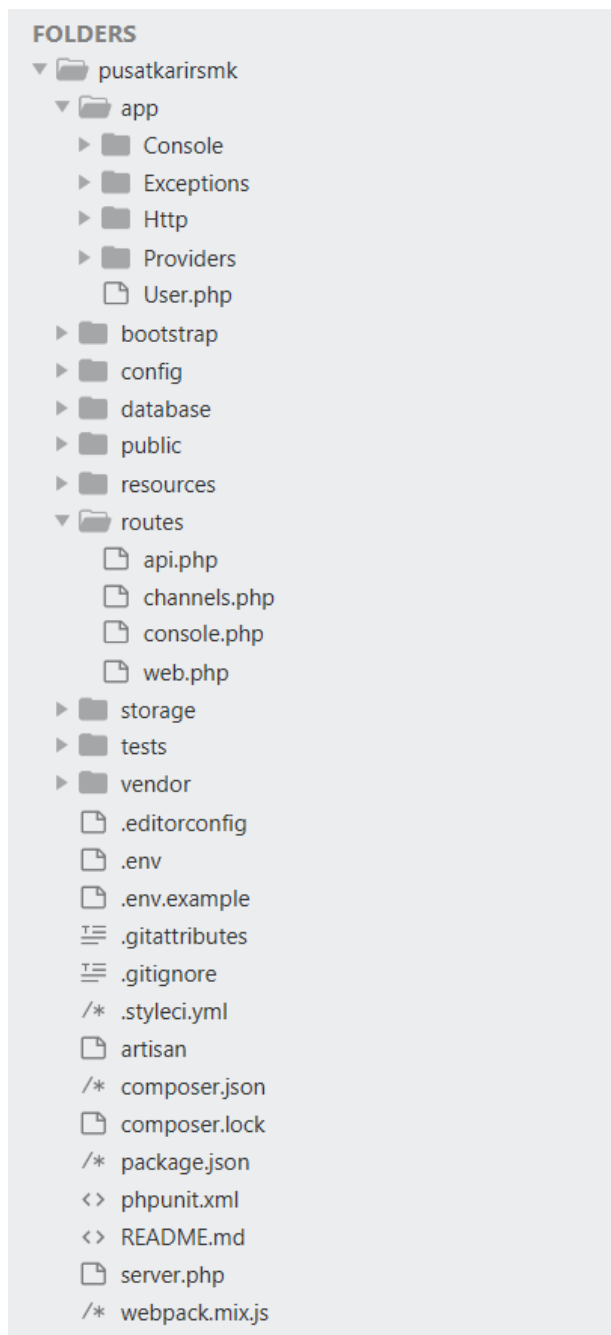
7. Coding & Pembuatan Tabel.

Ini adalah tahapan utama dan paling lama dalam pengembangan perangkat lunak. Pada tahap ini dilakukan penulisan kode program berdasarkan dokumen analisis kebutuhan. Pada tahap ini, penulisan kode bertujuan untuk implementasi proses bisnis dan desain tampilan untuk pengguna. Selain itu, dilakukan juga pembuatan tabel dan query. Kegiatan pada tahapan ini antara lain:

- Menulis kode program di framework Larvel
- Menulis kode program untuk tampilan aplikasi berbasis Bootstrap
- Menulis kode program untuk validasi, interaksi, dan pemrosesan disisi client menggunakan JavaScript dan JQuery.
- Menyiapkan assets aplikasi baik dalam bentuk gambar, ikon, fonts, dan yang lainnya.
- Membuat tabel
- Membuat query pada tabel
- Menulis kode program untuk Error Handling
- Unggah ke GitHub dan web hosting

8. Testing

Tahapan *testing* bertujuan agar program yang dihasilkan terbebas dari kesalahan (*zero bug*). Metode testing yang dilakukan pada tahapan ini lebih banyak pada *unit testing* dan *blackbox testing*. Ketika ditemukan *error*, maka langsung dilakukan perbaikan pada kode program. Testing dan perbaikan dilakukan terus-menerus dan berulang sampai tidak ditemukan *bug* baik secara aplikasi maupun secara *logic* data dan proses.

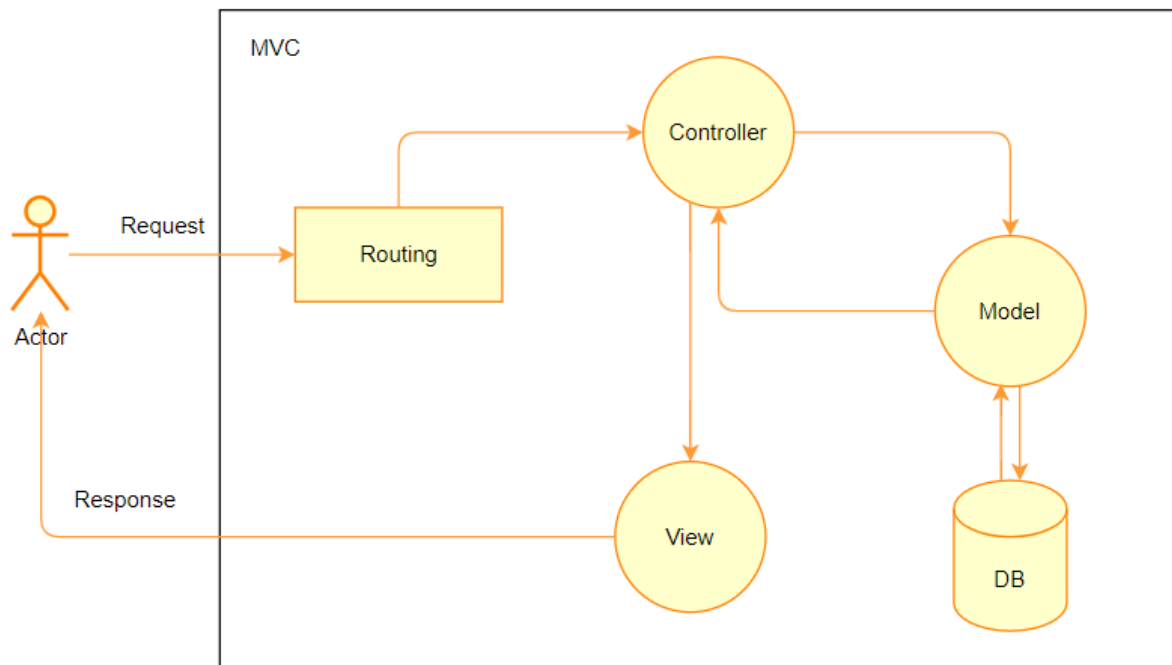


Gambar 4.9 Struktur direktori dan file di Laravel 6.x

4.4.2 Laravel Routing

Framework Laravel telah menerapkan konsep MVC (*Model View Controller*) seperti framework yang lain seperti CodeIgniter, Yii dan yang lain. Dalam MVC, terjadi pembagian dan pemisahan yang antar komponen aplikasi. Bagian logika hanya mengurus tugasnya, bagian tampilan juga hanya mengurus bagian tampilan. Akibatnya, kode program menjadi bersih karena tidak terjadi percampuran antara kode program untuk tampilan dengan kode

program untuk logika. Antar komponen dalam MVC saling berhubungan dalam memberikan data, manipulasi data, dan menggunakan data. Gambar 4.10 adalah alur aplikasi yang menggunakan konsep MVC.



Gambar 4.10 Routing dalam Model MVC

Berdasarkan gambar 4.10, ada enam (6) komponen yang terlibat dalam menerapkan konsep MVC yaitu (1) pengguna, (2) routing, (3) Controller, (4) Model, (5) view, (6) database. Antar komponen saling berhubungan dan berkomunikasi baik satu arah maupun duarah. Penjelasan dari gambar 4.10 adalah:

1. Pengguna mengirimkan request (permintaan)
2. Routing kemudian mengarahkan ke controller permintaan dari pengguna tersebut.
3. Controller melakukan request data ke Model
4. Model melakukan request ke database
5. Database memberikan respon ke Model
6. Model memberikan respon ke Controller
7. Controller mengirimkan data ke View
8. View memberikan respon ke pengguna dalam bentuk tampilan web sesuai yang diminta.

Routing berperan besar dalam siklus MVC karena komponen pertama yang menjawab permintaan (*request*) dari pengguna kemudian meneruskannya ke Controller, dan seterusnya. Di dalam folder *project laravel*, terdapat folder *routes*, dimana didalamnya terdapat 4 file yaitu *api.php*, *channels.php*, *console.php*, dan *web.php*. File yang digunakan dalam pengembangan aplikasi ini adalah *web.php*. Routing dalam pengembangan pusat karir SMK dapat dibagi menjadi beberapa yaitu routing umum, routing untuk IDUKA, routing untuk Admin, routing untuk Siswa, dan routing terkait halaman *error*. Gambar xx berikut adalah isi file *web.php* untuk routing umum.

```

15 date_default_timezone_set('Asia/Jakarta');
16
17 Route::get('/', 'Homepage_Controller@index');
18
19 Route::get('cari', ['as' => 'q', 'uses' => 'Homepage_Controller@cari']);
20
21 Route::get('portal/perusahaan', 'Homepage_Controller@portal_perusahaan');
22 Route::get('portal/perusahaan/login', 'Homepage_Controller@login_portal_perusahaan');
23 Route::get('portal/perusahaan/daftar', 'Homepage_Controller@daftar_perusahaan');
24 Route::post('portal/perusahaan/login', 'Authentication@login');
25
26 Route::post('portal/perusahaan/daftar', 'Homepbkk/siswa/tambah-tracer-studyage_PerusahaanPendaftaranController@store');
27
28 Route::get('lowongan', 'Homepage_LowonganController@index');
29 Route::get('lowongan/detail/{id}', 'Homepage_LowonganController@show');
30 Route::post('lowongan/filter/jenis', 'Homepage_LowonganController@filter_jenis');
31
32 Route::get('perusahaan', 'Homepage_PerusahaanController@index');
33 Route::get('perusahaan/detail/{id}', 'Homepage_PerusahaanController@show');
34 Route::post('perusahaan/filter/jenis', 'Homepage_PerusahaanController@filter_jenis');
35
36
37 Route::get('agenda', 'Homepage_AgendaController@index');
38 Route::get('agenda/detail/{id}', 'Homepage_AgendaController@show');
39
40 Route::get('tips-karir', 'Homepage_TipsKarirController@index');
41 Route::get('tips-karir/detail/{id}', 'Homepage_TipsKarirController@show');
42
43 Route::get('faq', 'Homepage_FAQController@index');
44
45 Route::get('siswa', 'HomepageSiswa_Controller@index');
46 Route::post('siswa/daftar', 'Homepage_PendaftaranSiswaController@store');
47 Route::post('siswa/login', 'Authentication@login');
48
49 Route::get('logout', function () { Session::flush(); return redirect()->intended('login'); });
50
51 Route::get('bkk/login', function () {
52     return View::make("home.login");
53 });

```

Gambar 4.11 Routing Aplikasi

Secara umum, penulisan route dlaravel adalah sebagai berikut:

```

Route::get('page/{page}/comments/{comment}', function ($page, $comment) {

```

```
//
});
```

Daftar routing yang tersedia antara lain:

- /, untuk halaman utama website
- Portal/perusahaan, *route* untuk perusahaan, yang terdiri dari: index, login, filter dan daftar.
- Agenda, *routing* untuk agenda kegiatan, yang terdiri dari: index dan detail agenda.
- Tips-karir, *routing* untuk tips, yang terdiri dari: index dan detail.
- Faq, *routing* untuk menampilkan FAQ
- Siswa, *routing* untuk siswa terdiri dari index, daftar dan login.
- Bkk/login, *routing* untuk login sebagai BKK.

Semua *route* menggunakan *method* *get*, dan ada beberapa yang berparameter, ada yang menggunakan alias, dan melakukan *redirect*. Gambar 4.12 berikut adalah *route* untuk pengguna dengan level akses IDUKA.

```
62 Route::group(['middleware' => 'role:I'], function () {
63     Route::get('iduka', 'IDUKA_Controller@index');
64
65     Route::get('iduka/pengguna', function () {
66         error_reporting(0);
67         return View::make("admin.dashboard.index");
68     });
69
70     Route::get('iduka/perusahaan', 'Admin_PerusahaanController@index');
71     Route::get('iduka/tambah-perusahaan', 'Admin_PerusahaanController@create');
72     Route::get('iduka/ubah-perusahaan/{id}', 'Admin_PerusahaanController@edit');
73     Route::post('iduka/simpan-perusahaan', 'Admin_PerusahaanController@store');
74     Route::post('iduka/hapus-perusahaan', 'Admin_PerusahaanController@destroy');
75     Route::post('iduka/update-perusahaan', 'Admin_PerusahaanController@update');
76     Route::get('iduka/detail-perusahaan/{id}', 'Admin_PerusahaanController@show');
77     Route::post('iduka/perusahaan/hapus-logo', 'Admin_PerusahaanController@destroy_logo');
78
79
80     Route::get('iduka/lowongan', 'Admin_LowonganController@index');
81     Route::get('iduka/tambah-lowongan', 'Admin_LowonganController@create');
82     Route::get('iduka/ubah-lowongan/{id}', 'Admin_LowonganController@edit');
83     Route::post('iduka/simpan-lowongan', 'Admin_LowonganController@store');
84     Route::post('iduka/hapus-lowongan', 'Admin_LowonganController@destroy');
85     Route::post('iduka/update-lowongan', 'Admin_LowonganController@update');
86     Route::get('iduka/detail-lowongan/{id}', 'Admin_LowonganController@show');
87     Route::post('iduka/tutup-lowongan', 'Admin_LowonganController@close_job');
88
89     Route::post('iduka/lowongan/simpan-deskripsi-pekerjaan', 'Admin_LowonganController@store_deskripsi');
90     Route::post('iduka/lowongan/deskripsi-pekerjaan/hapus', 'Admin_LowonganController@destroy_deskripsi');
91     Route::post('iduka/lowongan/syarat-pekerjaan/hapus', 'Admin_LowonganController@destroy_syarat');
92     Route::post('iduka/lowongan/keahlian-pekerjaan/hapus', 'Admin_LowonganController@destroy_keahlian');
93
94     Route::post('iduka/lowongan/simpan-syarat-pekerjaan', 'Admin_LowonganController@store_syarat');
95     Route::post('iduka/lowongan/simpan-keahlian-pekerjaan', 'Admin_LowonganController@store_keahlian');
96
97     //add to admin
98     Route::get('iduka/lowongan/lihat-pelamar/{id}', 'Admin_LowonganController@show_pelamar');
99     Route::get('iduka/detail-pelamar/{id}', 'Admin_LowonganController@detail_pelamar');
100    //Route::get('iduka/lowongan/pelamar/validasi', 'Admin_LowonganController@validasi');
101    Route::get('iduka/lowongan/panggilan-tes/{idlowongan}', 'Admin_LowonganController@set_panggilan_tes');
102    Route::post('iduka/lowongan/panggilan-tes/simpan', 'Admin_LowonganController@store_panggilan_tes');
```

Gambar 4.12 Routing Groups untuk Iduka

Middleware Laravel adalah penengah antara *request* yang masuk dengan *controller* yang dituju yang berpengaruh pada *request* dan *respons*. *Middleware* dibuat sebagai *filter* dari *request* yang masuk web. Sebagai contoh, *middleware* untuk memverifikasi apakah user pada website statusnya sudah login, jika belum login maka *middleware* akan mengarahkan user pada halaman login. Tapi jika user sudah login maka *middleware* akan mengarahkan ke halaman *dashboard*. Route pada gambar xx menggunakan *route groups* dan *middleware* dengan role I yang artinya IDUKA. Semua route akan berjalan jika pengguna telah melakukan proses login dengan level akses IDUKA. Semua route menggunakan method *get*, dan ada beberapa yang berparamter.

Sedangkan route dan *middleware* dengan level akses S atau siswa dapat dilihat pada gambar 4.13 berikut. Semua route akan berfungsi jika pengguna telah melakukan proses autentikasi atau login dengan level akses Siswa. Semua route menggunakan method *get*.

```

123 Route::group(['middleware' => 'role:S'], function () {
124
125     Route::get('siswa/berkas', 'Siswa_BerkasController@index');
126     Route::post('siswa/berkas/upload', 'Siswa_BerkasController@store');
127     Route::post('siswa/berkas/hapus', 'Siswa_BerkasController@destroy');
128
129     Route::get('siswa/profil', 'Siswa_ProfilController@index');
130     Route::post('siswa/profil/profil-dasar', 'Siswa_ProfilController@update_profil_dasar');
131     Route::post('siswa/profil/info-personal', 'Siswa_ProfilController@update_info_personal');
132     Route::post('siswa/profil/kontak-medsos', 'Siswa_ProfilController@update_kontak_medsos');
133     Route::post('siswa/profil/info-kesehatan', 'Siswa_ProfilController@update_info_kesehatan');
134     Route::post('siswa/profil/riwayat-pendidikan', 'Siswa_ProfilController@update_riwayat_pendidikan');
135     Route::post('siswa/profil/keahlian', 'Siswa_ProfilController@update_keahlian');
136     Route::post('siswa/profil/foto', 'Siswa_ProfilController@update_photo');
137
138     Route::post('siswa/lowongan/lamar-pekerjaan', 'Siswa_LamarPekerjaanController@store');
139
140     Route::post('siswa/profil/riwayat-pendidikan/hapus', 'Siswa_ProfilController@destroy_riwayat_pendidikan');
141     Route::post('siswa/profil/keahlian/hapus', 'Siswa_ProfilController@destroy_keahlian');
142
143     Route::get('siswa/profil/cetak/{idsiswa}', 'Siswa_ProfilController@cetak_cv');
144
145     Route::get('siswa/lamaran', 'Siswa_LamaranController@index');
146     Route::post('siswa/lowongan/lamar-pekerjaan/hapus', 'Siswa_LamaranController@destroy');
147
148     Route::get('siswa/panggilan-tes', 'Siswa_PanggilanTesController@index');
149
150     Route::get('siswa/tracer-study', 'Siswa_TracerStudyController@index');
151     Route::post('siswa/tracer-study/info-kelulusan/simpan', 'Siswa_TracerStudyController@store_info_kelulusan');
152     Route::post('siswa/tracer-study/info-pekerjaan/simpan', 'Siswa_TracerStudyController@store_info_pekerjaan');
153     Route::post('siswa/tracer-study/info-perkuliahan/simpan', 'Siswa_TracerStudyController@store_info_perkuliahan');
154     Route::post('siswa/tracer-study/info-wirusaha/simpan', 'Siswa_TracerStudyController@store_info_wirusaha');
155     Route::post('siswa/tracer-study/info-lainnya/simpan', 'Siswa_TracerStudyController@store_info_lainnya');
156     Route::post('siswa/tracer-study/info-kepuasan-sekolah/simpan', 'Siswa_TracerStudyController@store_kepuasan_sekolah');
157     Route::get('siswa/tracer-study/cetak/{idtracerstudy}', 'Siswa_TracerStudyController@cetak');
158     //Route::post('siswa/tracer-study/simpan', 'Siswa_TracerStudyController@store');
159
160     Route::get('siswa/logout', function () {
161         Session::flush();
162         return redirect()->intended('siswa/login');
163     });
164 }
165
166 );

```

Gambar 4.13 Routing Group untuk Siswa

Route dan *middleware* dengan level akses A atau Admin dapat dilihat pada gambar 4.14 berikut. Semua route akan berfungsi jika pengguna telah melakukan proses autentikasi atau login dengan level akses Siswa. Semua route menggunakan *method* get.

```

169 Route::group(['middleware' => 'role:A'], ['middleware' => 'role:BKK'], function () {
170     Route::get('bkk', 'Admin_Controller@index');
171
172
173     Route::get('bkk/lowongan/lihat-pelamar/{id}', 'Admin_LowonganController@show_pelamar');
174     Route::get('bkk/detail-pelamar/{id}', 'Admin_LowonganController@detail_pelamar');
175     Route::get('bkk/lowongan/panggilan-tes/{idlowongan}', 'Admin_LowonganController@set_panggilan_tes');
176     Route::post('bkk/lowongan/panggilan-tes/simpan', 'Admin_LowonganController@store_panggilan_tes');
177
178     Route::get('bkk/panggilan-tes', 'Admin_PanggilanTesController@index');
179     Route::get('bkk/panggilan-tes/detail/{id}', 'Admin_PanggilanTesController@show');
180     Route::get('bkk/panggilan-tes/seleksi/{id}', 'Admin_PanggilanTesController@seleksi');
181     Route::post('bkk/panggilan-tes/seleksi/validasi', 'Admin_PanggilanTesController@validasi');
182     Route::post('bkk/panggilan-tes/hapus', 'Admin_PanggilanTesController@destroy');
183
184
185     Route::get('bkk/pengguna', 'Admin_PenggunaController@index');
186     Route::get('bkk/tambah-pengguna', 'Admin_PenggunaController@create');
187     Route::post('bkk/simpan-pengguna', 'Admin_PenggunaController@store');
188     Route::get('bkk/ubah-pengguna/{id}', 'Admin_PenggunaController@edit');
189     Route::get('bkk/reset-password/{id}', 'Admin_PenggunaController@edit_password');
190     Route::post('bkk/update-pengguna', 'Admin_PenggunaController@update');
191     Route::post('bkk/hapus-pengguna', 'Admin_PenggunaController@destroy');
192     Route::post('bkk/update-password-pengguna', 'Admin_PenggunaController@update_password');
193     Route::get('bkk/pengguna/tambah/siswa/{idsiswa}', 'Admin_PenggunaController@create_siswa');
194
195     Route::get('bkk/berkas', 'Admin_BerkasController@index');
196     Route::get('bkk/berkas/tambah', 'Admin_BerkasController@create');
197     Route::get('bkk/ubah-berkas/{id}', 'Admin_BerkasController@edit');
198     Route::post('bkk/simpan-berkas', 'Admin_BerkasController@store');
199     Route::post('bkk/hapus-berkas', 'Admin_BerkasController@destroy');
200     Route::post('bkk/update-berkas', 'Admin_BerkasController@update');
201
202     Route::get('bkk/laporan/siswa', 'Admin_LaporanSiswaController@index');
203
204     Route::get('bkk/logout', function () {
205         Session::flush();
206         return redirect()->intended('/login');
207     });
208
209
210 });

```

Gambar 4.14 Routing Group untuk Admin

4.4.3 Konfigurasi Database

Tahapan persiapan sebelum mulai menuliskan kode program adalah melakukan konfigurasi database. Database yang digunakan dalam pengembangan aplikasi ini adalah MySQL. Dalam *mode development*, DBMS MySQL tersimpan dalam komputer yang sama dengan web server. Sebelum melakukan konfigurasi database, harus dibuat terlebih dahulu databasenya di MySQL. Informasi penting yang diperlukan dalam konfigurasi database yaitu: driver database, server database, port, username, password dan nama database.

Secara default, konfigurasi database di laravel dapat dilakukan pada file `.env` pada direktori utama dan file `config/database.php`. Konfigurasi database pada file `database.php`

dapat dilihat pada gambar 4.15 berikut. Konfigurasi yang diisikan yaitu: *host*, *database*, *username*, dan *password*. Isikan data tersebut sesuai dengan konfigurasi pada DBMS MySQL yang digunakan. Jika terdapat ketidaksesuaian, maka koneksi ke database akan gagal dilakukan.

```
34 'connections' => [  
35  
36     'sqlite' => [  
37         'driver' => 'sqlite',  
38         'database' => env('DB_DATABASE', database_path('database.sqlite')),  
39         'prefix' => '',  
40     ],  
41  
42     'mysql' => [  
43         'driver' => 'mysql',  
44         'host' => 'localhost',  
45         'port' => '3307',  
46         'database' => 'pusatkar_bkk',  
47         'username' => 'pusatkar_bkk',  
48         'password' => 'Z@nu2h%haw&&',  
49         'charset' => 'utf8',  
50         'collation' => 'utf8_unicode_ci',  
51         'prefix' => '',  
52         'strict' => false,  
53         'engine' => null,  
54     ],  
55  
56     'pgsql' => [  
57         'driver' => 'pgsql',  
58         'host' => env('DB_HOST', '127.0.0.1'),  
59         'port' => env('DB_PORT', '5432'),  
60         'database' => env('DB_DATABASE', 'forge'),  
61         'username' => env('DB_USERNAME', 'forge'),  
62         'password' => env('DB_PASSWORD', ''),  
63         'charset' => 'utf8',  
64         'prefix' => '',  
65         'schema' => 'public',  
66         'sslmode' => 'prefer',  
67     ],  
68 ]
```

Gambar 4.15 Pengaturan database di Laravel

4.4.4 Proses Login atau Autentikasi

Implementasi proses login dikelola oleh *controller Authentication* dengan *method login*. *Method login* dijalankan ketika pengguna telah mengisi *username* (email) dan *password* dan klik tombol login. informasi awal yang dikumpulkan yaitu:

- Email yang diisikan pengguna
- Password
- Browser yang digunakan
- Alamat IP
- Status remember bernilai true atau false

Potongan kode program sebagai berikut:

```
1. $ua = browser();
2. $ip = ip();
3.
4. $credentials = [
5.     "email" => $request->input("email"),
6.     "password" => $request->input("password")
7. ];
8.
9. $remember = ($request->input('remember')) ? true : false;
```

Proses berikutnya adalah pengecekan username dan password yang diisikan oleh pengguna. Jika sukses, maka data: *level akses*, *nama user*, *iduser*, dan *idperusahaan* disimpan di variable untuk digunakan pada proses selanjutnya.

```
3 public function login(Request $request)
4 {
5     $ua = browser();
6     $ip = ip();
7
8     $credentials = [
9         "email" => $request->input("email"),
10        "password" => $request->input("password")
11    ];
12
13    if(Auth::attempt($credentials))
14    {
15        ..
16        if (auth()->attempt(['email' => $email, 'password' => $password], $remember))
17        {
18            $level = auth()->user()->level;
19            $nama = auth()->user()->nama;
20            $userid = auth()->user()->id;
21            $idperusahaan = auth()->user()->idperusahaan;
22
23            if(($level = 'A') || ($level = 'BKK')){
24                ..
25                //level akses BKK
26            }
27        }
28        else if($level = 'I'){
29            ..
30            //level akses IDUKA
31        }
32        else if($level = 'S'){
33            ..
34            //level akses Siswa dan Alumni
35        }
36        else {
37            $pesan_gagal = "Maaf, username atau password salah.";
38            return redirect()->intended('login')->with('pesan_gagal', $pesan_gagal);
39        }
40    }
41 }
```

Berdasarkan data level akses, tahap berikutnya dilakukan pengecekan level akses pengguna yang terdiri dari A (admin), BKK, I (Iduka), dan S (Siswa). Jika level akses

pengguna adalah BKK, maka akan dibawah ke halaman dengan level akses BKK. jika tidak ada yang sesuai, maka ditampilkan pesan gagal dan *redirect* ke halaman login. Sebelum dilakukan *redirect* ke laman *dashboard* dari masing-masing level pengguna dilakukan proses berikut:

- Membuat variable *session*
- Mengambil data dari *database* sesuai dengan pengguna yang login
- Melakukan update tabel *users* untuk menyimpan data *log login* yaitu alamat IP, waktu login terakhir, dan browser yang digunakan.
- *Redirect* ke laman utama dari masing-masing level akses pengguna.

Berikut adalah potongan kode program dari proses tersebut untuk pengguna dengan level akses siswa/alumni.

```
...
else if($level == 'S'){
    $idsiswa = $klaster = $idbkk = "";
    $siswa = DB::table('ms_siswa')->where('email',$email)->first();
    if(isset($siswa))
    {
        $idsiswa = $siswa->id; $idbkk = $siswa->idbkk;
    }else{ $idsiswa = ""; $idbkk = ""; }

    $pendaftar = DB::table('ms_pendaftar')->where('email',$email)->first();
    if(isset($pendaftar))
    { $klaster = $pendaftar->klaster; }
    else{ $klaster = ""; }

    $request->session()->put('level', 'S');
    $request->session()->put('email', $email);
    $request->session()->put('nama', $nama);
    $request->session()->put('userid', $userid);
    $request->session()->put('idsiswa', $idsiswa);
    $request->session()->put('klaster', $klaster);
    $request->session()->put('idbkk', $idbkk);
    $request->session()->put('namalevel', 'Siswa');

    $ubah = DB::table('users')
        ->where('email',$email)
        ->update(['ip_address' => $ip,
            'last_login'=>date('Y-m-d H:i:s'),
            'browser' => $browser
        ]);

    $pesan_sukses = "Login berhasil.";
    return redirect()->intended('siswa')->with('pesan_sukses',
    $pesan_sukses);
}
```

....

Alur untuk level pengguna yang lain tidak jauh berbeda. Walaupun ada perbedaan, hanya pada bagian kecil saja. Terlihat bahwa setiap pengguna yang berhasil login, maka dibuat log dengan melakukan update pada tabel *users*. Proses *redirect* ke laman utama pengguna disertai dengan mengirimkan pesan sukses.

4.4.5 Laman Utama Aplikasi

Ketika pengunjung membuka website pusat karir SMK di <http://pusatkarirsmk.id> maka akan ditampilkan laman utama aplikasi. Laman utama terdiri dari beberapa tautan dan form pencarian lowongan pekerjaan. *Controller* untuk laman utama adalah *Homepage_Controller* dengan method utama *index()* untuk laman utama dan *cari()* untuk pencarian lowongan kerja. Isi dari *method index()* adalah:

- Membaca tabel *ms_profil*
- Menyimpan isi data dari tabel *ms_profile* ke session
- Membaca data dari tabel *ms_sosmed* dan menyimpan dalam session
- Membaca jumlah data pada tabel *users*, *ms_lowongan*, dan *ms_perusahaan* kemudian disimpan pada variabel
- Menampilkan file *index.php* yang terdapat pada direktori home

Potongan kode programnya adalah sebagai berikut:

```
..  
$profil = DB::table('ms_profil')->first();  
if(isset($profil))  
{  
    Session::put('APP_NAME', $profil->nama);  
    Session::put('APP_TELP', $profil->telp);  
    Session::put('APP_ALAMAT', $profil->alamat);  
    Session::put('APP_EMAIL', $profil->email);  
}  
..  
$sosmed = DB::table('ms_sosmed')->select('link', 'nama')->get();  
if(count($sosmed))  
{  
    Session::put('APP_SOSMED', '1');  
    foreach ($sosmed as $key => $value) {  
        Session::put('APP_SOSMED_' . strtoupper($value->nama), $value->link);  
    }  
}  
}
```

```

..
$jml_user = DB::table('users')->where('isactive','1')->count();
$jml_lowongan = DB::table('ms_lowongan')->count();
$jml_perusahaan = DB::table('ms_perusahaan')->count();
$jenis = "UMUM";
return
view('home.index',compact('jml_user','jml_lowongan','jml_perusahaan',
,'jenis'));
..

```

Sedangkan isi utama dari *method cari()* adalah melakukan *query* dari tabel *ms_lowongan* *left join* dengan tabel *ms_perusahaan*, *ms_kategori_perusahaan*, dan *ms_gaji* berdasarkan *keyword* pencarian. Berikut potongan kode programnya.

```

..
$lowongan = DB::table('ms_lowongan')->select('ms_lowongan.*','ms_perusahaan.nama as namaperusahaan',
    'ms_perusahaan.alamat as alamatperusahaan',
    'ms_kategori_perusahaan.nama as bidangperusahaan',
    'ms_perusahaan.logo as logoperusahaan','ms_gaji.nama as gaji')
->leftjoin('ms_perusahaan','ms_perusahaan.id','=','ms_lowongan.idperusahaan')
->leftjoin('ms_kategori_perusahaan','ms_kategori_perusahaan.id','=','ms_perusahaan.bidang')
->leftjoin('ms_gaji','ms_gaji.id','=','ms_lowongan.idgaji')
->where('ms_lowongan.status','1')
->whereRaw("$where AND ms_lowongan.lowongan LIKE '%$q.%' ")
->whereRaw("$where")
->paginate(10);
..

```

4.4.6 Kelola Data Perusahaan

Laravel telah menyediakan artisan untuk membuat *Controller* beserta *template method default* untuk keperluan CRUD (*Create, Read, Update, Delete*), yaitu *index()*, *create()*, *show(\$id)*, *edit(\$id)*, *store()*, *update()*, dan *destroy()*. Tugas *web developer* adalah mengisi *method* tersebut dengan kode program sehingga dapat berfungsi sesuai kebutuhan. Perintah Artisan untuk membuat *Controller* adalah:

```
php artisan make:controller Admin_PerusahaanController -r
```

Perintah Laravel Artisan CLI tersebut digunakan untuk membuat *Controller* baru dengan nama *Admin_PerusahaanController*. Opsi *-r* digunakan untuk *generate template method* standar. Jika menginginkan tanpa *template method default*, hilangkan opsi *-r* sehingga perintah untuk membuat *Controller* menjadi seperti berikut:

```
php artisan make:controller Admin_PerusahaanController
```

Isi *Controller* tanpa opsi *-r* adalah sebagai berikut:

```

<?php

namespace App\Http\Controllers;
use Illuminate\Http\Request;
class Admin_PerusahaanController extends Controller

```

```
{
    //
}
```

Terlihat bahwa *Controller* masih kosong, belum ada satu *method* sama sekali. *Method* yang diperlukan dapat ditambahkan sendiri secara manual. Isi *Controller* dengan *template method default* tidak ditampilkan karena isinya terlalu banyak. *Controller* yang digunakan untuk kelola data perusahaan dibuat dengan menggunakan opsi -r pada perintah *Artisan*. *Template method default* yang otomatis ada dapat dilihat pada tabel 4.1 berikut.

Tabel 4.1 Method Controller Perusahaan

No	Nama Method	Keterangan
1	Index()	Method untuk menampilkan halaman utama perusahaan
2	Create()	Method untuk menampilkan form tambah data perusahaan
3	Show(\$id)	Method untuk menampilkan detail data perusahaan
4	Edit(\$id)	Method untuk menampilkan form edit data perusahaan
5	Store()	Method untuk proses penyimpanan (<i>insert</i>) data perusahaan baru
6	Update()	Method untuk proses perubahan (<i>update</i>) data perusahaan
7	Destroy()	Method untuk proses hapus (<i>delete</i>) data perusahaan

Kode program untuk tiap *method* pada tabel 4.1 terlalu banyak kalau dimasukkan semua dalam panduan teknis ini, hanya potongan kode program penting saja. Berikut adalah kerangka dari *Controller* Admin_PerusahaanController:

```
<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\MasPeriodeModel;
use App\Http\Requests;
use App\Http\Controllers\Controller;
use Illuminate\Contracts\Auth\Guard;
use DB;
use Session;
use Validator;
use Input;
use Ramsey\Uuid\Uuid;
use Crypt;
use Response;
```

```

class Admin_PerusahaanController extends Controller
{
    public function index()
    { }
    public function create()
    { }

    public function edit($id)
    { }

    public function show($id)
    { }

    public function store(Request $request)
    { }
    public function destroy_logo(Request $request)
    { }

    public function destroy(Request $request)
    { }

    public function update(Request $request)
    { }
}

```

Terlihat bahwa ada beberapa *Package* yang digunakan yaitu session, DB, validator, Crypt, dan Ramsey\Uuid untuk membuat UUID. *Method* yang ditampilkan hanya *header method* saja, untuk isi *method* akan dijelaskan satu persatu.

Method index() digunakan untuk menampilkan data perusahaan. Data yang ditampilkan berasal dari tabel *ms_perusahaan* *left join* dengan tabel *ms_kategori_perusahaan*. Berikut adalah potongan kode program *method index()*.

```
<?php
```

```

public function index()
{
    if(Session::get('level')=='A')
    {
        ..
    }
    else if(Session::get('level')=='BKK')
    {
        ..
    }
}

```

```

    }
    else if (Session::get('level') == 'I') {
        ..
    }
    return view
('admin.data_perusahaan.index', compact('perusahaan'));
}

```

Terdapat pengecekan level akses pengguna berdasarkan session login. Jika level="A" maka semua data perusahaan ditampilkan, jika level="BKK" maka data perusahaan yang ditampilkan yang terdaftar pada BKK tersebut, dan jika level="I" maka hanya perusahaan yang ditampilkan sesuai ID perusahaan. Data perusahaan kemudian ditampilkan pada file index.php yang berada di direktori admin\data_perusahaan. Kode program dari masing-masing level akses: select data dari tabel ms_perusahaan, left joint dengan tabel ms_kategori_perusahaan, memberikan kondisi where, dan mengurutkan data berdasarkan kolom created_at secara descending. berikut potongan kode programnya.

```

$perusahaan = DB::table('ms_perusahaan')->select('kolom yang
ditampilkan')
-
>leftjoin('ms_kategori_perusahaan','primary_key','=','forieng_key')
->where('ms_perusahaan.idbkk',Session::get('idbkk'))
->orderBy('created_at','desc')->get();

```

Method create() dipanggil ketika pengguna klik tombol tambah di laman perusahaan. Isi dari method ini adalah menampilkan form tambah data dengan cara memanggil file create.php yang berada pada direktori admin\data_perusahaan.

```

public function create()
{
    return view('admin.data_perusahaan.create');
}

```

Method edit(\$id) dipanggil ketika pengguna klik tombol edit. inti dari method ini adalah membaca data perusahaan berdasarkan parameter \$id kemudian menampilkan dalam file edit.php yang berada pada direktori admin\data_perusahaan.

```

public function edit($id)
{
    try{
        $n = Crypt::decrypt($id);
        $data = DB::table('ms_perusahaan')
        ->where('id',$n)
        ->first();
    }
}

```

```

    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }
    return view('admin.data_perusahaan.edit', compact('data'));
}

```

Method `show($id)` dipanggil ketika pengguna klik tombol detail. inti dari method ini adalah membaca satu data perusahaan left join dengan data kategori perusahaan berdasarkan parameter `$id`. Data kemudian ditampilkan pada file `show.php` pada direktori `admin\data_perusahaan`. jika data perusahaan tidak ada, maka ditampilkan laman `error 404`.

```

public function show($id)
{
    try{
        $n = Crypt::decrypt($id);
        $data = DB::table('ms_perusahaan')-
>select('ms_perusahaan.*','ms_kategori_perusahaan.nama as namabidang')
        -
>leftjoin('ms_kategori_perusahaan','ms_kategori_perusahaan.id','=','ms_peru
sahaan.bidang')
        ->where('ms_perusahaan.id',$n)
        ->first();
    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }
    return view('admin.data_perusahaan.show', compact('data'));
}

```

Method `store()` dipanggil ketika pengguna klik tombol simpan setelah selesai mengisi data pada form tambah data perusahaan. Alur program pada *method* ini adalah:

- Menentukan *rules* validasi data
- Melakukan pengecekan validasi berdasarkan *rules* yang telah ditentukan
- Melakukan proses unggah data berdasarkan file logo perusahaan
- Membaca data yang diisikan pada form tambah data
- Melakukan insert data ke tabel `ms_perusahaan`
- Redirect ke laman data perusahaan

Potongan kode program dari masing-masing tahapan diatas adalah sebagai berikut.

```

// 1. rule validasi data
$rules = array(

```

```

        'nama' => 'required|max:255',
        'alamat' => 'required|max:255',
        'jenis' => 'required|max:255',
        'bidang' => 'required|max:255'
    );

// 2. Pengecekan Validasi
$validator = Validator::make($request->all(), $rules, $pesan_error);
if($validator->fails())
{
    $pesan_error = $validator->messages();
    echo "<script> window.self.history.back(); </script>";
}
else
{
    // isi program utama
    // 1. Upload logo perusahaan
    // 2. Baca data dari form tambah data
    // 3. Insert Data
    // 4. Redirect
}

// 3. Upload file logo
$namafile = null;
$size = $_FILES["file"]["size"];
if($size>0)
{
    $filename = $_FILES['file']['name'];
    $ext = pathinfo($filename, PATHINFO_EXTENSION);
    $explode = explode(".", $filename);
    $uuid = Uuid::uuid4();
    $namafile = md5($uuid.$explode[0]).'.'.$ext;

    $path_dir = 'assets/logo/perusahaan/';

    $request->file('file')->move($path_dir, $namafile);
}
else{
    $namafile = null;
}

// 4. Baca data dari form tambah data
$kota = DB::table('ms_kota')->where('kode', $request->input('idkota'))->first();
if(isset($kota))
{
    $namakota = $kota->nama;
}

```

```

}else{ $namakota = null; }

$id = Uuid::uuid4();
$insert = array(
    'id'=>$id,
    'created_at'=>date ('Y-m-d H:i:s'),
    'userid_created' => Session::get('userid'),
    'nama' => $request->input('nama'),
    'status' => $request->input('status'),
    'alamat' => $request->input('alamat'),
    'email' => strtolower($request->input('email')),
    'jenis' => $request->input('jenis'),
    'telp' => $request->input('telp'),
    'idkota' => $request->input('idkota'),
    'profil' => $request->input('profil'),
    'jenis' => $request->input('jenis'),
    'bidang' => $request->input('bidang'),
    'logo' => $namafile,
    'gaya_berpakaian' => $request->input('gaya_berpakaian'),
    'tunjangan' => $request->input('tunjangan'),
    'web' => $request->input('web'),
    'idwaktubekerja' => $request->input('idwaktubekerja'),
    'idjmlkaryawan' => $request->input('idjmlkaryawan'),
    'kota' => $namakota,
    'nama_pj' => $request->input('nama_pj'),
    'hp_pj' => $request->input('hp_pj'),
    'metode_daftar' => $request->input('metode_daftar'),
    'dilihat' => 0,
    'idbkk' => Session::get('idbkk')
);

// 5. Insert data
DB::table('ms_perusahaan')->insert([$insert]);

// 6. Redirect
$pesan_sukses = "Data berhasil ditambahkan";
return redirect()->intended('bkk/perusahaan')->with('pesan_sukses',
$pesan_sukses);

```

Method `update()` dipanggil ketika pengguna melakukan edit data dan klik tombol simpan. Alur program pada method ini sebagian besar sama dengan `method store()`. Yang berbeda hanya pada proses *insert* diganti menjadi *update* data. berikut potongan kode program untuk proses *update* data.

```

// Update data
$kota = DB::table('ms_kota')->where('kode',$request->input('idkota'))->first();
if(isset($kota))

```

```

{
    $namakota = $kota->nama;
}else{ $namakota = null; }

$ubah = DB::table('ms_perusahaan')
->where('id',$n)
->update(['nama' => $request->input('nama'),
        'status' => $request->input('status'),
        'alamat' => $request->input('alamat'),
        'email' => strtolower($request->input('email')),
        'jenis' => $request->input('jenis'),
        'telp' => $request->input('telp'),
        'idkota' => $request->input('idkota'),
        'profil' => $request->input('profil'),
        'jenis' => $request->input('jenis'),
        'bidang' => $request->input('bidang'),
        'updated_at'=>date('Y-m-d H:i:s'),
        'userid_updated' => Session::get('userid'),
        'logo'=>$namafile,
        'kota' => $namakota,
        'gaya_berpakaian' => $request->input('gaya_berpakaian'),
        'tunjangan' => $request->input('tunjangan'),
        'web' => $request->input('web'),
        'idwaktubekerja' => $request->input('idwaktubekerja'),
        'idjmlkaryawan' => $request->input('idjmlkaryawan'),
        'nama_pj' => $request->input('nama_pj'),
        'hp_pj' => $request->input('hp_pj'),
        'metode_daftar' => $request->input('metode_daftar')
    ]);

```

Method destroy() dipanggil ketika pengguna klik tombol hapus data kemudian klik tombol Yes pada konfirmasi yang muncul. Alur program pada method ini:

- Hapua data perusahaan berdasarkan parameter \$id
- Hapus logo perusahaan
- Jika proses proses hapus berhasil tampilkan pesan sukses dan jika gagal tampilkan pesan gagal

Kode program pada method destroy() adalah sebagai berikut:

```

<?php

public function destroy(Request $request)
{
    $data = array();
    $id = Crypt::decrypt($request->input('id'));

```

```

    $ms_perusahaan = DB::table('ms_perusahaan')->select('logo')->
    >where('id',$id)->delete();
    $del = DB::table('ms_perusahaan')->where('id',$id)->delete();

    $path_dir = 'assets/logo/perusahaan/';
    if($ms_perusahaan->logo!="")
    {
        error_reporting(0);
        unlink($path_dir.'/'.$ms_perusahaan->logo);
    }

    if($del)
    {
        $pesan_sukses = "Data berhasil dihapus.";
        Session::flash('pesan_sukses',$pesan_sukses);
        $data = array('msg'=>$pesan_sukses, 'status' => 1);
    }
    else{
        $pesan_error = "Data gagal dihapus.";
        Session::flash('pesan_error',$pesan_error);
        $data = array('msg'=>$pesan_error, 'status' => 0);
    }
    return Response::json($data);
}

```

Selain *Controller*, yang terlihat dalam kelola data perusahaan adalah file yang berada pada direktori `\resources\views\admin\data_perusahaan`. terdapat empat file yaitu:

1. `Index.blade.php`, ditunakan untuk menampilkan data perusahaan.
2. `Create.blade.php`, digunakan untuk menampilkan form tambah data perusahaan.
3. `Edit.blade.php`, digunakan untuk menampilkan form edit data perusahaan.
4. `Show.blade.php`, digunakan untuk menampilkan detail data perusahaan.

