



**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN**

LEMBAGA PENGEMBANGAN DAN PEMBERDAYAAN PENDIDIK DAN TENAGA KEPENDIDIKAN

BIDANG KELAUTAN PERIKANAN TEKNOLOGI INFORMASI DAN KOMUNIKASI

(LPPPTK KPTK) GOWA

2018

# RPL

REKAYASA  
PERANGKAT LUNAK

Mengimplementasikan Pemrograman Terstruktur

Penulis:

Alun Sujjadah, S.Kom., MT.

**Modul,**  
Pelatihan  
Berbasis  
Kompetensi



**MODUL**

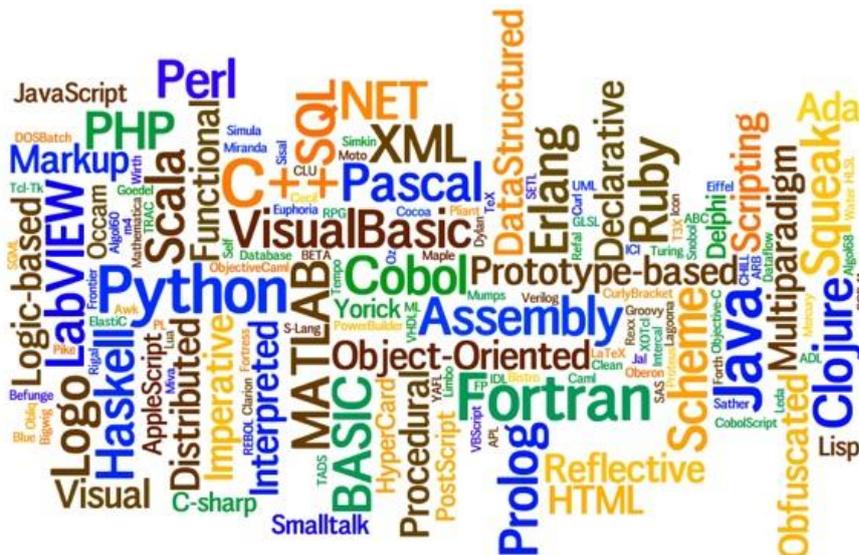
**PENGEMBANGAN KEPROFESIAN**

**BERKELANJUTAN BERBASIS KOMPETENSI**

**MENGIMPLEMENTASIKAN PEMROGRAMAN**

**TERSTRUKTUR**

**J.620100.017.02**



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN R.I.  
**DIREKTORAT JENDERAL GURU DAN TENAGA KEPENDIDIKAN**  
LEMBAGA PENGEMBANGAN DAN PEMBERDAYAAN PENDIDIK DAN TENAGA KEPENDIDIKAN BIDANG  
KELAUTAN, PERIKANAN, TEKNOLOGI INFORMASI DAN KOMUNIKASI  
GOWA  
2018

# MODUL PENGEMBANGAN KEPROFESIAN BERKELANJUTAN BERBASIS KOMPETENSI

## BIDANG KEAHLIAN REKAYASA PERANGKAT LUNAK (RPL)

### PROFESIONAL:

Mengimplementasikan Pemrograman Terstruktur

### Penulis:

Alun Sujjadah, S.Kom., MT. ([alun.sujjada@gmail.com](mailto:alun.sujjada@gmail.com))

### Penelaah:

Novi Nurlaela, S.Kom. ([java.alex@gmail.com](mailto:java.alex@gmail.com))

Yohan Pribadi ([Yue4hn@gmail.com](mailto:Yue4hn@gmail.com))

### Desain Grafis dan Ilustrasi:

Azhar Waliullah Syafir ([azhar.syafir@gmail.com](mailto:azhar.syafir@gmail.com))

*Copyright* © 2018

Direktorat Pembinaan Guru Pendidikan Dasar Direktorat Jenderal Guru dan Tenaga  
Kependidikan Kementerian Pendidikan dan Kebudayaan

Hak Cipta Dilindungi Undang-Undang Dilarang mengcopy sebagian atau keseluruhan isi  
buku ini untuk kepentingan komersial tanpa izin tertulis dari Kementerian Pendidikan  
Kebudayaan.

## KATA PENGANTAR

Modul pengembangan keprofesian berkelanjutan (PKB) berbasis kompetensi merupakan salah satu media pembelajaran yang dapat digunakan sebagai media transformasi pengetahuan, keterampilan dan sikap kerja kepada peserta pelatihan untuk mencapai kompetensi tertentu berdasarkan program pelatihan yang mengacu kepada Standar Kompetensi.

Modul pelatihan ini berorientasi kepada pelatihan berbasis kompetensi (*Competence Based Training*) diformulasikan menjadi 3 (tiga) buku, yaitu Buku Informasi, Buku Kerja dan Buku Penilaian sebagai satu kesatuan yang tidak terpisahkan dalam penggunaannya sebagai referensi dalam media pembelajaran bagi peserta pelatihan dan instruktur, agar pelaksanaan pelatihan dapat dilakukan secara efektif dan efisien. Untuk memenuhi kebutuhan pelatihan berbasis kompetensi tersebut, maka disusunlah modul pelatihan berbasis kompetensi dengan judul "**Mengimplementasikan Pemrograman Terstruktur**".

Kami menyadari bahwa modul yang kami susun ini masih jauh dari sempurna. Oleh karena itu, kami sangat mengharapkan saran dan masukan untuk perbaikan agar tujuan dari penyusunan modul ini menjadi lebih efektif.

Demikian kami sampaikan, semoga Tuhan YME memberikan tuntunan kepada kita dalam melakukan berbagai upaya perbaikan dalam menunjang proses pelaksanaan pembelajaran di lingkungan direktorat guru dan tenaga kependidikan.

Gowa, April 2018  
Kepala LPPPTK KPTK,

Prof. Dr. Irwan, M.Pd

## DAFTAR ISI

KATA PENGANTAR.....	3
DAFTAR ISI.....	4
ACUAN STANDAR KOMPETENSI KERJA DAN SILABUS DIKLAT .....	5
A. Acuan Standar Kompetensi Kerja .....	5
B. KEMAMPUAN YANG HARUS DIMILIKI SEBELUMNYA.....	8
C. Silabus Diklat.....	9
LAMPIRAN .....	14
1. BUKU INFORMASI .....	14
2. BUKU KERJA .....	14
3. BUKU PENILAIAN.....	14

## ACUAN STANDAR KOMPETENSI KERJA DAN SILABUS DIKLAT

### A. Acuan Standar Kompetensi Kerja

Materi modul pelatihan ini mengacu pada unit kompetensi terkait yang disalin dari Standar Kompetensi Kerja Sub-golongan Pemrograman dengan uraian sebagai berikut:

**Kode Unit** : **J.620100.017.02**

**Judul Unit** : **Mengimplementasikan Pemrograman Terstruktur**

**Deskripsi Unit** : Unit kompetensi ini berhubungan dengan sikap, pengetahuan, dan keterampilan yang dibutuhkan untuk membuat program terstruktur atau prosedural.