Semua file tersebut merupakan extends dari file `layout_admin.php` yang merupakan layout utama dari aplikasi.

4.4.7 Kelola Data Siswa

Kelola data siswa dilakukan oleh *Controller* `Admin_SiswaController` yang terdapat pada direktori `\app\Http\Controllers` dan 10 file View pada direktori `\resources\views\data_siswa`. Daftar method pada `Admin_SiswaController` dapat dilihat pada tabel 4.2 berikut.

Tabel 4.2 Method Controller Siswa

No	Nama Method	Keterangan
1	Index()	Method untuk menampilkan halaman utama data siswa
2	Create()	Method untuk menampilkan form tambah data siswa
3	Import_data()	Method untuk menampilkan import data siswa dari Excel
4	Edit(\$id)	Method untuk menampilkan form edit data siswa
5	Create_tracer_study(\$id)	Method untuk menampilkan form tracer study
6	Nilai_akademik(\$id)	Method untuk menampilkan nilai akademik siswa
7	Create_nilai_akademik()	Method untuk menampilkan form isi nilai akademik
8	Show()	Method untuk menampilkan detail data siswa
9	Tracer_study()	Method untuk menampilkan data hasil tracer study
10	Store()	Method Untuk proses insert data siswa
11	Destroy()	Method untuk menghapus data siswa
12	Update()	Method untuk proses update data siswa
13	Store_nilai()	Method untuk proses insert nilai akademik
14	Destroy_nilai()	Method untuk hapus data nilai akademik
15	Hasil_import()	Method untuk menampilkan hasil import data siswa
16	Destroy_import()	Method untuk menghapus data temporary import data siswa

File View yang digunakan sebagai antarmuka dengan pengguna yaitu:

1. Create.blade.php
2. Create_nilai_akademik.blade.php
3. Create_tracer_study.blade.php
4. Edit.blade.php
5. Hasil_import.blade.php

6. Import.blade.php
7. Index.blade.php
8. Nilai_akademik.blade.php
9. Show.blade.php
10. Tracer_study.blade.php

Method index() dipanggil ketika laman data siswa ditampilkan. Data yang ditampilkan berasal dari tabel ms_siswa *left join* dengan tabel ms_jurusan. Data siswa ditampilkanurut berdasarkan kolom created_at secara *descending*. Jika level akses pengguna “A” (admin), maka seluruh data siswa ditampilkan, dan jika level akses pengguna “BKK” maka data siswa yang ditampilkan adalah yang berada pada BKK tersebut.

```
public function index()
{
    if(Session::get('level')=='A')
    {
        $siswa = DB::table('ms_siswa')
        ->select('ms_siswa.*','ms_jurusan.nama as jurusan')
        -
>leftjoin('ms_jurusan','ms_jurusan.id','=','ms_siswa.idjurusan')
        ->orderBy('created_at','desc')
        ->get();
    }
    else if(Session::get('level')=='BKK')
    {
        $siswa = DB::table('ms_siswa')
        ->select('ms_siswa.*','ms_jurusan.nama as jurusan')
        -
>leftjoin('ms_jurusan','ms_jurusan.id','=','ms_siswa.idjurusan')
        ->where('ms_siswa.idbkk',Session::get('idbkk'))
        ->orderBy('created_at','desc')
        ->get();
    }

    return view ('admin.data_siswa.index',compact('siswa'));
}
```

Method Create() dipanggil ketika menggunakan klik tombol tambah untuk menampilkan form tambah data siswa. Method import() dipanggil ketika pengguna klik tombol import untuk menampilkan form import data siswa.

```
public function create()
```

```

{
    return view('admin.data_siswa.create');
}

public function import()
{
    return view('admin.data_siswa.import');
}

```

Method `edit()` dipanggil ketika pengguna melakukan edit data siswa. Yang dilakukan pada method `edit()` adalah:

1. Membaca parameter `$id`
2. Melakukan dekripsi dengan menggunakan Package `Crypt`
3. Mengambil satu data dari tabel `ms_siswa` berdasarkan parameter `$id` yang telah didekripsi.
4. Jika data tidak ditemukan, maka redirect ke router 404.
5. Menampilkan data siswa pada file `edit.blade.php` yang berada pada direktori View.

Kode program untuk method `edit($id)` adalah sebagai berikut:

```

public function edit($id)
{
    try{
        $n = Crypt::decrypt($id);
        $data = DB::table('ms_siswa')
            ->where('id',$n)
            ->first();
    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }
    return view('admin.data_siswa.edit',compact('data'));
}

```

Method `show($id)` dipanggil ketika pengguna menampilkan detail data siswa dengan klik tombol detail. Yang dilakukan pada method `show($id)` adalah:

1. Membaca parameter `$id`
2. Melakukan dekripsi dengan menggunakan Package `Crypt`
3. Mengambil satu data dari tabel `ms_siswa` berdasarkan parameter `$id` yang telah didekripsi.
4. Jika data tidak ditemukan, maka redirect ke router 404.

5. Membaca data dari tabel `tr_nilai_akademik` left join dengan tabel `ms_matpel` berdasarkan parameter `$id`
6. Membaca data dari tabel `ms_file_siswa` berdasarkan parameter `$id`
7. Menampilkan detail data siswa pada file `show.blade.php` yang berada pada direktori View.

Kode program untuk method ini adalah:

```
public function show($id)
{
    try{
        $n = Crypt::decrypt($id);
        $data = DB::table('ms_siswa')
            ->where('id',$n)
            ->first();
    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }

    $nilai= DB::table('tr_nilai_akademik')
        ->select('tr_nilai_akademik.nilai','ms_matpel.nama',
            'tr_nilai_akademik.tahun','tr_nilai_akademik.id',
            'tr_nilai_akademik.tahun')
        -
    >leftjoin('ms_matpel','ms_matpel.id','=','tr_nilai_akademik.idmatpel')
        ->where('idsiswa',$n)->orderby('nilai','desc')->get();

    $berkas = DB::table('ms_file_siswa')
        ->where('idsiswa',$n)->get();

    return view('admin.data_siswa.show',
        compact('data','nilai','berkas'));
}
```

Method `store()` dipanggil ketika tombol simpan pada form tambah data siswa di-klik. Yang dilakukan pada method `store()` adalah:

1. Mendefinisikan rule validasi
2. Melakukan validasi berdasarkan rule yang telah didefinisikan. Jika hasil validasi bernilai *true*, maka dilakukan proses simpan data. Jika bernilai *false* maka akan ditampilkan pesan error.
3. Insert data ke tabel `ms_pendaftar` dan tabel `ms_siswa`.
4. Insert dat ke tabel `users` untuk data login siswa.

Berikut ini adalah kerangka kode program pada method store(). Bagian yang diberi komentar sebenarnya berisi kode program. Sengaja tidak ditampilkan agar memudahkan membaca alur program pada method store().

```
public function store(Request $request)
{
    // definisi validasi data

    $validator = Validator::make($request-
>all(), $rules, $pesan_error);

    if($validator->fails())
    {
        $pesan_error = $validator->messages();
        echo "<script> window.self.history.back(); </script>";
    }
    else
    {
        // insert data ke tabel ms_pendaftar
        // insert data ke tabel ms_siswa

        $users = DB::table('users')
        ->where('email', $email)->first();
        if(isset($users))
        {
            $pesan_error = "Data siswa berhasil ditambahkan.
Tetapi data login gagal karena email
sudah terdaftar sebelumnya.";
            return redirect()->intended('bkk/siswa')
            ->with('pesan_error', $pesan_error);

        }else{
            // insert data ke tabel users
        }
        $pesan_sukses = "Data siswa berhasil ditambahkan";
        return redirect()->intended('bkk/siswa')
        ->with('pesan_sukses', $pesan_sukses);
    }
}
```

Definisi validasi berisi *rule* untuk proses validasi terhadap data yang diisikan oleh pengguna pada form tambah data.

```
$rules = array(
    'klaster' => 'required|max:10',
    'nama' => 'required|max:255',
    'nik' => 'required|max:255',
    'email' => 'required|max:255',
    'jk' => 'required|max:10',
```

```

        'tempat_lahir' => 'required|max:255',
        'tgl_lahir' => 'required|max:255',
        'idagama' => 'required|max:255',
        'password' => 'required|max:100',
    );

    $pesan_error = array(
        'required' => 'Form input harus diisi.'
    );

```

Ketika hasil proses validasi bernilai *true*, maka dilakukan proses *insert* data ke tabel *ms_pendaftar* dan tabel *ms_siswa*. Proses *insert* data ke tabel *ms_pendaftar* adalah sebagai berikut.

```

$idpendaftar = Uuid::uuid4();
$insert_pendaftar = array(
    'id'=>$idpendaftar,
    'created_at'=>date ('Y-m-d H:i:s'),
    'userid_created' => Session::get('userid'),
    'nama' => $request->input('nama'),
    'email' => $email,
    'idkelas' => $request->input('idkelas'),
    'password' => bcrypt($request->input('password')),
    'tgl_pendaftaran' => date ('Y-m-d H:i:s'),
    'klaster' => $request->input('klaster'),
    'idbkk' => $idbkk,
    'ismanual' => '1'
);
DB::table('ms_pendaftar')->insert([$insert_pendaftar]);

```

Sedangkah potongan kode program untuk melakukan *insert* data ke tabel *ms_siswa* adalah sebagai berikut.

```

$id = Uuid::uuid4();
$ms_siswa = DB::table('ms_siswa')->where('email',$email)->first();
if(isset($ms_siswa))
{
    //do nothing
}else{
    $insert = array(
        'id'=>$id,
        'created_at'=>date ('Y-m-d H:i:s'),
        'userid_created' => Session::get('userid'),
        'nama' => $request->input('nama'),
        'nisn' => $request->input('nisn'),
        'idjurusan' => $request->input('idjurusan'),
        'tgl_lahir' => $tgl_lahir,
        'tempat_lahir' => $request->input('tempat_lahir'),
        'alamat' => $request->input('alamat'),
    );
}

```

```

        'telp' => $request->input('telp'),
        'hp' => $request->input('hp'),
        'email' => $email,
        'idpendaftar' => $idpendaftar,
        'nik' => $request->input('nik'),
        'jk' => $request->input('jk'),
        'idagama' => $request->input('idagama'),
        'hobi' => $request->input('hobi'),
        'alamat_tinggal' => $request->input('alamat_tinggal'),
        'idkota' => $request->input('idkota'),
        'idkelas' => $request->input('idkelas'),
        'goldarah' => $request->input('goldarah'),
        'ismenikah' => $request->input('ismenikah'),
        'idbkk' => $idbkk,
        'nik' => $request->input('nik'),
        'klaster' => $request->input('klaster'),
    );
    DB::table('ms_siswa')->insert([$insert]);
}

```

Agar siswa dapat melakukan proses login, maka harus dilakukan juga proses insert data pada tabel *users*. Potongan kode programnya adalah sebagai berikut.

```

$data_user = array(
    'nama'=> $request->input('nama'),
    'email'=> $email,
    'userid'=> $request->input('email'),
    'password'=>bcrypt($request->input('password')),
    'isactive'=>'1',
    'level'=>'S',
    'created_at' => date('Y-m-d H:i:s'),
    'userid_created' => Session::get('userid'),
    'idbkk'=>$idbkk
);

DB::table('users')->insert([$data_user]);

```

Method `update()` dipanggil ketika pengguna klik tombol simpan pada form edit data siswa. Method ini melakukan proses update data berdasarkan parameter yang diberikan. Secara proses bisnis, kode program untuk method `update()` mirip dengan method `store()`. Yang berbeda hanya pada proses *insert* diganti menjadi *update* data. berikut potongan kode program untuk proses *update* data.

```

// update ke tabel ms_pendaftar
$siswa = DB::table('ms_siswa')->select('idpendaftar')
    ->where('id',$n)->first();

```

```

$idpendaftar = $siswa->idpendaftar;
$update_pendaftar = array(
    'id'=>$idpendaftar,
    'updated_at'=>date ('Y-m-d H:i:s'),
    'userid_updated' => Session::get('userid'),
    'idbkk' => $idbkk
);
DB::table('ms_pendaftar')->where('id',$idpendaftar)
->update($update_pendaftar);

//update ke tabel ms_siswa
$id = Uuid::uuid4();
$update_siswa = array(
    'updated_at'=>date ('Y-m-d H:i:s'),
    'userid_updated' => Session::get('userid'),
    'nama' => $request->input('nama'),
    'nisn' => $request->input('nisn'),
    'idjurusan' => $request->input('idjurusan'),
    'tgl_lahir' => $tgl_lahir,
    'tempat_lahir' => $request->input('tempat_lahir'),
    'alamat' => $request->input('alamat'),
    'telp' => $request->input('telp'),
    'hp' => $request->input('hp'),
    'email' => $email,
    //'idpendaftar' => $idpendaftar,
    'nik' => $request->input('nik'),
    'jk' => $request->input('jk'),
    'idagama' => $request->input('idagama'),
    'hobi' => $request->input('hobi'),
    'alamat_tinggal' => $request->input('alamat_tinggal'),
    'idkota' => $request->input('idkota'),
    'idkelas' => $request->input('idkelas'),
    'goldarah' => $request->input('goldarah'),
    'ismenikah' => $request->input('ismenikah'),
    'idbkk' => $idbkk,
    'nik' => $request->input('nik'),
    'klaster' => $request->input('klaster'),
);
DB::table('ms_siswa')->where('id',$n)
->update($update_siswa);

```

Method `destroy()` dipanggil ketika menggunakan klik tombol hapus data siswa. Pada method ini dilakukan proses hapus data dari tabel `ms_siswa`. Berikut adalah kode programnya.

```

public function destroy(Request $request)
{
    $data = array();
    $id = Crypt::decrypt($request->input('id'));
    $del = DB::table('ms_siswa')->where('id',$id)->delete();
}

```

```

if($del)
{
    $pesan_sukses = "Data berhasil dihapus.";
    Session::flash('pesan_sukses', $pesan_sukses);
    $data = array('msg'=>$pesan_sukses, 'status' => 1);
}
else{
    $pesan_error = "Data gagal dihapus.";
    Session::flash('pesan_error', $pesan_error);
    $data = array('msg'=>$pesan_error, 'status' => 0);
}
return Response::json($data);
}

```

Method `store_nilai()` dipanggil ketika pengguna klik tombol simpan pada form entri nilai akademik. Isi dari kode programnya adalah melakukan insert data ke tabel `tr_nilai_akademik`.

```

$idsiswa = Crypt::decrypt($request->input('idsiswa'));
$idpendaftar = Uuid::uuid4();
$insert_pendaftar = array(
    'id'=>$idpendaftar,
    'created_at'=>date('Y-m-d H:i:s'),
    'userid_created' => Session::get('userid'),
    'idsiswa' => $idsiswa,
    'nilai' => $request->input('nilai'),
    'idmatpel' => $request->input('idmatpel'),
    'tahun' => $request->input('tahun')
);
DB::table('tr_nilai_akademik')->insert([$insert_pendaftar]);

$pesan_sukses = "Data nilai akademik siswa berhasil ditambahkan";
return redirect()->
    intended('bkk/siswa/nilai-akademik/' . Crypt::encrypt($idsiswa))
    ->with('pesan_sukses', $pesan_sukses);

```

4.4.8 Kelola Tracer Study

Tracer study merupakan salah satu fitur penting dalam aplikasi pusat karir. Kegiatan *tracer study* diisi oleh alumni. Terdapat tiga jenis pilihan dalam pengisian *tracer study* yaitu bekerja, studi lanjut dan berwirausaha. *Tracer study* dikelola *Controller Admin_TracerStudyController* yang terdiri dari enam method yaitu:

- Method `store_info_kelulusan`
- Method `store_info_pekerjaan`
- Method `store_info_perkuliahan`

- d. Method store_info_wirusaha
- e. Method store_info_lainnya
- f. Method store_kepuasan_sekolah

Berikut adalah potongan kode program untuk kelola tracer study.

```
public function store_info_kelulusan(Request $request)
{
    $data = array();
    $idaktivitas = $request->input('idaktivitas');
    $tahun_lulus = $request->input('tahun_lulus');
    $idpendaftar = Crypt::decrypt($request->input('idpendaftar'));

    $rules = array(
        'idaktivitas' => 'required|max:50',
        'tahun_lulus' => 'required|max:50',
        'idpendaftar' => 'required|max:255'
    );
    $pesan_error = array(
        'required' => 'Form input harus diisi.'
    );
    $validator = Validator::make($request->all(), $rules, $pesan_error);

    if($validator->fails())
    {
        $pesan_error = "Data gagal disimpan.";
        Session::flash('pesan_error', $pesan_error);
        $data = array('msg'=>$pesan_error, 'status' => 0);
    }
    else
    {
        $id = Uuid::uuid4();

        $tr_tracer_study = DB::table('tr_tracer_study')
        ->where('idpendaftar', $idpendaftar)->count('*');
        if($tr_tracer_study>0)
        {
            $update = DB::table('tr_tracer_study')
            ->where('idpendaftar', $idpendaftar)
            ->update([
                'idpendaftar' => $idpendaftar,
                'tahun_lulus' => $tahun_lulus,
                'idaktivitas' => $idaktivitas,
                'updated_at'=>date('Y-m-d H:i:s'),
                'userid_updated' => Session::get('userid')
            ]);
        }
    }
}
```

```

    }
    else{
        $insert = DB::table('tr_tracer_study')
        ->insert([
            'id' => $id,
            'idpendaftar' => $idpendaftar,
            'tahun_lulus' => $tahun_lulus,
            'idaktivitas' => $idaktivitas,
            'created_at'=>date ('Y-m-d H:i:s'),
            'userid_created' => Session::get('userid')
        ]);
    }

    $pesan_sukses = "Data berhasil disimpan.";
    Session::flash('pesan_sukses',$pesan_sukses);
    $data = array('msg'=>$pesan_sukses, 'status' => 1);
}

return Response::json($data);
}

public function store_info_pekerjaan(Request $request)
{
    $data = array();
    $nama_perusahaan = $request->input('nama_perusahaan');
    $idkategoriperusahaan = $request->input('idkategoriperusahaan');
    $posisi_perusahaan = $request->input("posisi_perusahaan");
    $idstatuskontrak = $request->input("idstatuskontrak");
    $ispekerjaanpertama = $request->input("ispekerjaanpertama");
    $idrelevan = $request->input("idrelevan");
    $isluar = $request->input("isluar");
    $idkota = $request->input("idkota");
    $tglmulaibekerja = $request->input("tglmulaibekerja");

    if($request->input('tglmulaibekerja')!="")
    {
        $explode = explode("/", $request->input('tglmulaibekerja'));
        $tglmulaibekerja = $explode[2].'-' . $explode[0].'-' . $explode[1];
    }else{ $tglmulaibekerja = null; }

    $idgaji = $request->input("idgaji");
    $isjurusanrelevan = $request->input("isjurusanrelevan");
    $idperolehinfo = $request->input("idperolehinfo");
    $idpendaftar = Crypt::decrypt($request->input('idpendaftar'));

    $tracer = DB::table('tr_tracer_study')
    ->where('idpendaftar',$idpendaftar)->first();

```

```

$rules = array(
    'nama_perusahaan' => 'required|max:255',
    'idkategoriperusahaan' => 'required|max:255',
    'posisi_perusahaan' => 'required|max:255',
    'idstatuskontrak' => 'required|max:255',
    'ispekerjaanpertama' => 'required|max:255',
    'idrelevan' => 'required|max:255',
    'isluar' => 'required|max:255',
    'idkota' => 'required|max:255',
    'tglmulaibekerja' => 'required|max:255',
    'idgaji' => 'required|max:255',
    'isjurusanrelevan' => 'required|max:255',
    'idperolehinfo' => 'required|max:255'
);
$pesan_error = array(
    'required' => 'Form input harus diisi.'
);
$validator = Validator::make($request-
>all(), $rules, $pesan_error);

    if($validator->fails())
    {
        $pesan_error = "Data gagal disimpan. Isian data tidak
boleh kosong.";
        Session::flash('pesan_error', $pesan_error);
        $data = array('msg'=>$pesan_error, 'status' => 0);
    }
    else
    {
        if(isset($tracer))
        {
            $update = DB::table('tr_tracer_study')
            ->where('idpendaftar', $idpendaftar)
            ->update([
                'nama_perusahaan' => $nama_perusahaan,
                'idkategoriperusahaan' => $idkategoriperusahaan,
                'posisi_perusahaan' => $posisi_perusahaan,
                'idstatuskontrak' => $idstatuskontrak,
                'ispekerjaanpertama' => $ispekerjaanpertama,
                'idrelevan' => $idrelevan,
                'isluar' => $isluar,
                'idkota' => $idkota,
                'tglmulaibekerja' => $tglmulaibekerja,
                'idgaji' => $idgaji,
                'isjurusanrelevan' => $isjurusanrelevan,
                'idperolehinfo' => $idperolehinfo
            ]);

            $pesan_sukses = "Data berhasil disimpan.";
            Session::flash('pesan_sukses', $pesan_sukses);

```

```

        $data = array('msg'=>$pesan_sukses, 'status' => 1);
    }
    else{
        $pesan_error = "Data gagal disimpan.";
        Session::flash('pesan_error',$pesan_error);
        $data = array('msg'=>$pesan_error, 'status' => 0);
    }
}

return Response::json($data);
}

public function store_info_perkuliahahan(Request $request)
{
    $data = array();
    $nama_pt = $request->input('nama_pt');
    $idjenjang = $request->input('idjenjang');
    $idjurusan = $request->input("idjurusan");
    $idkotapt = $request->input("idkotapt");
    $tahun_masuk_pt = $request->input("tahun_masuk_pt");
    $islinear = $request->input("islinear");
    $deskripsi_pt = $request->input("deskripsi_pt");

    $idpendaftar = Crypt::decrypt($request->input('idpendaftar'));

    $tracer = DB::table('tr_tracer_study')-
>where('idpendaftar',$idpendaftar)->first();

    $rules = array(
        'nama_pt' => 'required|max:255',
        'idjenjang' => 'required|max:255',
        'idjurusan' => 'required|max:255',
        'idkotapt' => 'required|max:255',
        'tahun_masuk_pt' => 'required|max:255',
        'islinear' => 'required|max:255'
    );
    $pesan_error = array(
        'required' => 'Form input harus diisi.'
    );
    $validator = Validator::make($request-
>all(),$rules,$pesan_error);

    if($validator->fails())
    {
        $pesan_error = "Data gagal disimpan. Isian data tidak
boleh kosong.";
        Session::flash('pesan_error',$pesan_error);
        $data = array('msg'=>$pesan_error, 'status' => 0);
    }
    else

```

```

        {
            if(isset($tracer))
            {
                $update = DB::table('tr_tracer_study')
                    ->where('idpendaftar',$idpendaftar)
                    ->update([
                        'nama_pt' => $nama_pt,
                        'idjenjang' => $idjenjang,
                        'idjurusan' => $idjurusan,
                        'idkotapt' => $idkotapt,
                        'tahun_masuk_pt' => $tahun_masuk_pt,
                        'islinear' => $islinear,
                        'deskripsi_pt' => $deskripsi_pt
                    ]);

                $pesan_sukses = "Data berhasil disimpan.";
                Session::flash('pesan_sukses',$pesan_sukses);
                $data = array('msg'=>$pesan_sukses, 'status' => 1);
            }
            else{
                $pesan_error = "Data gagal disimpan.";
                Session::flash('pesan_error',$pesan_error);
                $data = array('msg'=>$pesan_error, 'status' => 0);
            }
        }

        return Response::json($data);
    }

    public function store_info_wirausaha(Request $request)
    {
        $data = array();
        $nama_perusahaan_wira = $request->input('nama_perusahaan_wira');
        $idbidangwira = $request->input('idbidangwira');
        $idkotawira = $request->input("idkotawira");
        // $tglmulaiwira = $request->input("tglmulaiwira");
        $islinearwira = $request->input("islinearwira");
        $idomsetwira = $request->input("idomsetwira");
        $idrelevanwira = $request->input("idrelevanwira");
        $deskripsi_wira = $request->input("deskripsi_wira");
        if($request->input('tglmulaiwira')!="")
        {
            $explode = explode("/", $request->input('tglmulaiwira'));
            $tglmulaiwira = $explode[2].'-'. $explode[0].'-'. $explode[1];
        } else{ $tglmulaiwira = null; }

        $idpendaftar = Crypt::decrypt($request->input('idpendaftar'));

        $tracer = DB::table('tr_tracer_study')
            ->where('idpendaftar',$idpendaftar)->first();
    }

```

```

$rules = array(
    'nama_perusahaan_wira' => 'required|max:255',
    'idbidangwira' => 'required|max:255',
    'idkotawira' => 'required|max:255',
    'tglmulaiwira' => 'required|max:255',
    'islinearwira' => 'required|max:255',
    'idomsetwira' => 'required|max:255',
    'idrelewanwira' => 'required|max:255'
);
$pesan_error = array(
    'required' => 'Form input harus diisi.'
);
$validator = Validator::make($request-
>all(), $rules, $pesan_error);

    if($validator->fails())
    {
        $pesan_error = "Data gagal disimpan. Isian data tidak
boleh kosong.";
        Session::flash('pesan_error', $pesan_error);
        $data = array('msg'=>$pesan_error, 'status' => 0);
    }
    else
    {
        if(isset($tracer))
        {
            $update = DB::table('tr_tracer_study')
            ->where('idpendaftar', $idpendaftar)
            ->update([
                'nama_perusahaan_wira' =>
$nama_perusahaan_wira,
                'idbidangwira' => $idbidangwira,
                'idkotawira' => $idkotawira,
                'tglmulaiwira' => $tglmulaiwira,
                'islinearwira' => $islinearwira,
                'idomsetwira' => $idomsetwira,
                'idrelewanwira'=> $idrelewanwira,
                'deskripsi_wira'=> $deskripsi_wira
            ]);

            $pesan_sukses = "Data berhasil disimpan.";
            Session::flash('pesan_sukses', $pesan_sukses);
            $data = array('msg'=>$pesan_sukses, 'status' => 1);
        }
        else{
            $pesan_error = "Data gagal disimpan.";
            Session::flash('pesan_error', $pesan_error);
            $data = array('msg'=>$pesan_error, 'status' => 0);
        }
    }

```

```

    }
    return Response::json($data);
}

public function store_info_lainnya(Request $request)
{
    $data = array();
    $idkarir_lainnya = $request->input('idkarir_lainnya');
    $nama_instansi = $request->input('nama_instansi');

    if($request->input('tglmulaikarir')!="")
    {
        $explode = explode("/", $request->input('tglmulaikarir'));
        $tglmulaikarir = $explode[2].'-'.$explode[0].'-'.$explode[1];
    }else{ $tglmulaikarir = null; }

    if($request->input('tglselesaikarir')!="")
    {
        $explode2 = explode("/", $request->input('tglselesaikarir'));
        $tglselesaikarir = $explode2[2].'-'.$explode2[0].'-'.
        $explode2[1];
    }else{ $tglselesaikarir = null; }

    $idpendaftar = Crypt::decrypt($request->input('idpendaftar'));

    $tracer = DB::table('tr_tracer_study')
        ->where('idpendaftar',$idpendaftar)->first();

    $rules = array(
        'idkarir_lainnya' => 'required|max:255',
        'nama_instansi' => 'required|max:255',
        'tglmulaikarir' => 'required|max:25',
        'tglselesaikarir' => 'required|max:25'
    );
    $pesan_error = array(
        'required' => 'Form input harus diisi.'
    );
    $validator = Validator::make($request-
    >all(),$rules,$pesan_error);

    if($validator->fails())
    {
        $pesan_error = "Data gagal disimpan. Isian data tidak boleh
        kosong.";
        Session::flash('pesan_error',$pesan_error);
        $data = array('msg'=>$pesan_error, 'status' => 0);
    }
    else
    {
        if(isset($tracer))

```

```

        {
            $update = DB::table('tr_tracer_study')
                ->where('idpendaftar',$idpendaftar)
                ->update([
                    'idkarir_lainnya' => $idkarir_lainnya,
                    'nama_instansi' => $nama_instansi,
                    'tglmulaikarir' => $tglmulaikarir,
                    'tglselesaikarir' => $tglselesaikarir
                ]);

            $pesan_sukses = "Data berhasil disimpan.";
            Session::flash('pesan_sukses',$pesan_sukses);
            $data = array('msg'=>$pesan_sukses, 'status' => 1);
        }
        else{
            $pesan_error = "Data gagal disimpan.";
            Session::flash('pesan_error',$pesan_error);
            $data = array('msg'=>$pesan_error, 'status' => 0);
        }
    }
    return Response::json($data);
}

public function store_kepuasan_sekolah(Request $request)
{
    $data = array();
    $kritik_saran = $request->input('kritik_saran');
    $kualitas_pendidikan = $request->input('kualitas_pendidikan');
    $kualitas_pembelajaran = $request->
>input('kualitas_pembelajaran');
    $kualitas_fasilitas = $request->input('kualitas_fasilitas');
    $kualitas_lab = $request->input('kualitas_lab');
    $kualitas_guru = $request->input('kualitas_guru');
    $kualitas_ketrampilan = $request->input('kualitas_ketrampilan');
    $kualitas_kedisiplinan = $request->
>input('kualitas_kedisiplinan');
    $kualitas_kebersihan = $request->input('kualitas_kebersihan');
    $kualitas_bkk = $request->input('kualitas_bkk');
    $kualitas_sistem = $request->input('kualitas_sistem');

    $idpendaftar = Crypt::decrypt($request->input('idpendaftar'));

    $tracer = DB::table('tr_tracer_study')
        ->where('idpendaftar',$idpendaftar)->first();

    $rules = array(
        'kritik_saran' => 'required|max:1000',
        'kualitas_pendidikan' => 'required|max:255',
        'kualitas_pembelajaran' => 'required|max:255',
        'kualitas_fasilitas' => 'required|max:255',
    );
}

```

```

        'kualitas_lab' => 'required|max:255',
        'kualitas_guru' => 'required|max:255',
        'kualitas_ketrampilan' => 'required|max:255',
        'kualitas_kedisiplinan' => 'required|max:255',
        'kualitas_kebersihan' => 'required|max:255',
        'kualitas_bkk' => 'required|max:255',
        'kualitas_sistem' => 'required|max:255'
    );
    $pesan_error = array(
        'required' => 'Form input harus diisi.'
    );
    $validator = Validator::make($request-
>all(), $rules, $pesan_error);

    if($validator->fails())
    {
        $pesan_error = "Data gagal disimpan. Isian data tidak boleh
kosong.";
        Session::flash('pesan_error', $pesan_error);
        $data = array('msg'=>$pesan_error, 'status' => 0);
    }
    else
    {
        if(isset($tracer))
        {
            $update = DB::table('tr_tracer_study')
                ->where('idpendaftar', $idpendaftar)
                ->update([
                    'kritik_saran' => $kritik_saran,
                    'kualitas_pendidikan' => $kualitas_pendidikan,
                    'kualitas_pembelajaran' => $kualitas_pembelajaran,
                    'kualitas_fasilitas' => $kualitas_fasilitas,
                    'kualitas_lab' => $kualitas_lab,
                    'kualitas_guru' => $kualitas_guru,
                    'kualitas_ketrampilan' => $kualitas_ketrampilan,
                    'kualitas_kedisiplinan' => $kualitas_kedisiplinan,
                    'kualitas_kebersihan' => $kualitas_kebersihan,
                    'kualitas_bkk' => $kualitas_bkk,
                    'kualitas_sistem' => $kualitas_sistem
                ]);

            $pesan_sukses = "Data berhasil disimpan. Terima kasih
sudah ikut berpartisipasi dalam pengisian tracer
study.";
            Session::flash('pesan_sukses', $pesan_sukses);
            $data = array('msg'=>$pesan_sukses, 'status' => 1);
        }
        else{
            $pesan_error = "Data gagal disimpan.";
            Session::flash('pesan_error', $pesan_error);

```

```

        $data = array('msg'=>$pesan_error, 'status' => 0);
    }
    return Response::json($data);
}

```

4.4.9 Kelola Lowongan Pekerjaan

Kelola data lowongan pekerjaan dilakukan oleh *Controller* Admin_LowonganController yang terdapat pada direktori \app\http\Controllers dan delapan file View pada direktori \resources\views\data_lowongan. Daftar method pada Admin_LowonganController dapat dilihat pada tabel 4.3 berikut.