ELEMEN KOMPETENSI		KRITERIA UNJUK KERJA	
1.	Menggunakan tipe data dan control program	1.1	Tipe data yang sesuai standar ditentukan.
		1.2	Syntax program yang dikuasai digunakan sesuai standar.
		1.3	Struktur kontrol program yang dikuasai digunakan sesuai standar.
2.	Membuat program sederhana	2.1	Program baca tulis untuk memasukkan data dari keyboard dan menampilkan ke layar monitor termasuk variasinya sesuai standar masukan/keluaran telah dibuat.
		2.2	Struktur kontrol percabangan dan pengulangan dalam membuat program telah digunakan.
3.	Membuat program menggunakan prosedur dan fungsi	3.1	Program dengan menggunakan prosedur dibuat sesuai aturan penulisan program.

		3.2	Program dengan menggunakan fungsi dibuat sesuai aturan penulisan program.
		3.3	Program dengan menggunakan prosedur dan fungsi secara bersamaan dibuat sesuai aturan penulisan program.
		3.4	Keterangan untuk setiap prosedur dan fungsi telah diberikan.
4.	Membuat program menggunakan array	4.1	Dimensi array telah ditentukan.
		4.2	Tipe data array telah ditentukan.
		4.3	Panjang array telah ditentukan.
		4.4	Pengurutan array telah digunakan.
5.	Membuat program untuk akses file	5.1	Program untuk menulis data dalam media penyimpan telah dibuat.
		5.2	Program untuk membaca data dari media penyimpan telah dibuat.
6.	Mengkompilasi Program	6.1	Kesalahan program telah dikoreksi.
		6.2	Kesalahan syntax dalam program telah dibebaskan.

## BATASAN VARIABEL

### 1. Konteks Variabel:

- 1.1 Unit kompetensi ini berhubungan dengan penerapan bahasa pemrograman terstruktur atau prosedural
- 1.2 Bahasa Pemrograman yang digunakan berupa bahasa pemrograman berparadigma terstruktur atau prosedural.

### 2. Peralatan dan perlengkapan

#### 2.1 Peralatan

- 2.1.1 Perangkat lunak pemrograman yang sesuai
- 2.1.2 Perangkat keras yang sesuai

#### 2.2 Perlengkapan

- 2.2.1 Petunjuk teknis bahasa pemrograman terkait

#### 2.2.2 Algoritma program

### 3. Peraturan yang diperlukan

- 3.1 Undang-Undang Nomor 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik
- 3.2 Undang-Undang Nomor 14 Tahun 2008 tentang Keterbukaan Informasi Publik

### 4. Norma dan standar

#### 4.1 Norma

- 4.1.1 Legalitas dan etika yang terkait dengan profesi bidang teknologi informasi

#### 4.2 Standar

- 4.2.1 SNI ISO/IEC 20000-1:2009 Teknologi informasi Manajemen layanan Bagian 1: Spesifikasi
- 4.2.2 SNI ISO/IEC 20000-2:2009 Teknologi informasi Manajemen layanan Bagian 2: Aturan Praktik
- 4.2.3 Standar Pemrograman terstruktur atau prosedural yang ada.

## **PANDUAN PENILAIAN**

### 1. Konteks penilaian

- 1.1 Konteks penilaian merupakan aspek dalam penilaian yang sangat berpengaruh atas tercapainya kompetensi ini terkait dengan menerapkan bahasa pemrograman terstruktur sesuai dengan skema sertifikasi.
- 1.2 Penilaian dapat dilakukan dengan cara demonstrasi/praktik, baik di TUK dan/atau di tempat kerja.
- 1.3 Penilaian unit ini mencakup pengetahuan, keterampilan dan sikap kerja yang dipersyaratkan.

### 2. Persyaratan kompetensi

- 2.1 J.620100.022.02: Mengimplementasikan Algoritma Pemrograman

### 3. Pengetahuan dan keterampilan yang dibutuhkan

#### 3.1 Pengetahuan

- 3.1.1 Algoritma pemrograman
- 3.1.2 Struktur data
- 3.1.3 Spesifikasi program
- 3.1.4 Membuat program dengan bahasa terstruktur

#### 3.2 Keterampilan

(Tidak ada)

4. Sikap kerja yang diperlukan

4.1 Kerjasama

4.2 Komunikatif

4.3 Analitik

5. Aspek kritis

5.1 Ketepatan penggunaan struktur kontrol percabangan dan pengulangan dalam pembuatan program sederhana yang dapat dieksekusi.

**B. KEMAMPUAN YANG HARUS DIMILIKI SEBELUMNYA**

(Tidak ada)

### C. Silabus Diklat

Judul Unit Kompetensi : Mengimplementasikan Pemrograman Terstruktur

Kode Unit Kompetensi : J.620100.017.02

Deskripsi Unit Kompetensi : Unit kompetensi ini berhubungan dengan sikap, pengetahuan, dan keterampilan yang dibutuhkan untuk membuat program terstruktur atau prosedural.

Perkiraan Waktu Pelatihan :

Tabel Silabus Unit Kompetensi :

Elemen Kompetensi	Kriteria Unjuk Kerja	Indikator Unjuk Kerja	Materi Diklat			Perkiraan Waktu Diklat (JP)	
			Pengetahuan (P)	Keterampilan (K)	Sikap (S)	P	K
1. Menggunakan tipe data dan control program	1.1 Tipe data yang sesuai standar ditentukan.	<ul style="list-style-type: none"> <li>✓ Mampu memahami tipe data</li> <li>✓ Mampu menentukan tipe data yang sesuai standar</li> <li>✓ Harus benar, efisien dan sesuai dengan kaidah bahasa pemrograman</li> </ul>	Jenis-jenis tipe data	<ul style="list-style-type: none"> <li>✓ Penentuan tipe data</li> <li>✓ Deklarasi tipe data</li> </ul>		10'	15'
	1.2 Syntax program yang dikuasai digunakan sesuai standar.	<ul style="list-style-type: none"> <li>✓ Mampu memahami syntax program.</li> <li>✓ Mampu menggunakan syntax program sesuai standar.</li> <li>✓ Harus benar dan sesuai dengan kaidah pemrograman</li> </ul>	Aturan syntax Bahasa pemrograman.	Penggunaan aturan syntax dalam membuat program		10'	15'
	1.3 Struktur kontrol program yang dikuasai	<ul style="list-style-type: none"> <li>✓ Dapat memahami struktur kontrol</li> </ul>	Jenis-jenis struktur kontrol	Penggunaan struktur control dalam membuat program		30'	60'