Tabel 4.3 Method Controller Lowongan

No	Nama Method	Keterangan
1	Index()	Method untuk menampilkan halaman utama data lowongan pekerjaan
2	Create()	Method untuk menampilkan form tambah data lowongan pekerjaan
3	edit(\$id)	Method untuk menampilkan form edit lowongan pekerjaan
4	show(\$id)	Method untuk menampilkan detail data lowongan pekerjaan
5	Show_pelamar(\$id)	Method untuk menampilkan data pelamar pada lowongan pekerjaan tertentu
6	set_panggilan_tes(\$id)	Method untuk membuat panggilan tes
7	detail_pelamar(\$id)	Method untuk menampilkan detail pelamar pekerjaan
8	store(Request \$request)	Method untuk menyimpan lowongan pekerjaan
9	destroy(Request \$request)	Method untuk hapus lowongan pekerjaan
10	update(Request \$request)	Method untuk update lowongan pekerjaan
11	close_job(Request \$request)	Method untuk close lowongan pekerjaan
12	store_deskripsi(Request \$request)	Method untuk insert deskripsi lowongan pekerjaan

No	Nama Method	Keterangan
13	destroy_deskripsi(Request \$request)	Method untuk hapus deskripsi lowongan pekerjaan
14	store_syarat(Request \$request)	Method untuk insert persyaratan lowongan pekerjaan
15	destroy_syarat(Request \$request)	Method untuk hapus persyaratan lowongan pekerjaan
16	store_keahlian(Request \$request)	Method untuk insert data keahlian pada lowongan pekerjaan
17	destroy_keahlian(Request \$request)	Method untuk hapus data keahlian pada lowongan pekerjaan
18	store_panggilan_tes(Request \$request)	Method untuk insert panggilan tes

Kode program untuk masing-masing method adalah sebagai berikut.

```
public function index()
{
    if(Session::get('level')=='A')
    {
        $lowongan = DB::table('ms_lowongan')
        ->select('ms_lowongan.*','ms_perusahaan.nama as
nama_perusahaan',
        'ms_kota.nama as kota','ms_gaji.nama as gaji')
        ->leftjoin('ms_perusahaan','ms_perusahaan.id','=',
        'ms_lowongan.idperusahaan')
        ->leftjoin('ms_kota','ms_kota.kode','=', 'ms_lowongan.idkota')
        ->leftjoin('ms_gaji','ms_gaji.id','=', 'ms_lowongan.idgaji')
        ->orderBy('created_at','desc')->get();
    }
    else if(Session::get('level')=='BKK')
    {
        $lowongan = DB::table('ms_lowongan')
        ->select('ms_lowongan.*','ms_perusahaan.nama as
nama_perusahaan',
        'ms_kota.nama as kota','ms_gaji.nama as gaji')
        ->leftjoin('ms_perusahaan','ms_perusahaan.id','=',
        'ms_lowongan.idperusahaan')
        ->leftjoin('ms_kota','ms_kota.kode','=', 'ms_lowongan.idkota')
        ->leftjoin('ms_gaji','ms_gaji.id','=', 'ms_lowongan.idgaji')
        ->where('ms_perusahaan.idbkk',Session::get('idbkk'))
    }
}
```

```

        ->orderBy('created_at','desc')->get();
    }
    else if(Session::get('level')=='I'){
        $lowongan = DB::table('ms_lowongan')
        ->select('ms_lowongan.*','ms_perusahaan.nama as
nama_perusahaan',
        'ms_kota.nama as kota','ms_gaji.nama as gaji')
        ->leftjoin('ms_perusahaan','ms_perusahaan.id','=',
        'ms_lowongan.idperusahaan')
        ->leftjoin('ms_kota','ms_kota.kode','=', 'ms_lowongan.idkota')
        ->leftjoin('ms_gaji','ms_gaji.id','=', 'ms_lowongan.idgaji')
        ->where('ms_perusahaan.id',Session::get('idperusahaan'))
        ->orderBy('created_at','desc')->get();
    }
    return view ('admin.data_lowongan.index',compact('lowongan'));
}

public function create()
{
    return view('admin.data_lowongan.create');
}

public function edit($id)
{
    try{
        $arr_jur = array();
        $n = Crypt::decrypt($id);
        $data = DB::table('ms_lowongan')
        ->where('id',$n)
        ->first();

        if($data->tgl_mulai!="")
        { $tglmulai = date('m/d/Y',strtotime($data->tgl_mulai)); }

        if($data->tgl_selesai!="")
        { $tglselesai = date('m/d/Y',strtotime($data->tgl_selesai)); }
    }

    if($data->idjurusan!="")
    { $jurusan = explode(",", $data->idjurusan);
      if(count($jurusan)>0)
      {
          for ($i=0; $i < count($jurusan); $i++) {
              $arr_jur[] = "".$jurusan[$i]."";
          }
          $idjurusan = implode(",", $arr_jur);
      }
      else{
          $idjurusan = "0";
      }
    }
}

```

```

    }
}
catch (\Exception $id) {
    return redirect()->route('404');
}

$deskripsi = DB::table('ms_lowongan_deskripsi')
    ->where('idlowongan',$n)
    ->orderBy('created_at','asc')->get();
$syarat = DB::table('ms_lowongan_syarat')
    ->where('idlowongan',$n)
    ->orderBy('created_at','asc')->get();
$keahlian = DB::table('ms_lowongan_keahlian')
    ->where('idlowongan',$n)
    ->orderBy('created_at','asc')->get();

return view('admin.data_lowongan.edit',
    compact('data','tglmulai','tglselesai',
        'idjurusan','deskripsi','syarat','keahlian'));
}

public function show($id)
{
    try{
        $n = Crypt::decrypt($id);
        $data = DB::table('ms_lowongan')
            ->where('id',$n)
            ->first();

        if($data->tgl_mulai!="")
        { $tglmulai = date('m/d/Y',strtotime($data->tgl_mulai)); }

        if($data->tgl_mulai!="")
        { $tglselesai = date('m/d/Y',strtotime($data->tgl_mulai)); }

        if($data->idjurusan!="")
        { $jurusan = explode(",", $data->idjurusan);
            if(count($jurusan)>0)
            {
                for ($i=0; $i < count($jurusan); $i++) {
                    $arr_jur[] = "".$jurusan[$i]."";
                }
                $idjurusan = implode(",", $arr_jur);
            }
            else{
                $idjurusan = "0";
            }
        }

        $deskripsi = DB::table('ms_lowongan_deskripsi')
            ->where('idlowongan',$n)

```

```

        ->orderBy('created_at','asc')->get();
        $syarat = DB::table('ms_lowongan_syarat')
        ->where('idlowongan',$n)
        ->orderBy('created_at','asc')->get();
        $keahlian = DB::table('ms_lowongan_keahlian')
        ->where('idlowongan',$n)
        ->orderBy('created_at','asc')->get();
    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }
    return view('admin.data_lowongan.show',
        compact('data','tglmulai','tglselesai','idjurusan',
            'deskripsi','syarat','keahlian'));
}

public function show_pelamar($id)
{
    $idlowongan = "";
    try{
        $n = Crypt::decrypt($id);
        $data = DB::table('ms_lowongan')
        ->where('id',$n)
        ->first();
        $idlowongan = $n;

        if($data->tgl_mulai!="")
        { $tglmulai = date('m/d/Y',strtotime($data->tgl_mulai)); }

        if($data->tgl_mulai!="")
        { $tglselesai = date('m/d/Y',strtotime($data->tgl_mulai)); }

        if($data->idjurusan!="")
        { $jurusan = explode(",",$data->idjurusan);
            if(count($jurusan)>0)
            {
                for ($i=0; $i < count($jurusan); $i++) {
                    $arr_jur[] = "".$jurusan[$i]."";
                }
                $idjurusan = implode(",",$arr_jur);
            }
            else{
                $idjurusan = "'0'";
            }
        }

        $deskripsi = DB::table('ms_lowongan_deskripsi')
        ->where('idlowongan',$n)
        ->orderBy('created_at','asc')->get();
    }
}

```

```

        $syarat = DB::table('ms_lowongan_syarat')
            ->where('idlowongan',$n)
            ->orderBy('created_at','asc')->get();
        $keahlian = DB::table('ms_lowongan_keahlian')
            ->where('idlowongan',$n)
            ->orderBy('created_at','asc')->get();

        $pelamar = DB::table('tr_lamaran')
            ->select('tr_lamaran.*','ms_siswa.nama as namasiswa',
                'ms_siswa.nik','ms_siswa.hp',
                'ms_jurusan.nama as namajurusan')
            -
>leftjoin('ms_siswa','ms_siswa.id','=','tr_lamaran.idsiswa')
            -
>leftjoin('ms_jurusan','ms_jurusan.id','=','ms_siswa.idjurusan')
            ->where('tr_lamaran.idlowongan',$n)
            ->orderBy('tr_lamaran.created_at','asc')->get();

    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }
    return view('admin.data_lowongan.show_pelamar',
        compact('data','tglmulai','tglselesai',
            'idjurusan','deskripsi','syarat',
            'keahlian','pelamar','idlowongan'));
}

public function set_panggilan_tes($id)
{
    try{
        $n = Crypt::decrypt($id);
        $data = DB::table('ms_lowongan')
            ->where('id',$n)
            ->first();

        $idlowongan = $n;
    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }
    return view('admin.data_lowongan.set_panggilan_tes',
        compact('data','idlowongan'));
}

public function detail_pelamar($id)
{
    try{
        $n = Crypt::decrypt($id);
    }

```

```

        catch (\Exception $id) {
            return redirect()->route('404');
        }
        error_reporting(0);
        $pelamar = DB::table('tr_lamaran')
            ->select('tr_lamaran.*','ms_siswa.nama as namasiswa',
                'ms_siswa.nik','ms_siswa.hp','ms_jurusan.nama as
namajurusan')
            -
>leftjoin('ms_siswa','ms_siswa.id','=','tr_lamaran.idsiswa')
            -
>leftjoin('ms_jurusan','ms_jurusan.id','=','ms_siswa.idjurusan')
            ->where('tr_lamaran.id',$n)
            ->orderBy('tr_lamaran.created_at','asc')->first();
        $sidlowongan = $pelamar->idlowongan;

        $siswa = DB::table('ms_siswa')->where('id',$pelamar->idsiswa)-
>first();

        return view('admin.data_lowongan.detail_pelamar',
            compact('pelamar','siswa','idlowongan'));
    }

    public function store(Request $request)
    {
        $rules = array(
            'lowongan' => 'required|max:255',
            'idperusahaan' => 'required|max:255',
            'idkota' => 'required|max:50'
        );
        $pesan_error = array(
            'required' => 'Form input harus diisi.'
        );
        $validator = Validator::make($request-
>all(),$rules,$pesan_error);

        if($validator->fails())
        {
            $pesan_error = $validator->messages();
            echo "<script> window.self.history.back(); </script>";
        }
        else
        {
            $namafile = null;
            $size = $_FILES["file"]["size"];
            if($size>0)
            {
                $filename = $_FILES['file']['name'];
                $ext = pathinfo($filename, PATHINFO_EXTENSION);

```

```

$explode = explode(".", $filename);
$uuid = Uuid::uuid4();
$namafile = md5($uuid.$explode[0]).'.'.$ext;

// menentukan direktori tempat gambar akan disimpan
$path_dir = 'assets/lowongan/poster/';

// upload ke direktori yang sudah ditentukan
$request->file('file')->move($path_dir,$namafile);
}
else{
    $namafile = null;
}

$arr_jur = array();
if(count($request->input('jurusan')>0)){
    foreach ($request->input('jurusan') as $key => $data_jur) {
        $arr_jur[] = $data_jur;
    } $jurusan = implode(",", $arr_jur);
}

$explode_mulai = explode("/", $request->input('tgl_mulai'));
$tanggal_mulai = $explode_mulai[2].'-'.$explode_mulai[0].'-
'.$explode_mulai[1];

$explode_selesai = explode("/", $request-
>input('tgl_selesai'));
$tanggal_selesai = $explode_selesai[2].'-
'.$explode_selesai[0].'-'.$explode_selesai[1];

$id = Uuid::uuid4();
$insert = array(
    'id'=>$id,
    'created_at'=>date ('Y-m-d H:i:s'),
    'userid_created' => Session::get('userid'),
    'lowongan' => $request->input('lowongan'),
    'status' => $request->input('status'),
    'tgl_mulai' => $tanggal_mulai,
    'tgl_selesai' => $tanggal_selesai,
    'idperusahaan' => $request->input('idperusahaan'),
    'idkota' => $request->input('idkota'),
    'idgaji' => $request->input('idgaji'),
    'idjurusan' => $jurusan,
    'poster' => $namafile,
    'dilihat' => 0,
    'jeniswaktu'=>$request->input('jeniswaktu'),
    "levelpekerjaan" => $request->input('levelpekerjaan'),
    "jk" => $request->input('jk'),
    "syarat_tinggi_laki" => $request-
>input('syarat_tinggi_laki'),

```

```

        "syarat_tinggi_perempuan" => $request->input('syarat_tinggi_perempuan'),
        "isbuta" => $request->input('isbuta'),
        "iskacamata" => $request->input('iskacamata'),
        "batas_usia" => $request->input('batas_usia'),
        "alamat_penempatan" => $request->input('alamat_penempatan'),
        "tipelowongan" => $request->input('tipelowongan'),
        "isnilai" => $request->input('isnilai'),
        "isdokumen" => $request->input('isdokumen'),
        'idbkk' => Session::get('idbkk'),
        "ispublic" => $request->input('ispublic'),
        'isgaji' => $request->input('isgaji')
    );
    DB::table('ms_lowongan')->insert([$insert]);

    $pesan_sukses = "Data berhasil ditambahkan";
    return redirect()
        ->intended(app_url().'/ubah-lowongan/'.Crypt::encrypt($id))
        ->with('pesan_sukses', $pesan_sukses);
    }
}

public function destroy(Request $request)
{
    $data = array();
    $id = Crypt::decrypt($request->input('id'));
    $del = DB::table('ms_lowongan')->where('id',$id)->delete();

    if($del)
    {
        $pesan_sukses = "Data berhasil dihapus.";
        Session::flash('pesan_sukses',$pesan_sukses);
        $data = array('msg'=>$pesan_sukses, 'status' => 1);
    }
    else{
        $pesan_error = "Data gagal dihapus.";
        Session::flash('pesan_error',$pesan_error);
        $data = array('msg'=>$pesan_error, 'status' => 0);
    }
    return Response::json($data);
}

public function update(Request $request)
{
    $id = $request->input('id');
    $n = Crypt::decrypt($id);

    $namafilename = null;

```

```

$size = $_FILES["file"]["size"];
if($size>0)
{
    $filename = $_FILES['file']['name'];
    $ext = pathinfo($filename, PATHINFO_EXTENSION);
    $explode = explode(".", $filename);
    $uuid = Uuid::uuid4();
    $namafile = md5($uuid.$explode[0]).'.'.$ext;

    // menentukan direktori tempat gambar akan disimpan
    $path_dir = 'assets/lowongan/poster/';

    // upload ke direktori yang sudah ditentukan
    $request->file('file')->move($path_dir,$namafile);
}
else{

    $ms_lowongan = DB::table('ms_lowongan')
        ->where('id',$n)->first();
    if($ms_lowongan->poster!="")
    {
        $namafile = $ms_lowongan->poster;
    }else{ $namafile = null; }
}

$arr_jur = array();
if(count($request->input('jurusan')>0)){
    foreach ($request->input('jurusan') as $key => $data_jur)
    {
        $arr_jur[] = $data_jur;
    } $jurusan = implode(",", $arr_jur);
}

    $explode_mulai = explode("/", $request->input('tgl_mulai'));
    $tanggal_mulai = $explode_mulai[2].'-'.
    $explode_mulai[0].'-'. $explode_mulai[1];

    $explode_selesai = explode("/", $request->input('tgl_selesai'));
    $tanggal_selesai = $explode_selesai[2].'-'.
    $explode_selesai[0].'-'. $explode_selesai[1];

    $ubah = DB::table('ms_lowongan')
        ->where('id',$n)
        ->update(['lowongan' => $request->input('lowongan'),
            'status' => $request->input('status'),
            'tgl_mulai' => $tanggal_mulai,

```

```

        'tgl_selesai' => $tanggal_selesai,
        'idperusahaan' => $request->input('idperusahaan'),
        'idkota' => $request->input('idkota'),
        'idgaji' => $request->input('idgaji'),
        'idjurusan' => $jurusan,
        'poster' => $namafile,
        'updated_at'=>date ('Y-m-d H:i:s'),
        'userid_updated' => Session::get('userid'),
        'jeniswaktu'=>$request->input('jeniswaktu'),
        "levelpekerjaan" => $request->
>input('levelpekerjaan'),
        "jk" => $request->input('jk'),
        "syarat_tinggi_laki" => $request->
>input('syarat_tinggi_laki'),
        "syarat_tinggi_perempuan" => $request->
>input('syarat_tinggi_perempuan'),
        "isbuta" => $request->input('isbuta'),
        "iskacamata" => $request->input('iskacamata'),
        "batas_usia" => $request->input('batas_usia'),
        "alamat_penempatan" => $request->
>input('alamat_penempatan'),
        "tipelowongan" => $request->input('tipelowongan'),
        "isnilai" => $request->input('isnilai'),
        "isdokumen" => $request->input('isdokumen'),
        "ispublic" => $request->input('ispublic'),
        'isgaji' => $request->input('isgaji')
    ]);

```

```

    $pesan_update = "Data lowongan berhasil diubah.";
    \Session::flash('pesan_update', $pesan_update);

```

```

    return redirect()
        ->intended(app_url().'/lowongan')
        ->with('pesan_update', $pesan_update);
}

```

```

public function close_job(Request $request)
{

```

```

    $data = array();
    $id = Crypt::decrypt($request->input('id'));
    $status = $request->input('status');
    $update = DB::table('ms_lowongan')
        ->where('id',$id)
        ->update(['status'=>$status]);

```

```

    if($status=='1')
    {
        $pesan_sukses = "Lowongan berhasil dibuka.";
        Session::flash('pesan_sukses', $pesan_sukses);
        $data = array('msg'=>$pesan_sukses, 'status' => 1);
    }

```

```

    }
    else if($status=='0') {
        $pesan_sukses = "Lowongan berhasil ditutup.";
        Session::flash('pesan_sukses',$pesan_sukses);
        $data = array('msg'=>$pesan_sukses, 'status' => 1);
    }

    return Response::json($data);
}

public function store_deskripsi(Request $request)
{
    $rules = array(
        'deskripsi' => 'required|max:500'
    );
    $pesan_error = array(
        'required' => 'Form input harus diisi.'
    );
    $validator = Validator::make($request-
    >all(),$rules,$pesan_error);

    if($validator->fails())
    {
        $pesan_error = $validator->messages();
        echo "<script> window.self.history.back(); </script>";
    }
    else
    {
        $idlowongan = Crypt::decrypt($request->input('idlowongan'));
        $id = Uuid::uuid4();
        $insert = array(
            'id'=>$id,
            'created_at'=>date ('Y-m-d H:i:s'),
            'userid_created' => Session::get('userid'),
            'deskripsi' => $request->input('deskripsi'),
            'idlowongan' => $idlowongan
        );
        DB::table('ms_lowongan_deskripsi')->insert([$insert]);

        $pesan_sukses = "Data deskripsi lowongan berhasil
        ditambahkan";
        return redirect()
            ->intended(app_url().'/ubah-lowongan/'.$request
            ->input('idlowongan'))
            ->with('pesan_sukses', $pesan_sukses);
    }
}

public function destroy_deskripsi(Request $request)

```

```

{
    $data = array();
    $idlowongan = Crypt::decrypt($request->input('idlowongan'));
    $id = $request->input('id');
    $del = DB::table('ms_lowongan_deskripsi')
        ->where('id', $id)->delete();

    if($del)
    {
        $pesan_sukses = "Data berhasil dihapus.";
        Session::flash('pesan_sukses', $pesan_sukses);
        $data = array('msg'=>$pesan_sukses, 'status' => 1);
    }
    else{
        $pesan_error = "Data gagal dihapus.";
        Session::flash('pesan_error', $pesan_error);
        $data = array('msg'=>$pesan_error, 'status' => 0);
    }

    return Response::json($data);
}

public function store_syarat(Request $request)
{
    $rules = array(
        'syarat' => 'required|max:500'
    );
    $pesan_error = array(
        'required' => 'Form input harus diisi.'
    );
    $validator = Validator::make($request-
    >all(), $rules, $pesan_error);

    if($validator->fails())
    {
        $pesan_error = $validator->messages();
        echo "<script> window.self.history.back(); </script>";
    }
    else
    {
        $idlowongan = Crypt::decrypt($request->input('idlowongan'));
        $id = Uuid::uuid4();
        $insert = array(
            'id'=>$id,
            'created_at'=>date('Y-m-d H:i:s'),
            'userid_created' => Session::get('userid'),
            'syarat' => $request->input('syarat'),
            'idlowongan' => $idlowongan
        );
        DB::table('ms_lowongan_syarat')->insert([$insert]);
    }
}

```

```

        $pesan_sukses = "Data syarat/kualifikasi lowongan berhasil
ditambahkan";
        return redirect()->intended(app_url().'/ubah-
lowongan/'.$request
        ->input('idlowongan'))
        ->with('pesan_sukses', $pesan_sukses);
    }
}

```

```

public function destroy_syarat(Request $request)
{
    $data = array();
    $idlowongan = Crypt::decrypt($request->input('idlowongan'));
    $id = $request->input('id');
    $del = DB::table('ms_lowongan_syarat')
        ->where('id',$id)->delete();

```

```

    if($del)
    {
        $pesan_sukses = "Data berhasil dihapus.";
        Session::flash('pesan_sukses',$pesan_sukses);
        $data = array('msg'=>$pesan_sukses, 'status' => 1);
    }
    else{
        $pesan_error = "Data gagal dihapus.";
        Session::flash('pesan_error',$pesan_error);
        $data = array('msg'=>$pesan_error, 'status' => 0);
    }

```

```

    return Response::json($data);
}

```

```

public function store_keahlian(Request $request)
{

```

```

    $rules = array(
        'keahlian' => 'required|max:500'
    );
    $pesan_error = array(
        'required' => 'Form input harus diisi.'
    );
    $validator = Validator::make($request-
>all(),$rules,$pesan_error);

```

```

    if($validator->fails())
    {
        $pesan_error = $validator->messages();
        echo "<script> window.self.history.back(); </script>";
    }
    else

```

```

    {
        $idlowongan = Crypt::decrypt($request->input('idlowongan'));
        $id = Uuid::uuid4();
        $insert = array(
            'id'=>$id,
            'created_at'=>date('Y-m-d H:i:s'),
            'userid_created' => Session::get('userid'),
            'keahlian' => $request->input('keahlian'),
            'idlowongan' => $idlowongan
        );
        DB::table('ms_lowongan_keahlian')->insert([$insert]);

        $pesan_sukses = "Data keahlian lowongan berhasil
        ditambahkan";
        return redirect()->intended(app_url().'/ubah-
        lowongan/'.$request
            ->input('idlowongan'))
            ->with('pesan_sukses', $pesan_sukses);
    }
}

public function destroy_keahlian(Request $request)
{
    $data = array();
    $idlowongan = Crypt::decrypt($request->input('idlowongan'));
    $id = $request->input('id');
    $del = DB::table('ms_lowongan_keahlian')
        ->where('id',$id)->delete();

    if($del)
    {
        $pesan_sukses = "Data berhasil dihapus.";
        Session::flash('pesan_sukses',$pesan_sukses);
        $data = array('msg'=>$pesan_sukses, 'status' => 1);
    }
    else{
        $pesan_error = "Data gagal dihapus.";
        Session::flash('pesan_error',$pesan_error);
        $data = array('msg'=>$pesan_error, 'status' => 0);
    }

    return Response::json($data);
}

public function store_panggilan_tes(Request $request)
{
    $rules = array(
        'idlowongan' => 'required|max:255',
        'nama' => 'required|max:255',

```

```

        'keterangan' => 'required|max:10000',
        'tanggal_mulai' => 'required|max:50',
        'waktu_mulai' => 'required|max:50'
    );
    $pesan_error = array(
        'required' => 'Form input harus diisi.'
    );
    $validator = Validator::make($request-
>all(), $rules, $pesan_error);

    if($validator->fails())
    {
        $pesan_error = $validator->messages();
        echo "<script> window.self.history.back(); </script>";
    }
    else
    {
        $explode_mulai = explode("/", $request-
>input('tanggal_mulai'));
        $tanggal_mulai = $explode_mulai[2].'-'. $explode_mulai[0].'-
'. $explode_mulai[1];

        $waktu_mulai = $request->input('waktu_mulai').':00';

        if($request->input('waktu_selesai')!="")
        { $waktu_selesai = $request->input('waktu_selesai').':00'; }
        else{ $waktu_selesai = null; }

        $idlowongan = Crypt::decrypt($request->input('idlowongan'));

        $id = Uuid::uuid4();
        $insert = array(
            'id'=>$id,
            'created_at'=>date ('Y-m-d H:i:s'),
            'userid_created' => Session::get('userid'),

            'idlowongan' => $idlowongan,
            'nama' => $request->input('nama'),
            'keterangan' => $request->input('keterangan'),
            'waktu_mulai' => $waktu_mulai,
            'tgl_mulai' => $tanggal_mulai,
            'waktu_selesai' => $waktu_selesai
        );
        DB::table('ms_panggilan_tes')->insert([$insert]);
        $jml_siswa = $request->input('idsiswa');
        if(count($jml_siswa)>0)
        {
            foreach ($request->input('idsiswa') as $key =>
$data_pelamar) {
                $iddetailpelamar = Uuid::uuid4();

```

```

        if($data_pelamar!="")
        {
            $insert_pelamar = array('id'=>$iddetailpelamar,
            'idsiswa'=> Crypt::decrypt($data_pelamar),
            'idpanggilan'=>$id,
            'idlowongan' => $idlowongan
            );
            DB::table('tr_panggilan_tes')
                ->insert([$insert_pelamar]);
        }
    }

    $pesan_sukses = "Data berhasil ditambahkan";
    return redirect()
        ->intended(app_url().'/panggilan-
tes/detail/'.Crypt::encrypt($id))
        ->with('pesan_sukses', $pesan_sukses);
    }
}

```

4.4.10 Kelola Pendaftaran Perusahaan

Kelola pendaftaran perusahaan dilakukan oleh Admin_PendaftaranPerusahaanController yang terdapat pada direktori \app\http\Controllers dan file View pada direktori \resources\views\admin\data_pendaftaran_perusahaan. Daftar Method yang ada dapat dilihat pada tabel 4.4 berikut.

Tabel 4.4 Method Controller Pendaftar Iduka

No	Nama Method	Keterangan
1	Index()	Method untuk menampilkan halaman utama pendaftar perusahaan
2	validasi(Request \$request)	Method untuk validasi dan insert pendaftaran perusahaan
3	destroy(Request \$request)	Method untuk hapus data pendaftar perusahaan

Kode program untuk masing-masing method adalah sebagai berikut.

```

public function index()
{
    if(Session::get('level')=='A')
    {
        $pendaftar = DB::table('tr_pendaftar_perusahaan')
            ->select('tr_pendaftar_perusahaan.*',
                'ms_perusahaan.id as idperusahaan')
            ->leftjoin('ms_perusahaan','ms_perusahaan.idpendaftar',
                '=' , 'tr_pendaftar_perusahaan.id')
            ->orderBy('created_at','desc')->get();
    }
    else if(Session::get('level')=='BKK')
    {
        $pendaftar = DB::table('tr_pendaftar_perusahaan')
            ->select('tr_pendaftar_perusahaan.*',
                'ms_perusahaan.id as idperusahaan')
            ->leftjoin('ms_perusahaan','ms_perusahaan.idpendaftar',
                '=' , 'tr_pendaftar_perusahaan.id')
            ->orderBy('created_at','desc')->get();
    }

    return view ('admin.data_pendaftar_perusahaan.index',
        compact('pendaftar'));
}

public function validasi(Request $request)
{
    $data = array();
    $id = Crypt::decrypt($request->input('id'));
    $status = 1;
    $update = DB::table('tr_pendaftar_perusahaan')
        ->where('id',$id)->update(['status'=>$status]);
    $pendaftar = DB::table('tr_pendaftar_perusahaan')
        ->where('id',$id)->first();
    $email = $pendaftar->email;
    $idperusahaan = Uuid::uuid4();

    $cek_users = DB::table('users')->where('email',$email)->count();
    if($cek_users>0)
    { }else{
        $insert = array(
            'created_at'=>date ('Y-m-d H:i:s'),
            'userid_created' => Session::get('userid'),
            'nama' => $pendaftar->nama,
            'email' => trim(strtolower($pendaftar->email)),
            'password' => $pendaftar->password_en,
            'level' => 'I',
            'isactive' => '1',
            'idperusahaan' => $idperusahaan,
        );
    }
}

```

```

        DB::table('users')->insert([$insert]);
    }

    $cek_perusahaan = DB::table('ms_perusahaan')
        ->where('email',$email)
        ->where('nama', $pendaftar->nama)->count();
    if($cek_perusahaan>0)
    { }else{

        $insert = array(
            'created_at'=>date ('Y-m-d H:i:s'),
            'userid_created' => Session::get('userid'),
            'nama' => $pendaftar->nama,
            'email' => trim(strtolower($pendaftar->email)),
            'userid' => trim(strtolower($pendaftar->email)),
            'idpendaftar' => $id,
            'id'=>$idperusahaan,
            'dilihat' => 0
        );
        DB::table('ms_perusahaan')->insert([$insert]);
    }

    if($status=='1')
    {
        $pesan_sukses = "Pendaftaran berhasil divalidasi.";
        Session::flash('pesan_sukses',$pesan_sukses);
        $data = array('msg'=>$pesan_sukses, 'status' => 1);
    }
    else if($status=='0') {
        $pesan_sukses = "Pendaftaran gagal divalidasi.";
        Session::flash('pesan_sukses',$pesan_sukses);
        $data = array('msg'=>$pesan_sukses, 'status' => 1);
    }

    return Response::json($data);
}

public function destroy(Request $request)
{
    $data = array();
    $id = Crypt::decrypt($request->input('id'));
    $del = DB::table('tr_pendaftaran_perusahaan')
        ->where('id',$id)->delete();

    if($del)
    {
        $pesan_sukses = "Data berhasil dihapus.";
        Session::flash('pesan_sukses',$pesan_sukses);
        $data = array('msg'=>$pesan_sukses, 'status' => 1);
    }
}

```

```

    }
    else{
        $pesan_error = "Data gagal dihapus.";
        Session::flash('pesan_error', $pesan_error);
        $data = array('msg'=>$pesan_error, 'status' => 0);
    }

    return Response::json($data);
}

```

4.4.11 Kelola Agenda Kegiatan

Kelola agenda kegiatan dilakukan oleh Controller Admin_AgendaController yang terdapat pada direktori `\app\http\Controllers` dan file View pada direktori `\resources\views\admin\data_agenda`. Daftar method pada Admin_AgendaController dapat dilihat pada tabel 4.5 berikut.

Tabel 4.5 Method Controller Agenda

No	Nama Method	Keterangan
1	Index()	Method untuk menampilkan halaman utama agenda kegiatan
2	create()	Method untuk menampilkan form tambah agenda kegiatan
3	edit(\$id)	Method untuk menampilkan form edit agenda kegiatan
4	show(\$id)	Method untuk menampilkan detail agenda kegiatan
5	store(Request \$request)	Method untuk proses insert agenda kegiatan
6	destroy(Request \$request)	Method untuk proses hapus agenda kegiatan
7	update(Request \$request)	Method untuk proses update agenda kegiatan

Kode program untuk masing-masing method adalah sebagai berikut.

```

public function index()
{
    if(Session::get('level')=='A')
    {
        $agenda = DB::table('ms_agenda')
            ->orderBy('created_at','desc')->get();
    }
}

```

```

else if (Session::get('level') == 'BKK')
{
    $agenda = DB::table('ms_agenda')
        ->where('ms_agenda.idbkk', Session::get('idbkk'))
        ->orderBy('created_at', 'desc')->get();
}
return view('admin.data_agenda.index', compact('agenda'));
}

public function create()
{
    return view('admin.data_agenda.create');
}

public function edit($id)
{
    try{
        $n = Crypt::decrypt($id);
        $edit = DB::table('ms_agenda')
            ->where('id', $n)
            ->first();
    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }
    return view('admin.data_agenda.edit', compact('edit'));
}

public function show($id)
{
    try{
        $n = Crypt::decrypt($id);
        $show = DB::table('ms_agenda')
            ->where('id', $n)
            ->first();
    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }

    return view('admin.data_agenda.show', compact('show'));
}

public function store(Request $request)
{
    $rules = array(
        'tanggal_mulai' => 'required|max:25',
        'tanggal_selesai' => 'required|max:25',
        'status' => 'required|max:10',
        'judul' => 'required|max:255',
    );

```

```

        'lokasi' => 'required|max:255'
    );
    $pesan_error = array(
        'required' => 'Form Input Ini Tidak
            Boleh Kosong / Harus Diisi',

    );
    $validator = Validator::make($request-
>all(), $rules, $pesan_error);

    if($validator->fails())
    {
        $pesan_error = $validator->messages();
        echo "<script> window.self.history.back(); </script>";
    }
    else
    {
        $explode_mulai = explode("/", $request-
>input('tanggal_mulai'));
        $tanggal_mulai = $explode_mulai[2].'-'. $explode_mulai[0].'-
'. $explode_mulai[1];

        $explode_selesai = explode("/", $request-
>input('tanggal_selesai'));
        $tanggal_selesai = $explode_selesai[2].'-
'. $explode_selesai[0].'-'. $explode_selesai[1];

        $id = Uuid::uuid4();
        $insert = array(
            'created_at'=>date ('Y-m-d H:i:s'),
            'agenda' => $request->input('judul'),
            'lokasi' => $request->input('lokasi'),
            'deskripsi' => $request->input('keterangan'),
            'biaya' => $request->input('biaya'),
            'tgl_mulai' => $tanggal_mulai,
            'tgl_selesai' => $tanggal_selesai,
            'waktu_mulai' => $request->input('waktu_mulai'),
            'waktu_selesai' => $request->input('waktu_selesai'),
            //'status' => $request->input('status'),
            'userid_created' => Session::get('id'),
            'id' => $id,
            'dilihat' => 0,
            'penyelenggara' => $request->input('penyelenggara')
        );
        DB::table('ms_agenda')->insert([$insert]);

        $pesan_sukses = "Data berhasil ditambahkan";
        return redirect()->intended('bkk/agenda')
            ->with('pesan_sukses', $pesan_sukses);
    }
}

```

```

}

public function destroy(Request $request)
{
    $data = array();
    $id = Crypt::decrypt($request->input('id'));
    $del = DB::table('ms_agenda')
        ->where('id', $id)->delete();

    if($del)
    {
        $pesan_sukses = "Data berhasil dihapus.";
        Session::flash('pesan_sukses', $pesan_sukses);
        $data = array('msg'=>$pesan_sukses, 'status' => 1);
    }
    else{
        $pesan_error = "Data gagal dihapus.";
        Session::flash('pesan_error', $pesan_error);
        $data = array('msg'=>$pesan_error, 'status' => 0);
    }
    return Response::json($data);
}

public function update(Request $request)
{
    $id = $request->input('id');

    $explode_mulai = explode("/", $request->input('tanggal_mulai'));
    $tanggal_mulai = $explode_mulai[2].'-'. $explode_mulai[0].'-'.
    $explode_mulai[1];

    $explode_selesai = explode("/", $request->input('tanggal_selesai'));
    $tanggal_selesai = $explode_selesai[2].'-'.
    $explode_selesai[0].'-'. $explode_selesai[1];

    $n = Crypt::decrypt($id);

    $ubah = DB::table('ms_agenda')
    ->where('id', $n)
    ->update(['agenda' => $request->input('judul'),
        'lokasi' => $request->input('lokasi'),
        'deskripsi' => $request->input('keterangan'),
        'biaya' => $request->input('biaya'),
        'tgl_mulai' => $tanggal_mulai,
        'tgl_selesai' => $tanggal_selesai,
        'updated_at'=>date ('Y-m-d H:i:s'),
        'userid_updated' => Session::get('userid'),
        'waktu_mulai' => $request->input('waktu_mulai'),
        'waktu_selesai' => $request->input('waktu_selesai'),

```

```

        'penyelenggara' => $request->input('penyelenggara')
    });

    $pesan_update = "Data agenda berhasil diubah.";
    \Session::flash('pesan_update', $pesan_update);
    return redirect()->intended('bkk/agenda')
    ->with('pesan_update', $pesan_update);
}

```

4.4.12 Kelola Pengguna

Kelola data pengguna dilakukan oleh *Controller* Admin_PenggunaController yang terdapat pada direktori `\app\http\Controllers` dan file *View* pada direktori `\resources\views\admin\data_pengguna`. Daftar method pada Admin_BerkasController dapat dilihat pada tabel 4.6 berikut.

Tabel 4.6 Method Controller Pengguna

No	Nama Method	Keterangan
1	Index()	Method untuk menampilkan halaman utama data pengguna
2	create()	Method untuk menampilkan form tambah pengguna
3	create_siswa(\$id)	Method untuk menampilkan form data pengguna siswa
4	show(\$id)	Method untuk menampilkan detail pengguna
5	edit(\$id)	Method untuk menampilkan form edit pengguna
6	edit_password(\$id)	Method untuk menampilkan form edit password pengguna
7	update(Request \$request)	Method untuk proses update pengguna
8	update_password(Request \$request)	Method untuk proses update password pengguna
9	destroy(Request \$request)	Method untuk proses hapus pengguna
10	store(Request \$request)	Method untuk proses insert data pengguna

Kode program dari masing-masing method adalah sebagai berikut.

```
public function index()
{
    if(Session::get('level')=='A')
    {
        $users = DB::table('users')->select('users.*',
            'ms_hak_akses.nama as hakakses')
            ->leftjoin('ms_hak_akses','ms_hak_akses.level',
                '=' , 'users.level')
            ->orderBy('id','ASC')->get();
    }
    else if(Session::get('level')=='BKK')
    {
        $users = DB::table('users')
            ->select('users.*','ms_hak_akses.nama as hakakses')
            ->leftjoin('ms_hak_akses','ms_hak_akses.level',
                '=' , 'users.level')
            ->leftjoin('ms_perusahaan','ms_perusahaan.id',
                '=' , 'users.idperusahaan')
            ->whereRaw(" ( ms_perusahaan.idbkk =
                '".Session::get('idbkk') ."'
                OR users.idbkk = '".Session::get('idbkk') ."' or
                users.userid_created = '".Session::get('userid') ."'
            ) ")
            ->orderBy('id','ASC')
            ->get();
    }
    return view ('admin.data_pengguna.index',
        compact('users'));
}

public function create()
{
    return view('admin.data_pengguna.create');
}

public function create_siswa($id)
{
    try{
        $n = Crypt::decrypt($id);
        $data = DB::table('ms_siswa')
            ->where('id',$n)
            ->first();
    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }

    return view('admin.data_pengguna.create_siswa',
        compact('data'));
}
```

```

}

public function show($id)
{
    try{
        $n = Crypt::decrypt($id);
        $data = DB::table('users')
            ->where('id',$n)
            ->first();
    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }
    return view('admin.data_pengguna.show',
        compact('data'));
}

public function edit($id)
{
    try{
        $n = Crypt::decrypt($id);
        $data = DB::table('users')
            ->where('id',$n)
            ->first();
    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }
    return view('admin.data_pengguna.edit',
        compact('data'));
}

public function edit_password($id)
{
    try{
        $n = Crypt::decrypt($id);
        $data = DB::table('users')
            ->where('id',$n)
            ->first();
    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }
    return view('admin.data_pengguna.edit_password',
        compact('data'));
}

public function update(Request $request)
{
    $id = $request->input('id');

```

```

    try{
        $n = Crypt::decrypt($id);

        $updated_at = date ('Y-m-d H:i:s');
        $userid_updated = Session::get('userid');

        $data = array(
            'nama'=>$request->input('nama'),
            'email'=>$request->input('email'),
            'isactive'=>$request->input('status'),
            'updated_at' => $updated_at,
            'userid_updated' => $userid_updated
        );
        DB::table('users')->where('id',$n)->update($data);
        $pesan_update = "Data pengguna berhasil diubah.";

        \Session::flash('pesan_update',$pesan_update);

        return redirect()->intended('bkk/pengguna')
            ->with('pesan_update', $pesan_update);
    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }
}

public function update_password(Request $request)
{
    $id = $request->input('id');
    $geturl = $request->input('url');
    if($geturl!="")
    {
        $url = Crypt::decrypt($geturl);
    }
    try{
        $n = Crypt::decrypt($id);

        $data = array(
            'password'=>bcrypt($request->input('password'))
        );
        DB::table('users')->where('id',$n)->update($data);

        $pesan_update = "Data password pengguna berhasil diubah.";
        \Session::flash('pesan_update',$pesan_update);

        if($geturl!="")
        {
            return redirect()->intended($url)
                ->with('pesan_update', $pesan_update);
        }
    }
}

```

```

    }
    else{
        return redirect()->intended(app_url().'/pengguna')
        ->with('pesan_update', $pesan_update);
    }
}

catch (\Exception $id) {
    return redirect()->route('404');
}
}

public function destroy(Request $request)
{
    $data = array();
    $id = Crypt::decrypt($request->input('id'));
    $del = DB::table('users')->where('id',$id)->delete();

    if($del)
    {
        $pesan_sukses = "Data berhasil dihapus.";
        Session::flash('pesan_sukses',$pesan_sukses);
        $data = array('msg'=>$pesan_sukses, 'status' => 1);
    }
    else{
        $pesan_error = "Data gagal dihapus.";
        Session::flash('pesan_error',$pesan_error);
        $data = array('msg'=>$pesan_error, 'status' => 0);
    }

    return Response::json($data);
}

public function store(Request $request)
{
    $rules = array(
        //'userid' => 'required|max:100',
        'password' => 'required|max:100',
        'nama' => 'required|max:100'
    );
    $pesan_error = array(
        'required' => 'Form Input Ini Tidak
        Boleh Kosong / Harus Diisi',
    );
    $validator = Validator::make($request-
    >all(),$rules,$pesan_error);

    if($validator->fails())
    {

```

```

        $pesan_error = $validator->messages();
        echo "<script>window.self.history.back();</script>";
    }
    else
    {
        $email = strtolower($request->input('email'));
        $users = DB::table('users')->where('email',$email)->first();
        if(isset($users))
        {
            $pesan_simpan = "Email sudah terdaftar.";
            \Session::flash('pesan_simpan',$pesan_simpan);
        } else {
            if($request->input('level')=="BKK")
            {
                $userid = $request->input('email');
            }
            else { $userid = $request->input('userid'); }

            $insert = array(
                'userid'=>$userid,
                'nama'=>$request->input('nama'),
                'email'=>$email,
                'isactive'=>$request->input('status'),
                'level'=>$request->input('level'),
                'created_at'=>date('Y-m-d H:i:s'),
                'userid_created'=>\Session::get('userid'),
                'password' => bcrypt($request->input('password')),
                'idbkk' => \Session::get('idbkk')
            );
            DB::table('users')->insert([$insert]);

            $pesan_simpan = "Tambah pengguna berhasil ditambahkan.";
            \Session::flash('pesan_simpan',$pesan_simpan);
        }
        return redirect()->intended('bkk/pengguna')
        ->with('pesan_simpan', $pesan_simpan);
    }
}

```

4.4.13 Kelola FAQ

Kelola data FAQ dilakukan oleh *Controller Admin_FAQController* yang terdapat pada direktori `\app\http\Controllers` dan file `View` pada direktori `\resources\views\admin\data_faq`. Daftar method pada *Admin_BerkasController* dapat dilihat pada tabel 4.7 berikut.

Tabel 4.7 Method Controller FAQ

No	Nama Method	Keterangan
1	Index()	Method untuk menampilkan halaman utama data FAQ
2	create()	Method untuk menampilkan form tambah FAQ
3	edit(\$id)	Method untuk menampilkan form edit FAQ
4	show(\$id)	Method untuk menampilkan detail FAQ
5	store(Request \$request)	Method untuk proses insert data FAQ
6	destroy(Request \$request)	Method untuk proses hapus data FAQ
7	update(Request \$request)	Method untuk proses update data FAQ

Kode program untuk masing-masing method adalah sebagai berikut.

```

public function index()
{
    $faq = DB::table('ms_faq')
        ->orderBy('created_at', 'desc')->get();
    return view('admin.data_faq.index', compact('faq'));
}

public function create()
{
    return view('admin.data_faq.create');
}

public function edit($id)
{
    try{
        $n = Crypt::decrypt($id);
        $data = DB::table('ms_faq')
            ->where('id', $n)
            ->first();
    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }
    return view('admin.data_faq.edit', compact('data'));
}

public function show($id)
{

```

```

    try{
        $n = Crypt::decrypt($id);
        $data = DB::table('ms_faq')
            ->where('id',$n)
            ->first();
    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }
    return view('admin.data_faq.show',compact('data'));
}

public function store(Request $request)
{
    $rules = array(
        'pertanyaan' => 'required|max:1000',
        'jawaban' => 'required|max:1000'
    );
    $pesan_error = array(
        'required' => 'Form Input Ini
        Tidak Boleh Kosong / Harus Diisi',
    );
    $validator = Validator::make($request-
    >all(),$rules,$pesan_error);

    if($validator->fails())
    {
        $pesan_error = $validator->messages();
        echo "<script> window.self.history.back(); </script>";
    }
    else
    {
        $id = Uuid::uuid4();
        $insert = array(
            'created_at'=>date ('Y-m-d H:i:s'),
            'userid_created' => Session::get('id'),
            'pertanyaan' => $request->input('pertanyaan'),
            'jawaban' => $request->input('jawaban'),
            'status' => $request->input('status'),
            'id' => $id
        );
        DB::table('ms_faq')->insert([$insert]);

        $pesan_sukses = "Data berhasil ditambahkan";
        return redirect()->intended('bkk/faq')
        ->with('pesan_sukses', $pesan_sukses);
    }
}

public function destroy(Request $request)

```

```

{
    $data = array();
    $id = Crypt::decrypt($request->input('id'));
    $del = DB::table('ms_faq')->where('id',$id)->delete();

    if($del)
    {
        $pesan_sukses = "Data berhasil dihapus.";
        Session::flash('pesan_sukses',$pesan_sukses);
        $data = array('msg'=>$pesan_sukses, 'status' => 1);
    }
    else{
        $pesan_error = "Data gagal dihapus.";
        Session::flash('pesan_error',$pesan_error);
        $data = array('msg'=>$pesan_error, 'status' => 0);
    }
    return Response::json($data);
}

public function update(Request $request)
{
    $id = $request->input('id');

    $n = Crypt::decrypt($id);

    $ubah = DB::table('ms_faq')
    ->where('id',$n)
    ->update([ 'pertanyaan' => $request->input('pertanyaan'),
        'jawaban' => $request->input('jawaban'),
        'status' => $request->input('status'),
        'updated_at'=>date('Y-m-d H:i:s'),
        'userid_updated' => Session::get('userid')
        ]);

    $pesan_update = "Data FAQ berhasil diubah.";
    \Session::flash('pesan_update',$pesan_update);
    return redirect()->intended('bkk/faq')
        ->with('pesan_update', $pesan_update);
}

```

4.4.14 Kelola Panggilan Tes

Kelola panggilan tes dan seleksi dilakukan oleh *Controller Admin_PanggilanController* yang terdapat pada direktori `\app\http\Controllers` dan file *View* pada direktori `\resources\views\admin\data_panggilan_tes`. Daftar method pada *Admin_PanggilanController* dapat dilihat pada tabel 4.8 berikut.

Tabel 4.8 Method Controller Panggilan Tes

No	Nama Method	Keterangan
1	Index()	Method untuk menampilkan halaman utama data panggilan tes
2	show(\$id)	Method untuk menampilkan detail panggilan tes
3	seleksi(\$id)	Method untuk menampilkan data yang ikut seleksi
4	validasi(Request \$request)	Method untuk proses validasi pada panggilan tes

Kode program untuk masing-masing method adalah sebagai berikut.

```
public function index()
{
    if(Session::get('level')=='A')
    {
        $panggilan = DB::table('ms_panggilan_tes')
        ->select('ms_panggilan_tes.*','ms_lowongan.lowongan',
            'ms_perusahaan.nama as nama_perusahaan',
            'ms_lowongan.lowongan')
        ->leftjoin('ms_lowongan','ms_lowongan.id',
            '=' , 'ms_panggilan_tes.idlowongan')
        ->leftjoin('ms_perusahaan','ms_perusahaan.id',
            '=' , 'ms_lowongan.idperusahaan')
        ->orderBy('ms_panggilan_tes.created_at','desc')
        ->get();
    }
    else if(Session::get('level')=='BKK')
    {
        $panggilan = DB::table('ms_panggilan_tes')
        ->select('ms_panggilan_tes.*','ms_lowongan.lowongan',
            'ms_perusahaan.nama as
nama_perusahaan','ms_lowongan.lowongan')
        ->leftjoin('ms_lowongan','ms_lowongan.id',
            '=' , 'ms_panggilan_tes.idlowongan')
        ->leftjoin('ms_perusahaan','ms_perusahaan.id',
            '=' , 'ms_lowongan.idperusahaan')
        ->where('ms_lowongan.idbkk',Session::get('idbkk'))
        ->orderBy('ms_panggilan_tes.created_at','desc')
        ->get();
    }
    else if(Session::get('level')=='I'){
        $panggilan = DB::table('ms_panggilan_tes')
        ->select('ms_panggilan_tes.*','ms_lowongan.lowongan',
            'ms_perusahaan.nama as
nama_perusahaan','ms_lowongan.lowongan')
```

```

        ->leftjoin('ms_lowongan','ms_lowongan.id',
            '=', 'ms_panggilan_tes.idlowongan')
        ->leftjoin('ms_perusahaan','ms_perusahaan.id',
            '=', 'ms_lowongan.idperusahaan')
    }
    -
>where('ms_panggilan_tes.userid_created',Session::get('userid'))
    ->orderBy('ms_panggilan_tes.created_at','desc')->get();
    }
    return view ('admin.data_panggilan_tes.index',
        compact('panggilan'));
    }

```

```

public function show($id)
{
    try{
        $n = Crypt::decrypt($id);
    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }
}

```

```

$ Panggilan = DB::table('ms_panggilan_tes')
    ->select('ms_panggilan_tes.*','ms_lowongan.lowongan',
        'ms_perusahaan.nama as nama_perusahaan',
        'ms_lowongan.lowongan')
    ->leftjoin('ms_lowongan','ms_lowongan.id',
        '=', 'ms_panggilan_tes.idlowongan')
    ->leftjoin('ms_perusahaan','ms_perusahaan.id',
        '=', 'ms_lowongan.idperusahaan')
    ->where('ms_panggilan_tes.id',$n)
    ->orderBy('ms_panggilan_tes.created_at','desc')
    ->first();
$idlowongan = $ Panggilan->idlowongan;

$ Pelamar = DB::table('tr_panggilan_tes')
    ->select('tr_panggilan_tes.*','ms_siswa.nama as namasiswa',
        'ms_siswa.nik','ms_siswa.hp','ms_jurusan.nama as
nama_jurusan')
    ->leftjoin('ms_panggilan_tes','ms_panggilan_tes.id',
        '=', 'tr_panggilan_tes.idpanggilan')
    ->leftjoin('ms_siswa','ms_siswa.id',
        '=', 'tr_panggilan_tes.idsiswa')
    ->leftjoin('ms_jurusan','ms_jurusan.id',
        '=', 'ms_siswa.idjurusan')

    ->where('tr_panggilan_tes.idlowongan',$idlowongan)
    ->where('ms_panggilan_tes.id',$n)
    ->orderBy('tr_panggilan_tes.created_at','asc')->get();

    return view ('admin.data_panggilan_tes.show',

```

```

        compact('panggilan','pelamar','idlowongan')));
    }

    public function seleksi($id)
    {
        try{
            $n = Crypt::decrypt($id);
        }
        catch (\Exception $id) {
            return redirect()->route('404');
        }

        $panggilan = DB::table('ms_panggilan_tes')
            ->select('ms_panggilan_tes.*','ms_lowongan.lowongan',
                'ms_perusahaan.nama as
nama_perusahaan','ms_lowongan.lowongan')
            ->leftjoin('ms_lowongan','ms_lowongan.id',
                '=' , 'ms_panggilan_tes.idlowongan')
            ->leftjoin('ms_perusahaan','ms_perusahaan.id',
                '=' , 'ms_lowongan.idperusahaan')
            ->where('ms_panggilan_tes.id',$n)
            ->orderBy('ms_panggilan_tes.created_at','desc')->first();
        $idlowongan = $panggilan->idlowongan;

        $pelamar = DB::table('tr_panggilan_tes')
            ->select('tr_panggilan_tes.*','ms_siswa.nama as namasiswa',
                'ms_siswa.nik','ms_siswa.hp','ms_jurusan.nama as
nama_jurusan')
            ->leftjoin('ms_panggilan_tes','ms_panggilan_tes.id',
                '=' , 'tr_panggilan_tes.idpanggilan')
            ->leftjoin('ms_siswa','ms_siswa.id',
                '=' , 'tr_panggilan_tes.idsiswa')
            ->leftjoin('ms_jurusan','ms_jurusan.id',
                '=' , 'ms_siswa.idjurusan')

            ->where('tr_panggilan_tes.idlowongan',$idlowongan)
            ->where('ms_panggilan_tes.id',$n)
            ->orderBy('tr_panggilan_tes.created_at','asc')->get();

        return view ('admin.data_panggilan_tes.seleksi',
            compact('panggilan','pelamar','idlowongan'));
    }

    public function validasi(Request $request)
    {
        $data = array();
        $id = Crypt::decrypt($request->input('id'));
        $idlowongan = Crypt::decrypt($request->input('idlowongan'));
        $update = DB::table('tr_panggilan_tes')
            ->where('id',$id)

```

```

        ->where('idlowongan',$idlowongan)
        ->update(['status'=>'1']);

    if($update)
    {
        $pesan_sukses = "Data pelamar berhasil disetujui.";
        Session::flash('pesan_sukses',$pesan_sukses);
        $data = array('msg'=>$pesan_sukses, 'status' => 1);
    }
    else{
        $pesan_error = "Data pelamar gagal disetujui.";
        Session::flash('pesan_error',$pesan_error);
        $data = array('msg'=>$pesan_error, 'status' => 0);
    }
    return Response::json($data);
}

public function destroy(Request $request)
{
    $data = array();
    $id = Crypt::decrypt($request->input('id'));
    $del = DB::table('ms_panggilan_tes')->where('id',$id)-
>delete();

    if($del)
    {
        $pesan_sukses = "Data berhasil dihapus.";
        Session::flash('pesan_sukses',$pesan_sukses);
        $data = array('msg'=>$pesan_sukses, 'status' => 1);
    }
    else{
        $pesan_error = "Data gagal dihapus.";
        Session::flash('pesan_error',$pesan_error);
        $data = array('msg'=>$pesan_error, 'status' => 0);
    }

    return Response::json($data);
}

```

4.4.15 Kelola Profil Website

Kelola data profil website dilakukan oleh Controller Admin_ProfilController yang terdapat pada direktori \app\http\Controllers dan file View pada direktori \resources\views\admin\data_profil. Daftar method pada Admin_ProfilController dapat dilihat pada tabel xx berikut.

Tabel 4.9 Method Controller Profil

No	Nama Method	Keterangan
1	Index()	Method untuk menampilkan data profil website
2	create()	Method untuk menampilkan form tambah profil website
3	edit(\$id)	Method untuk menampilkan form edit profil website
4	show(\$id)	Method untuk menampilkan detail profil website
5	destroy_logo(Request \$request)	Method untuk menghapus hapus logo profil website
6	update(Request \$request)	Method untuk proses update profil website

Kode program untuk masing-masing method adalah sebagai berikut.

```
public function index()
{
    error_reporting(0);
    $profil = DB::table('ms_profil')
        ->orderBy('created_at', 'desc')->first();
    return view ('admin.data_profil.index', compact('profil'));
}
```

```
public function create()
{
    return view('admin.data_profil.create');
}
```

```
public function edit($id)
{
    error_reporting(0);
    try{
        $n = Crypt::decrypt($id);
        $data = DB::table('ms_profil')
            ->first();
    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }

    return view('admin.data_profil.edit',
        compact('data'));
}
```

```
public function show($id)
```

```

{
    try{
        $n = Crypt::decrypt($id);
        $data = DB::table('ms_profil')
            ->where('id',$n)
            ->first();
    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }
    return view('admin.data_profil.show',
        compact('data'));
}

public function destroy_logo(Request $request)
{
    $data = array();
    $id = Crypt::decrypt($request->input('id'));
    $ms_profil = DB::table('ms_profil')
        ->where('id',$id)->first();

    $path_dir = 'assets/images/';
    if(isset($ms_profil))
    {
        if($ms_profil->logo!="")
        {
            error_reporting(0);
            unlink($path_dir.'/'.$ms_profil->logo);
        }
    }

    $data_update = array('logo'=>null);
    $update = DB::table('ms_profil')
        ->where('id',$id)->update($data_update);

    if($update)
    {
        $pesan_sukses = "Data berhasil dihapus.";
        Session::flash('pesan_sukses',$pesan_sukses);
        $data = array('msg'=>$path_dir, 'status' => 1);
    }
    else{
        $pesan_error = "Data gagal dihapus.";
        Session::flash('pesan_error',$pesan_error);
        $data = array('msg'=>$path_dir, 'status' => 0);
    }
    return Response::json($data);
}

public function update(Request $request)

```

```

{
    $id = $request->input('id');
    $n = Crypt::decrypt($id);

    $namafile = null;
    $size = $_FILES["file"]["size"];
    if($size>0)
    {
        $filename = $_FILES['file']['name'];
        $ext = pathinfo($filename, PATHINFO_EXTENSION);
        $explode = explode(".", $filename);
        $uuid = Uuid::uuid4();
        $namafile = 'logo.png';

        // menentukan direktori tempat gambar akan disimpan
        $path_dir = 'assets/images/';

        // upload ke direktori yang sudah ditentukan
        $request->file('file')->move($path_dir,$namafile);
    }
    else{
        $namafile = null;
    }

    $ubah = DB::table('ms_profil')
    ->where('id',$n)
    ->update(['nama' => $request->input('nama'),
        'email' => trim(strtolower($request->input('email'))),
        'alamat' => $request->input('alamat'),
        'telp' => $request->input('telp'),
        'logo' => $namafile,
        'updated_at'=>date ('Y-m-d H:i:s'),
        'userid_updated' => Session::get('userid')
    ]);

    $pesan_update = "Data profil BKK berhasil diubah.";
    \Session::flash('pesan_update',$pesan_update);

    return redirect()->intended('bkk/profil')
    ->with('pesan_update', $pesan_update);
}

```

4.4.16 Kelola Tips Karir

Kelola data tips karir dilakukan oleh *Controller* Admin_TipsKarirController yang terdapat pada direktori \app\http\Controllers dan file View pada direktori \resources\views\admin\data_tips_karir. Daftar method pada Admin_TipsKarirController dapat dilihat pada Tabel 4.10 berikut.

Tabel 4.10 Method Controller Tips Karir

No	Nama Method	Keterangan
1	Index()	Method untuk menampilkan halaman utama tips karir
2	create()	Method untuk menampilkan form tambah tips karir
3	edit(\$id)	Method untuk menampilkan form edit tips karir
4	show(\$id)	Method untuk menampilkan detail tips karir
5	store(Request \$request)	Method untuk proses insert data tips karir
6	destroy(Request \$request)	Method untuk proses hapus data tips karir
7	update(Request \$request)	Method untuk proses update data tips karir

Kode program untuk masing-masing method adalah sebagai berikut.

```
public function index()
{
    if(Session::get('level')=='A')
    {
        $tips_karir = DB::table('ms_tips_karir')
        ->orderBy('created_at','desc')->get();
    }
    else if(Session::get('level')=='BKK')
    {
        $tips_karir = DB::table('ms_tips_karir')
        ->where('idbkk',Session::get('idbkk'))
        ->orderBy('created_at','desc')->get();
    }
    return view('admin.data_tips_karir.index',
        compact('tips_karir'));
}
```

```
public function create()
{
    return view('admin.data_tips_karir.create');
}
```

```
public function edit($id)
{
    try{
        $n = Crypt::decrypt($id);
        $data = DB::table('ms_tips_karir')
        ->where('id',$n)
        ->first();
    }
```

```

        catch (\Exception $id) {
            return redirect()->route('404');
        }

        return view('admin.data_tips_karir.edit',
            compact('data'));
    }

    public function show($id)
    {
        try{
            $n = Crypt::decrypt($id);
            $data = DB::table('ms_tips_karir')
                ->where('id',$n)
                ->first();
        }
        catch (\Exception $id) {
            return redirect()->route('404');
        }

        return view('admin.data_tips_karir.show',
            compact('data'));
    }

    public function store(Request $request)
    {
        $rules = array(
            'penulis' => 'required|max:255',
            'judul' => 'required|max:255',
            'tips' => 'required|max:20000'
        );
        $pesan_error = array(
            'required' => 'Form Input Ini Tidak
                Boleh Kosong / Harus Diisi',
        );
        $validator = Validator::make($request-
            >all(),$rules,$pesan_error);

        if($validator->fails())
        {

            $pesan_error = $validator->messages();
            echo "<script> window.self.history.back(); </script>";
        }
        else
        {
            if(Session::get('level')=="A")
            {
                $idbkk = $request->input('idbkk');
            }
        }
    }

```

```

    }
    else{
        $idbkk = Session::get('idbkk');
    }

    $id = Uuid::uuid4();
    $insert = array(
        'created_at'=>date ('Y-m-d H:i:s'),
        'userid_created' => Session::get('id'),
        'penulis' => $request->input('penulis'),
        'judul' => $request->input('judul'),
        'tips' => $request->input('tips'),
        'status' => $request->input('status'),
        'tipe' => $request->input('tipe'),
        'id' => $id,
        'dilihat' => 0,
        'idbkk'=>$idbkk
    );
    DB::table('ms_tips_karir')->insert([$insert]);

    $pesan_sukses = "Data berhasil ditambahkan";
    return redirect()->intended('bkk/tips-karir')
        ->with('pesan_sukses', $pesan_sukses);
}

}

public function destroy(Request $request)
{
    $data = array();
    $id = Crypt::decrypt($request->input('id'));
    $del = DB::table('ms_tips_karir')
        ->where('id',$id)->delete();

    if($del)
    {
        $pesan_sukses = "Data berhasil dihapus.";
        Session::flash('pesan_sukses',$pesan_sukses);
        $data = array('msg'=>$pesan_sukses, 'status' => 1);
    }
    else{
        $pesan_error = "Data gagal dihapus.";
        Session::flash('pesan_error',$pesan_error);
        $data = array('msg'=>$pesan_error, 'status' => 0);
    }

    return Response::json($data);
}

public function update(Request $request)
{

```

```

        $id = $request->input('id');

        $n = Crypt::decrypt($id);

        $ubah = DB::table('ms_tips_karir')
        ->where('id',$n)
        ->update(['penulis' => $request->input('penulis'),
        'judul' => $request->input('judul'),
        'tips' => $request->input('tips'),
        'status' => $request->input('status'),
        'tipe' => $request->input('tipe'),
        'updated_at'=>date('Y-m-d H:i:s'),
        'userid_updated' => Session::get('userid')
        ]);

        $pesan_update = "Data tips/artikel berhasil diubah.";
        \Session::flash('pesan_update',$pesan_update);
        return redirect()->intended('bkk/tips-karir')
        ->with('pesan_update', $pesan_update);
    }

```

4.4.17 Kelola Berkas BKK

Kelola data berkas BKK dilakukan oleh *Controller* Admin_BerkasController yang terdapat pada direktori `\app\http\Controllers` dan dua file *View* pada direktori `\resources\views\data_berkas`. Daftar method pada Admin_BerkasController dapat dilihat pada tabel 4.11 berikut.

Tabel 4.11 Method Controller Berkas BKK

No	Nama Method	Keterangan
1	Index()	Method untuk menampilkan halaman utama berkas BKK
2	create()	Method untuk menampilkan form tambah berkas BKK
3	store(Request \$request)	Method untuk proses insert data berkas BKK
4	show(\$id)	Method untuk menampilkan detail berkas BKK
5	edit(\$id)	Method untuk menampilkan form edit berkas BKK
6	update(Request \$request)	Method untuk proses update berkas BKK
7	destroy(Request \$request)	Method untuk hapus berkas BKK

Kode program untuk masing-masing method adalah sebagai berikut.

```
public function index()
{
    if(Session::get('level')=='A')
    {
        $file = DB::table('ms_file')->get();
    }
    else if(Session::get('level')=='BKK')
    {
        $file = DB::table('ms_file')
        ->where('idbkk',Session::get('idbkk'))
        ->get();
    }
    return view('admin.data_berkas.index', compact('file'));
}

public function create()
{
    return view('admin.data_berkas.create');
}

public function show($id)
{
    try{
        $n = Crypt::decrypt($id);
        $ms_file = DB::table('ms_file')
        ->where('id',$n)->first();
        return view('admin.data_berkas.show', compact('ms_file'));
    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }
}

public function edit($id)
{
    try{
        $n = Crypt::decrypt($id);
        $file = DB::table('ms_file')
        ->where('id',$n)
        ->first();

        return view('admin.data_galeri.edit', compact('file'));
    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }
}
```

```

public function update(Request $request)
{
    $id = $request->input('id');

    try{
        $n = Crypt::decrypt($id);
        $nama = $request->input('nama');

        $status = $request->input('status');
        $updated_at = date('Y-m-d H:i:s');
        $userid_updated = Session::get('userid');

        $ubah = DB::table('ms_file')
        ->where('id', $n)
        ->update(['nama'=> $nama,
                'status' => $status,
                'updated_at' => $updated_at,
                'userid_updated' => $userid_updated
                ]);

        $galeri= DB::table('ms_file')
        ->orderBy('created_at', 'DESC')
        ->get();

        $pesan_update = "Data berkas berhasil diubah.";
        \Session::flash('pesan_update', $pesan_update);

        return redirect()->intended('bkk/berkas')
        ->with('pesan_update', $pesan_update);

    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }
}

public function destroy(Request $request)
{
    error_reporting(0);
    $data = array();
    $id = Crypt::decrypt($request->input('id'));
    $data_hapus = DB::table('ms_file')->where('id', $id)->first();

    $hapus_berkas = unlink('assets/berkas/'. $data_hapus->namafile);

    $del = DB::table('ms_file')->where('id', $id)->delete();
    if($del)
    {
        $pesan_sukses = "Data berhasil dihapus.";
    }
}

```

```

        Session::flash('pesan_sukses', $pesan_sukses);
        $data = array('msg'=>$pesan_sukses, 'status' => 1);
    }
    else{
        $pesan_error = "Data gagal dihapus.";
        Session::flash('pesan_error', $pesan_error);
        $data = array('msg'=>$pesan_error, 'status' => 0);
    }

    return Response::json($data);
}

```

4.4.18 Kelola Pendaftar Siswa/Alumni

Kelola data daftar akun Siswa dan alumni dilakukan oleh *Controller* Admin_PendaftarSiswaController yang terdapat pada direktori \app\http\Controllers dan delapan file View pada direktori \resources\views\admin\data_pendaftar. Daftar method pada Admin_PendaftarSiswaController dapat dilihat pada tabel xx berikut.

Tabel 4.12 Method Controller Pendaftar Siswa

No	Nama Method	Keterangan
1	Index()	Method untuk menampilkan halaman utama data pendaftar siswa/alumni
2	create()	Method untuk menampilkan form pendaftaran siswa/alumni
3	edit(\$id)	Method untuk edit data pendaftar siswa/alumni
4	show(\$id)	Method untuk menampilkan detail siswa/alumni
5	store(Request \$request)	Method untuk proses insert data pendaftar siswa/alumni
6	destroy(Request \$request)	Method untuk hapus pendaftar siswa/alumni
7	update(Request \$request)	Method untuk proses update siswa/alumni
8	validasi(Request \$request)	Method untuk validasi pendaftar siswa/alumni
9		

Kode program untuk masing-masing method adalah sebagai berikut.

```
public function index()
```

```

{
    if(Session::get('level')=='A')
    {
        $pendaftar = DB::table('ms_pendaftar')
            ->orderBy('created_at','desc')->get();
    }
    else if(Session::get('level')=='BKK'){
        $pendaftar = DB::table('ms_pendaftar')
            ->where('idbkk',Session::get('idbkk'))
            ->orderBy('created_at','desc')->get();
    }

    return view ('admin.data_pendaftar_siswa.index',
        compact('pendaftar'));
}

public function create()
{
    return view('admin.data_pendaftar_siswa.create');
}

public function edit($id)
{
    try{
        $n = Crypt::decrypt($id);
        $data = DB::table('ms_siswa')
            ->where('id',$n)
            ->first();
    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }

    return view('admin.data_pendaftar_siswa.edit',
        compact('data'));
}

public function show($id)
{
    try{
        $n = Crypt::decrypt($id);
        $data = DB::table('ms_siswa')
            ->where('id',$n)
            ->first();
    }
    catch (\Exception $id) {
        return redirect()->route('404');
    }
}

```

```

        return view('admin.data_pendaftar_siswa.show', compact('data'));
    }
    public function store(Request $request)
    {
        $rules = array(
            'nama' => 'required|max:255'
        );
        $pesan_error = array(
            'required' => 'Form input harus diisi.'
        );
        $validator = Validator::make($request->all(), $rules, $pesan_error);

        if($validator->fails())
        {
            $pesan_error = $validator->messages();
            echo "<script> window.self.history.back(); </script>";
        }
        else
        {
            $id = Uuid::uuid4();
            $insert = array(
                'id'=>$id,
                'created_at'=>date('Y-m-d H:i:s'),
                'userid_created' => Session::get('userid'),
                'nama' => $request->input('nama'),
                'status' => $request->input('status')
            );
            DB::table('ms_siswa')->insert([$insert]);

            $pesan_sukses = "Data berhasil ditambahkan";
            return redirect()->intended('bkk/siswa/pendaftar')
                ->with('pesan_sukses', $pesan_sukses);
        }
    }

    public function destroy(Request $request)
    {
        $data = array();
        $id = Crypt::decrypt($request->input('id'));
        $del = DB::table('ms_pendaftar')->where('id', $id)->delete();

        if($del)
        {
            $pesan_sukses = "Data berhasil dihapus.";
            Session::flash('pesan_sukses', $pesan_sukses);
            $data = array('msg'=>$pesan_sukses, 'status' => 1);
        }
    }

```

```

else{
    $pesan_error = "Data gagal dihapus.";
    Session::flash('pesan_error', $pesan_error);
    $data = array('msg'=>$pesan_error, 'status' => 0);
}
return Response::json($data);
}

public function update(Request $request)
{
    $id = $request->input('id');
    $n = Crypt::decrypt($id);

    $ubah = DB::table('ms_siswa')
    ->where('id', $n)
    ->update(['nama' => $request->input('nama'),
            'status' => $request->input('status'),
            'updated_at'=>date ('Y-m-d H:i:s'),
            'userid_updated' => Session::get('userid')
            ]);

    $pesan_update = "Data siswa berhasil diubah.";
    \Session::flash('pesan_update', $pesan_update);

    return redirect()->intended('bkk/siswa/pendaftar')
    ->with('pesan_update', $pesan_update);
}

public function validasi(Request $request)
{
    $data = array();
    $id = Crypt::decrypt($request->input('id'));
    $status = 1;
    $update = DB::table('ms_pendaftar')
    ->where('id', $id)->update(['status'=>$status]);
    $pendaftar = DB::table('ms_pendaftar')
    ->where('id', $id)->first();
    $email = $pendaftar->email;

    $cek_users = DB::table('users')->where('email', $email)->count();
    if($cek_users>0)
    { }else{
        $insert = array(
            'created_at'=>date ('Y-m-d H:i:s'),
            'userid_created' => Session::get('userid'),
            'nama' => $pendaftar->nama,
            'email' => trim(strtolower($pendaftar->email)),
            'password' => $pendaftar->password,

```

```

        'level' => 'S',
        'isactive' => '1'
    );
    DB::table('users')->insert([$insert]);
}

$ccek_siswa = DB::table('ms_siswa')->where('email',$email)->count();
if($ccek_siswa>0)
{ }else{
    $idsiswa = Uuid::uuid4();
    $insert_siswa = array(
        'created_at'=>date('Y-m-d H:i:s'),
        'userid_created' => Session::get('userid'),
        'nama' => $pendaftar->nama,
        'email' => trim(strtolower($pendaftar->email)),
        'idpendaftar' => $id,
        'id'=>$idsiswa
    );
    DB::table('ms_siswa')->insert([$insert_siswa]);
}

if($status=='1')
{
    $pesan_sukses = "Validasi berhasil.";
    Session::flash('pesan_sukses',$pesan_sukses);
    $data = array('msg'=>$pesan_sukses, 'status' => 1);
}
else if($status=='0') {
    $pesan_sukses = "Validasi gagal.";
    Session::flash('pesan_sukses',$pesan_sukses);
    $data = array('msg'=>$pesan_sukses, 'status' => 1);
}

return Response::json($data);
}

```

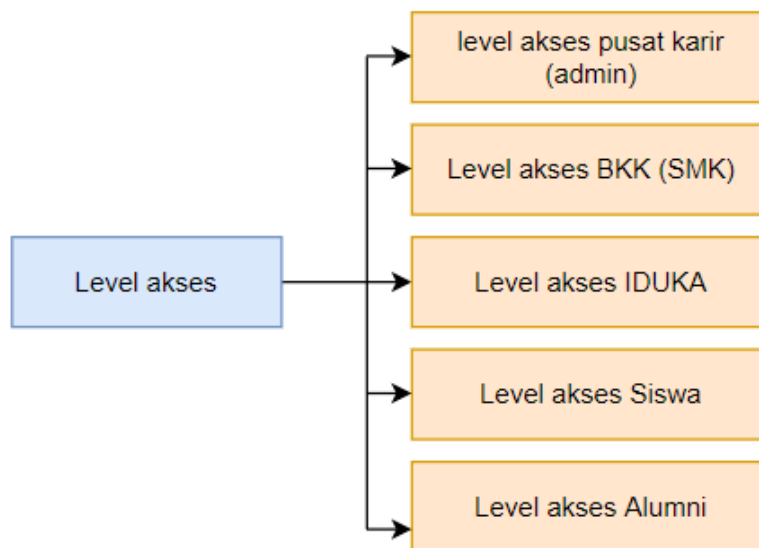
BAB 5

LEVEL AKSES DAN ANTARMUKA GRAFIS

Pada bab ini dibahas hasil dari pengembangan aplikasi pusat karir SMK dalam bentuk website yang telah online dan telah dilakukan pengujian oleh beberapa BKK dari SMK di Bojonegoro.

5.1 Level Akses

Terdapat lima (5) level akses pada aplikasi pusat karir SMK dengan fitur yang berbeda-beda. Level akses tersebut terlihat pada gambar 5.1 berikut.



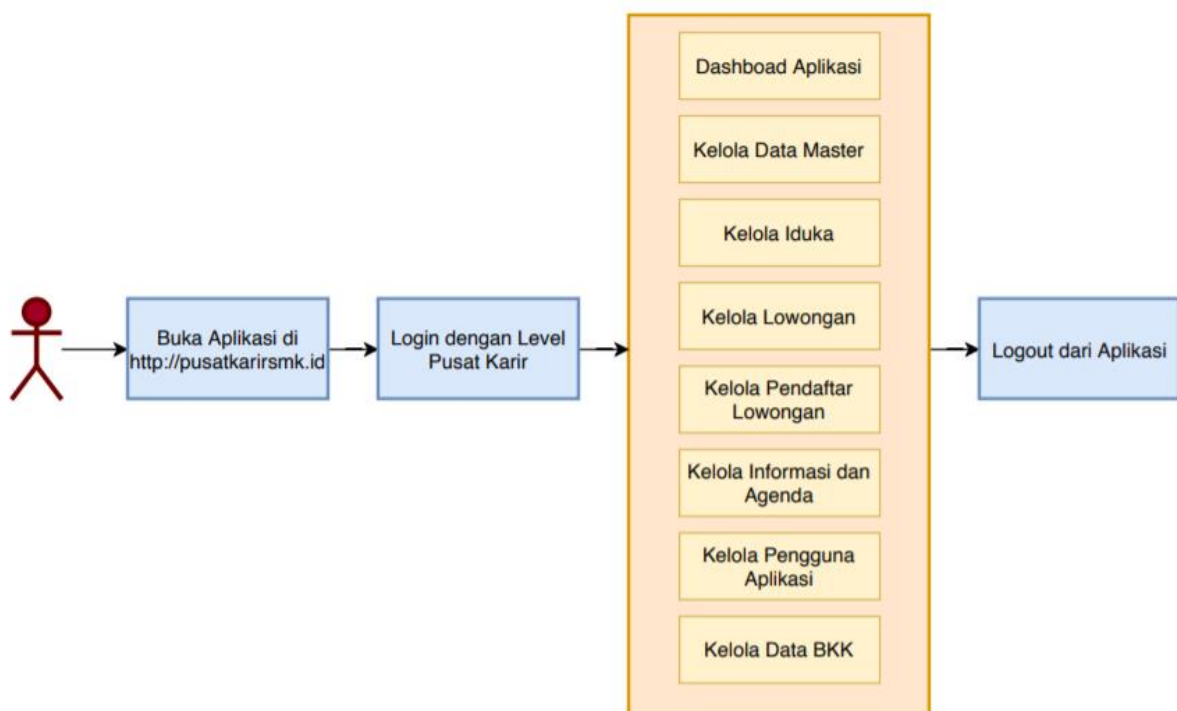
Gambar 5.1 Level Akses Aplikasi

Level akses tertinggi adalah pusat karir (admin) yang memiliki hak akses atas keseluruhan fitur pada aplikasi pusat karir. Level akses berikutnya adalah BKK yang juga mewakili SMK, yang memiliki hak akses sesuai kebutuhan pengelolaan BKK. Level akses IDUKA digunakan oleh industri terkait pengelolaan rekrutmen tenaga kerja meliputi publish lowongan kerja, apply lowonga kerja, dan panggilan tes. Level akses siswa dan alumni memiliki fitur yang hampir sama, bedanya hanya pada proses *tracer study*. Level akses tidak sama dengan user, karena level akses tidak bisa digunakan untuk login. Level akses bisa

diibaratkan seperti *group*, sedangkan *user* adalah anggota *group*. Untuk proses login, yang digunakan adalah *user*, bukan level akses.

5.2 Fitur Aplikasi sesuai Level Akses

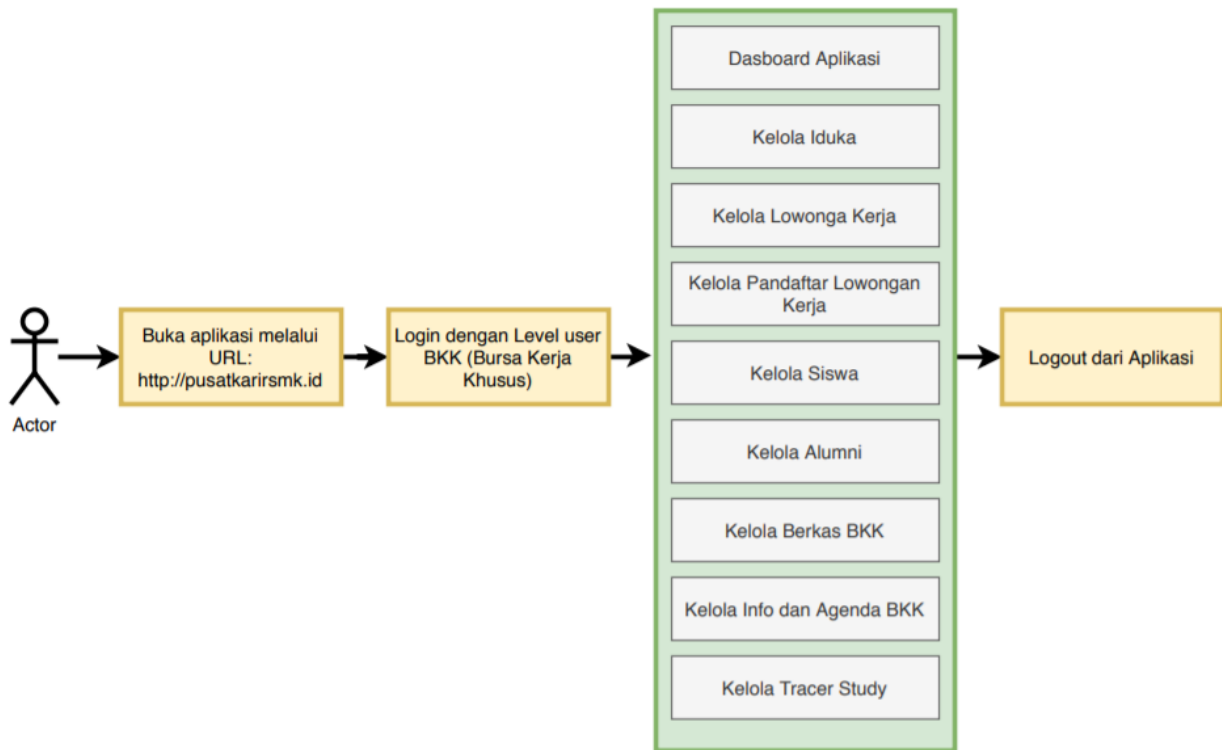
Setiap level akses memiliki fitur yang berbeda antar satu dengan yang lain. Setiap *user* menjadi anggota dari satu level akses, artinya tidak ada satu *user* yang menjadi anggota lebih dari satu level akses. Gambar 5.2 berikut adalah fitur untuk level akses pusat karir (admin).



Gambar 5.2 Fitur Level Akses Admin

Berdasarkan gambar xx, *user* dengan level akses pusatkarir membuka website pusat karir SMK melalui laman <http://pusatkarirsmk.id>. Langkah berikutnya adalah melakukan proses login dengan mengisi username dan password yang telah dimiliki. Setelah proses login berhasil, maka pengguna dengan level akses pusat karir SMK dapat menggunakan fitur-fitur seperti pada gambar 5.2. Jangan lupa logout, untuk menjamin keamanan akun yang dimiliki.

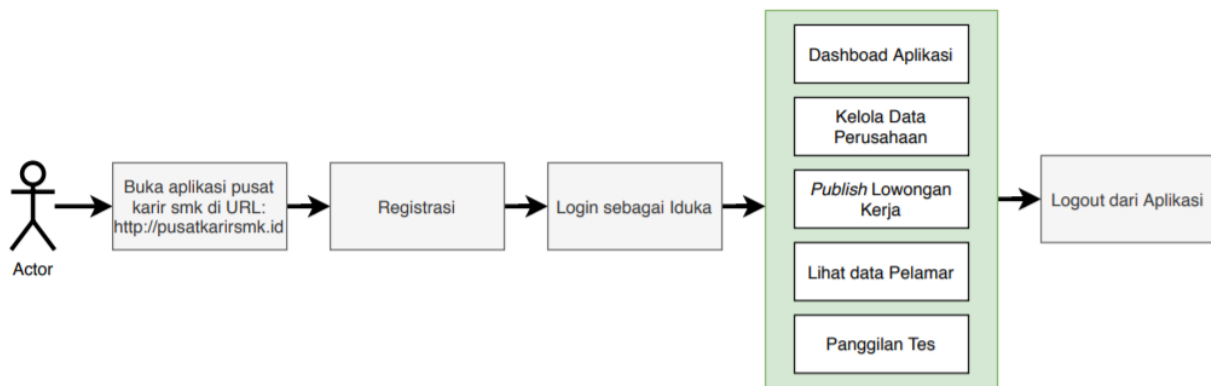
Pengguna dengan level BKK memiliki fitur-fitur sesuai kebutuhan dalam pengelolaan BKK disekolah. Fitur tersebut tersaji pada gambar 5.3 berikut.



Gambar 5.3 Fitur Level Akses BKK

Berdasarkan Gambar 5.3, akun level BKK membuka website pusat karir dengan alamat <http://pusatkarirsmk.id>, kemudian melakukan proses login dengan memasukkan username dan password pada form login yang tersedia. Ketika proses login berhasil, maka pengguna dapat menggunakan semua fitur untuk level BKK yaitu dashboard, Kelola Iduka, Kelola lowongan kerja, Kelola pendaftar lowongan kerja, Kelola siswa, Kelola alumni, Kelola berkas BKK, Kelola info dan agenda BKK, dan kelalola *tracer study*.

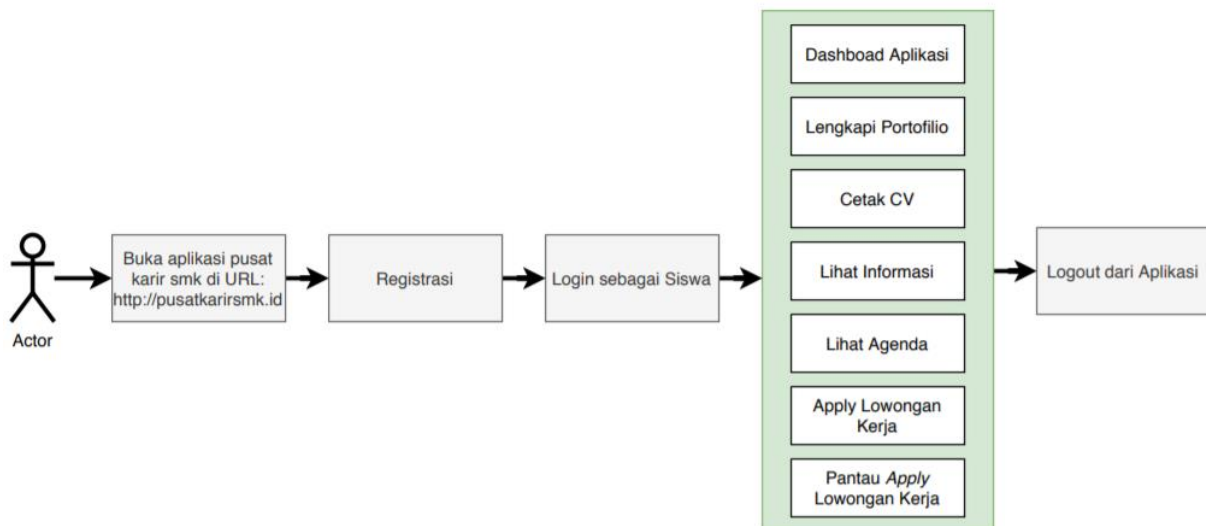
Pengguna dengan level IDUKA memiliki fitur sesuai kebutuhan yakni terkait dengan lowongan kerja, apply lowongan kerja, dan panggilan tes. Gambar 5.4 menampilkan fitur untuk IDUKA.