	digunakan sesuai standar.	<ul style="list-style-type: none"> <li>✓ Mampu menggunakan struktur kontrol program yang dikuasai sesuai standar.</li> <li>✓ Cermat, teliti dan efisien</li> </ul>					
2. Membuat program sederhana	2.1 Program baca tulis untuk memasukkan data dari keyboard dan menampilkan ke layar monitor termasuk variasinya sesuai standar masukan/keluaran telah dibuat.	<ul style="list-style-type: none"> <li>✓ Mampu memahami proses input/output dari keyboard ke layar monitor</li> <li>✓ Mampu membuat program baca tulis untuk memasukkan data dari keyboard dan menampilkan ke layar monitor termasuk variasinya sesuai standar masukan/keluaran</li> <li>✓ Harus efisien</li> </ul>	Konsep input/output	Pembuatan program baca tulis dari keyboard ke layer monitor		15'	60'
	2.2 Struktur kontrol percabangan dan pengulangan dalam membuat program telah digunakan.	<ul style="list-style-type: none"> <li>✓ Mampu menjelaskan jenis-jenis kontrol percabangan dan pengulangan beserta kegunaannya</li> <li>✓ Mampu menggunakan struktur kontrol percabangan dan pengulangan dalam membuat program</li> <li>✓ Cermat, tekun dan teliti.</li> </ul>	<ul style="list-style-type: none"> <li>✓ Jenis-jenis kontrol percabangan dan pengulangan.</li> <li>✓ Fungsi dari masing-masing struktur kontrol</li> </ul>	Penggunaan <i>kontrol</i> percabangan dan pengulangan dalam membuat program		15'	60'
3. Membuat program menggunakan	3.1 Program dengan menggunakan prosedur dibuat	<ul style="list-style-type: none"> <li>✓ Dapat memahami konsep tentang prosedur.</li> </ul>	Konsep prosedur	Pembuatan program dengan menggunakan		15'	30'

prosedur dan fungsi	sesuai aturan penulisan program.	<ul style="list-style-type: none"> <li>✓ Mampu membuat program dengan menggunakan prosedur sesuai aturan penulisan program.</li> <li>✓ Tekun dan teliti</li> </ul>		prosedur yang sesuai aturan penulisan			
	3.2 Program dengan menggunakan fungsi dibuat sesuai aturan penulisan program.	<ul style="list-style-type: none"> <li>✓ Dapat memahami konsep tentang fungsi</li> <li>✓ Mampu membuat program dengan menggunakan fungsi sesuai aturan penulisan program.</li> <li>✓ Tekun dan teliti</li> </ul>	Konsep fungsi	Pembuatan program dengan menggunakan fungsi yang sesuai aturan penulisan		15'	30'
	3.3 Program dengan menggunakan prosedur dan fungsi secara bersamaan dibuat sesuai aturan penulisan program.	<ul style="list-style-type: none"> <li>✓ Mampu mengidentifikasi prosedur dan fungsi</li> <li>✓ Mampu membuat program dengan menggunakan prosedur dan fungsi secara bersamaan sesuai aturan penulisan program.</li> <li>✓ Tekun dan teliti</li> </ul>	Identifikasi bagian fungsi dan prosedur pada sebuah program	Pembuatan program dengan menggunakan prosedur dan fungsi sesuai aturan penulisan program		15'	30'
	3.4 Keterangan untuk setiap prosedur dan fungsi telah diberikan.	<ul style="list-style-type: none"> <li>✓ Mampu memahami fungsi keterangan dalam program</li> <li>✓ Mampu memberikan keterangan dalam setiap prosedur dan fungsi.</li> </ul>	Kegunaan pemberian keterangan pada program	Pemberian keterangan dalam setiap prosedur dan fungsi		5'	10'
4. Membuat program	4.1 Dimensi array telah ditentukan.	<ul style="list-style-type: none"> <li>✓ Mampu memahami konsep dimensi array</li> </ul>	Konsep dimensi array	Penentuan dimensi array		10'	15'

menggunakan array		<ul style="list-style-type: none"> <li>✓ Mampu menentukan dimensi array</li> <li>✓ Sesuai kaidah bahasa pemrograman</li> </ul>					
	4.2 Tipe data array telah ditentukan.	<ul style="list-style-type: none"> <li>✓ Mampu memahami tipe data array</li> <li>✓ Mampu menentukan tipe data array</li> <li>✓ Sesuai kaidah bahasa pemrograman</li> </ul>	Tipe data Array	Penentuan tipe data array		15'	15'
	4.3 Panjang array telah ditentukan.	Mampu menentukan Panjang array		Penentuan Panjang array			5'
	4.4 Pengurutan array telah digunakan.	<ul style="list-style-type: none"> <li>✓ Mampu memahami konsep pengurutan array</li> <li>✓ Mampu menggunakan pengurutan array</li> <li>✓ Sesuai kaidah bahasa pemrograman</li> </ul>	Jenis-jenis pengurutan array	Penggunaan pengurutan array		30'	60'
5. Membuat program untuk akses file	5.1 Program untuk menulis data dalam media penyimpan telah dibuat.	<ul style="list-style-type: none"> <li>✓ Mampu memahami konsep penyimpanan data.</li> <li>✓ Mampu membuat program untuk menulis data dalam media penyimpanan</li> <li>✓ Harus sesuai syntax Bahasa pemrograman</li> </ul>	Konsep penyimpanan data dalam media	Pembuatan program untuk menulis data dalam media penyimpanan		30'	45'
	5.2 Program untuk membaca data dari media penyimpan telah dibuat.	<ul style="list-style-type: none"> <li>✓ Mampu memahami konsep pembacaan data.</li> <li>✓ Mampu membuat program untuk membaca data dari media penyimpanan</li> <li>✓ Harus sesuai syntax Bahasa pemrograman</li> </ul>	Konsep pembacaan data dari media penyimpanan	Pembuatan program untuk membaca data dari media penyimpanan		30'	45'

6. Mengkompilasi Program	6.1 Kesalahan program telah dikoreksi.	<ul style="list-style-type: none"> <li>✓ Dapat memahami jenis-jenis kesalahan dalam program</li> <li>✓ Mampu mengoreksi kesalahan</li> <li>✓ Cermat, tekun dan teliti</li> </ul>	Jenis-jenis kesalahan dalam program	Koreksi kesalahan		15'	30'
	6.2 Kesalahan syntax dalam program telah dibebaskan.	<ul style="list-style-type: none"> <li>✓ Aturan syntax Bahasa pemrograman</li> <li>✓ Mampu menghasilkan program yang bebas dari kesalahan sintaks</li> <li>✓ Cermat, tekun dan teliti</li> </ul>	Syntax Bahasa pemrograman	Penanggulangan kesalahan syntax		15'	30'

## LAMPIRAN

1. BUKU INFORMASI
2. BUKU KERJA
3. BUKU PENILAIAN



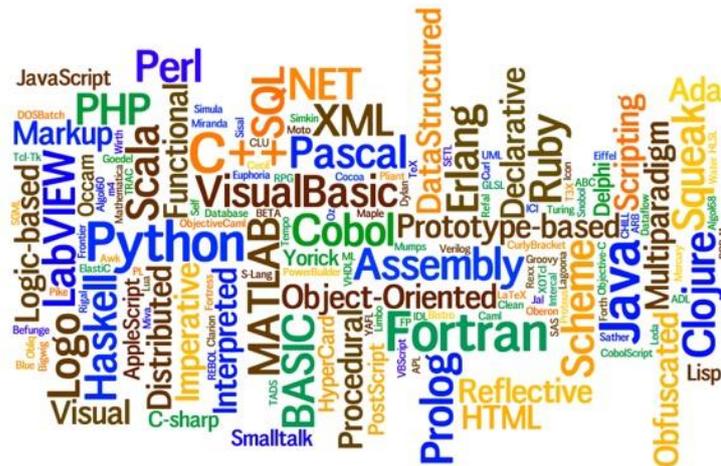
**1**

**LAMPIRAN  
BUKU INFORMASI**



## BUKU INFORMASI

# MENGIMPLEMENTASIKAN PEMROGRAMAN TERSTRUKTUR J.620100.017.02



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN R.I.  
**DIREKTORAT JENDERAL GURU DAN TENAGA KEPENDIDIKAN**  
LEMBAGA PENGEMBANGAN DAN PEMBERDAYAAN PENDIDIK DAN TENAGA KEPENDIDIKAN BIDANG  
KELAUTAN, PERIKANAN, TEKNOLOGI INFORMASI DAN KOMUNIKASI  
GOWA

## DAFTAR ISI

DAFTAR ISI .....	2
DAFTAR SCRIPT.....	5
DAFTAR GAMBAR.....	6
DAFTAR TABEL .....	7
BAB I PENDAHULUAN .....	8
A. TUJUAN UMUM MENGIMPLEMENTASIKAN PEMROGRAMAN TERSTRUKTUR .....	8
B. TUJUAN KHUSUS MENGIMPLEMENTASIKAN PEMROGRAMAN TERSTRUKTUR.....	8
BAB II MENGGUNAKAN TIPE DATA DAN CONTROL PROGRAM .....	9
A. Pengetahuan yang Diperlukan dalam Menggunakan Tipe Data dan Control Program .....	9
1. Variabel JavaScript .....	9
2. Tipe Data .....	9
3. Tipe Data Numerik .....	9
4. Control Program .....	11
5. Instalasi Program .....	17
6. Implementasi Program.....	20
B. Keterampilan yang Diperlukan dalam Menggunakan Tipe Data dan Control Program .....	22
C. Sikap Kerja yang Diperlukan dalam Menggunakan Tipe Data dan Control Program .....	23
BAB III MEMBUAT PROGRAM SEDERHANA .....	24
A. Pengetahuan yang Diperlukan dalam dalam Membuat Program Sederhana .....	24
1. Input Output Program .....	24
2. Program Menggunakan JavaScript.....	24
3. Program dengan Struktur Kontrol .....	26
B. Keterampilan yang Diperlukan dalam Membuat Program Sederhana....	27
C. Sikap Kerja yang Diperlukan dalam Membuat Program Sederhana .....	27

<b>BAB IV MEMBUAT PROGRAM MENGGUNAKAN PROSEDUR DAN FUNGSI .....</b>	<b>29</b>
A. Pengetahuan yang Diperlukan dalam dalam Membuat Program Menggunakan Prosedur dan Fungsi .....	29
1. Prosedur dan Fungsi .....	29
2. Pembuatan Fungsi pada JavaScript .....	30
3. Implementasi Fungsi dan Prosedur .....	31
B. Keterampilan yang Diperlukan dalam Membuat Program Menggunakan Prosedur dan Fungsi .....	36
C. Sikap Kerja yang Diperlukan dalam Membuat Program Menggunakan Prosedur dan Fungsi.....	36
<b>BAB V MEMBUAT PROGRAM MENGGUNAKAN ARRAY .....</b>	<b>37</b>
A. Pengetahuan yang Diperlukan dalam dalam Membuat Program Menggunakan Array .....	37
1. Pengertian Array .....	37
2. Array pada JavaScript.....	39
3. Program Sorting Menggunakan Array .....	42
B. Keterampilan yang Diperlukan dalam Membuat Program Menggunakan Array .....	46
C. Sikap Kerja yang Diperlukan dalam Membuat Program Menggunakan Array .....	46
<b>BAB VI MEMBUAT PROGRAM UNTUK AKSES FILE .....</b>	<b>48</b>
A. Pengetahuan yang Diperlukan dalam dalam Membuat Program Untuk Akses File .....	48
1. Akses File .....	48
2. Akses File Menggunakan JavaScript.....	48
B. Keterampilan yang Diperlukan dalam Membuat Program Untuk Akses File.....	52
C. Sikap Kerja yang Diperlukan dalam Membuat Program Untuk Mengakses File.....	52
<b>BAB VII MENGKOMPILASI PROGRAM .....</b>	<b>53</b>
A. Pengetahuan yang Diperlukan dalam Mengkompilasi Program.....	53
1. Perbaikan Kesalahan .....	53

2. Program Debugging .....	56
B. Keterampilan yang Diperlukan dalam Mengkompilasi Program .....	61
C. Sikap Kerja yang Diperlukan dalam Mengkompilasi Program .....	61
DAFTAR PUSTAKA .....	62
A. Buku Referensi .....	62
B. Referensi Lainnya .....	62
DAFTAR ALAT DAN BAHAN .....	63
A. DAFTAR PERALATAN/MESIN .....	63
B. DAFTAR BAHAN .....	63
DAFTAR PENYUSUN .....	64