Gambar 5.4 Fitur Level Akses IDUKA

Sesuai gambar 5.4, pengguna dengan level IDUKA dapat melakukan login bagi yang telah memiliki akun, dan dapat melakukan registrasi bagi yang belum. Proses login dilakukan dengan memasukkan username dan password yang telah dimiliki. Username yang digunakan adalah alamat email. Setelah proses login berhasil, pengguna level IDUKA dapat menggunakan fitur dashboard, Kelola data perusahaan, publish lowongan kerja, lihat data pelamar, dan Kelola panggilan tes. Logout dari aplikasi jika sudah selesai.

Pengguna dengan level akses siswa dan alumni memiliki kemiripan fitur, bedanya status siswa jika pengguna tersebut masih aktif disekolah. Fitur pengguna untuk level akses siswa dan alumni tersaji pada gambar 5.5 berikut.



Gambar 5.5 Fitur Level Akses Siswa

Sesuai Gambar 5.5, siswa dan alumni dapat melakukan registrasi terlebih dahulu untuk memiliki akun. Setelah itu, dapat melakukan login menggunakan akun yang telah dimiliki

dengan mengisi username yang berupa alamat email dan password. Setelah berhasil login, siswa dan alumni dapat menggunakan fitur yang tersedia yaitu dashboard, lengkapi profil, cetak CV (*Curriculum Vitae*), lihat informasi, lihat agenda, *apply* lowongan kerja, pantau lowongan kerja, dan pengelolaan berkas portofolio. Melakukan *logout* jika telah selesai, demi keamanan akun yang dimiliki.

5.3 Hasil dan Pembahasan

Pada bagian ini, dibahas hasil dari pengembangan aplikasi pusat karir SMK disertai dengan penjelasannya. Setiap pembahasan disertai dengan tampilan aplikasi, fungsinya, dan data yang digunakan. Selain itu, dijelaskan juga kegunaan tautan dan kaitannya dengan tampilan yang lain. Hasil dari pembahasan dari aplikasi pusat karir SMK adalah sebagai berikut.

1. Laman Utama

Merupakan tampilan utama ketika pengguna membuka website pusat karir SMK melalui alamat <http://pusatkarirsmk.id>. Tampilan laman utama dapat dilihat pada gambar 5.6 berikut.



Gambar 5.6 Laman Utama Pusat Karir SMK

Informasi yang ditampilkan pada laman utama antara lain:

- a. Form pencarian lowongan kerja
- b. Jumlah pengguna aktif, jumlah lowongan, dan jumlah perusahaan

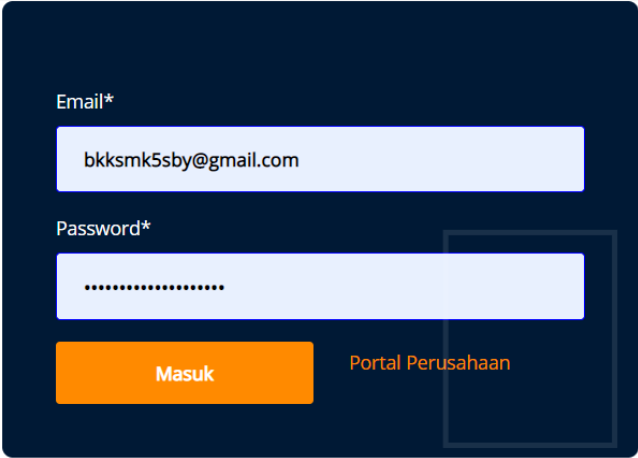
- c. Tautan: Beranda, Lowongan, Perusahaan, Agenda, Tips/Artikel, FAQ, Login, siswa/alumni, dan perusahaan.

Penjelasan dari masing-masing tautan pada laman utama adalah sebagai berikut:

- a. Beranda, tautan untuk menampilkan laman utama aplikasi pusat karir SMK
- b. Lowongan, tautan untuk menampilkan data semua lowongan yang aktif dan siswa dapat *apply* lowongan.
- c. Perusahaan, tautan untuk menampilkan data perusahaan beserta jumlah lowongan yang tersedia pada perusahaan tersebut.
- d. Agenda, tautan untuk menampilkan agenda pusat karir atau BKK.
- e. Tips/Artikel, tautan untuk menampilkan informasi dan tips terkait pekerjaan, kepribadian, dan memasuki dunia kerja.
- f. FAQ, tautan untuk menampilkan daftar pertanyaan yang sering ditanyakan agar membantu pengguna yang lain.
- g. Login, tautan untuk menampilkan form login yang digunakan oleh semua pengguna dalam menggunakan aplikasi pusatkarir.
- h. Siswa/alumni, tautan untuk menampilkan laman registrasi siswa/alumni dalam aplikasi ini.
- i. Perusahaan, tautan bagi perusahaan yang akan melakukan registrasi pada aplikasi ini.

2. Laman Login

Laman ini berfungsi sebagai pintu masuk pada aplikasi pusat karir SMK. Semua pengguna yang akan menggunakan fitur sesuai level akses harus melakukan proses login terlebih dahulu.



Email*

bkksmk5sby@gmail.com

Password*

.....

Masuk

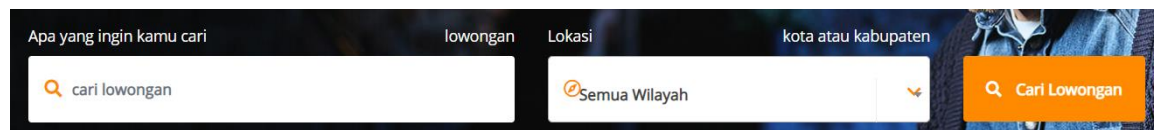
Portal Perusahaan

Gambar 5.7 Form Login Pusat Karir SMK

Pada laman ini, pengguna harus memasukkan username dan password. Untuk semua pengguna, username yang digunakan adalah berupa alamat email. Setelah memasukkan username dan password, kemudian klik tombol Masuk, maka pengguna dibawa ke laman *dashboard*. Jika username dan password ada yang salah, maka ditampilkan pesan kesalahan.

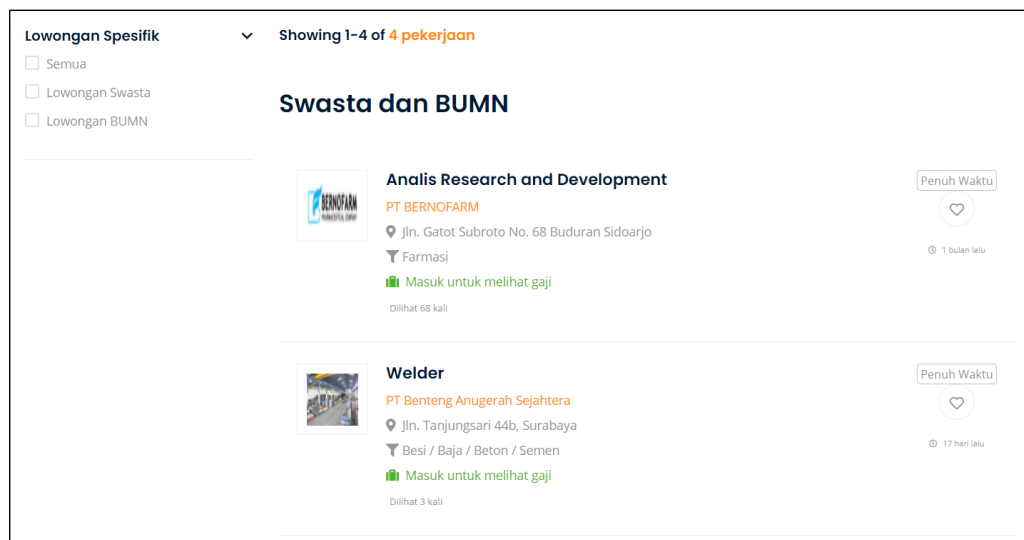
3. Laman Lowongan kerja

Laman ini menampilkan data lowongan pekerjaan yang tersedia. Lowongan pekerjaan yang bersifat publik akan ditampilkan untuk semua pengguna, sedangkan yang local BKK hanya dapat dilihat oleh siswa dan alumni hanya yang berasal dari SMK dimana BKK tersebut berada. Pada laman ini (gambar 5.8) terdapat fitur cari lowongan kerja dan dapat difilter berdasarkan kota.



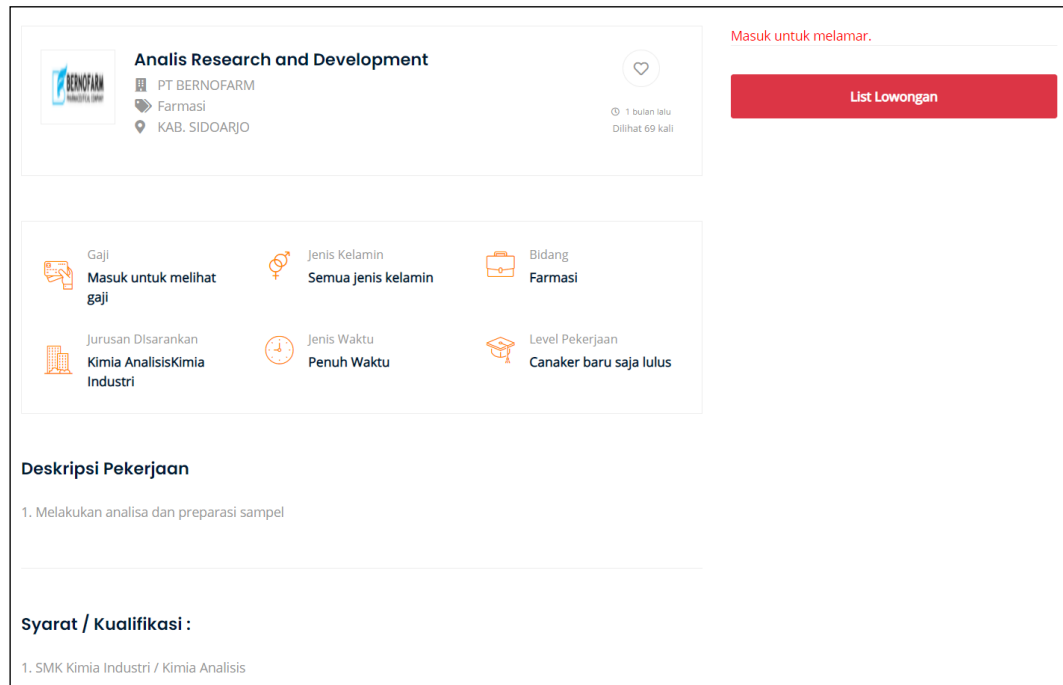
Gambar 5.8 Pencarian Logongan Kerja

Selain itu, terdapat juga fitur untuk filter data lowongan berdasarkan jenis perusahaan, apakah swasta atau BUMN. Setiap data lowongan yang ditampilkan, disertai juga dengan identitas dan logo perusahaan yang membua lowongan kerja tersebut. Detail lowongan kerja dapat dilihat melalui klik judul lowongan sehingga ditampilkan detail lowongan kerja seperti gambar 5.9.



Gambar 5.9 Laman Lowongan Kerja

Untuk mengetahui informasi detail lowongan pekerjaan, klik tautan pada tautan judul lowongan pekerjaan sehingga tampil seperti gambar 5.10 berikut.

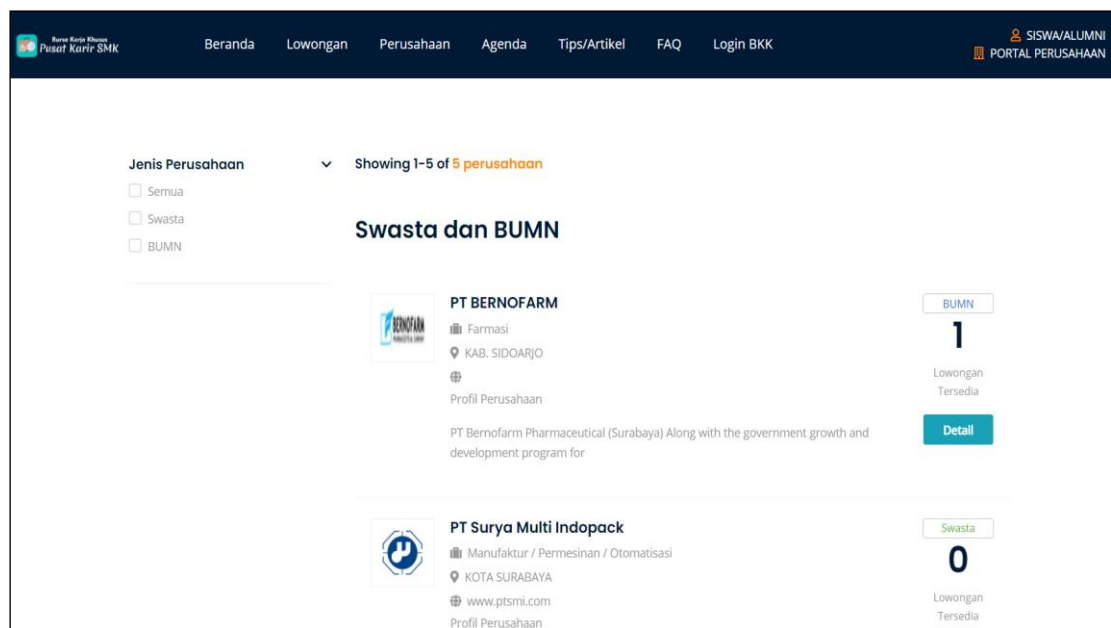


Gambar 5.10 Detail Lowongan Kerja

Pada gambar 5.10, pengguna dapat melihat detail informasi lowongan pekerjaan dan melakukan apply lowongan dengan syarat pengguna telah melakukan login terlebih dahulu.

4. Laman Perusahaan (IDUKA)

Laman ini digunakan untuk menampilkan data IDUKA yang telah terdaftar di pusat karir SMK. Agar terdaftar, IDUKA dapat melakukan registrasi secara mandiri atau didaftarkan oleh pusat karir maupun BKK. Tampilan data IDUKA untuk umum tersaji pada gambar 5.11 berikut.



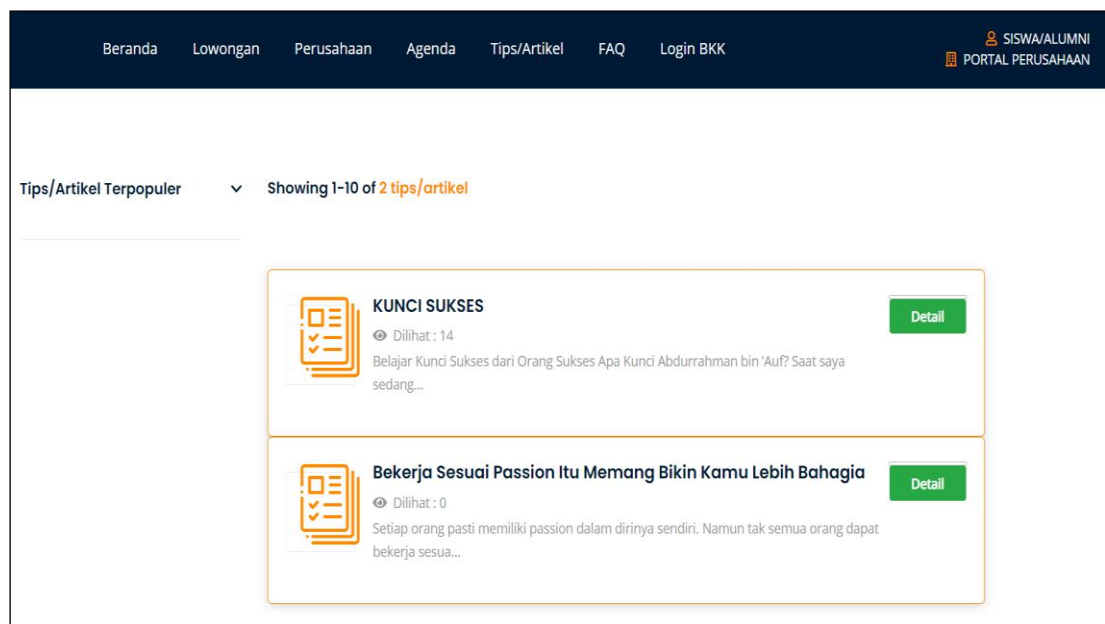
Gambar 5.11 Laman Perusahaan

Pada gambar 5.11, pengguna dapat melihat data IDUKA beserta jumlah lowongan yang tersedia. Jika jumlah lowongan yang tersedia bernilai 0, artinya IDUKA tersebut sedang tidak membuka lowongan pekerjaan.

5. Laman Tips/Artikel

Laman ini menampilkan informasi terkait tips dan artikel terkait dunia kerja, motivasi, memasuki dunia kerja, dan lain-lain. Tampilan utama seperti yang terjadi pada gambar 5.12 menampilkan artikel yang ditampilkan berdasarkan urutan waktu. Detail artikel dapat ditampilkan dengan klik tombol Detail. Pada laman detail, ditampilkan keseluruhan artikel, jumlah dilihat, dan pengirim artikel.

Pada gambar 5.12, ditampilkan juga tips/artikel yang paling populer. Penentuan tips/artikel populer didasarkan pada jumlah dilihat. Semakin tinggi nilai jumlah lihat, maka semakin populer, dan sebaliknya.



Gambar 5.12 Laman Tips dan Artikel

6. Laman Registrasi Siswa dan Alumni

Laman ini digunakan oleh siswa atau alumni untuk melakukan registrasi pada pusat karir SMK. Registrasi dilakukan agar siswa atau alumni terdaftar, memiliki akun, dan dapat menggunakan fitur-fitur yang tersedia. Data yang harus diisi pada saat registrasi antara lain:

- Nama depan
- Nama belakang
- Email, nantinya digunakan sebagai username pada saat login.
- Sandi, password untuk login.
- Jenis pendaftaran, terdapat pilihan: siswa dan alumni.
- BKK, pilihan BKK dimana siswa atau alumni tersebut sekolah.

Setelah mengisi semua isian dengan benar, klik tombol Daftar sehingga ditampilkan pesan sukses registrasi. Setelah sukses registrasi, siswa atau alumni dapat melakukan login dengan level akses siswa/alumni. Selain dengan registrasi, data siswa dan alumni dapat juga dimasukkan oleh admin BKK.

Pendaftaran Akun Siswa

Mengapa kamu harus mendaftar akun siswa di portal Pusat Karir SMK?

- Akun yang terhubung dengan BKK (Bursa Kerja Khusus) sekolah yang nantinya akan berguna untuk prakerind dan karir kamu di masa mendatang.
- Dengan fitur pengenalan diri, portal siswa Bursa Kerja Khusus - Pusat Karir SMK akan dapat memberikan kamu saran tentang kepribadian, saran pengembangan diri serta potensi kamu dalam menunjang pembelajaran serta karir kamu di kemudian hari.
- Unduh Curriculum Vitae mu secara otomatis dan dapat langsung dicetak.

Dengan mendaftar, saya telah membaca dan menyetujui [Ketentuan Penggunaan](#) dan [Kebijakan Privasi](#)

Nama Depan

Nama Belakang

Email

Sandi*

Jenis Pendaftaran

Siswa

Kelas

- Pilih Kelas -

BKK

BKK SMKN1 Baureno

Daftar

Gambar 5.13 Laman Pendaftaran Akun Siswa

Dalam registrasi siswa atau alumni sengaja tidak ada fitur aktivasi melalui email demi kemudahan dan mengurasi proses. Siswa dan alumni tidak mau melakukan registrasi jika prosesnya sulit dan panjang.

7. Laman Registrasi IDUKA

Laman ini digunakan untuk registrasi IDUKA agar terdaftar, memiliki akun dan dapat melakukan proses login. Informasi yang diisikan pada registrasi IDUKA tidak banyak yakni:

- Nama perusahaan
- Email perusahaan, isian email nantinya digunakan sebagai username pada saat login.
- Kategori, pilihan bidang usaha dari perusahaan.
- Sandi, kata sandi atau password yang digunakan pada saat login.

Tampilan laman registrasi IDUKA terlihat pada gambar 5.14.

The screenshot shows a registration form titled "Pendaftaran Perusahaan". It includes fields for company name, email, category, and password. The company name is "PT. Trugee Inovasi", the email is "admin@trugee.co.id", and the category is "Teknologi Informatika / Komputer". There are buttons for "Daftar", "Login", and "Beranda".

Pendaftaran Perusahaan

Cari karyawan terbaik untuk perusahaan anda?
Daftar dan temukan sekarang.

Nama Perusahaan
PT. Trugee Inovasi

Email Perusahaan
admin@trugee.co.id

Kategori Perusahaan
Teknologi Informatika / Komputer

Sandi*
.....

Daftar Login Beranda

Gambar 5.14 Laman Pendaftaran Perusahaan

Setelah terdaftar dan melakukan login, IDUKA dapat melengkapi data sehingga menjadi lengkap dan terbaru. Hal ini diperlukan karena informasi terkait IDUKA ikut tampil pada lowongan pekerjaan.

8. Dashboard siswa dan alumni

Laman ini merupakan tampilan ketika pengguna dengan level akses siswa atau alumni melakukan login. Informasi yang ditampilkan adalah profil, berkas pendukung, lamaran, dan kelengkapan profil.

The screenshot shows a dashboard for a student named Arif Kurniawan. It includes a profile section with a photo and contact information, an "Informasi" section with a table of job applications, and a "Kelengkapan Profil" section showing a 50% completion rate.

Arif Kurniawan
BKK SMKN1 Baureno
arifkur@pusatkarismk.id
Cetak CV

Informasi
Gunakan fitur secara penuh dengan lengkapi profil
Resume Riwayat

Total Lamaran	Lamaran Tersimpan	Ketidakhadiran
3	0	0

Kelengkapan Profil
50%

- Informasi Dasar ✓
- Profil Singkat ✓
- Informasi Kontak
- Informasi Kesehatan ✓
- Riwayat Pendidikan Formal
- Keahlian / Passion

Beranda **Profilku** Berkas Pendukung Keluar

Gambar 5.15 Dashboad Siswa dan Alumni

Berdasarkan gambar 5.13, tombol cetak CV masih tidak aktif disebabkan kelengkapan profil masih 50%. Jika kelengkapan profil telah mencapai 100%, maka siswa/alumni dapat melakukan cetak CV. Setiap tahapan melengkapi profil, maka nilai kelengkapan profil akan bertambah.

9. Profil Siswa dan Alumni

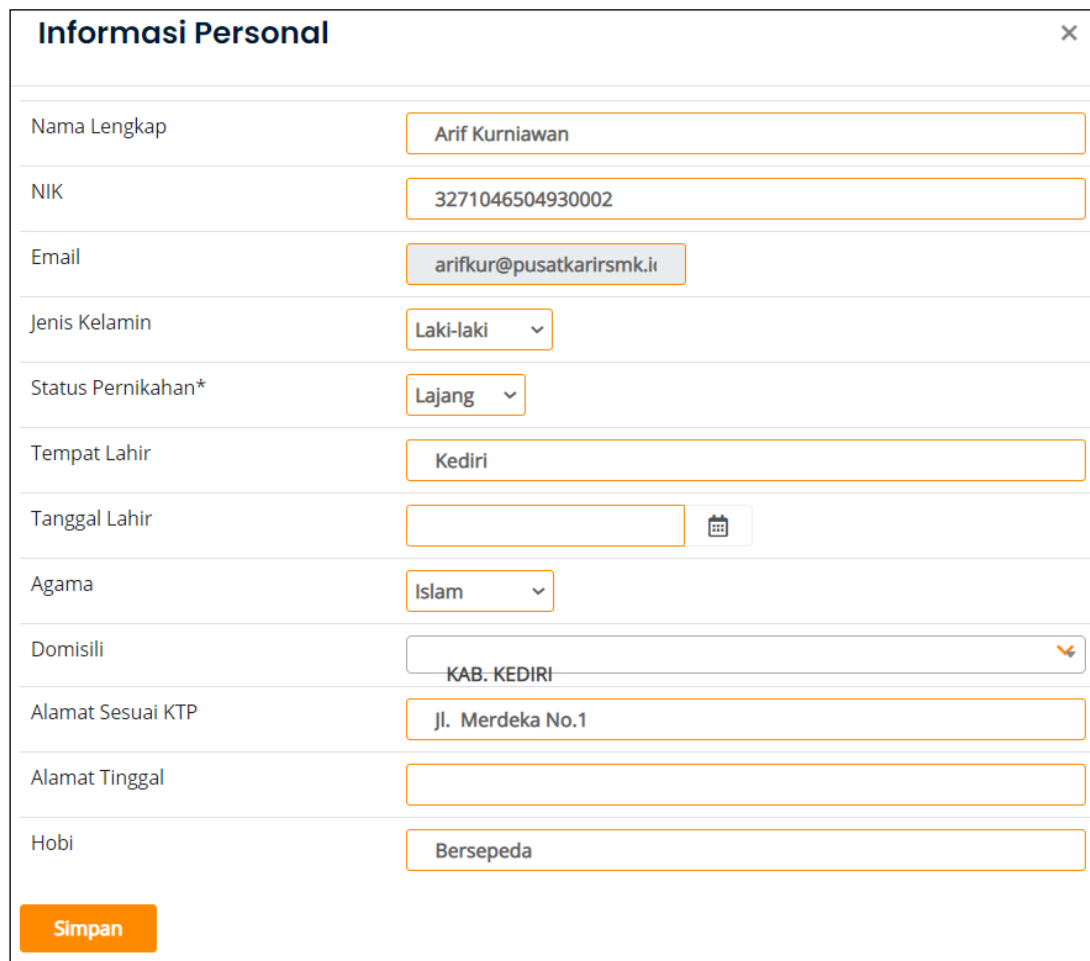
Profil siswa dan alumni terdiri dari enam (6) yakni profil dasar, informasi personal, kontak dan media sosial, informasi Kesehatan, riwayat Pendidikan formal dan keahlian atau skill, yang dapat dilihat pada gambar 5.16, 5.17, 5.18 dan 5.19.

Konfigurasi Dasar Profil*		
Jurusan	Rekayasa Perangkat Lunak	 Edit
Klaster	Alumni	
Informasi Personal*		
Nama Lengkap	Arif Kurniawan	 Edit
NIK	3271046504930002	
Email	arifkur@pusatkarirsmk.id	
Jenis Kelamin	Laki-laki	
Tempat/Tanggal Lahir	Kediri, 22-11-2003	
Agama	Islam	
Domisili	KAB. KEDIRI	
Alamat Sesuai KTP	Jl. Merdeka No.1	
Alamat Tinggal		
Hobi	Bersepeda	

Gambar 5.16 Laman Konfigurasi Profil

Berdasarkan gambar 5.16, profil dasar terdiri dari jurusan dan kluster (siswa atau alumni). Sedangkan informasi personal meliputi: nama lengkap, NIK, jenis kelamin, tempat/tgl lahir, agama, domisili, alamat sesuai ktp, alamat tinggal, dan hobi. Untuk melakukan perubahan data, klik tombol Edit seperti gambar xx berikut. Tombol Edit

ini berfungsi sebagai tambah dan edit sekaligus, artinya ketika data masih kosong maka ditambahkan dan ketika sudah terisi, maka diubah.



The image shows a web form titled "Informasi Personal" with a close button (X) in the top right corner. The form contains the following fields:

Field Label	Value
Nama Lengkap	Arif Kurniawan
NIK	3271046504930002
Email	arifkur@pusatkarismk.i
Jenis Kelamin	Laki-laki
Status Pernikahan*	Lajang
Tempat Lahir	Kediri
Tanggal Lahir	[Empty field with calendar icon]
Agama	Islam
Domisili	KAB. KEDIRI
Alamat Sesuai KTP	Jl. Merdeka No.1
Alamat Tinggal	[Empty field]
Hobi	Bersepeda

At the bottom left of the form is an orange button labeled "Simpan".

Gambar 5.17 Informasi Personal

Profil kontak dan media sosial (gambar 5.18) digunakan untuk mengisi data nomor hp dan alamat media sosial yang dimiliki. Media sosial yang harus ditambahkan yaitu facebook, twitter, linkedin, dan Instagram. Untuk menambahkan data, klik tombol Edit kemudian isikan data kontak dan media sosial.

Profil informasi Kesehatan (gambar 5.19) digunakan untuk mengelola data Kesehatan siswa dan alumni. Informasi yang dikelola yaitu golongan darah, tinggi, berat, buta warna, berkacamata, bertato, bertindik, penyandang difabilitas, dan riwayat penyakit yang pernah diderita. Untuk mengelola profil Informasi Kesehatan,

klik tautan Edit sehingga ditampilkan tambah atau edit. Isikan datanya kemudian simpan agar data yang diisikan tersimpan secara permanen didatabase.

Kontak & Media Sosial*

Facebook	 Edit
Twitter	
Linkedin	
Instagram	
Nomor HP	

Gambar 5.18 Kontak dan Media Sosial

Informasi Kesehatan*

Golongan Darah	O	 Edit
Tinggi Badan	170 cm	
Berat Badan	70 kg	
Buta Warna	Tidak	
Berkaca Mata	Tidak	
Bertato	Tidak	
Bertindik	Tidak	
Penyandang Difabilitas	Tidak	
Riwayat Penyakit	Tidak ada	

Gambar 5.19 Informasi Kesehatan

Profil riwayat Pendidikan formal dapat diisi lebih dari satu data, mulai jenjang SMK sampai jenjang di atasnya. Untuk menambahkan data, klik tombol Tambah kemudian isikan data pada form yang tersedia, seperti gambar 5.20 berikut.

Riwayat Pendidikan Formal	
Nama Sekolah / Instansi*	SMKN 1 Surabaya
Jenjang Pendidikan*	SMK
Jurusan*	Rekayasa Perangkat Lunak
Tahun Masuk*	1980
Tahun Lulus (Kosongkan jika masih aktif)	1983
Simpan Tutup	

Gambar 5.20 Tambah Riwayat Pendidikan Format

Jika telah diisi selanjutnya klik tombol simpan. Data tersebut kemudian tersimpan didatabase dan ditampilkan pada profil riwayat Pendidikan formal seperti gambar 5.21 berikut.

Riwayat Pendidikan Formal*		
1980 - 1983	SMKN 1 Surabaya SMK Jurusan : Rekayasa Perangkat Lunak	+ Tambah
1983 - 1987	D4 PENS D4 Jurusan : Komputer dan Teknologi Teknik Informatika	

Gambar 5.21 Data Riwayat Pendidikan Formal

Profil keahlian atau Skill digunakan untuk mengelola data keahlian yang dimiliki oleh siswa atau alumni. Untuk menambah profil keahlian, klik tombol Tambah kemudian isikan datanya seperti gambar 5.22.

Klik tombol Simpan sehingga data profil keahlian tersimpan dan ditampilkan Kembali seperti gambar 5.23, dimana tanda bintang menunjukkan level keahliannya.

Keahlian atau Skill

×

Bidang Keahlian*

Menguasai PHP Programming

Level Keahlian*

☐ Cukup Baik
☒ Baik
☐ Sangat Baik

Deskripsikan Keahlianmu

Memiliki skill yang baik pada pemrograman web PHP

Simpan

Tutup

Gambar 5.22 Tambah Keahlian atau Skill

Keahlian atau Skill*

Menguasai PHP Programming

Level : **

Memiliki skill yang baik pada pemrograman web PHP

Mahir HTML dan CSS

Level : ***

Menguasai secara baik HTML dan CSS

Mahir Bahasa Inggris

Level : **

Menguasai Bahasa Inggris baik pasif maupun aktif dengan baik

+ Tambah

Gambar 5.23 Data Keahlian atau Skill

10. Cetak CV

Ketika semua profil telah diisi, maka nilai kelengkapak profil menjadi 100%, artinya telah lengkap dan siap untuk dicetak. Hal ini dibuktikan dengan tombol Cetak CV menjadi aktif.



Gambar 5.24 Progress Kelengkapan Profil

Untuk melakukan cetak CV, klik tombol Cetak CV yang terletak dibagian atas, dekat foto siswa. Gambar 5.25 merupakan hasil cetak CV dari profil siswa, dimana tidak semua informasi ditampilkan.

Arif Kurniawan

Nama:
Arif Kurniawan

Email:
arifkur@pusatkarirsmk.id

Telp/HP:
085712345621

Tanggal Lahir:
22-11-2003

Alamat:
Jl. Merdeka No.1

Jenis Kelamin:
Laki-laki

Riwayat Pendidikan:

- 1980 - 1983
SMK – Rekayasa Perangkat Lunak
SMKN 1 Surabaya
- 1983 - 1987
D4 – Komputer dan Teknologi Teknik Informatika
D4 PENS

Keahlian :

Menguasai PHP Programming	66%
Mahir HTML dan CSS	100%
Mahir Bahasa Inggris	66%

Bursa Kerja Khusus
Pusat Karir SMK

Gambar 5.25 Cetak CV

11. Berkas Pendukung (Portofolio) Siswa

Siswa dan alumni dapat melengkapi profil dengan mengisi portofolio yang disertai dengan bukti pendukungnya. Bukti pendukung dapat berupa file teks, video, audio, dan foto. Gambar 5.26 adalah tampilan dari berkas portofolio siswa dan alumni.

Data Berkas				
 Upload Berkas				
Nama Berkas	Tanggal Upload	File	Ekstensi	Menu
Sertifikat Pelatihan BNSP di LSP1	23-11-2020 04:16:23		pdf	
Juara 1 Lomba Penengembangan Aplikasi Bisnis Dinas Kominfo Jatim tahun 2020	23-11-2020 04:17:34		pdf	

Gambar 5.26 Laman Berkas Portofolio Siswa

Untuk menambah data berkas portofolio, klik tombol Upload berkas kemudian isikan data pada tampilan yang tersedia. Setelah disimpan, maka data berkas portofolio akan tersimpan dan ditampilkan seperti pada gambar 5.26. Untuk membuka file berkas, klik tombol File, sedangkan untuk menghapus dapat menggunakan tombol Hapus.

12. Pantau Lamaran

Siswa atau alumni yang telah *apply* lowongan pekerjaan dapat memantau perkembangan dari lamaran yang telah dilakukan. Informasi yang ditampilkan yaitu nama lowongan, nama IDUKA, dan status lamaran (gambar 5.27).

Lamaranku			
Lowongan	Perusahaan	Status	Menu
Operator Produksi Tanggal Selesai: 31-12-2020 Alamat Penempatan: Jl. Margomulyo Indah C1, Tandes, Surabaya, Kota Surabaya, Jawa Timur	PT. INDOWIRE PRIMA INDUSTRINDO	menunggu	
Welder Tanggal Selesai: 31-12-2020 Alamat Penempatan: Jln. Tanjungsari 44b, Surabaya, Kota Surabaya, Jawa Timur	PT Benteng Anugerah Sejahtera	menunggu	
Analisis Research and Development Tanggal Selesai: 23-12-2020 Alamat Penempatan: Jln. Gatot Subroto No. 68 Buduran Sidoarjo, Sidoarjo, Jawa Timur	PT BERNOFARM	panggilan tes tanggal 22-11-2020	

Gambar 5.27 Laman Pantau Lamaranku

Berdasarkan gambar 5.27, siswa dan alumni dapat mengetahui perkembangan tahapan dari lowongan yang telah dilamar, apakah statusnya masih menunggu atau panggilan tes. Panggilan tes merupakan tahapan dan seleksi yang dilakukan oleh IDUKA dalam proses pencarian tenaga kerja.

13. Panggilan Tes

Siswa dan alumni dapat menampilkan panggilan tes yang telah dilakukan. Informasi yang ditampilkan yaitu nama panggilan tes, keterangan, lowongan, tanggal dan waktu, nama IDUKA dan status dari panggilan tes (gambar 5.28).

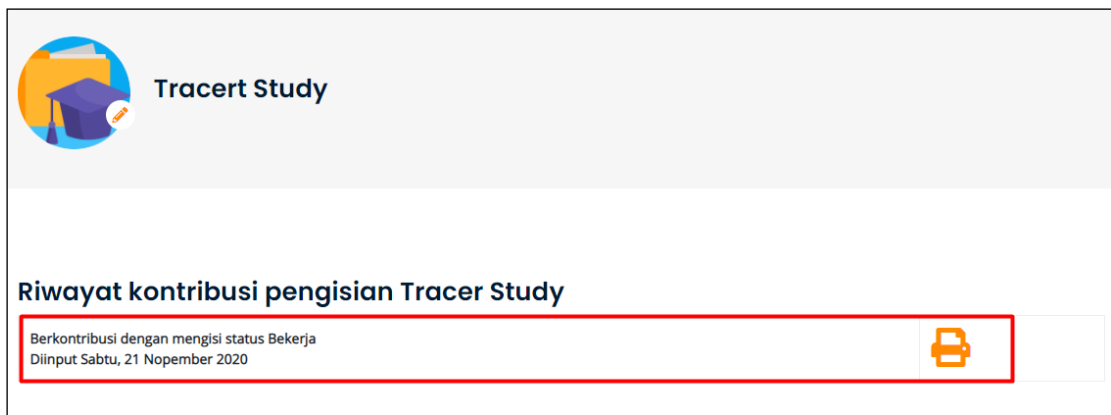
Panggilan Tes						
Panggilan Tes	Keterangan	Lowongan	Tanggal	Waktu	Perusahaan	Status
Tes Kesehatan	Tes kesehatan	Analisis Research and Development	2020-11-22	09:00 - selesai	PT BERNOFARM	diterima
Tes Wawancara	diharapkan kepada semua pelajar untuk hadir pada tes wawancara yang akan dilaksanakan di kantor. Harap datang sesuai dengan waktu yang ditentukan. Terlambat hadir, maka dianggap mengundurkan diri.	Analisis Research and Development	2020-11-22	09:00 - selesai	PT BERNOFARM	diterima
Tes Praktik	Pemanggilan calon pelamar untuk mengikuti tes tahap kedua yakni praktik atau skill	Analisis Research and Development	2020-11-22	10:00 - selesai	PT BERNOFARM	diterima

Gambar 5.28 Laman Panggilan Tes

Melalui laman panggilan tes, siswa dan alumni dapat mengetahui perkembangan *apply* lowonga kerja yang telah dilakukan. Hal ini dapat diketahui karena panggilan tes dapat diketahui status hasil tes. Jika hasil tes diterima, maka calon karyawan dapat mengikuti proses panggilan tes berikutnya. Status diterima sebagai karyawan jika dalam semua panggilan tes dinyatakan diterima.

14. Tracer Study

Setiap SMK melalui BKK secara rutin melaksanakan kegiatan *tracer study* terhadap alumni. Hal ini dilakukan untuk mengetahui kondisi alumni, apakah bekerja, studi lanjut atau berwirausaha. Melalui aplikasi pusat karir SMK, proses tracer study dilakukan secara online dan bukti telah mengisi tracer study langsung dapat dilihat oleh alumni dan sekolah seperti gambar 5.29 berikut.



Gambar 5.29 Bukti telah mengisi Tracer Study

Gambar 5.29 menampilkan riwayat pengisian tracer study yang dilakukan oleh alumni. Terlihat bahwa alumni telah mengisi kegiatan tracer pada hari sabtu, 21 Nopember 2020. Selain itu, siswa dan alumni dapat cetak bukti pengisian tracer study seperti gambar 5.30 berikut.

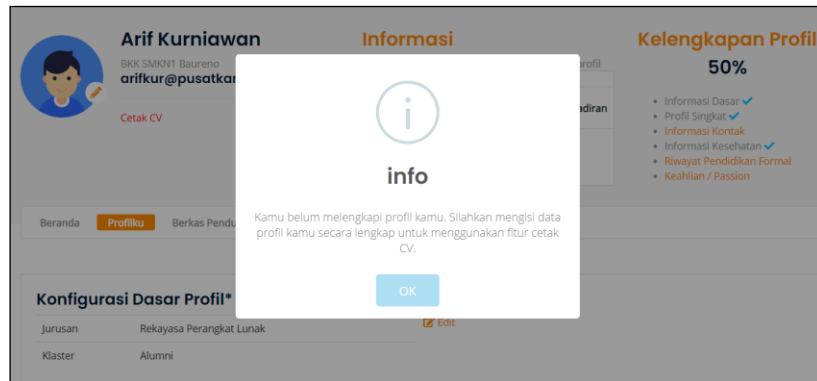
BUKTI PENGISIAN TRACER STUDY	
Pusat Karir SMK	
Nama Lengkap	: Arif Kurniawan
Jenis Kelamin	: Laki-laki
Email	: arifkur@pusatkarirsmk.id
Jurusan	: Rekayasa Perangkat Lunak
Status	: Bekerja
No. Telp/HP	
Tanggal Cetak	: 09-12-2020 15:38:41
ID Pengisian	34a1660f-b34e-4dd5-a465-0a9a7fb8734c
Lulus Tahun	2020
Status Tracer Study	Bekerja
Informasi Mendapat Pekerjaan	Mencari Sendiri
Nama Perusahaan	PT. Maspion Surabaya
Posisi Pekerjaan	Staff Keuangan
Rentang Gaji	2 juta - 3 juta
Sektor Industri	Peralatan Elektronik
Status Kontrak	Karyawan Tetap (PKWTT)
Provinsi	KAB. SIDOARJO
Tanggal Mulai Bekerja	2020-11-21
Pekerjaan Pertama Setelah lulus	Ya
Kebermanfaatan Jurusan	Bermanfaat
Informasi Mendapat Pekerjaan	Mencari Sendiri
Kritik dan Saran	sangat baik

Gambar 5.30 Cetak data isian Tracer Study

Pada gambar xx, dapat diketahui detail isian tracer study yang telah diisi oleh alumni. Terlihat bahwa alumni bekerja di PT. Maspion Surabaya, jabatan staff keuangan dengan gaji 2-3 juta, dan merupakan pekerjaan pertama.

15. Cetak CV

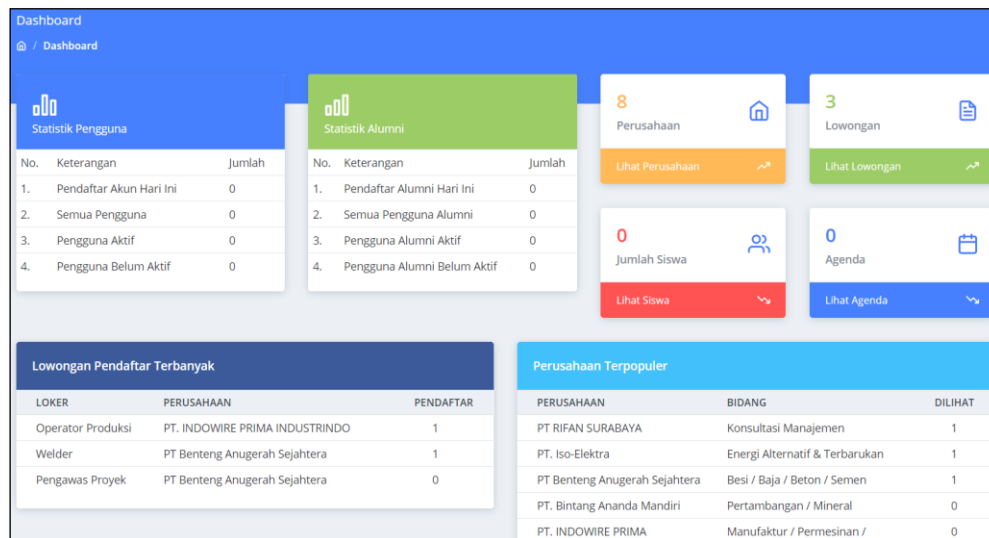
Ketika kelengkapan profil belum 100% dan akan mencetak CV maka ditampilkan pesan bahwa alumni harus melengkapi. Jika belum 100% akan ditampilkan pesan seperti gambar 5.31 berikut.



Gambar 5.31 Cetak CV

16. Dashboard BKK

Dashboard ini ditampilkan ketika pengguna dengan level akses BKK sukses melakukan proses login. Tampilan dashboard BKK tersaji pada gambar 5.32.



Gambar 5.32 Dashboard Login BKK

Berdasarkan gambar xx, banyak informasi penting yang ditampilkan yaitu statistik pengguna, statistik alumni, perusahaan, lowongan, jumlah siswa, agenda, lowongan dengan pendaftar terbanyak, dan perusahaan terpopuler. Semua informasi dapat berubah sesuai dengan proses dan transaksi yang terjadi. Informasi terkait perusahaan, lowongan, jumlah siswa dan agenda merupakan tautan yang dapat diklik sehingga akan dibawa ke laman tujuan.

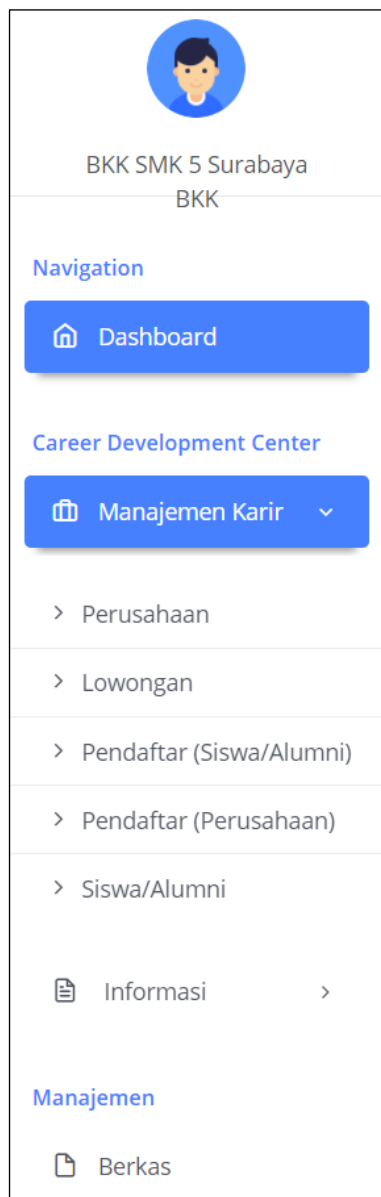
17. Menu utama level akses BKK

Selain *dashboard*, yang ditampilkan setelah pengguna level akses melakukan login adalah menu utama. Tampilan menu utama terlihat pada gambar 5.33 berikut. Menu utama untuk level akses BKK antara lain:

- a. Perusahaan
- b. Lowongan
- c. Pendaftar Siswa/Alumni
- d. Pendaftar Perusahaan
- e. Siswa/Alumni
- f. Berkas
- g. Pengguna
- h. Informasi

18. Data perusahaan

Laman ini digunakan oleh pengguna level BKK untuk pengelolaan data IDUKA. Fitur yang tersedia yaitu CRUD (*Create, Read, Update, Delete*), pencarian, paging, dan laman detail IDUKA. Karena laman ini pada level akses BKK, maka data IDUKA yang ditampilkan adalah yang telah terdaftar di BKK tersebut. Tampilan laman data IDUKA terlihat pada Gambar 5.33.



Penjelasan masing-masing tautan adalah sebagai berikut:

- **Perusahaan**, adalah tautan untuk pengelolaan data IDUKA.
- **Lowongan**, adalah tautan untuk pengelolaan lowongan pekerjaan.
- **Pendaftar (Siswa/Alumni)**, tautan untuk menampilkan data siswa/alumni yang melakukan pendaftaran secara online melalui website.
- **Pendaftar (perusahaan)**, adalah tautan untuk menampilkan data perusahaan yang melakukan pendaftar secara online melalui website.
- **Siswa/Alumni**, adalah tautan untuk pengelolaan data siswa dan alumni di BKK tersebut.
- **Berkas**, adalah tautan untuk pengelolan berkas *softcopy* BKK.
- **Pengguna**, adalah tautan untuk menampilkan data pengguna pada BKK tersebut. Melalui laman pengguna, BKK dapat melakukan edit data, reset password, dan hapus data.
- **Informasi**, adalah tautan untuk pengelolaan informasi tips/artikel dan agenda kegiatan BKK.

Gambar 5.33 Tautan Aplikasi Pusat Karir SMK

+ Tambah						
Show 10 entries			Search:			
LOGO	NAMA PERUSAHAAN	KATEGORI	BUMN/SWASTA	STATUS	MENU	
	PT WILLOW MANDIRI INTERNUSA	Perdagangan / Komoditas / Distribusi	Swasta	Aktif		
	PT RIFAN SURABAYA	Konsultasi Manajemen	Swasta	Aktif		
	PT. Iso-Elektra	Energi Alternatif & Terbarukan	Swasta	Aktif		
	PT. HARGO PRATAMA INDONESIA	Perdagangan / Komoditas / Distribusi	Swasta	Aktif		
	PT Surya Multi Indopack	Percetakan / Pengemasan	Swasta	Aktif		
	PT. Bintang Ananda Mandiri	Pertambangan / Mineral	Swasta	Aktif		
	PT. INDOWIRE PRIMA INDUSTRINDO	Manufaktur / Permesinan / Otomatisasi	Swasta	Aktif		

Gambar 5.34 Laman Kelola Perusahaan

Berdasarkan gambar 5.34, tombol tambah data digunakan untuk menambah data IDUKA baru, tombol dengan ikon search (warna biru) digunakan untuk menampilkan detail IDUKA, tombol dengan ikon pencil digunakan untuk melakukan edit data, dan tombol dengan ikon trash (warna merah) digunakan untuk menghapus data. Ketika menghapus data, sebelum data benar terhapus akan ditampilkan konfirmasi penghapusan. BKK juga dapat menentukan aktif atau tidaknya BKK berdasarkan kerjasama yang telah terjalin.

19. Detail data perusahaan

Laman ini ditampilkan ketika pengguna level akses BKK klik tombol detail. Laman ini menampilkan detail informasi BKK. tampilan detail laman IDUKA terlihat pada gambar 5.35 berikut.

Logo



Nama Perusahaan

PT WILLOW MANDIRI INTERNUSA

Alamat

Jl. Darmahusada Permai Blok N No.234

Email

internusa@gmail.com

Bidang Perusahaan

Perdagangan / Komoditas / Distribusi

Telp

031-5925777

Kota

KOTA SURABAYA

Profil

Willow Baby Shop merupakan Toko Perlengkapan Bayi dan Ibu. Toko kami juga menyediakan ribuan produk pilihan dengan 400+ brand favorit untuk menjadi bagian dari masa kehamilan Ibu, menyambut kedatangan sang buah hati, masa pertumbuhan bayi dan anak dan segala yang menggugurkan di dalamnya. Kami juga memiliki 50 - 100 karyawan yang handal dan lingkungan kerja yang nyaman

Jenis

Swasta

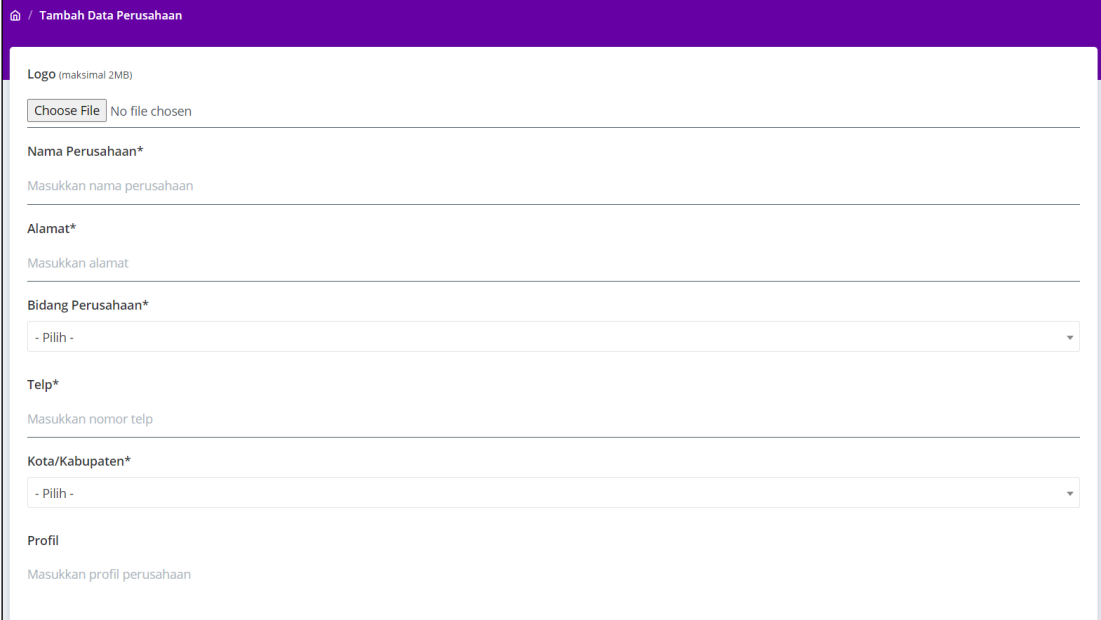
Status

Aktif

Gambar 5.35 Detail Data Perusahaan

20. Tambah Data Perusahaan

Laman ini digunakan oleh pengguna dengan level akses BKK untuk menambah data IDUKA. gambar 5.36 berikut adalah tampilan tambah data IDUKA.



The screenshot shows a web form titled "Tambah Data Perusahaan". The form is white with a purple header bar. It contains the following fields:

- Logo (maksimal 2MB)**: A file upload field with a "Choose File" button and the text "No file chosen".
- Nama Perusahaan***: A text input field with the placeholder "Masukkan nama perusahaan".
- Alamat***: A text input field with the placeholder "Masukkan alamat".
- Bidang Perusahaan***: A dropdown menu with the placeholder "- Pilih -".
- Telp***: A text input field with the placeholder "Masukkan nomor telp".
- Kota/Kabupaten***: A dropdown menu with the placeholder "- Pilih -".
- Profil**: A text input field with the placeholder "Masukkan profil perusahaan".

Gambar 5.36 Tambah Data Perusahaan

Berdasarkan gambar 5.36, data IDUK yang harus diisi yaitu:

- Logo perusahaan, digunakan untuk tampilan logo perusahaan
- Nama perusahaan
- Alamat
- Bidang perusahaan, pilihan bidang usaha dari perusahaan
- Telp
- Kota/Kabupaten, lokasi perusahaan.
- Profil
- Jenis, Swasta atau BUMN
- Status, Aktif atau Tidak Aktif
- Tunjangan yang ada, diisi dengan data tunjangan untuk karyawan yang disediakan oleh perusahaan.
- Gaya berpakaian, pilihan gaya berpakaian
- Jumlah karyawan, pilihan range jumlah karyawan pada perusahaan.
- Waktu bekerja, pilihan waktu bekerja.
- Email

- o. Website
- p. Penanggungjawab perusahaan
- q. Kontak/Telpon penanggungjawab perusahaan
- r. Metode mendaftar, pilihan perusahaan terdaftar dalam aplikasi pusat karir SMK apakah mendaftar sendiri atau didaftarkan oleh BKK.

Isian yang bertanda “*”, artinya harus diisi, tidak boleh dikosongi. Agar isian tersimpan pada database, maka klik tombol Simpan. Untuk membatalkan penambahan data, klik tombol Batal. Data IDUKA selanjutnya digunakan dalam proses lowongan kerja dan rekrutmen tenaga kerja.

21. Edit Data Perusahaan

Ketika terdapat kesalahan pada data IDUKA yang telah ditambahkan, BKK dapat ubah data. Untuk melakukan ubah data, klik tombol Edit pada laman data IDUKA sehingga ditampilkan laman seperti gambar 5.37 berikut.

Logo Perusahaan	
	
Hapus Logo	
Nama Perusahaan*	
PT WILLOW MANDIRI INTERNUSA	
Alamat*	
Jl. Dharmasada Permai Blok N No.234	
Bidang Perusahaan*	
Perdagangan / Komoditas / Distribusi	
Telp*	
031-5925777	
Kota*	
KOTA SURABAYA	
Profil	
Willow Baby Shop merupakan Toko Perlengkapan Bayi dan Ibu. Toko kami juga menyediakan ribuan produk pilihan dengan 400+ brand favorit untuk menjadi bagian dari masa kehamilan ibu, menyambut kedatangan sang buah hati, masa pertumbuhan bayi dan anak dan segala yang mengagumkan di dalamnya. Kami juga memiliki 50 - 100 karyawan yang handal dan lingkungan kerja yang nyaman	

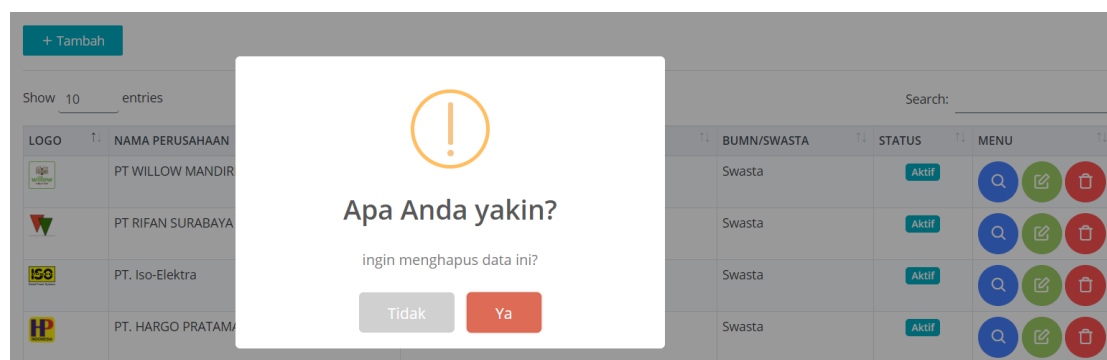
Gambar 5.37 Edit Data Perusahaan

Pada laman ubah data, pengguna dapat mengubah data yang salah dengan mengganti data tersebut dengan data yang benar. Tidak perlu semua data diubah, hanya data yang salah saja. Agar perubahan data tersimpan secara permanen,

jangan lupa klik tombol Simpan. Untuk membatalkan perubahan data, klik tombol Batal.

22. Hapus data Perusahaan

Fitur hapus data sebaiknya menjadi pilihan terakhir dalam pengelolaan data IDUKA. hapus data dilakukan ketika terjadi data ganda dan lama yang telah kedaluwarsa. Kalau tujuannya hanya untuk menyembunyikan data, terdapat fitur untuk pengaturan aktif dan tidak aktif.

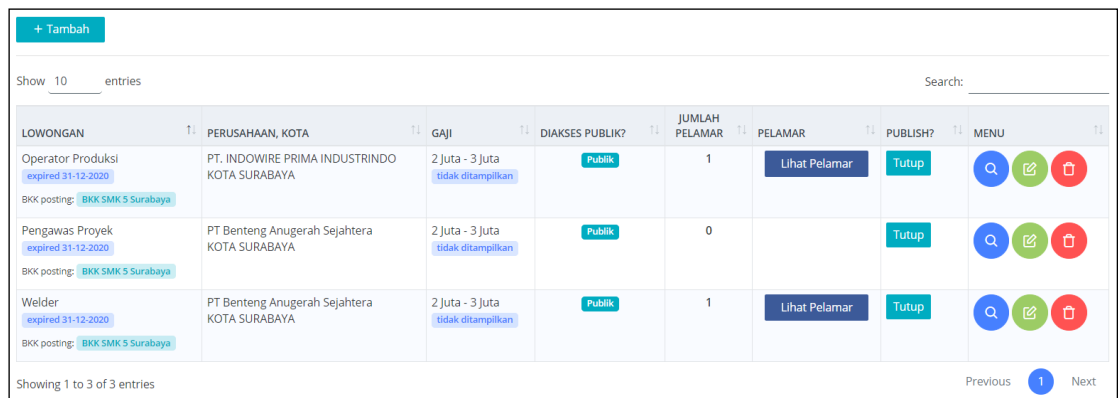


Gambar 5.38 Hapus Data Perusahaan

Karena hapus data memiliki resiko, maka sebelum data benar dihapus akan ditampilkan konfirmasi. Dengan konfirmasi hapus, pengguna diingatkan kembali agar memikirkan kembali proses yang dilakukan. Jika benar yakin ingin menghapus data, pengguna dapat klik tombol Yes sehingga data akan terhapus dari database.

23. Data Lowongan Kerja

Laman ini menampilkan data lowongan pekerjaan yang tersedia pada BKK. lowongan kerja ini hanya tampil pada BKK, siswa dan alumni di BKK tersebut. Tampilan data lowongan kerja terlihat pada gambar 5.39 berikut.



LOWONGAN	PERUSAHAAN, KOTA	GAJI	DIAKSES PUBLIK?	JUMLAH PELAMAR	PELAMAR	PUBLISH?	MENU
Operator Produksi expired 31-12-2020 BKK posting: BKK SMK 5 Surabaya	PT. INDOWIRE PRIMA INDUSTRINDO KOTA SURABAYA	2 Juta - 3 Juta tidak ditampilkan	Publik	1	Lihat Pelamar	Tutup	Search Edit Delete
Pengawas Proyek expired 31-12-2020 BKK posting: BKK SMK 5 Surabaya	PT Benteng Anugerah Sejahtera KOTA SURABAYA	2 Juta - 3 Juta tidak ditampilkan	Publik	0		Tutup	Search Edit Delete
Welder expired 31-12-2020 BKK posting: BKK SMK 5 Surabaya	PT Benteng Anugerah Sejahtera KOTA SURABAYA	2 Juta - 3 Juta tidak ditampilkan	Publik	1	Lihat Pelamar	Tutup	Search Edit Delete

Showing 1 to 3 of 3 entries

Previous **1** Next

Gambar 5.39 Kelola Data Lowongan Pekerjaan

Fitur yang tersedia yaitu CRUD (Create, Read, Update, Delete), pencarian, paging, lihat pelamar, set aktif dan tidak aktif lowongan pekerjaan. informasi yang ditampilkan antara lain:

- Nama lowongan kerja, ditampilkan juga waktu expire lowongan dan BKK yang posting lowongan kerja.
- IDUKA yang membuka lowongan kerja
- Gaji yang ditawarkan, disertai dengan status apakah ditampilkan pada lowongan kerja atau tidak. Kalau tertulis tidak ditampilkan, maka gaji tidak ditampilkan.
- Status dapat diakses publik atau tidak
- Jumlah yang telah melamar
- Manampilkan data pelamar
- Status lowongan, apakah masih buka atau sudah ditutup

24. Tambah Lowongan Kerja

Selain IDUKA dan pusat karir, BKK dapat melakukan pengelolaan lowongan pekerjaan dengan akses terbatas di BKK tersebut. Untuk menambah data lowongan pekerjaan, klik tombol Tambah pada laman data lowongan pekerjaan sehingga tampil seperti gambar 5.40 berikut.

INFORMASI LOWONGAN	PERSYARATAN
Diakses Publik?* - Pilih - Poster Choose File No file chosen Tanggal Mulai* 12/09/2020 Batas Pendaftaran (Expired)* 12/09/2020 Nama Lowongan* Masukkan nama lowongan Perusahaan* PT Benteng Anugerah Sejahtera Lokasi Kota* - Pilih - Gaji* - Pilih - Gaji Ditampilkan?*	Jurusan* Level Pekerjaan* - Pilih - Jenis Kelamin* - Pilih - Tinggi Minimal Syarat - Laki-laki (cm) Tinggi Minimal Syarat - Perempuan (cm) Buta?* - Pilih - Berkacamata?* Tidak Batas Usia Minimal (tahun) Alamat Penempatan*

Gambar 5.40 Tambah Lowongan Pekerjaan

Data yang diisikan antara lain:

- Status akses, publik atau local BKK
- Poster
- Tanggal mulai
- Batas pendaftaran (*expired*)
- Nama lowongan
- Perusahaan
- Lokasi kerja
- Gaji
- Status gaji ditampilkan atau tidak
- Jenis waktu, pilihan jenis waktu
- Status, aktif atau tidak aktif
- Jurusan
- Level pekerjaan, pilihan level
- Jenis kelamin
- Tinggi minimal laki-laki
- Tinggi minimal perempuan
- Opsi butawarna
- Opsi berkacamata
- Batas usia minimal

- t. Alamat penempatan
- u. Tipe lowongan
- v. Opsi perlu nilai akademik atau tidak
- w. Opsi membutuhkan kelengkapan scan dokumen atau tidak.

Isian yang bertanda bintang (*) artinya wajib diisi, tidak boleh dibiarkan kosong. Klik tombol Simpan untuk menyimpan data penambahan lowongan kerja, dan Batal untuk membatalkannya. Data lowongan pekerjaan yang telah tersimpan, otomatis ditampilkan pada laman lowongan kerja, dimana siswa dan alumni dapat melakukan *apply* pada lowongan kerja tersebut asalkan memenuhi persyaratan.

25. Detail Lowongan Kerja

Pengguna level BKK dapat menampilkan detail lowonga pekerjaan dengan klik tombol Detail sehingga ditampilkan seperti gambar 5.41 berikut.

INFORMASI LOWONGAN	PERSYARATAN
Poster	Jurusan
Tanggal Mulai	Teknik Audio Video, Teknik Elektronika Industri, Teknik Instalasi Tenaga Listrik, Teknik Otomasi Industri, Teknik Otomasi Manufaktur dan Mekatronika, Teknik Tenaga Listrik,
11/22/2020	
Batas Pendaftaran (Expired)	Level Pekerjaan
11/22/2020	Canaker baru saja lulus
Nama Lowongan	Jenis Kelamin
Operator Produksi	Laki-laki
Perusahaan	Tinggi Minimal Syarat - Laki-laki (cm)
PT. INDOWIRE PRIMA INDUSTRIINDO	170
Lokasi Kota	Tinggi Minimal Syarat - Perempuan (cm)
KOTA SURABAYA	0
Gaji	Buta?
2 Juta - 3 Juta	Tidak
Jenis Waktu	Berkacamata?
Penuh Waktu	Tidak
Status	Batas Usia Minimal (tahun)
Aktif	20
	Alamat Penempatan
	Jl. Margomulyo Indah C1, Tandes, Surabaya, Kota Surabaya, Jawa Timur
	Tipe Lowongan
	Umum
	Membutuhkan Data Nilai
	Ya

Gambar 5.41 Detail Lowongan Pekerjaan

Pada laman ini, data lowongan pekerjaan ditampilkan secara lengkap. Hal ini dapat digunakan oleh pengguna untuk mendapatkan informasi, apakah data lowongan telah benar atau masih perlu dilakukan perubahan. Selain itu, tampilan lowongan kerja secara detail ini digunakan untuk menampilkan data deskripsi pekerjaan, syarat/kualifikasi, dan keahlian.

26. Deskripsi, Syarat dan Keahlian

Data lowongan pekerjaan yang telah ditambahkan masih belum lengkap sebelum data deskripsi pekerjaan, syarat/kualifikasi, dan keahlian ditambahkan.

DESKRIPSI PEKERJAAN	SYARAT / KUALIFIKASI	KEAHLIAN
Tidak ada data	1. Laki-laki 2. Pendidikan SMK Teknik Listrik, Teknik Otomasi Industri / Relevan 3. Mampu menggunakan alat ukur 4. Berintegritas, disiplin, dan profesional 5. Tidak buta warna / tidak berkaca mata 6. Jujur dan bertanggung jawab tinggi	Tidak ada data
Kembali		

Gambar 5.42 Kelola Deskripsi, Syarat dan Keahlian

Laman ini menampilkan deskripsi pekerjaan, syarat/kualifikasi dan keahlian yang akan tampil pada lowongan pekerjaan. dengan melihat data ini, pelamar menjadi mengerti terkait pekerjaan yang akan dikerjakan, syarat untuk melakukan *apply* lowongan, dan keahlian apa yang harus dikuasai. Untuk menambah dan mengubah data deskripsi pekerjaan, syarat/kualifikasi, dan keahlian dapat dilakukan melalui fitur edit data.

27. Edit Lowongan Kerja

Laman ini digunakan untuk melakukan perubahan data atau edit data lowongan pekerjaan. proses ubah data dilakukan karena terdapat data yang salah dan perlu diperbaiki, atau ada data yang kurang sehingga perlu ditambah. Pada laman edit, data yang telah ada akan ditampilkan pada *form* sehingga untuk melakukan perubahan data cukup dilakukan dengan mengganti data yang salah. Tampilan edit data lowongan pekerjaan terlihat pada gambar 5.43 berikut.

Pada laman ini, pengguna dapat menambahkan kelengkapan lowongan pekerjaan yaitu deskripsi pekerjaan, syarat/kualifikasi dan keahlian. Informasi yang ditambahkan ini akan melengkapi lowongan pekerjaan sehingga memiliki spesifikasi, syarat pelamar dan syarat keahlian.

Gambar 5.43 Edit Lowongan Kerja

28. Tambah Deskripsi, Syarat dan Keahlian

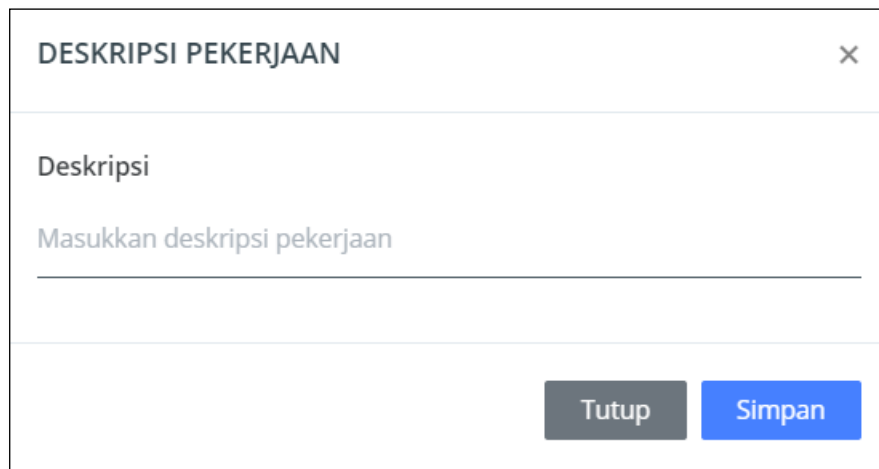
Gambar 5.44 terdapat pada laman edit lowongan pekerjaan, digunakan untuk menambah data deskripsi pekerjaan, syarat/kualifikasi, dan keahlian.

Gambar 5.44 Tambah Deskripsi, Syarat dan Keahlian

Untuk menambah, klik tombol Tambah kemudian isikan informasi pada tampilan yang muncul kemudian klik tombol Simpan. Informasi yang telah tersimpan akan ditampilkan seperti pada gambar 5.42.

29. Tambah Deskripsi

Tampilan ini muncul ketika pengguna klik tombol Tambah Deskripsi pada laman edit data pekerjaan. Isikan deskripsi pekerjaan kemudian klik tombol Simpan.

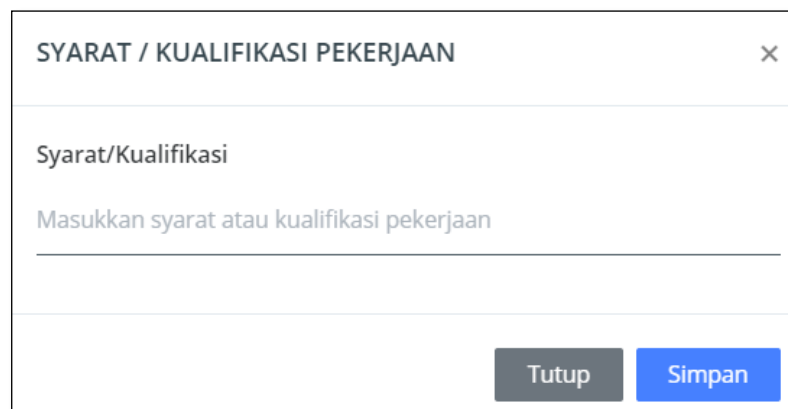


Gambar 5.45 Tambah Deskripsi Pekerjaan

Jika terdapat 10 deskripsi pekerjaan, maka pengisian deskripsi pekerjaan juga dilakukan sebanyak 10 kali. Tidak ada fitur ubah terhadap data deskripsi yang telah tersimpan. Jika ada data yang salah, maka dihapus dan ditambahkan lagi.

30. Tambah Syarat

Gambar 5.46 berikut digunakan untuk menambah data syarat/kualifikasi pekerjaan pada lowongan pekerjaan. Cara kerjanya mirip dengan tambah deskripsi pekerjaan, yang menjadi pembeda terletak pada data yang ditambahkan.



Gambar 5.46 Tambah Syarat/Kualifikasi Pekerjaan

31. Tambah Keahlian

Gambar 5.47 digunakan untuk menambah data keahlian pada lowongan pekerjaan. Cara kerjanya mirip dengan tambah deskripsi pekerjaan, yang menjadi pembeda terletak pada data yang ditambahkan.

KEAHLIAN

Keahlian

Masukkan keahlian

Tutup

Simpan

Gambar 5.47 Tambah Keahlian

32. Data Pelamar

Laman ini menampilkan data pelamar pada lowongan kerja tertentu. Pengguna yang dapat melihat data pelamar adalah IDUKA, BKK dan Pusat Karir. Laman data pelamar ditampilkan melalui data lowongan pekerjaan, kemudian klik tombol Lihat Pelamar.

Kembali

Tambah Panggilan Tes

INFORMASI PELAMAR

Show 10 entries

Search:

TANGGAL PENGAJUAN	NIK	NAMA	JURUSAN	HP	DETAIL PELAMAR
22-11-2020 14:58:03	3271046504930002	Arif Kurniawan	Rekayasa Perangkat Lunak		

Showing 1 to 1 of 1 entries

Previous

1

Next

Panggilan Tes Lowongan

NO.	NAMA TES	LOWONGAN	PERUSAHAAN, KOTA	TANGGAL	WAKTU	DETAIL
1.	Tes Kesehatan	Analisis Research and Development	PT BERNOFARM	22-11-2020	09:00 s/d selesai	
2.	Tes Praktik	Analisis Research and Development	PT BERNOFARM	22-11-2020	10:00 s/d selesai	
3.	Tes Wawancara	Analisis Research and Development	PT BERNOFARM	22-11-2020	09:00 s/d selesai	

Gambar 5.48 Data Pelamar

Data pelamar yang ditampilkan terdiri dari:

- Tanggal pangajuan lamaran kerja
- NIK
- Nama pelamar
- Jurusan

e. No HP

Terdapat tombol detail pelamar yang digunakan untuk menampilkan informasi detail portofolio pelamar. Dengan mengetahui detail portofolio pelamar maka IDUKA dapat referensi dalam penentuan calon pelamar yang diterima.

33. Panggilan Tes

Laman ini digunakan oleh IDUKA dalam melakukan pengelolaan panggilan tes dan seleksi pelamar. Dalam satu lowongan kerja, IDUKA dapat membuat beberapa kali panggilan tes, misalkan panggilan tes wawancara, panggilan tes Kesehatan, panggilan tes kemampuan akademik, dan lain-lain. Tampilan panggilan tes terlihat seperti gambar 5.49 berikut.

No.	Nama	No. HP	Dipanggil?	Hapus
1	Arif Kurniawan		<button>Ya</button>	

Gambar 5.49 Tambah Panggilan Tes

Pada setiap panggilan tes, terdapat daftar hadir yang harus diisi oleh pelamar. Jika tidak hadir, maka dinyatakan gugur. Selain itu, panggilan tes juga digunakan oleh IDUKA dalam seleksi dimana dalam tiap tahapan ditentukan siapa yang lolos dan gagal. Pelamar yang diterima adalah yang hadir dalam semua panggilan tes dan dinyatakan lolos.

34. Seleksi pada Panggilan Tes

Laman ini digunakan sebagai seleksi dalam panggilan tes. Melalui laman ini, pengguna dapat menentukan pelamar diterima atau tidak. Pelamar yang dinyatakan diterima dapat mengikuti panggilan tes dan seleksi tahap berikutnya.

Show 10 entries

Search:

NAMA TES	LOWONGAN	PERUSAHAAN, KOTA	TANGGAL	WAKTU	DETAIL	SELEKSI	HAPUS
Tes Kesehatan	Analisis Research and Development	PT BERNOFARM	22-11-2020	09:00 s/d selesai			
Tes Praktik	Analisis Research and Development	PT BERNOFARM	22-11-2020	10:00 s/d selesai			
Tes Wawancara	Analisis Research and Development	PT BERNOFARM	22-11-2020	09:00 s/d selesai			

Showing 1 to 3 of 3 entries

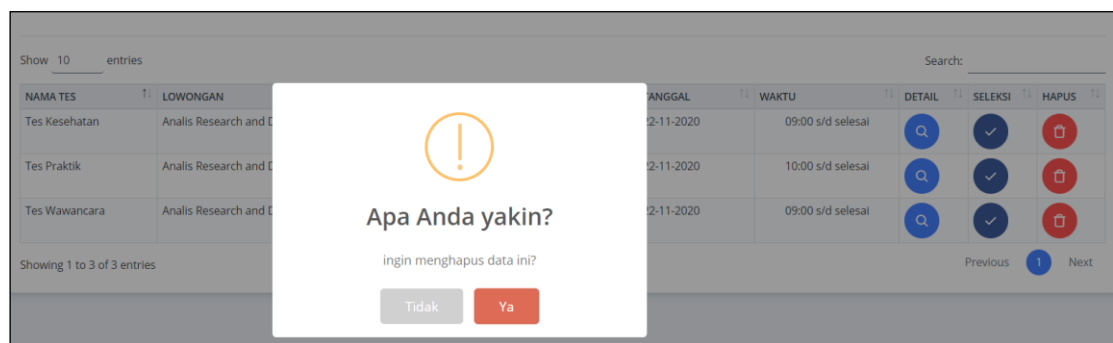
Previous 1 Next

Gambar 5.50 Seleksi pada Panggilan Tes

Informasi detail panggilan tes dapat ditampilkan dengan cara klik tombol pada kolom detail, sedangkan penentuan diterima atau tidak dalam panggilan tes dilakukan dengan klik tombol pada kolom seleksi. Pada seleksi, pengguna dapat menentukan pelamar yang diterima atau tidak. Untuk menghapus data panggilan tes, klik tombol pada kolom hapus sehingga data panggilan tes beserta data terkait ikut terhapus.

35. Hapus Panggilan Tes

Ketika tombol hapus panggilan tes dihapus, maka ditampilkan konfirmasi hapus seperti gambar 5.51 berikut.

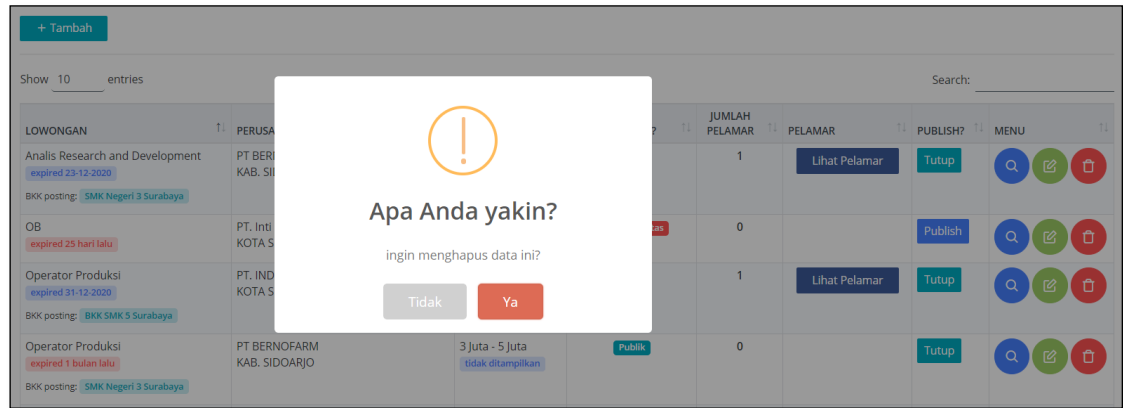


Gambar 5.51 Hapus Panggilan Tes

Karena hapus data sebaiknya dihindari, maka setiap hapus data selalui ditampilkan konfirmasi. Tujuan konfirmasi hapus adalah untuk memastikan pengguna dengan sadar akan menghapus data, bukan karena ketidaksengajaan.

36. Hapus Lowongan Kerja

Menghapus data lowongan pekerjaan sebaiknya sangat dihindari kecuali data tersebut adalah data ujicoba. Cara lain untuk menghindari hapus data adalah melalui mekanisme edit data jika data perlu diperbaiki atau membuat non-aktif.

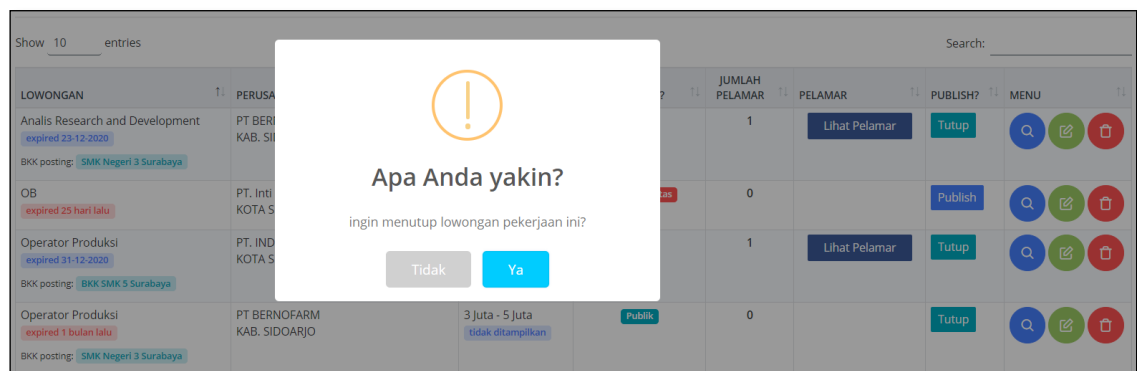


Gambar 5.52 Hapus Lowongan Kerja

Fitur hapus data tidak dapat dihilangkan dalam suatu aplikasi karena masih dibutuhkan. Pengguna harus bijak dalam menggunakan fitur hapus agar tidak mudah menggunakan penghapusan data. kenapa tombol hapus data selalui berwarna merah, karena hapus data merupakan kegiatan yang berbahaya dan sebaiknya dihindari.