## DAFTAR SCRIPT

Script 2.1 Penggunaan if .....	11
Script 2.2 Penggunaan if else.....	12
Script 2.3 Penggunaan else if.....	13
Script 2.4 Penggunaan <i>switch</i> .....	14
Script 2.5 Penggunaan perulangan for.....	16
Script 2.6 Penggunaan <i>while</i> .....	16
Script 2.7 Penggunaan <i>do while</i> .....	17
Script 2.8 Contoh program JavaScript.....	21
Script 3.1 Penyisipan <i>file</i> JavaScript.....	25
Script 3.2 Program pengkatagorian bilangan prima.....	27
Script 4.1 Fungsi pada JavaScript .....	29
Script 4.2 Expresi fungsi .....	30
Script 4.3 <i>Input Form</i> .....	32
Script 4.4 Fungsi cek_barang .....	34
Script 4.5 Pemberian komentar pada program.....	35
Script 5.1 Deklarasi array 1 (satu) dimensi.....	39
Script 5.2 Deklarasi array 2 (dua) dimensi.....	39
Script 5.3 Pemberian nilai array dari variabel lain .....	40
Script 5.4 Pemberian nilai pada variabel array .....	40
Script 5.5 Penggunaan <i>length of property</i> .....	41
Script 5.6 Menghapus array .....	41
Script 5.7 Penggunaan perulangan untuk mengakses array.....	42
Script 5.8 <i>Selection Sort</i> .....	45
Script 5.9 Sortir dengan <i>output</i> tiap langkah .....	46
Script 6.1 Membuka <i>file</i> gambar .....	49
Script 7.1 Contoh program sederhana .....	54

## DAFTAR GAMBAR

Gambar 2.1 Halaman Sublime Text .....	18
Gambar 2.2 Proses instalasi Sublime Text.....	18
Gambar 2.3 Pemilihan lokasi instalasi .....	19
Gambar 2.4 Pemilihan fitur tambahan .....	19
Gambar 2.5 Proses Memulai Instalasi .....	20
Gambar 2.6 <i>Menu File</i> .....	21
Gambar 2.7 Penyimpanan <i>File</i> .....	22
Gambar 2.8 Hasil <i>output</i> program .....	22
Gambar 3.1 I/O Proses.....	24
Gambar 3.2 Hasil Program.....	26
Gambar 3.3 Hasil <i>output</i> program bilangan prima .....	28
Gambar 4.1 Hasil <i>output program</i> .....	36
Gambar 5.1 Ilustrasi variabel array .....	38
Gambar 5.2 Ilustrasi array 2 (dua) dimensi.....	38
Gambar 5.3 ilustrasi <i>selection sort</i> .....	43
Gambar 5.4 Ilustrasi pertukaran data .....	44
Gambar 6.1 Tampilan hasil program.....	49
Gambar 6.2 <i>File Dialog</i> pemilihan gambar.....	50
Gambar 7.1 Hasil <i>output</i> program pembagian .....	54
Gambar 7.2 <i>output</i> program pembagian dengan bilangan 0.....	55
Gambar 7.3 Proses <i>debugging</i> .....	59

## DAFTAR TABEL

Tabel 2.1 Penggunaan tipe data dan variabel.....	10
Tabel 6.1 API Filereader .....	47

## **BAB I PENDAHULUAN**

### **A. TUJUAN UMUM MENGIMPLEMENTASIKAN PEMROGRAMAN TERSTRUKTUR**

Setelah mempelajari modul ini peserta diharapkan mampu Mengimplementasikan Pemrograman Terstruktur

### **B. TUJUAN KHUSUS MENGIMPLEMENTASIKAN PEMROGRAMAN TERSTRUKTUR**

Adapun tujuan mempelajari unit kompetensi melalui buku informasi Mengimplementasikan Pemrograman Terstruktur ini guna memfasilitasi peserta sehingga pada akhir diklat diharapkan memiliki kemampuan sebagai berikut:

1. Menggunakan tipe data dan *control* program
2. Membuat program sederhana
3. Membuat program menggunakan prosedur dan fungsi
4. Membuat program menggunakan *array*
5. Mengkompilasi program untuk akses *file*

## **BAB II**

### **MENGGUNAKAN TIPE DATA DAN CONTROL PROGRAM**

#### **A. PENGETAHUAN YANG DIPERLUKAN DALAM MENGGUNAKAN TIPE DATA DAN CONTROL PROGRAM**

##### **1. Variabel JavaScript**

Variabel merupakan tempat yang digunakan untuk menyimpan suatu nilai atau data secara sementara pada ruang memori yang terdapat dalam komputer. Sementara berarti nilainya dapat diubah sesuai keperluan, dan ketika program dihentikan maka nilai dari variabel akan hilang. JavaScript mengizinkan variabel yang dideklarasikan dapat diisi dengan nilai apapun, berbeda dengan bahasa pemrograman lain seperti C, C++, Java, Visual Basic.net, Delphi dan lain-lain, dimana pemberian nilai pada variabel harus sesuai dengan tipe data yang dideklarasikan. Aturan penamaan variabel pada JavaScript adalah:

- a. Harus diawali dengan huruf atau garis bawah (*underscore*).
- b. Tidak boleh menggunakan spasi.
- c. *Case sensitive* atau membedakan antara huruf besar dan huruf kecil.
- d. Tidak diperbolehkan menggunakan *reserve words* yaitu kata-kata yang telah digunakan oleh JavaScript, misal: exit, for, if, switch dan lain-lain.

##### **2. Tipe Data**

Pada pemrograman, tipe data adalah konsep yang sangat penting. Untuk dapat mengoperasikan variabel, maka dibutuhkan pemahaman tentang tipe data, karena tanpa adanya tipe data, maka komputer tidak dapat memecahkan permasalahan seperti `var a = 100 + "Nilai String"`. JavaScript tidak mempunyai tipe data secara eksplisit seperti bahasa pemrograman yang lain. Tipe data yang digunakan sesuai dengan nilai yang diberikan terhadap sebuah variabel. Meskipun JavaScript tidak memiliki tipe data eksplisit, akan tetapi memiliki tipe data implisit. Terdapat beberapa macam tipe data implisit yaitu:

- a. Numerik, seperti 1000000, 1234, 567.890, 3.14 dan sebagainya.
- b. String, seperti "Apa kabar pagi ini?", "Jl. Bendungan Sutami I/387"

- c. Boolean bernilai true dan false
- d. Array, sekumpulan nilai yang mempunyai indeks
- e. Null, variabel yang tidak diinisialisasi.
- f. Object

Beberapa contoh representasi tipe data menggunakan bahasa pemrograman JavaScript seperti pada tabel 2.1

### 3. Tipe Data Numerik

Pada JavaScript terdapat 2(dua) tipe data numerik, yaitu bilangan bulat (*integer*) dan bilangan pecahan (*float/real*). Bilangan bulat dapat direpresentasikan dalam bentuk basis desimal, octal dan heksadesimal.