37. Menutup atau Publish Lowongan Kerja

Selain dihapus, lowongan pekerjaan dapat ditentukan status *publish* atau tutup. Status publish artinya lowongan pekerjaan dapat dibuka oleh pelamar, sedangkan status tutup artinya lowongan pekerjaan tidak dapat dibuka.



Gambar 5.53 Publish Lowongan Kerja

Pengguna yang dapat melakukan pengaturan ini adalah BKK dan IDUKA. Tutup dan publish lowongan pekerjaan diperlukan ketika jumlah pelamar terlalu banyak atau lowongan pekerjaan tersebut tidak jadi dibuka.

38. Data Pendaftar

Laman ini menampilkan data siswa dan alumni yang melakukan pendaftaran secara mandiri melalui aplikasi. Data yang ditampilkan yaitu tanggal pendaftaran, nama siswa/alumni, email, klaster, dan status pendaftaran. klaster terdiri dua pilihan yakni siswa atau alumni.

Data pendaftaran siswa yang dilakukan secara offline.

Show 10 entries

Search: _____

TANGGAL PENDAFTARAN	NAMA	EMAIL	KLASTER	STATUS PENDAFTARAN	MENU
07-11-2020 09:57:40	Ali Imron	ali_imron@gmail.com	SIWA	daftar online	
07-11-2020 10:09:20	Budi Fatonil	budifatonil@gmail.com	ALUMNI	daftar online	
07-11-2020 13:58:19	Sugeng Raharjo	genksue@gmail.com	SIWA	daftar online	
07-11-2020 13:58:22	ADI MARTA	bksmkn3bojonegoro@gmail.com	SIWA	daftar online	
07-11-2020 13:58:35	Mohamad Yudianto	yud21@gmail.com	SIWA	daftar online	
07-11-2020 13:58:50	Ahmad Albar	ahmad@gmail.com	SIWA	daftar online	
07-11-2020 14:00:12	Arif Yudiono	ardionoo2013@gmail.com	SIWA	daftar online	
07-11-2020 14:03:41	Betty Retnosari	bettyretnosari@gmail.com	SIWA	daftar online	
21-11-2020 19:49:06	Arif Kurniawan	arifkur@pusatkarirsmk.id	ALUMNI	daftar online	

Showing 1 to 9 of 9 entries

Previous 1 Next

Gambar 5.54 Data Pendaftar

39. Pendaftar Perusahaan

Laman ini menampilkan data perusahaan atau IDUKA yang melakukan pendaftaran secara mandiri melalui aplikasi. Data yang ditampilkan yaitu tanggal pendaftaran, nama IDUKA, email, status, akun login, dan validasi.

Data Pendaftar Perusahaan

Data pendaftaran IDUKA yang dilakukan secara offline.

Show 10 entries

Search: _____

TANGGAL PENDAFTARAN	NAMA	EMAIL	STATUS	AKUN LOGIN	VALIDASI	MENU
01-11-2020 15:28:49	PT. Inti Perkasa Indonesia	intiperkasaindo@gmail.com	Sudah di Acc	✓		
07-11-2020 14:39:22	Nyak nyik Nyuk	nyaknyiknyuk@gmail	Belum di Acc	✓	✓	
07-11-2020 14:40:08	Cekot Cekot	cekot2@gmail.com	Belum di Acc	✓	✓	

Showing 1 to 3 of 3 entries

Previous 1 Next

Gambar 5.55 Pendaftar Perusahaan

Setiap perusahaan atau IDUKA yang melakukan registrasi harus divalidasi terlebih dahulu oleh pusat karir dan BKK. Validasi bertujuan untuk memastikan bahwa yang melakukan registrasi adalah benar IDUKA bukan orang iseng. Proses validasi dilakukan dengan klik tombol pada kolom validasi, kemudian pada konfirmasi yang tampil klik tombol Ya. IDUKA yang telah tervalidasi selanjutnya dapat melakukan proses login dengan menggunakan email dan password yang telah diisikan. Semua password yang digunakan dalam aplikasi pusat karir ini tersimpan dalam database dalam bentuk enkripsi sehingga tidak ada orang lain yang mengetahui. Ketika terjadi lupa password, yang dapat dilakukan adalah melakukan reset password, bukan menampilkan password yang telah tersimpan.

40. Data Siswa

Laman ini digunakan untuk pengelolaan data siswa. Data siswa menjadi salah satu data penting dalam aplikasi pusat karir. Pengelolaan data siswa meliputi:

- CRUD data siswa, digunakan untuk melakukan tambah, ubah, dan hapus data siswa.
- Import Data, digunakan untuk menambahkan data siswa dalam jumlah banyak yang telah tersimpan dalam file Excel sesuai format yang ditentukan.
- Download Template Excel, digunakan untuk download template excel.
- Tambah Portofolio Siswa, digunakan untuk pengelolaan data berkas portofolio siswa.
- Tambah nilai akademik, digunakan untuk pengelolaan data portofolio akademik siswa dalam bentuk nilai pelajaran pada mata kuliah tertentu.
- Tracer study, digunakan untuk proses tracer study.

NAMA	JURUSAN / SMK	HP/EMAIL	PORTOFOLIO	NILAI AKADEMIK	TAMBAH PENGGUNA	TRACER STUDY	MENU
ADI MARTA	Administrasi Keuangan BKK SMKN 3 Bojonegoro	laksimkn3bojonegoro@gmail.com			Akun sudah ada		
Ahmad Albar	Rekayasa Perangkat Lunak BKK SMK Purwosari	ahmad@gmail.com			Akun sudah ada	Tidak ada data tracer	
Ali Imron	Rekayasa Perangkat Lunak BKK SMK Purwosari	ali_imron@gmail.com			Akun sudah ada	Tidak ada data tracer	
Arif Kurniawan	Rekayasa Perangkat Lunak BKK SMKN1 Baureno	arifkur@pusatkarirsmk.id			Akun sudah ada		
Arif Yudianto	Accounting Information BKK SMKN1 Baureno	ardionos2013@gmail.com			Akun sudah ada	Tidak ada data tracer	
Betty Retnosari	Jurusan belum dipilih BKK SMKN 2 Surabaya	bettyretnosari@gmail.com			Akun sudah ada	Tidak ada data tracer	
Budi Fatoni	Jurusan belum dipilih BKK SMK Purwosari	budifatoni@gmail.com			Akun sudah ada		

Gambar 5.56 Data Siswa

Pengguna yang dapat mengelola data siswa adalah level akses BKK dan Admin. Karena data siswa merupakan salah satu data penting, maka data yang tersimpan harus dijamin kebenaran dan kelengkapannya. Jika sekolah telah memiliki data siswa secara lengkap dalam format Excel, maka cara terbaik untuk memasukkan data siswa adalah melalui fasilitas import. Sekolah juga dapat menentukan waktu khusus dan diawasi agar siswa dapat secara bersama-sama melakukan registrasi data siswa secara mandiri. Data siswa yang telah tersimpan di database masih dapat diubah oleh siswa sendiri dan BKK.

41. Tambah data siswa

Laman ini digunakan untuk menambah data siswa yang dilakukan oleh BKK. Data yang diisikan antara lain:

- a. BKK, dipilih BKK dimana siswa tersebut sekolah
- b. Status personal, pilihan siswa atau alumni
- c. Nama lengkap
- d. NIK (Nomor Induk Kependudukan)
- e. Jurusan, pilihan jurusan siswa
- f. Email
- g. Password, digunakan untuk login
- h. Jenis kelamin
- i. Tempat lahir
- j. Tanggal lahir
- k. Agama
- l. Domisili
- m. Alamat sesuai KTP
- n. Alamat tinggal
- o. No Telpn
- p. HP
- q. Golongan darah
- r. Hobi

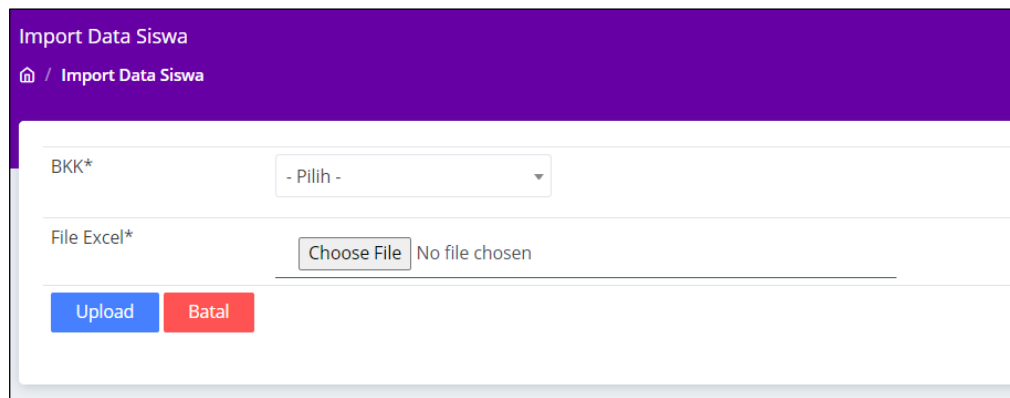
Semua isian sebaiknya diisi dengan benar dan lengkap agar data siswa menjadi baik dan valid.

Gambar 5.57 Tambah Data Siswa

Agar data siswa yang telah diisikan tersimpan permanen didatabase, klik tombol Simpan. Untuk membatalkan, klik tombol Batal. Data siswa yang telah tersimpan selanjutnya banyak digunakan dalam proses dan transaksi yang lain yaitu apply lowongan kerja, portofolio siswa, CV, tracer study dan rekrutmen tenaga kerja. Data siswa yang kurang lengkap akan berpengaruh pada siswa itu sendiri. Bagi BKK, data siswa yang lengkap akan sangat membantu dalam memberikan referensi ketika ada IDUKA melakukan permintaan tenaga kerja. Bagi IDUKA, data siswa yang lengkap akan menjadi pertimbangan tersendiri dalam penentuan diterima tidaknya pelamar.

42. Import Data Siswa

Fitur import data digunakan untuk menambahkan data siswa secara masal berdasarkan data yang tersimpan dalam format Excel. Fitur ini sangat cocok digunakan oleh BKK yang umumnya telah memiliki data siswa, dan tidak ingin memasukkan data siswa satu persatu.



Gambar 5.58 Import Data Siswa

Sebelum melakukan *import* data siswa, harus dipastikan format data dalam file Excel telah sesuai dengan template yang tersedia. Jika belum, harus disesuaikan terlebih dahulu. Untuk melakukan proses *import*, pilih BKK kemudian tentukan file data siswa yang telah tersimpan di Excel, kemudian klik tombol Upload. Data siswa yang telah di-*import* selanjutnya dapat dilihat pada laman data siswa.

43. Lihat Portofolio Siswa

Laman ini digunakan untuk menampilkan data portofolio yang dimiliki siswa. Data ini diisi oleh masing-masing siswa. Semakin banyak dan lengkap data portofolio siswa maka data yang ditampilkan juga semakin banyak.

Unggah Berkas

Kembali

Nama

Arif Kurniawan

Jurusan

Rekayasa Perangkat Lunak

Tanggal Unggah	Nama Berkas	Instansi	Tahun	Tempat	File	Hapus
23-11-2020 04:16:23	Sertifikat Pelatihan BNSP di LSP1				Lihat Berkas	
23-11-2020 04:17:34	Juara 1 Lomba Penengembangan Aplikasi Bisnis Dinas Kominfo Jatim tahun 2020				Lihat Berkas	

Gambar 5.59 Data Portofolio Siswa

Selain siswa, pengguna yang dapat melihat dan mengelola data portofolio siswa adalah BKK. Hal ini bertujuan untuk mengetahui profil dari siswa sehingga ketika terdapat permintaan siswa, BKK dapat dengan mudah dan cepat memberikan referensi. BKK juga dapat melakukan CRUD pada portofolio siswa.

44. Nilai Akademik Siswa

Laman ini digunakan untuk mengelola data portofolio akademik siswa. Nilai akademik masih menjadi salah syarat pada banyak lowongan pekerjaan.

MATA PELAJARAN	NILAI	TAHUN	MENU
Bhs. Indonesia	90	2020	

Gambar 5.60 Nilai Akademik Siswa

Melalui laman ini, BKK dapat menambah nilai akademik berdasarkan mata pelajaran yang ditentukan. Tidak semua nilai akademik ditambahkan, yang banyak diperlukan saja misalkan matematika, bahasa inggris, bahasa Indonesia, dan maple keilmuan. Ketika IDUKA melakukan permintaan ke sekolah dengan persyaratan nilai akademik tertentu, maka BKK dapat dengan mudah mencari.

45. Lihat data Tracer Study

Sekolah dan BKK sangat memerlukan data *tracer study* karena dapat digunakan oleh sekolah dalam berbagai kebutuhan. Oleh karena itu, BKK secara rutin harus melakukan kegiatan *tracer study* pada alumni. Kendala utama yang banyak dialami oleh sekolah dalam kegiatan *tracer study* adalah partisipasi dari alumni. Banyak alumni yang tidak tertarik dalam terlihat dalam proses *tracer study*.

Tracer Study Siswa

Nama Lengkap	:	Arif Kurniawan
Jenis Kelamin	:	Laki-laki
Email	:	arifkur@pusatkarirsmk.id
Jurusan	:	Rekayasa Perangkat Lunak
Status	:	Bekerja
No. Telp	:	
Tahun Lulus	:	2020
Aktivitas setelah lulus	:	Bekerja
Nama Perusahaan	:	PT. Maspion Surabaya
Bidang atau Sektor Industri	:	Peralatan Elektronik
Posisi dalam perusahaan	:	Staff Keuangan
Status Pegawai	:	Karyawan Tetap (PKWTT)
Pekerjaan pertama lulus?	:	Ya
Kebermanfaatan jurusan bagi pekerjaan	:	Bermanfaat
Lokasi Perusahaan	:	Dalam Negeri
Domisili Perusahaan	:	KAB. SIDOARJO
Tanggal Mulai Bekerja	:	21/11/2020
Kisaran Gaji per Bulan	:	2 Juta - 3 Juta
Pekerjaan sesuai dengan jurusan di Universitas	:	Ya
Informasi memperoleh pekerjaan	:	Mencari Sendiri

Kembali

Gambar 5.61 Lihat Isian Data Tracer Study

Alumni yang telah mengisi data tracer study dapat dilihat dikolom tracer study data siswa. Jika pada kolom tersebut terisi Tambah, maka siswa dan alumni belum mengisi tracer study. Untuk yang belum mengisi data *tracer study*, akun BKK dapat mengisi. Hal ini digunakan untuk mengantisipasi ketika terdapat siswa/alumni yang mengisi kegiatan *tracer study* dalam bentuk *offline*, dengan ditulis dikertas.

46. Detail Data Siswa

Laman ini digunakan untuk menampilkan data detail data siswa beserta nilai portofolio akademik dan portofolio prestasi dan keahlian. Melalui laman ini, BKK dapat dengan cepat dan mudah untuk mengetahui profil siswa/alumni sehingga ketika ada permintaan dari IDUKA dengan kualifikasi tertentu maka dengan mudah BKK mencari dan memberikan referensi siswa/alumni.

Detail Data Siswa

INFORMASI BIO DATA	
BKK	BKK SMKN 3 Bojonegoro
Klaster	-
Nama Lengkap	ADI MARTA
NIK	123456789
Jurusan	Administrasi Keuangan
Email	bkksmkn3bojonegoro@gmail.com
Jenis Kelamin	Laki-laki
Status Pernikahan	Lajang
Tempat Lahir	bojonegoro
Tanggal Lahir	02/21/2001
Agama	Islam
Domisili	KOTA MOJOKERTO
Alamat Sesuai KTP	Jawa timur
Alamat Tinggal	Jawa timur
No. Telp	-
HP	-
Gol. darah	B
Hobi	futsal

[Kembali](#)

NILAI AKADEMIK		
TAHUN	MATA PELAJARAN	NILAI
2020	Bhs. Indonesia	90

BERIKAS PORTOFOLIO			
NAMA BERKAS	TANGGAL UPLOAD	FILE	EKSTENSI
Tidak ada data.			

Gambar 5.62 Detail data Siswa

Laman ini juga bermanfaat ketika sekolah mencari siswa dengan kualifikasi akademik tertentu, prestasi dan keahlian tertentu. Ketika data portofolio siswa terisi dengan lengkap dan benar, maka manfaatnya akan kembali ke siswa sendiri.

47. Edit data Siswa

Laman ini digunakan untuk melakukan perubahan data siswa yang tidak benar. Cara melakukannya, klik tombol Edit sehingga ditampilkan seperti gambar 5.63 berikut.

Tambah Data Siswa

[Tambah Data Siswa](#)

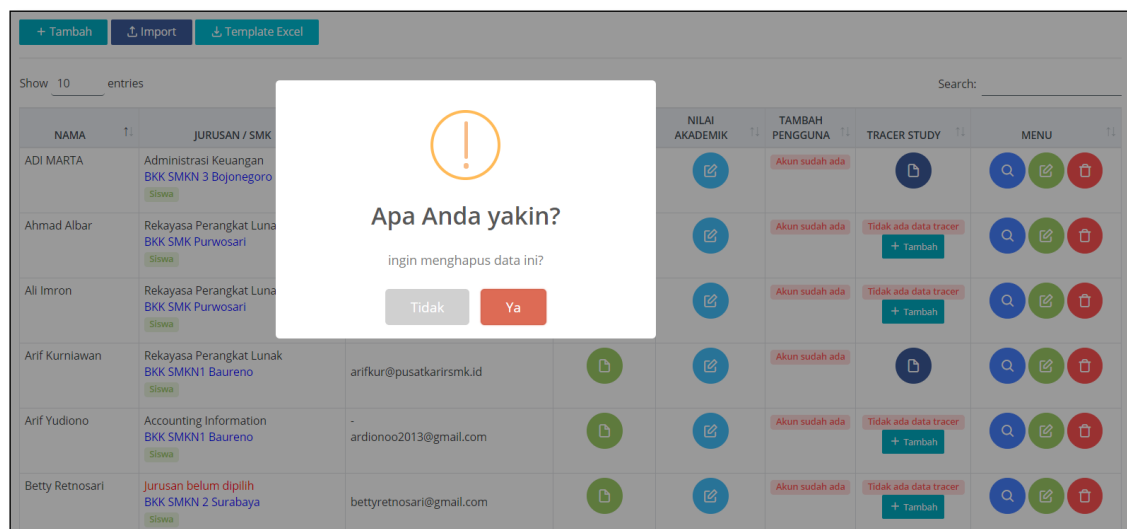
BKK*	BKK SMKN 3 Bojonegoro
Status Personal*	- Pilih -
Nama Lengkap*	ADI MARTA
NIK*	123456789
Jurusan*	Administrasi Keuangan
Email*	bkksmkn3bojonegoro@gmail.com
Jenis Kelamin*	Laki-laki
Status Pernikahan*	Lajang
Tempat Lahir*	bojonegoro
Tanggal Lahir*	02/21/2001
Agama*	Islam
Domisili*	KOTA MOJOKERTO
Alamat Sesuai KTP*	Jawa timur

Gambar 5.63 Edit Data Siswa

Lakukan perubahan seperlunya dan tidak perlu semua karena data yang diubah dan tersimpan, tidak dapat dikembalikan lagi. Lakukan perubahan pada data yang salah dengan mengganti data yang baru dan benar. Setelah melakukan perubahan data, jangan lupa untuk melakukan penyimpanan dengan klik tombol Simpan. Setelah tombol Simpan di-klik, maka perubahan data siswa tersimpan secara permanen pad database. Untuk membatalkan perubahan data, klik tombol Batal.

48. Hapus Data Siswa

Hapus data sebaiknya menjadi solusi terakhir terhadap data siswa, karna data yang telah dihapus tidak dapat dikembalikan lagi. Paga gambar 5.64, ditampilkan konfirmasi ketika pengguna akan menghapus data siswa. Hal ini bertujuan untuk memastikan kembali bahwa hapus data dilakukan dengan sengaja dan sadar.



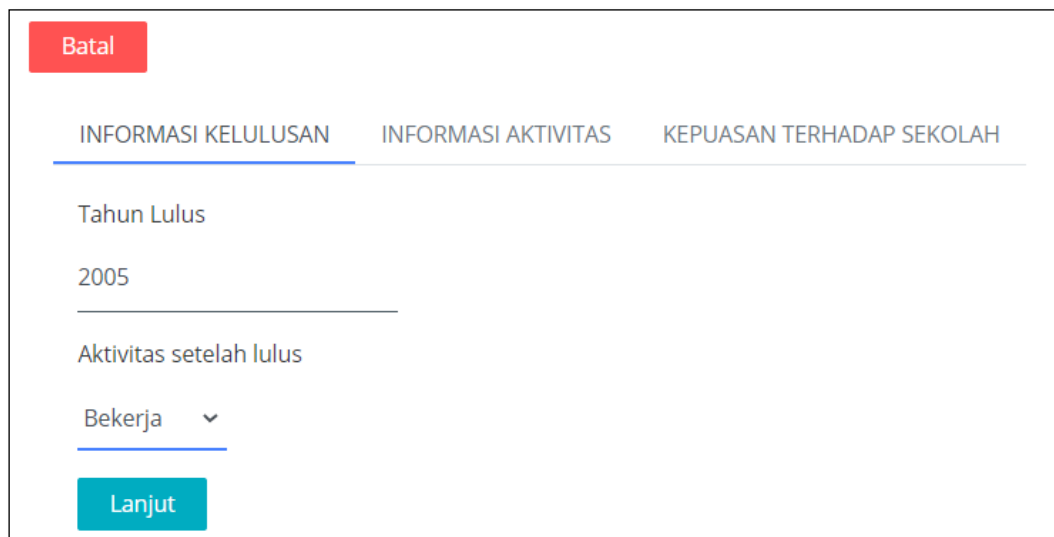
Gambar 5.64 Hapus Data Siswa

Solusi lain agar tidak terjadi proses hapus data adalah melalui mekanisme edit. Kalau data masih bisa diperbaiki, maka lakukan proses edit data agar menjadi benar. Proses hapus data akan menghapus juga data yang berkaitan. Sebagai contoh ketika data siswa dihapus, maka data portofolio akademik, portofolio keahlian dan prestasi, data tracer study, dan data CV juga menjadi tidak dapat diakses lagi karena induk datanya telah dihapus.

49. Tambah Tracer Study

Laman ini digunakan untuk menambah data tracer study yang dapat dilakukan oleh siswa/alumni dan BKK. Gambar 5.65 adalah tampilan pertama yang ditampilkan pada proses tracer study. Data pertama yang diisikan adalah tahun lulus dan

aktivitas setelah lulus. Isian aktivitas setelah lulus berupa pilihan: Bekerja, kuliah, wirausaha, dan lainnya.



The image shows a web-based form for a tracer study. At the top left is a red 'Batal' button. Below it are three tabs: 'INFORMASI KELULUSAN' (active), 'INFORMASI AKTIVITAS', and 'KEPUASAN TERHADAP SEKOLAH'. Under the active tab, there is a 'Tahun Lulus' label followed by a text input field containing '2005'. Below that is an 'Aktivitas setelah lulus' label followed by a dropdown menu showing 'Bekerja'. At the bottom left of the form is a teal 'Lanjut' button.

Gambar 5.65 Mengisi Tracer Study

Pilihan aktivitas setelah lulus sangat berpengaruh pada tahapan selanjutnya. Secara garis besar, terdapat tiga kelompok isian dalam tracer study yaitu:

- Informasi kelulusan
- Informasi Aktivitas
- Kepuasan terhadap sekolah

Isian informasi aktivitas sangat bergantung pada informasi kelulusan. Jika informasi kelulusan terisi kuliah, maka informasi aktivitas berisi isian terkait studi lanjut. Ketika aktivitas setelah lulus dipilih bekerja, maka ditampilkan isian tracer study terkait dengan pekerjaan seperti gambar 5.66 berikut. Data yang diisikan antara lain:

- a. Bidang atau sector Industri
- b. Posisi dalam perusahaan
- c. Status kontrak
- d. Pekerjaan ini merupakan pekerjaan pertama?
- e. Kebermanfaatan jurusan
- f. Luar atau dalam negeri
- g. Domisili perusahaan
- h. Tanggal mulai bekerja
- i. Kisaran gaji per-bulan
- j. Pekerjaan sesuai dengan jurusan?

k. Sumber mendapatkan informasi pekerjaan

The screenshot shows a web form titled 'Data Tracer Bekerja' with a 'Batal' button at the top left. The form is divided into three tabs: 'INFORMASI KELULUSAN', 'INFORMASI AKTIVITAS' (which is active), and 'KEPUASAN TERHADAP SEKOLAH'. The form contains the following fields and options:

- Nama Perusahaan***: Text input field.
- Bidang atau Sektor Industri***: Dropdown menu with the selected option 'Arsitektur / Bangunan / Konstruksi'.
- Posisi dalam perusahaan***: Text input field with the value 'Staff Keuangan'.
- Status Kontrak***: Dropdown menu with the selected option 'Karyawan Paruh Waktu'.
- Pekerjaan ini merupakan pekerjaan pertama?***: Dropdown menu with the selected option 'Ya'.
- Kebermanfaatn jurusan bagi pekerjaan kamu saat ini***: Dropdown menu with the selected option 'Sangat Bermanfaat'.
- Luar atau Dalam Negeri?***: Dropdown menu with the selected option 'Dalam Negeri'.
- Domisili Perusahaan***: Dropdown menu with the selected option 'KOTA SURABAYA'.
- Tanggal Mulai Bekerja***: Text input field with the value '20/08/2020' and a calendar icon.
- Kisaran Gaji per Bulan***: Dropdown menu with the selected option '2 Juta - 3 Juta'.
- Apakah pekerjaanmu sesuai dengan jurusanmu di Universitas?***: Dropdown menu with the selected option 'Ya'.
- Saya memperoleh pekerjaan ini melalui***: Dropdown menu with the selected option 'BPC Amikom'.

A 'Lanjut' button is located at the bottom right of the form.

Gambar 5.66 Mengisi Data Tracer Bekerja

Jika aktivitas setelah lulus dipilih kuliah, maka tampilan selanjutnya seperti gambar 5.67 berikut. Informasi yang diisikan terkait dengan pendidikan pada jenjang selanjutnya yaitu:

- Nama perguruan tinggi
- Jenjang pendidikan
- Program studi
- Lokasi perguruan tinggi
- Mulai studi
- Linier dengan jurusan yang di SMK atau tidak
- Deskripsi

Informasi studi lanjut yang diisikan dengan lengkap dan benar sangat bermanfaat bagi sekolah guna mengetahui alumni yang melanjutkan studi ke jenjang lebih tinggi. Fakta menunjukkan bahwa tidak semua alumni SMK langsung bekerja, ada juga yang lebih memilih untuk melanjutkan ke jenjang yang lebih tinggi. Perguruan tinggi yang dipilih umumnya adalah politeknik, sekolah vokasi dan sarjana terapan.

Terdapat kasus dimana lulusan SMK bekerja dan kuliah. Terhadap kasus ini, pusat karir SMK memberikan fasilitas untuk melakukan pengisian data tracer study sebanyak dua kali, satu untuk aktivitas bekerja dan satu untuk aktivitas studi lanjut.

The screenshot shows a web form titled "Mengisi data Tracer kuliah". At the top left is a red "Batal" button. Below it are three tabs: "INFORMASI KELULUSAN", "INFORMASI AKTIVITAS" (which is selected), and "KEPUASAN TERHADAP SEKOLAH". The form contains several fields: "Nama Universitas / Perguruan Tinggi*" (text input), "Saya Meneruskan Studi pada Tahun*" (text input with "2005" entered), "Jenjang Pendidikan*" (dropdown menu with "D4" selected), "Saya kira, jurusan sewaktu di SMK dengan Kuliah*" (text input with "Linier atau sesuai" entered), "Prodi atau Jurusan Kamu*" (dropdown menu with "Komputer dan Teknologi Teknik Informatika" selected), "Domisili Perguruan Tinggi*" (dropdown menu with "KOTA SURABAYA" selected), and "Deskripsi Studi" (text input with the placeholder text "tidak ada deskripsi"). At the bottom left is a green "Lanjut" button.

Gambar 5.67 Mengisi data Tracer kuliah

Jika aktivitas setelah lulus dipilih wirausaha, maka tampilan selanjutnya seperti gambar 5.68 berikut. Data yang diisikan terkait dengan profil wirausaha yang telah dirintis dan didirikan, yaitu:

- Nama perusahaan
- Bidang usaha
- Lokasi usaha
- Tanggal mulai perusahaan didirikan
- Omzet perusahaan
- Status bidang usaha liner dengan jurusan yang diambil pada waktu sekolah
- Status kebermanfaatan jurusan
- Deskripsi

Isian yang bertanda bintang (*), artinya wajib diisi sedangkan yang tidak bersifat tidak wajib. Sebaiknya, semua data diisi dengan lengkap dan benar sehingga bermanfaat bagi alumni dan sekolah. jumlah data wirausaha yang banyak dan lengkap dapat

dijadikan referensi bagi sekolah untuk melakukan kerjasama, bursa kerja, dan profil sukses alumni.

Kegiatan *tracer study* harus dilakukan secara rutin dan terus menerus. Sekolah harus aktif melakukan jemput bola, dan tidak kalah pentingnya dalam melakukan perekaman data baik secara *offline* maupun *online*. Alumni juga sering dilibatkan dalam kegiatan BKK misalkan sebagai narasumber untuk memberikan motivasi kepada adik kelas.

The screenshot shows a web form with the following elements:

- Buttons:** A red 'Batal' button at the top left and a teal 'Lanjut' button at the bottom left.
- Tabs:** Three tabs are visible: 'INFORMASI KELULUSAN' (blue), 'INFORMASI AKTIVITAS' (blue and underlined), and 'KEPUASAN TERHADAP SEKOLAH' (grey).
- Form Fields:**
 - Nama Perusahaan*:** Text input with 'CV. Maju Bersama' entered.
 - Rentang Omzet per Bulan*:** Dropdown menu with '2 juta - 5 juta / bulan' selected.
 - Bidang / Sektor*:** Dropdown menu with 'Arsitektur / Bangunan / Konstruksi' selected.
 - Jurusan kuliah dengan bidang usaha (Linier / Tidak linier)*:** Dropdown menu with 'Linier (usaha sesuai dengan jurusan kuliah)' selected.
 - Domisili Perusahaan*:** Dropdown menu with 'KOTA SURABAYA' selected.
 - Saya memulai usaha tersebut pada*:** Date input with '20/07/2018' and a calendar icon.
 - Kebermanfaatan jurusan kuliah bagi usaha kamu saat ini*:** Dropdown menu with 'Sangat Bermanfaat' selected.
 - Deskripsi Singkat Usaha:** Text input with the placeholder 'tidak ada deskripsi'.

Gambar 5.68 Mengisi Data Tracer Berwirausaha

Setelah mengisi data profil bekerja atau profil studi lanjut atau profil berwirausaha, klik tombol lanjut untuk mengisi data kepuasan terhadap sekolah. Tampilan laman untuk mengisi kepuasan terhadap sekolah terlihat pada gambar 5.69 berikut.

INFORMASI KELULUSAN	INFORMASI AKTIVITAS	KEPUASAN TERHADAP SEKOLAH
Kepuasan dengan kualitas pendidikan*		
☐	☐	☐
Tidak Puas	Kurang Puas	Cukup
		☒
		Puas
		☐
		Sangat Puas
Menurut saya, nilai aspek Kurikulum Pendidikan adalah*		
☐	☐	☐
Sangat Buruk	Buruk	Cukup
		☒
		Baik
		☐
		Sangat Baik
Menurut saya, nilai aspek Fasilitas (Sarana Prasarana) adalah*		
☐	☐	☐
Sangat Buruk	Buruk	Cukup
		☐
		Baik
		☒
		Sangat Baik
Menurut saya, nilai aspek Lab Praktikum / Bengkel Produktif adalah*		
☐	☐	☐
Sangat Buruk	Buruk	Cukup
		☐
		Baik
		☒
		Sangat Baik
Menurut saya, nilai aspek Guru atau Tenaga Pendidik adalah*		
☐	☐	☐
Sangat Buruk	Buruk	Cukup
		☐
		Baik
		☒
		Sangat Baik

Gambar 5.69 Kepuasan terhadap Sekolah

Cara mengisi data kepuasan terhadap sekolah dilakukan dengan memilih satu pilihan dari lima opsi pilihan yang tersedia. Terdapat 10 pertanyaan yang harus dijawab terkait kepuasan siswa terhadap sekolah, yaitu:

- Kualitas pendidikan
- Kurikulum pendidikan
- Fasilitas (sarana prasarana)
- Lab praktikum/bengkel produktif
- Guru atau tenaga pendidik
- Keterampilan dan inovasi siswa
- Kedisiplinan siswa disekolah
- Kebersihan lingkungan disekolah
- Bursa kerja khusus
- Layanan dan sistem digital BKK

Semua pertanyaan dijawab dengan memilih opsi pilihan yang tersedia. Setelah semua pertanyaan dijawab, alumni dapat mengisi kritik dan saran terhadap sekolah kemudian klik tombol lanjut sehingga ditampilkan pesan bahwa pengisian *tracer study*

telah selesai dilakukan. Alumni dan BKK selanjutnya dapat menampilkan data isian tracer study yang telah dilakukan seperti gambar 5.70 berikut.

Nama Lengkap	:	Ali Imron
Jenis Kelamin	:	Laki-laki
Email	:	ali_imron@gmail.com
Jurusan	:	Rekayasa Perangkat Lunak
Status	:	Wirausaha
No. Telp	:	e
Tahun Lulus	:	2000
Aktivitas setelah lulus	:	Wirausaha
Nama Perusahaan	:	CV. Maju Bersama
Bidang / Sektor	:	Arsitektur / Bangunan / Konstruksi
Domisili Perusahaan	:	KOTA SURABAYA
Mulai Usaha	:	30/11/-0001
Rentang Omzet per Bulan	:	2 juta - 5 juta / bulan
Jurusan kuliah dengan bidang usaha	:	Linier (usaha sesuai dengan jurusan kuliah)
Kebermanfaatan jurusan kuliah bagi usaha	:	Sangat Bermanfaat
Deskripsi Singkat Usaha	:	tidak ada deskripsi

[Kembali](#)

Gambar 5.70 Data Isian Tracer Study

50. Data Agenda

Laman ini digunakan untuk mengelola data agenda kegiatan pusat karir dan BKK. pada laman ini, pengguna dapat melakukan CRUD (Create, Read, Update, Delete) data agenda kegiatan.

Data Agenda						
+ Tambah						
PENYELENGGARA	JUDUL AGENDA	LOKASI/TEMPAT	TANGGAL SELESAI	WAKTU	BIAYA	MENU
SMKN 5 SURABAYA	SURVEI DISNAKER KOTA SURABAYA	Disnaker Provinsi Jatim	Sabtu, 31 Oktober 2020	10:00 s/d 17:00	gratis	  

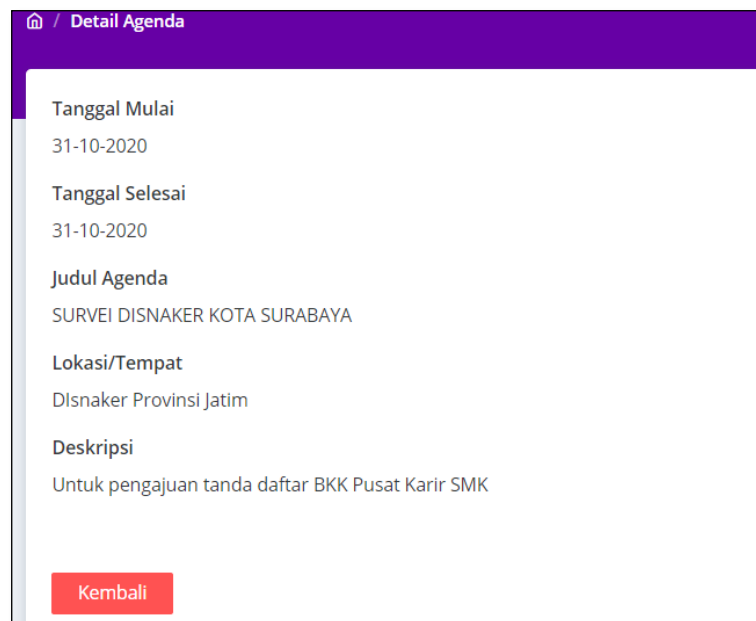
Gambar 5.71 Data Agenda Kegiatan

Pengguna yang dapat melakukan pengelolaan data agenda adalah level akses pusat karir dan BKK. Gambar 5.71 menampilkan data agenda yang ditampilkanurut berdasarkan waktu *publish* secara *descending*, artinya agenda terbaru akan ditampilkan paling atas. Data yang ditampilkan yaitu:

- Penyelenggara
- Judul agenda
- Lokasi/tempat
- Tanggal selesai
- Waktu
- Biaya

Pengelolaan agenda dapat menggunakan tombol yang tersedia yaitu tambah, edit dan hapus agenda. Tombol detail digunakan untuk menampilkan agenda tertentu secara detail seperti gambar 5.72 berikut. Pada laman detail agenda, tambahan informasi yang ditampilkan adalah deskripsi.

Data agenda yang dibuat oleh pusat karir pusat karir akan ditampilkan untuk semua, sedangkan yang dibuat oleh BKK akan ditampilkan pada internal BKK itu sendiri. Setiap BKK dapat membuat agenda kegiatan tanpa harus takut bentrok dengan kegiatan dari BKK yang lain karena secara *default*, agenda kegiatan BKK bersifat lokal.



The screenshot shows a web interface titled 'Detail Agenda'. It contains the following fields and values:

Field	Value
Tanggal Mulai	31-10-2020
Tanggal Selesai	31-10-2020
Judul Agenda	SURVEI DISNAKER KOTA SURABAYA
Lokasi/Tempat	Disnaker Provinsi Jatim
Deskripsi	Untuk pengajuan tanda daftar BKK Pusat Karir SMK

At the bottom of the form is a red button labeled 'Kembali'.

Gambar 5.72 Detail Data Agenda

51. Tambah Agenda

Laman ini digunakan untuk menambah data agenda kegiatan baru baik untuk pusat karir maupun BKK. Klik tombol tambah untuk menambah data agenda kegiatan kemudian isikan data agenda yang akan dibuat. Data yang diisikan pada tambah agenda kegiatan baru antara lain:

- Tanggal mulai
- Tanggal selesai
- Waktu mulai
- Waktu selesai
- Judul agenda kegiatan
- Lokasi/tempat
- Deskripsi
- Biaya

Tampilan tambah data agenda kegiatan dapat dilihat pada gambar xx berikut.

Penyelenggara Agenda
Masukkan nama penyelenggara agenda
Tanggal Mulai
12/09/2020
Tanggal Selesai
12/09/2020
Waktu Mulai
Format : hh:mm
Waktu Selesai
Format : hh:mm
Judul Agenda
Masukkan judul agenda
Lokasi/Tempat
Masukkan lokasi atau tempat
Deskripsi
Masukkan deskripsi agenda
Biaya
Masukkan biaya
Simpan Batal

Gambar 5.73 Tambah Agenda Kegiatan

Biasakan untuk mengisikan data secara lengkap dan benar agar data yang tersimpan bukan data sampah. Klik tombol simpan untuk melakukan penyimpanan data secara permanen didatabase, klik tombol batal untuk membatalkan. Jika data telah tersimpan dan terdapat kesalahan, maka data jangan dihapus tapi diubah melalui mekanisme edit data.

52. Edit Agenda

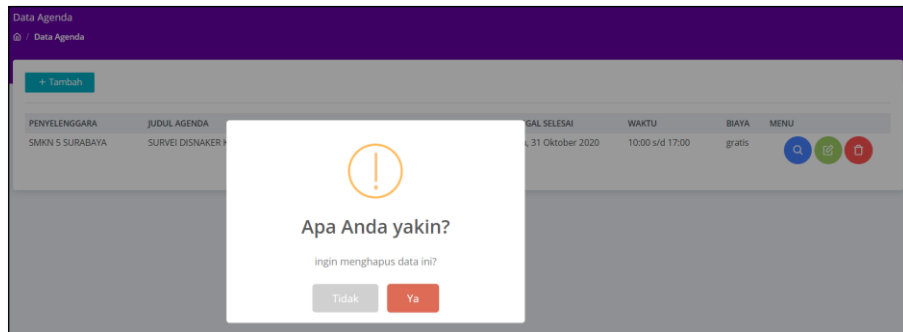
Digunakan untuk melakukan perubahan data ketika terdapat kesalahan pada data agenda atau terdapat perubahan jadwal agenda. Untuk melakukan edit agenda, klik tombol edit (warna hijau) yang terdapat kolom menu. Pada edit data agenda, hanya data yang salah saja yang dilakukan perubahan data, tidak semua data. Cara melakukan perubahan yakni dengan mengganti data yang salah dengan data yang benar (baru). Agar perubahan data tersimpan secara permanen, klik tombol simpan. Klik tombol batal untuk melakukan pembatalan peruban data.