Tabel 2. 2 Penggunaan tipe data dan variabel

CONTOH	KETERANGAN
<code>var nama;</code>	Variabel nama tanpa nilai awal
<code>var nama = "Edi Rahman";</code>	Variabel nama dengan nilai awal
<code>var tahun = 1985;</code>	Variabel tahun dengan tipe data integer
<code>var status = true;</code>	Variabel status dengan tipe data boolean
<code>var nilai = 85.5;</code>	Variabel nilai dengan tipe data float
<code>var bil = 1.23456E+3;</code>	Variabel bil dengan penulisan notasi ilmiah
<code>var nilai_oktal = 027;</code>	Variabel nilai dengan nilai oktal
<code>Var nilai_hexa = 0x2F;</code>	Variabel nilai dengan nilai hexadesimal
<code>var nama = {fname:"Adi",lname:"Wicaksono"};</code>	Variabel berbentuk objek
<code>var siswa = ["Andi","Budi","Cica","Deny"] ;</code>	Variabel array



#### DEKLARASI VARIABEL JAVASCRIPT

Variabel yang mempunyai nilai String dapat diisi dengan menggunakan *double quotes* (") atau *single quotes* (')

#### 4. Control Program

Pada semua bahasa pemrograman, dikenal istilah *control program* yaitu sebuah blok kode program yang digunakan untuk mengatur jalannya program sesuai rancangan yang telah disusun. Terdapat beberapa *control program* yang umum digunakan pada pemrograman, yaitu kondisi percabangan dan kondisi perulangan.

##### a. Kondisi Percabangan

Kondisi ini dapat diartikan sebagai pernyataan bersyarat yang digunakan untuk melakukan tindakan berbeda berdasarkan kondisi yang berbeda. Terdapat kondisi yang harus terpenuhi terlebih dahulu sehingga pernyataannya akan dijalankan. sintaks dari kondisi percabangan tersebut adalah:

##### i. *if*

##### a) *if*

Sintaksnya adalah:

```
if ( condition ) {  
    block of code to be executed if the condition is true  
}
```

Contoh: seorang ibu belanja di toko X, toko tersebut mempunyai promo pada bulan februari yaitu, untuk semua konsumen jika total belanja melebihi Rp 1.000.000,- maka akan mendapatkan potongan sebesar 10% dari total belanja. Bentuk kode programnya seperti *script 2.1*

```
<script>  
    var total_belanja = 1500000;  
    var discount;  
    if(total_belanja >1000000){  
        discount = 0.10;  
    }  
  
    document.write(total_belanja-(total_belanja * discount))  
</script>
```

Script 2. 3 Penggunaan if

Jika program tersebut dijalankan maka akan tampil pada *browser* nilai **1350000**

b) *else*

Sintaksnya adalah:

```
if ( condition ) {  
    block of code to be executed if the condition is true  
} else {  
    block of code to be executed if the condition is false  
}
```

Contoh : Terdapat sebuah program yang dapat melakukan pengecekan bilangan tersebut termasuk dalam kategori bilangan ganjil atau genap. Bentuk kode programnya seperti *script 2.2*

```
<script>  
var bilangan = 101;  
  
if(bilangan % 2==0){  
    document.write(bilangan + " adalah bilangan genap");  
}  
else{  
    document.write(bilangan + " adalah bilangan ganjil");  
}  
  
</script>
```

Script 2. 4 Penggunaan if else

Jika program tersebut dijalankan maka akan tampil pada *browser* sebuah kalimat **101 adalah bilangan ganjil.**

c) *else if*

Sintaksnya adalah:

```
if ( condition1 ) {  
    block of code to be executed if condition1 is true  
} else if ( condition2 ) {  
    block of code to be executed if the condition1 is false and  
condition2 is true  
} else {  
    block of code to be executed if the condition1 is false and  
condition2 is false  
}
```

Contoh: Siswa SMKN 1 Setia Janji ditugaskan untuk membuat program yang berfungsi untuk mengelompokkan usia seseorang termasuk dalam kategori Balita (0-4) tahun, Anak-anak (5-12) tahun, Remaja (13-17) tahun. Bentuk kode programnya seperti *script 2.3*

```
<script>
  var umur = 13;

  if(umur >=0 && umur <=4){
    document.write("Kategori Balita");
  }
  else if(umur >=5 && umur <=12){
    document.write("Kategori Anak-anak");
  }
  else if(umur >=13 && umur <=17){
    document.write("Kategori Remaja");
  }
</script>
```

Script 2. 5 Penggunaan else if

Jika program tersebut dijalankan maka akan tampil pada *browser* sebuah keterangan **Kategori Remaja**.

## ii. *switch*

Penggunaan *switch* adalah sebagai bentuk alternatif dari *if*. *Switch* sebaiknya digunakan jika terdapat banyak blok kode pernyataan yang akan dieksekusi. Cara kerja *switch* adalah ekspresi dievaluasi hanya sekali saja, nilai yang terdapat dalam ekspresi akan dibandingkan dengan nilai pada masing-masing *case*, jika terdapat kesamaan nilai, maka pernyataannya akan dieksekusi sampai batas *break*. Sintaksnya adalah:

```
switch(expression) {  
  case n:  
    code block  
    break;  
  case n:  
    code block  
    break;  
  default:  
    code block  
}
```

Contoh: dengan menggunakan *method* `getDay()` yang terdapat pada JavaScript akan menghasilkan nilai balik yaitu antara 0-6 yang berarti Minggu = 0, Senin = 1, Selasa = 2, Rabu = 3, Kamis = 4, Jum'at = 5 dan Sabtu = 6. Bentuk kode programnya seperti *script* 2.4

```
<script>  
var hari;  
switch (new Date().getDay()) {  
  case 0:  
    hari = "Minggu";  
    break;  
  case 1:  
    hari = "Senin";  
    break;  
  case 2:  
    hari = "Selasa";  
    break;  
  case 3:  
    hari = "Rabu";  
    break;  
  case 4:  
    hari = "Kamis";  
    break;  
  case 5:  
    hari = "Jum'at";  
    break;  
  case 6:  
    hari = "Sabtu";  
}  
document.write("Hari ini adalah " + hari);  
</script>
```

*Script* 2.4 Penggunaan *switch*

Jika kode program tersebut dijalankan, maka akan tampil keterangan **Hari ini adalah** diikuti dengan nama hari yang diambil dari sistem komputer.