Penyelenggara Agenda
SMKN 5 SURABAYA
Tanggal Mulai
10/31/2020
Tanggal Selesai
10/31/2020
Waktu Mulai
10:00
Waktu Selesai
17:00
Judul Agenda
SURVEI DISNAKER KOTA SURABAYA
Lokasi/Tempat
Disnaker Provinsi Jatim
Deskripsi
Untuk pengajuan tanda daftar BKK Pusat Karir SMK
Biaya
gratis
Simpan Batal

Gambar 5.74 Edit Agenda Kegiatan

53. Hapus Agenda

Digunakan untuk menghapus data agenda kegiatan. Hapus data adalah solusi terakhir terhadap data, dan sebaiknya dihindari. Data agenda kegiatan yang telah dihapus tidak dapat dikembalikan lagi karena belum ada fitur *softdelete* pada aplikasi ini. Jadi data yang dihapus akan terhapus juga dari database. Oleh karena itu, jangan mudah untuk melakukan hapus data.

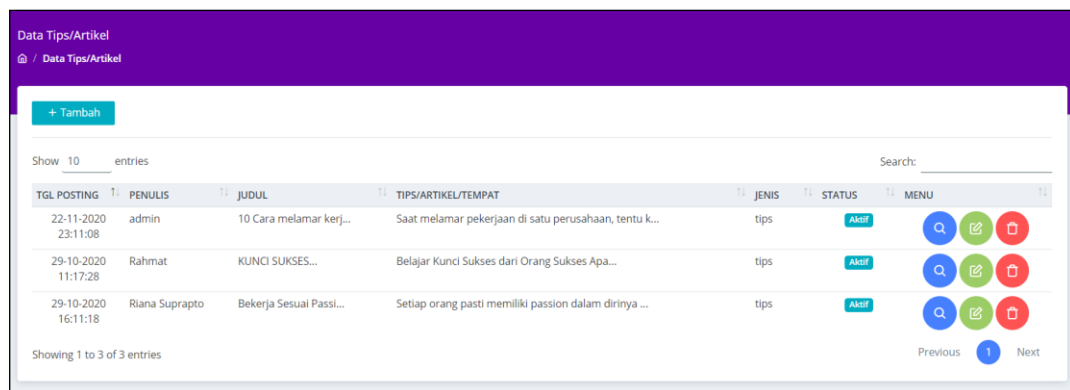


Gambar 5.75 Hapus Agenda Kegiatan

Ketika tombol hapus data di-klik, data tidak langsung dihapus akan tetapi ditampilkan konfirmasi seperti gambar xx. Hal ini bertujuan untuk menghindari salah klik dan untuk memastikan pengguna sadar akan melakukan hapus data.

54. Tips dan Artikel

Laman ini digunakan untuk mengelola tips dan artikel. Pada laman ini pengguna dapat melakukan CRUD (Create, Read, Update, Delete) melalui tombol yang tersedia. Level akses pengguna yang dapat melakukan pengelolaan tips dan artikel adalah pusat karir dan BKK.



Gambar 5.76 Tips dan Artikel

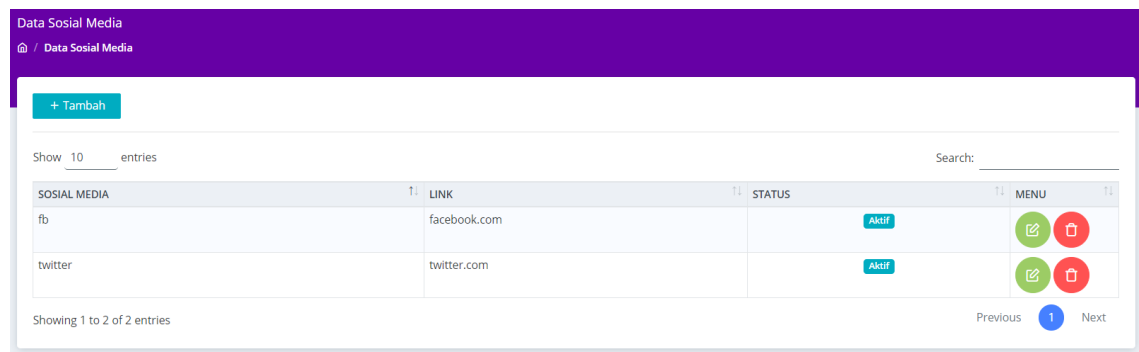
Informasi yang ditampilkan pada gambar 5.76 yaitu:

- Tanggal posting
- Penulis
- Judul
- Tips/artikel
- Jenis
- Status

Data ditampilkan berdasarkan waktu *posting* dan ditampilkan secara *descending*, artinya data terbaru ditampilkan pada paling atas.

55. Data Social Media

Laman ini digunakan untuk mengelola data *social media* yang dimiliki oleh pusat karir SMK. Jenis *social media* yang tersedia yaitu Facebook, Twitter, Instagram, dan LinkedIn.



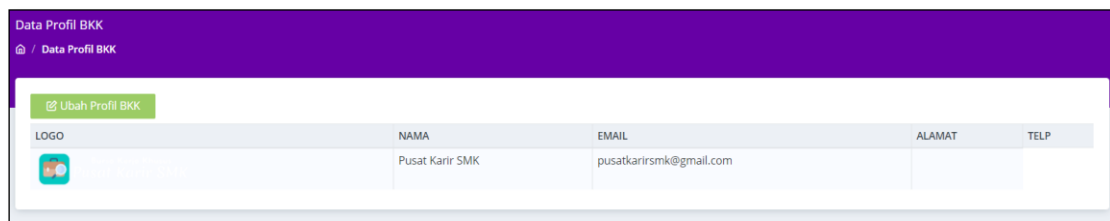
Gambar 5.77 Data Social Media

Data *social media* selanjutnya ditampilkan pada website. Untuk melakukan pengelolaan dapat menggunakan tombol tambah, edit dan hapus yang tersedia. Status aktif dan non-aktif digunakan untuk menentukan aktif atau tidaknya data *social media*.

56. Profil Aplikasi

Laman ini digunakan untuk pengelolaan data profil aplikasi. Hanya ada satu data dan tidak dapat melakukan penambahan. Hanya edit data yang dapat dilakukan untuk menjamin data hanya satu. Data profil aplikasi digunakan sebagai informasi pada aplikasi pusat karir dan ditampilkan sebagai judul website, logo, dan email pengelola. Fitur ini diperlukan dan berguna ketika aplikasi digunakan oleh banyak

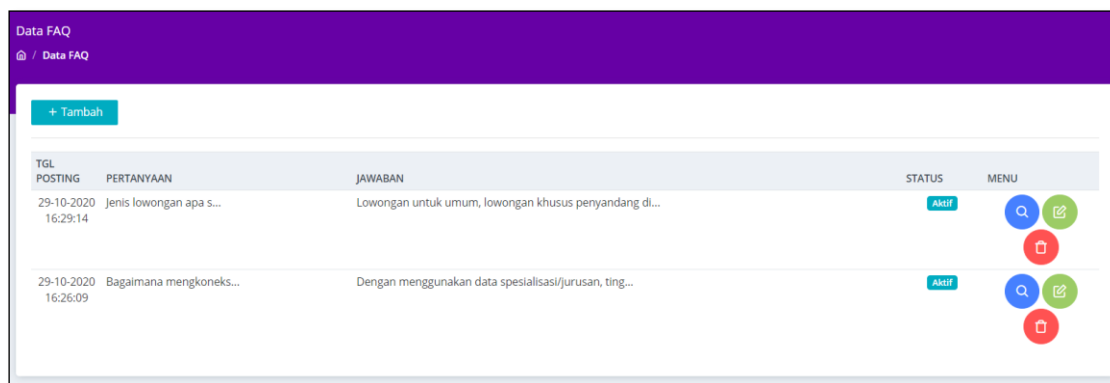
instansi dan harus menyesuaikan profilnya. Melalui fitur ini, perubahan data secara *hardcoding* yakni dengan membuka *source code* tidak perlu dilakukan.



Gambar 5.78 Profil Aplikasi

57. FAQ

Laman ini digunakan untuk pengelolaan FAQ (*Frequently Asked Questions*) atau pertanyaan yang sering ditanyakan. Informasi yang ditampilkan pada laman utama kalola FAQ yaitu tanggal posting, pertanyaan, jawaban, status, dan tombol untuk tambah, edit dan hapus.



Gambar 5.79 Data FAQ

Untuk menambah FAQ baru, klik tombol tambah kemudian isikan pertanyaan, jawaban, dan pilih status pada laman tambah data kemudian klik tombol simpan. Untuk ubah data, klik tombol edit kemudian ganti data yang salah dengan data yang benar lalu klik tombol simpan. Untuk hapus data, klik tombol hapus kemudian pilih Yes pada konfirmasi yang muncul, dan data otomatis terhapus secara permanen dari database.

58. Berkas BKK

Laman ini digunakan untuk pengelolaan data dokumen dan berkas BKK dalam bentuk *digital* atau *softcopy*. Fitur ini mendukung era industri 4.0 yakni *digitalisasi*.

Data Berkas
Data Berkas

+ Tambah

Show 10 entries

Search:

Nama Berkas	Nama File	Ukuran File	Ekstensi	Preview	Tampil?	Diakses Publik?	Hapus
berkas BKK	bac47a8de9b27e9a4a749cc3c525cfeb.docx	1902.65 Kb	docx	File	Ya	Ya	
Berkas Umum	f91db5837ecb5ae30de78de5e1ac068d.pdf	505.05 Kb	pdf	File	Ya	Tidak	
Berkas Umum	89defcda0981a5a4dc29471f5c7d131d.pdf	505.05 Kb	pdf	File	Ya	Tidak	
Berkas Visi Misi BKK	63edd87ab394c15da3c3bef950769a4a.docx	320.16 Kb	docx	File	Ya	Tidak	
Foto Rapat SAI	b4ec6096702734fa6f342f027a34e24d.JPG	4336.9 Kb	JPG	File	Tidak	Tidak	
Sertifikat BKK	3e4cd55e16505d37d916c72614a36a4e.pdf	61.08 Kb	pdf	File	Ya	Tidak	

Showing 1 to 6 of 6 entries

Previous 1 Next

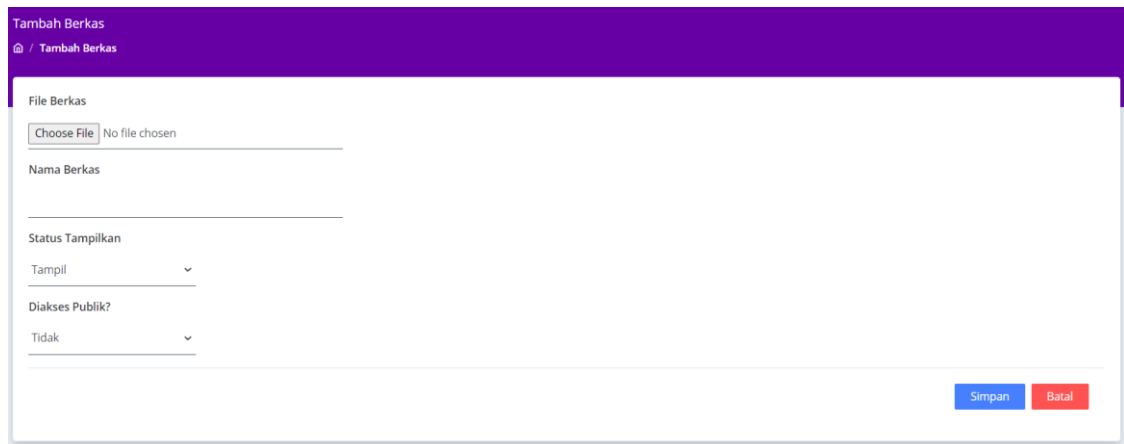
Gambar 5.80 Berkas BKK

Informasi yang ditampilkan pada laman utama data berkas yaitu nama berkas, nama file, ukuran file, dan ekstensi file. Status tampil “Ya” artinya ditampilkan, “Tidak” artinya tidak ditampilkan. Kolom diakses publik jika berisi “Ya” artinya data berkas digital dapat diakses oleh umum, dan jika berisi “Tidak” maka berkas digital tersebut hanya dapat diakses oleh pengelola BKK. Tombol pada kolom “Preview” digunakan untuk menampilkan atau *download* file berkas. Untuk menambah data berkas *softcopy* baru, klik tombol tambah. Untuk menghapus, klik tombol hapus. Tidak ada fitur untuk edit data berkas, jadi kalau terdapat berkas yang salah maka berkas tersebut dihapus dan ditambahkan lagi dengan yang benar.

59. Tambah Berkas

Laman ini digunakan untuk menambah berkas digital atau *softcopy* baru. Data yang diisi antara lain:

- File berkas, format file yang diperbolehkan: pdf, doc, jpg, jpeg, dan png.
- Nama berkas
- Status ditampilkan
- Status diakses publik



Gambar 5.81 Tambah Berkas BKK

Isikan data dengan benar dan lengkap. Agar berkas *softcopy* tersimpan secara permanen pada database, klik tombol simpan sehingga ditampilkan Kembali data berkas. klik tombol batal untuk membatalkan penambahan data berkas.

60. Data Pengguna

Laman ini digunakan untuk pengelolaan data *users* atau pengguna. Pengelolaan data *users* dapat dilakukan oleh pusat karir dan BKK. Pusat karir dapat melakukan Kelola semua data *users*, sedangkan BKK hanya dapat mengelola data *users* dimasing-masing BKK. Data *users* yang dapat dikelola oleh BKK adalah data siswa, alumni, dan Iduka yang berada pada BKK tersebut. Data yang ditampilkan pada laman ini yaitu:

- Username, isinya adalah alamat email
- Nama pengguna
- Email
- Hak akses
- Status

Data password tidak ditampilkan karena alasan kerahasiaan. Sebagai gantinya terdapat tombol *reset password*. Tombol yang tersedia yaitu tambah data, edit data, dan hapus data.

Data Pengguna

+ / Data Pengguna

+ Tambah IDUKA

Show 10 entries

Search:

USERNAME	NAMA PENGGUNA	EMAIL	HAK AKSES	STATUS	RESET PASSWORD	MENU
ahmad@gmail.com	Ahmad Albar	ahmad@gmail.com	Siswa/Alumni	Aktif		
ali_imron@gmail.com	Ali Imron	ali_imron@gmail.com	Siswa/Alumni	Aktif		
ardionoo2013@gmail.com	Arif Yudsono	ardionoo2013@gmail.com	Siswa/Alumni	Aktif		
arifkur@pusatkarismk.id	Arif Kurniawan	arifkur@pusatkarismk.id	Siswa/Alumni	Aktif		
asmunin@pusatkarismk.id	Asmunin	asmunin@pusatkarismk.id	Administrator	Aktif		
asmunin@unesa.ac.id	Asmunin	asmunin@unesa.ac.id	Administrator	Aktif		
bettyretnosari@gmail.com	Betty Retnosari	bettyretnosari@gmail.com	Siswa/Alumni	Aktif		
bkkismk5sby@gmail.com	BKK SMK 5 Surabaya	bkkismk5sby@gmail.com	Bursa Kerja Khusus	Aktif		

Gambar 5.82 Data Pengguna

Jumlah data users yang ditampilkan berbeda antara pusat karir dengan BKK dan berbeda juga antar tiap BKK. Sebagai contoh, jika yang login adalah admin BKK SMKN 5 Surabaya, maka data pengguna yang ditampilkan adalah siswa, alumni, dan iduka yang terdaftar untuk BKK tersebut.

61. Reset Password

Data *password* pengguna tersimpan didatabase dalam bentuk acak (*enkripsi*) sehingga tidak dapat dibaca oleh pengguna lain termasuk pemilik *password*. Ketika pengguna lupa *password*, maka *password* untuk pengguna tersebut tidak dapat diberitahukan. Sebagai solusinya, *password* harus diganti melalui fitur reset *password* ini. Yang dapat melakukan reset adalah pusat karir dan BKK. Tampilan reset password dapat dilihat pada gambar xx berikut. Cara melakukan reset adalah:

- Pilih users yang akan direset
- Klik tombol *reset password*
- Isikan password baru untuk users yang direset
- Klik tombol simpan

Gambar 5.83 Reset Password

Ketika *password users* telah diganti, maka *users* tersebut dapat melakukan login dengan menggunakan password yang baru.

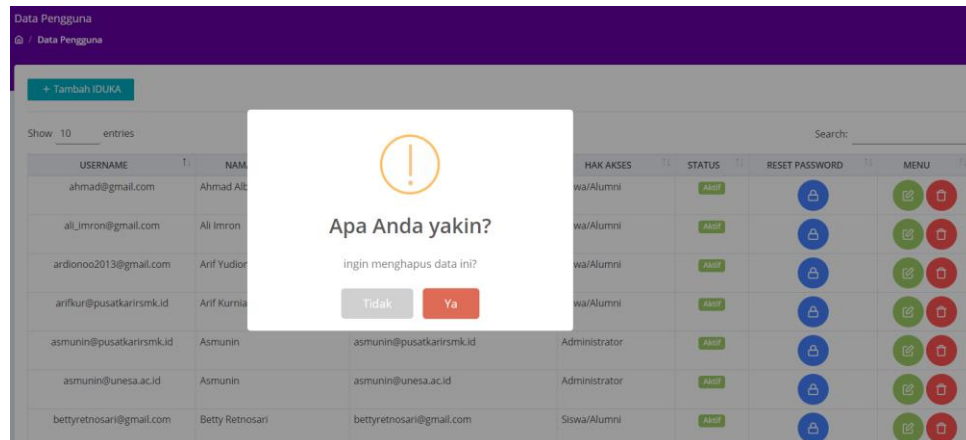
62. Edit data Pengguna

Laman ini digunakan untuk ubah data pengguna, bukan password pengguna. Cara yang dilakukan: klik tombol edit sehingga muncul pada gambar xx kemudian ganti data yang salah dengan data yang benar, kemudian klik tombol simpan agar perubahan data tersimpan secara permanen didatabase. Klik tombol batal untuk membatalkan ubah data pengguna.

Gambar 5.84 Edit Data Pengguna

63. Hapus Data Pengguna

Laman ini digunakan untuk menghapus data users atau pengguna. Fitur ini sebaiknya tidak mudah digunakan karena berbahaya. Proses hapus data merupakan solusi terakhir ketika data tidak dapat diubah atau data tersebut memang cacat.



Gambar 5.85 Hapus Data Pengguna

Cara menghapus data users: pilih data users yang akan dihapus, kemudian klik tombol hapus. Konfirmasi hapus akan ditampilkan untuk memastikan bahwa proses hapus dilakukan dengan sengaja. Klik tombol “Ya” pada konfirmasi yang muncul maka data otomatis telah terhapus dari database. Data yang telah dihapus tidak dapat dikembalikan lagi.

64. Data BKK

Laman ini digunakan untuk menampilkan data BKK yang terdaftar pada pusat karir. Kalau pusat karir SMK diibaratkan seperti Mall, maka BKK adalah toko-toko yang terdapat pada Mall tersebut. Aplikasi pusat karir SMK ini dapat menampilkan BKK di seluruh Indonesia. Data BKK hanya dapat ditambahkan oleh pusat karir SMK, belum ada pendaftaran BKK secara online dan mandiri. BKK yang datanya ingin ditambahkan dapat menghubungi pusat karir SMK sehingga ditambahkan dan mendapat akun yang digunakan untuk login. Akun login untuk BKK adalah email dan password yang telah ditentukan. Data yang ditampilkan pada gambar 5.86 yaitu:

- Nama BKK
- Email BKK
- Alamat
- Telpon
- Status user BKK, ada atau tidak.

NAMA	EMAIL	ALAMAT	TELP	USER BKK	TAMBAH USER	MENU
BKK SMK 5 Surabaya	bkksmk5sby@gmail.com	Surabaya	-	Ada		
BKK SMK Purwosari	bkksmknpurwosari@gmail.com	Bojonegoro	-	Ada		
BKK SMKN 2 Surabaya	bkksmkn2surabaya@gmail.com	Surabaya	-	Ada		
BKK SMKN 3 Bojonegoro	bkksmkn3bojonegoro@gmail.com	Bojonegoro	-	Ada		
BKK SMKN Sumberrejo	bkksmknsumberrejo@gmail.com	Bojonegoro	-	Ada		
BKK SMKN1 Baureno	bkksmkn1baureno@gmail.com	Bojonegoro	-	Ada		

Gambar 5.86 Data BKK

Tombol yang tersedia:

- Tambah, untuk menambahkan data BKK baru
- Tambah user, digunakan untuk menambahkan user BKK sehingga dapat melakukan login.
- Edit, untuk mengubah data BKK
- Hapus, untuk menghapus data BKK

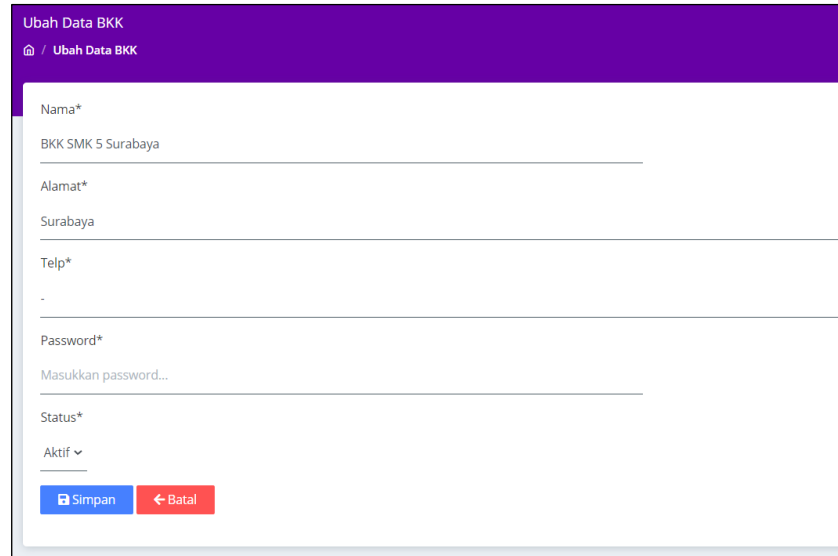
65. Tambah BKK

Laman ini digunakan untuk menambah data BKK. Data yang ditambahkan antara lain: nama BKK, alamat, telpon, dan email yang digunakan untuk login.

Gambar 5.87 Tambah BKK

66. Edit Data BKK

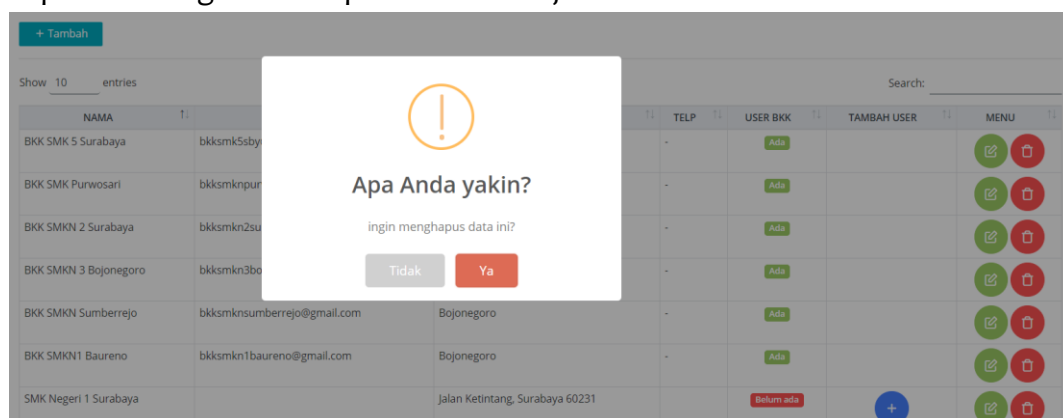
Laman ini digunakan untuk edit data BKK pada data yang salah. Cara melakukan: pilih data BKK kemudian klik tombol Edit sehingga muncul tampilan seperti gambar 5.88 berikut. Lakukan perubahan pada data yang salah kemudian klik tombol simpan.



Gambar 5.88 Edit BKK

67. Hapus Data BKK

Hapus data BKK sebaiknya sangat dihindari, karena data yang terhapus tidak dapat dikembalikan lagi. Dalam menggunakan fitur hapus harus hati-hati, dan data yang dihapus memang data sampah atau data ujicoba.



Gambar 5.89 Hapus data BKK

Meskipun hapus data disarankan untuk tidak banyak digunakan, akan tetapi fitur ini tidak bisa dihilangkan dalam suatu aplikasi. Untuk memastikan pengguna dalam hapus data, selalu ditampilkan konfirmasi. Ketika dikonfirmasi hapus klik tombol Ya, artinya pengguna secara sadar melakukan penghapusan data.

68. Kategori Perusahaan

Laman ini digunakan untuk pengelolaan data master kategori perusahaan. Setiap perusahaan atau IDUKA memiliki kategori bidang usaha. Pada proses CRUD data perusahaan, salah satu isian adalah bidang usaha.

KATEGORI PERUSAHAAN	STATUS	MENU
Arsitektur / Bangunan / Konstruksi	Tidak Aktif	[Edit] [Delete]
Asuransi	Tidak Aktif	[Edit] [Delete]
Besi / Baja / Beton / Semen	Tidak Aktif	[Edit] [Delete]
Civil Engineering	Tidak Aktif	[Edit] [Delete]
Desain Interior / Desain Grafis	Tidak Aktif	[Edit] [Delete]
Energi / Daya / Hidro / Panas / Minyak & Gas Bumi	Tidak Aktif	[Edit] [Delete]
Energi Alternatif & Terbarukan	Tidak Aktif	[Edit] [Delete]
Farmasi	Aktif	[Edit] [Delete]

Gambar 5.90 Kategori Perusahaan

Pengguna yang dapat melakukan pengelolaan data ini adalah dengan level akses admin atau pusat karir. Pengelolaan yang dapat dilakukan adalah CRUD data kategori perusahaan. Sangat disarankan untuk tidak menghapus data, akan akan terhapus secara permanen dari database. Sebagai gantinya, gunakan pengaturan aktif dan non-aktif.

69. Jumlah Karyawan

Data yang ditampilkan pada laman ini termasuk data master yang dijadikan referensi pada proses lain. Proses lain yang membutuhkan data jumlah karyawan adalah CRUD data perusahaan atau IDUKA. Tampilan laman jumlah karyawan terlihat pada gambar 5.91 berikut.

Jumlah Karyawan	STATUS	MENU
> 5000 karyawan	Aktif	
< 50 Karyawan	Aktif	
100 - 200 Karyawan	Aktif	
1000 - 2000 karyawan	Aktif	
200 - 500 karyawan	Aktif	
2000 - 5000 karyawan	Aktif	
50 - 100 Karyawan	Aktif	
500 - 1000 karyawan	Aktif	

Gambar 5.91 Jumlah Karyawan

Pengguna yang dapat melakukan pengelolaan data ini adalah dengan level akses admin atau pusat karir. Pengelolaan yang dapat dilakukan adalah CRUD data jumlah karyawan. Sangat disarankan untuk tidak menghapus data, akan akan terhapus secara permanen dari database. Sebagai gantinya, gunakan pengaturan aktif dan non-aktif.

70. Data Jurusan

Laman ini digunakan untuk pengelolaan data jurusan yang selanjutnya digunakan sebagai referensi dalam proses pengelolaan data siswa, alumni dan riwayat pendidikan format. Pada gambar 5.92, pengguna dapat melakukan CRUD, aktif dan non-aktifkan data jurusan.

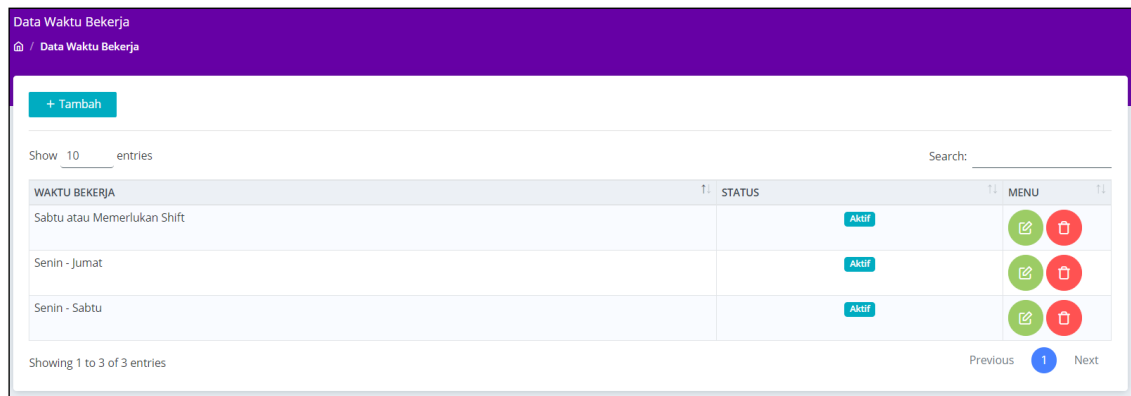
NAMA JURUSAN	STATUS	MENU
Accounting Information	Aktif	
Administrasi Bisnis	Aktif	
Administrasi Fiskal	Aktif	
Administrasi Keuangan	Aktif	
Administrasi Niaga	Aktif	
Administrasi Pendidikan	Aktif	
Administrasi Publik	Aktif	
Agribisnis (Sosial Ekonomi Pertanian)	Aktif	

Gambar 5.92 Data Jurusan

Data jurusan termasuk data master sehingga tidak banyak mengalami perubahan. Data jurusan lebih banyak digunakan dalam proses lain yang memerlukan data jurusan. Data jurusan bisa bertambah dan berkurang, olah karena itu terdapat fasilitas status aktif dan non-aktif. Sangat disarankan untuk tidak menghapus data, akan akan terhapus secara permanen dari database. Sebagai gantinya, gunakan pengaturan aktif dan non-aktif.

71. Waktu Bekerja

Laman ini digunakan untuk pengelolaan waktu bekerja di perusahaan atau IDUKA. data waktu bekerja merupakan data master yang dijadikan referensi pada waktu pengisian data IDUKA dan lowongan pekerjaan.



Gambar 5.93 Data Waktu Bekerja

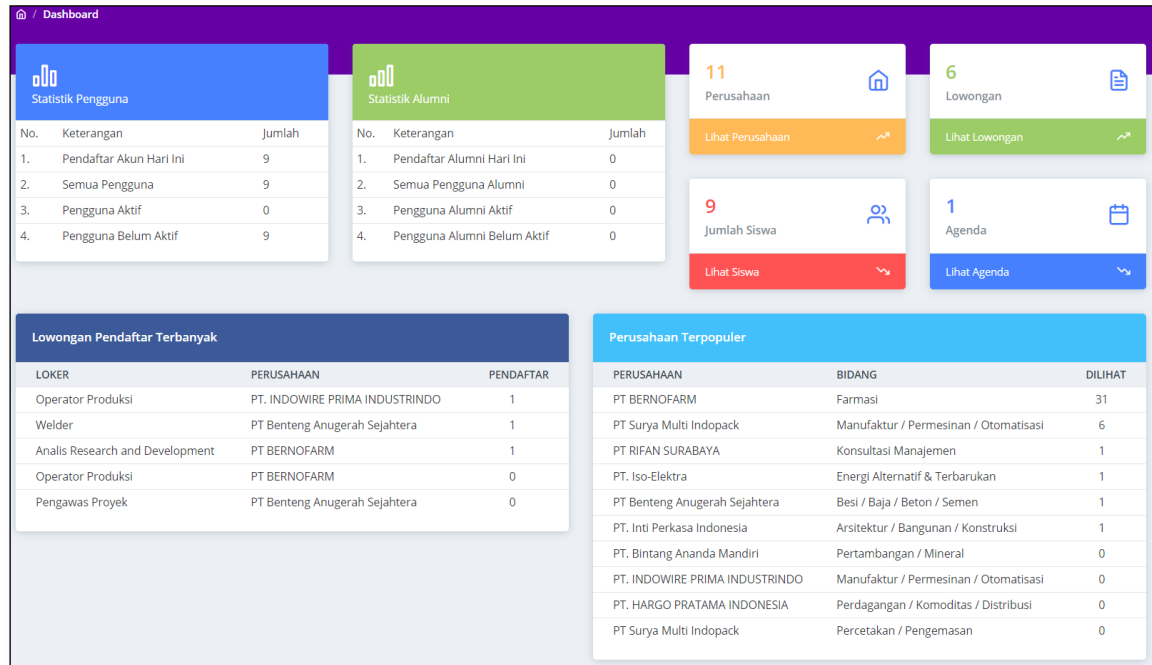
Pada gambar 5.93, pengguna dapat melakukan CRUD (Create, Read, Update, Delete) data waktu bekerja. Fitur untuk CRUD telah tersedia yakni tombol tambah, edit, dan hapus data.

72. Dashboard Pusat Karir

Merupakan tampilan pertama yang ditampilkan setelah pengguna dengan level akses admin atau pusat karir berhasil login. *Dashboard* ini (gambar 5.94) menampilkan semua informasi penting pada aplikasi pusat karir SMK. Informasi yang ditampilkan antara lain:

- Statistik pengguna
- Statistik alumni
- Jumlah perusahaan
- Jumlah lowongan pekerjaan
- Jumlah siswa
- Jumlah agenda kegiatan

- g. Lowongan dengan pendaftar terbanyak
- h. Perusahaan paling populer



Gambar 5.94 Dashboard Admin Pusat Karir

Statistik pengguna menampilkan jumlah: pendaftar hari ini, semua pengguna, pengguna aktif, dan pengguna belum aktif. Statistik alumni menampilkan jumlah: pendaftar alumni hari ini, semua alumni, akun alumni aktif, dan akun alumni tidak aktif. Tautan yang tersedia yaitu:

- Perusahaan, untuk menampilkan semua data perusahaan
- Lowongan, untuk menampilkan semua data lowongan pekerjaan
- Jumlah siswa, untuk menampilkan seluruh data siswa
- Agenda, untuk menampilkan agenda kegiatan

Tiap tautan dapat diklik dan akan dibawah ke laman tujuan tautan. Data lowongan dengan pendaftar terbanyak ditampilkan secara urut berdasarkan jumlah pelamar pada lowongan pekerjaan. lowongan pekerjaan dengan jumlah pelamar paling banyak ditampilkan pada urutan atas, semakin kebawah semakin sedikit. Perusahaan paling populer diurutkan berdasarkan jumlah lihat. Perusahaan dengan jumlah lihat paling banyak, ditampilkan pada urutan atas.

BAB 6

PENUTUP

6.1 Kesimpulan

Dimulai dari analisis kebutuhan dengan datang langsung ke SMK, pembuatan dokumen analisis kebutuhan, pembuatan spesifikasi kebutuhan, desain perangkat lunak, pengembangan aplikasi, pengujian, dan mengundang ketua BKK untuk melakukan ujicoba dalam bentuk FGD (*Focus Group Discussion*) dapat dirumuskan beberapa kesimpulan, antara lain:

1. Pusat karir SMK telah selesai dibuat dengan mengacu pada proses bisnis dan kebutuhan pengguna di SMK.
2. Semua fitur yang tersedia dan dibutuhkan oleh BKK telah terimplementasi sehingga dapat digunakan dalam proses nyata.
3. Pusat karir SMK adalah rumah besar bagi BKK, Iduka, Siswa dan Alumni bertemu dan saling berinteraksi sehingga dalam kurun waktu tertentu akan terkumpul data yang sangat besar (*big data*) terkait data lowongan pekerjaan, portofolio siswa, tracer study yang sangat bermanfaat bagi sekolah dan dinas terkait dalam membantu pengambilan keputusan.
4. Pusat karir SMK dapat menjadi solusi bagi BKK yang tidak memiliki dana untuk membeli perangkat lunak sejenis padahal itu sangat dibutuhkan oleh BKK.
5. Pusat karir SMK dapat menjadi solusi bagi BKK yang belum memiliki website apalagi semacam sistem informasi yang dapat membantu kegiatan operasional BKK. Pada faktanya, banyak BKK yang tidak memiliki aplikasi dan website.
6. Pusat karir SMK dapat digunakan oleh semua BKK diseluruh Indonesia tanpa perlu melakukan instalasi, menyiapkan server, dan menyediakan tenaga ahli. BKK cukup melakukan pendaftaran sehingga memiliki akun, maka langsung dapat menggunakan aplikasi.
7. Fitur utama aplikasi yakni lowongan pekerjaan, tracer study, portofolio siswa, cetak CV otomatis, apply lowongan kerja secara online telah terimplementasi dengan baik dan siap untuk digunakan.
8. Pusat karir SMK dikembangkan menggunakan teknologi *open-source* berbasis framework laravel dan DBMS MySQL.

6.2 Saran

Beberapa saran pengembangan pusat karir SMK berdasarkan masukan dan saran dari berbagai pihak yang terlibat langsung adalah sebagai berikut:

1. Dikembangkan versi mobile baik untuk OS Android maupun IOS agar pengguna dapat melakukan unduh dan install secara mandiri sehingga jumlah pengguna menjadi lebih banyak.
2. Aplikasi terintegrasi dengan dinas tenaga kerja sehingga dapat saling bekerja sama, menggunakan data bersama, saling melengkapi, dan memudahkan koordinasi.
3. Pengembangan aplikasi agar siap untuk pekerjaan jenis baru yang tidak mengharuskan jurusan dan program keahlian, artinya yang memiliki *skill* bagus dan memiliki *passion* akan diterima walaupun program keahliannya tidak sesuai.
4. Integrasi dengan aplikasi terkait data pokok pendidikan khususnya untuk data sekolah, program keahlian, siswa, mata pelajaran, dan guru.
5. Terdapat fitur notifikasi agar pengguna pusat karir SMK mendapatkan informasi terkait data dan interaksi yang telah dilakukan. Melalui fitur ini, pengguna tidak perlu secara aktif melakukan pengecekan pada aplikasi.

DAFTAR PUSTAKA

- ADB. (2020). *ADB Annual Report 2019*. www.adb.org/ar2019/digital
- Boell, S. K., & Cecez-Kecmanovic, D. (2015). What is an information system? *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2015-March, 4959–4968. <https://doi.org/10.1109/HICSS.2015.587>
- BPS. (2020). *Keadaan Angkatan Kerja di Indonesia*.
- Engelland, B. T., Workman, L., & Singh, M. (2000). Ensuring Service Quality for Campus Career Services Centers: A Modified SERVQUAL Scale. *Journal of Marketing Education*, 22(3), 236–245. <https://doi.org/10.1177/0273475300223007>
- Garis, J., Reardon, R. C., & Lenz, J. G. (2012). Current Status and Future Development of Career Centers in the United States. *Asian Journal of Counselling*, 19(2), 5–26.
- Harris-Bowlsbey, J., & Sampson, J. P. (2005). Use of technology in delivering career services worldwide. *Career Development Quarterly*, 54(1), 48–56. <https://doi.org/10.1002/j.2161-0045.2005.tb00140.x>
- International Monetary Fund. (2019). *World Economic Outlook, April 2019: Growth Slowdown, Precarious Recovery*. <https://www.imf.org/en/Publications/WEO/Issues/2019/03/28/world-economic-outlook-april-2019>
- Jurje, F., & Lavenex, S. (2015). ASEAN Economic Community : what model for labour mobility ? *NCCR Trade Regulation*, 1–26.
- Krumm, G., Arán Filippetti, V., Lemos, V., Koval, J., & Balabanian, C. (2016). Construct validity and factorial invariance across sex of the Torrance Test of Creative Thinking – Figural Form A in Spanish-speaking children. *Thinking Skills and Creativity*, 22, 180–189. <https://doi.org/10.1016/j.tsc.2016.10.003>
- McKinsey Global Institute. (2017). *JOBS LOST, JOBS GAINED: WORKFORCE TRANSITIONS IN A TIME OF AUTOMATION*. www.mckinsey.com/mgi.
- Menon, J., & Melendez, A. C. (2017). Realizing an Asean Economic Community: Progress and Remaining Challenge. *Singapore Economic Review*, 62(3), 681–702. <https://doi.org/10.1142/S0217590818400052>
- Qolik, A., Nuralasari, R., Yoto, Y., & Cornelia Tjiptady, B. (2020). The Role of Special Job

Fair in Distributing Competitive Graduates in the 21st Century. *4th International Conference on Vocational Education and Training, ICOVET 2020*, 115–118. <https://doi.org/10.1109/ICOVET50258.2020.9230064>

Sahraee, R. (2014). Determining Essential Steps in Designing a Successful System of Organizational Career Development. *European Online Journal of Natural and Social Sciences*, 3(3), 691–706.

Schutt, D. A. (2000). How to plan and develop a career center. In *Career Planning and Adult ...* (Issue June 2006).

Zacher, H., Rudolph, C. W., Todorovic, T., & Ammann, D. (2019). Academic career development: A review and research agenda. *Journal of Vocational Behavior*, 110(August), 357–373. <https://doi.org/10.1016/j.jvb.2018.08.006>