#### KONDISI PERCABANGAN

- Gunakan **if** untuk menentukan blok kode yang akan dieksekusi, jika kondisi yang ditentukan benar.
- Gunakan **else** untuk menentukan blok kode yang akan dieksekusi, jika kondisi yang sebelumnya salah.
- Gunakan **else if** untuk menentukan kondisi baru untuk menguji, jika kondisi pertama salah
- Gunakan **switch** untuk menentukan banyak blok kode alternatif yang akan dieksekusi.

#### b. Kondisi Perulangan

Pengulangan atau yang sering dikenal dengan istilah *looping* adalah kumpulan *set* instruksi khusus pada algoritma dan disemua bahasa pemrograman termasuk JavaScript yang digunakan untuk mengulang beberapa baris kode atau perintah sesuai dengan jumlah yang telah ditentukan. Tujuan penggunaannya adalah untuk mempermudah pengerjaan program dan untuk mengefisienkan baris-baris instruksi program. Dengan menggunakan pengulangan, maka instruksi program yang seharusnya ditulis dengan jumlah baris yang banyak dapat dipersingkat. Perulangan sering digunakan untuk mengakses data dalam bentuk *array*. Batas awal dan batas akhir indeks yang dimiliki oleh *array* digunakan sebagai variabel iterasi. Pemrograman JavaScript mengenal 3(tiga) jenis perulangan yaitu:

##### i. Perulangan *for*

Penggunaan perulangan *for* yaitu apabila sebuah permasalahan telah diketahui batas awal dan batas akhir. Sintaksnya adalah:

```
for (statement 1; statement 2; statement 3) {  
    code block to be executed  
}
```

Contoh: Terdapat sebuah program dimana *output* yang dihasilkan adalah angka 1 (satu) sampai dengan 100 (seratus). Jika permasalahan tersebut

dituliskan perbaris kode, maka akan menghabiskan 100(seratus) baris. Oleh karena itu untuk membuat program menjadi efisien, maka digunakan perulangan seperti pada *script 2.5*

```
<script>  
for (i = 1; i <= 100; i++) {  
    document.write("Looping yang ke " + i + "<br>");  
}  
</script>
```

*Script 2.5* Penggunaan perulangan for

## ii. Perulangan *while*

Perulangan *while* digunakan ketika permasalahan belum diketahui batas awal dan batas akhirnya, syarat kondisi perulangan berjalan dan berakhir adalah dengan pengecekan sebuah kondisi. Proses yang terjadi pada perulangan ini adalah dilakukan pengecekan kondisi terlebih dahulu, jika memenuhi persyaratan maka pernyataan akan dijalankan. Oleh karena itu perulangan ini minimal dikerjakan sebanyak 0(nol) kali. Sintaksnya adalah:

```
while (condition) {  
    code block to be executed  
}
```

Contoh: Program untuk menampilkan nilai variabel dari 0(nol) sampai 9(sembilan), kode programnya seperti *script 2.6*

```
<script>  
var i = 0;  
while (i < 10) {  
    document.write("Nilai i adalah :" + i + "<br>");  
    i++;  
}  
</script>
```

*Script 2.6* Perulangan *while*

### iii. Perulangan *do while*

Kebalikan dari perulangan *while*, cara kerja *do while* adalah akan mengeksekusi blok pernyataan terlebih dahulu, kemudian baru melakukan pengecekan kondisi. Jika kondisi terpenuhi maka proses akan melakukan perulangan selanjutnya. Oleh karena itu perulangan ini akan dijalankan minimal sebanyak 1(satu) kali. Sintaksnya adalah:

```
do {  
    code block to be executed  
}  
while (condition);
```

Contoh: Program untuk menampilkan nilai variabel dari 1(satu) sampai 10(sepuluh), kode programnya seperti *script 2.7*

```
<script>  
var i = 1;  
do {  
    document.write("Nilai i adalah : " + i + "<br>");  
    i++;  
}while (i <= 10)  
</script>
```

*Script 2.7 Penggunaan do while*

## 5. Instalasi Program

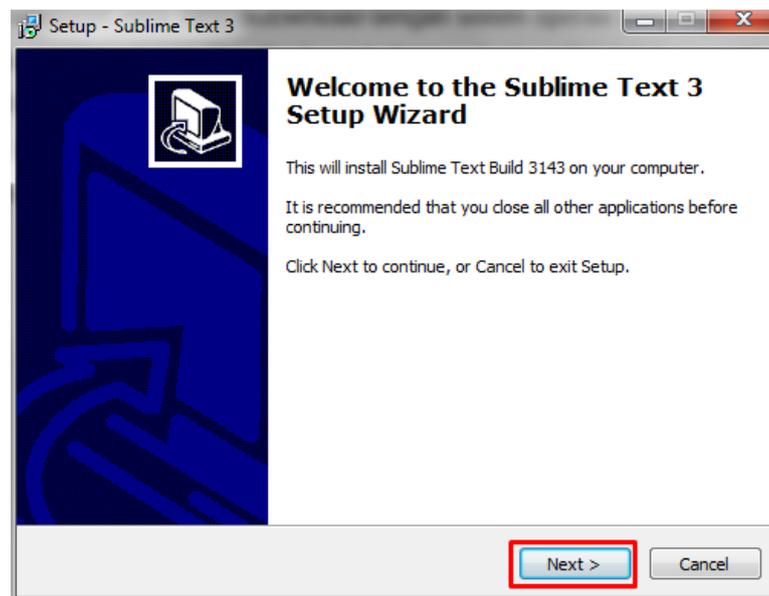
Penulisan kode program menggunakan bahasa JavaScript memerlukan *software* editor seperti Notepad++, Sublime text, Visual Code, Atom dan lain-lain. Penulisan JavaScript dapat disisipkan didalam *file* html menggunakan tag `<script>` atau terpisah. Adapun beberapa langkahnya adalah sebagai berikut:

- a. Unduh *software editor* sublime text di alamat url <https://www.sublimetext.com> klik pada menu *download* seperti pada gambar 2.1



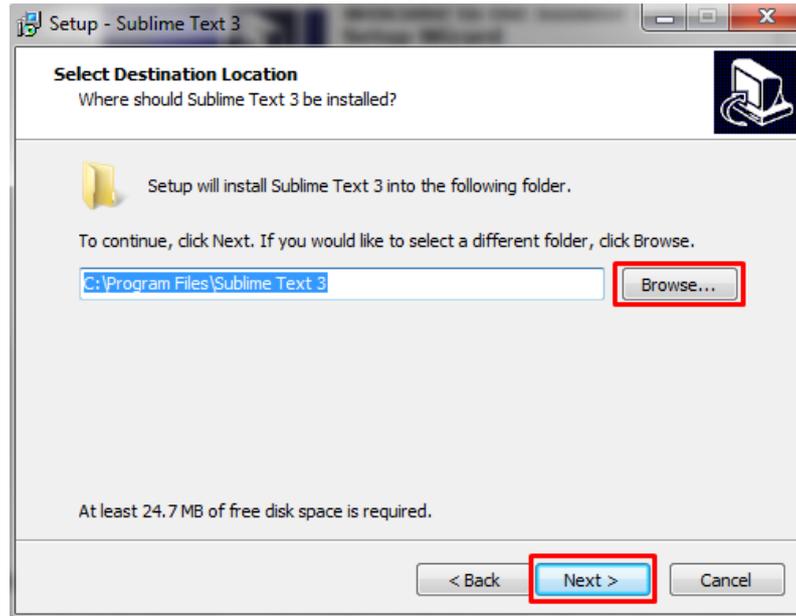
Gambar 2.1 Halaman Sublime Text

- b. Sesuaikan versi yang *didownload* dengan sistem operasi yang digunakan pada laptop atau komputer. Terdapat beberapa jenis, yaitu untuk versi Mac OS X, Windows 32 bit, Windows 64bit dan Linux.
- c. Lakukan instalasi dengan cara *double* klik pada *file installer* Sublime Text yang telah didownload dan klik tombol **next** seperti pada gambar 2.2



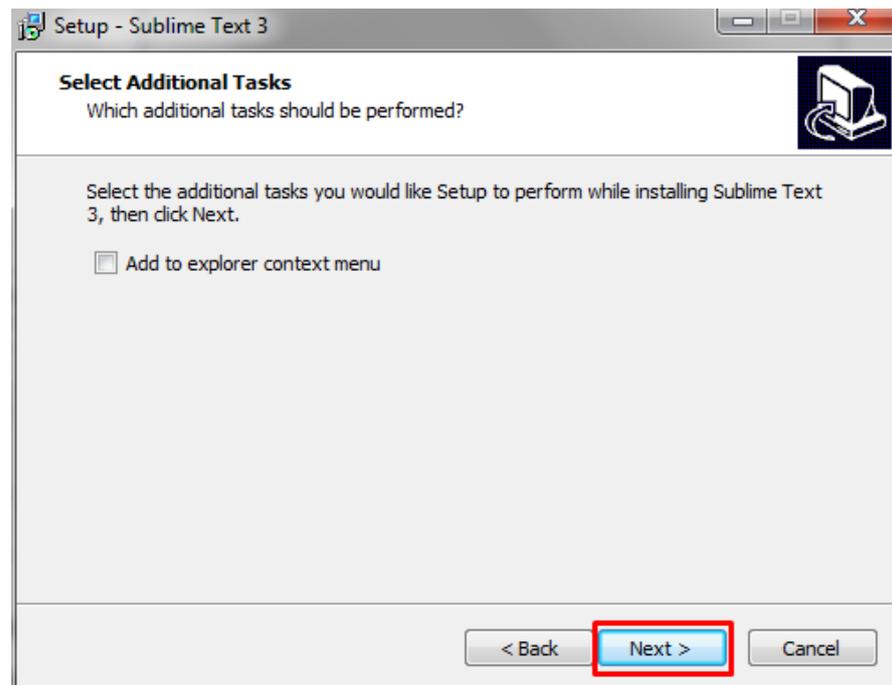
Gambar 2.2 Proses instalasi Sublime Text

- d. Proses selanjutnya adalah pemilihan direktori hasil instalasi, jika menggunakan lokasi *default* maka cukup klik tombol **next**, jika ingin meletakkan hasil instalasi sesuai keinginan maka klik tombol **browse** seperti pada gambar 2.3



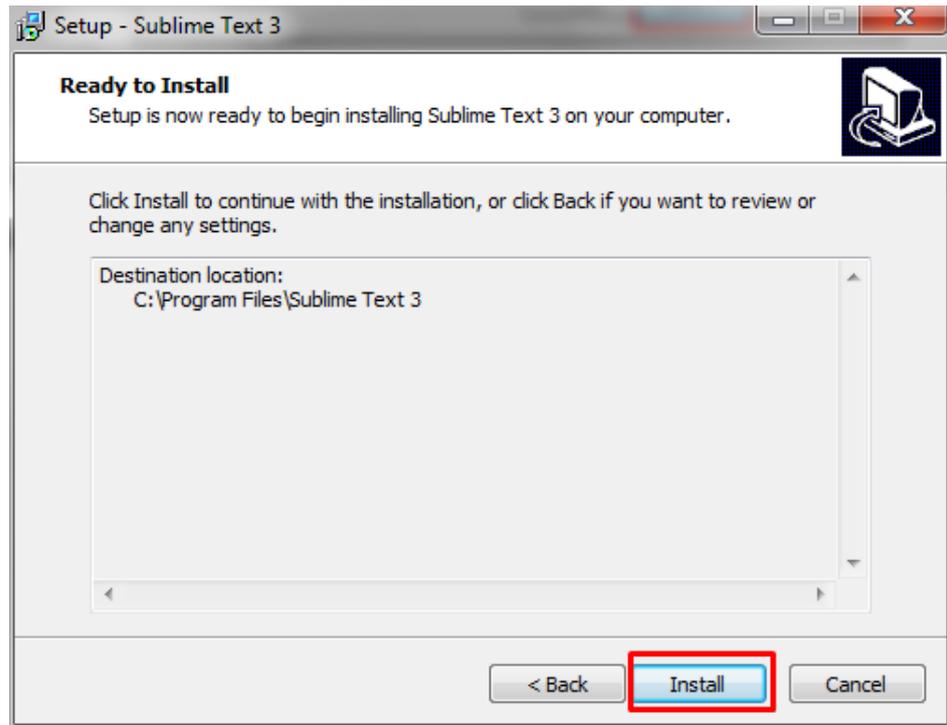
Gambar 2.3 Pemilihan lokasi instalasi

- e. Kemudian klik **next** untuk menambahkan pemilihan fitur tambahan, seperti pada gambar 2.4



Gambar 2.4 Pemilihan fitur tambahan

- f. Proses terakhir yaitu lakukan instalasi dengan klik tombol **Install** seperti pada gambar 2.5 dan tunggu sampai proses instalasi berhasil.



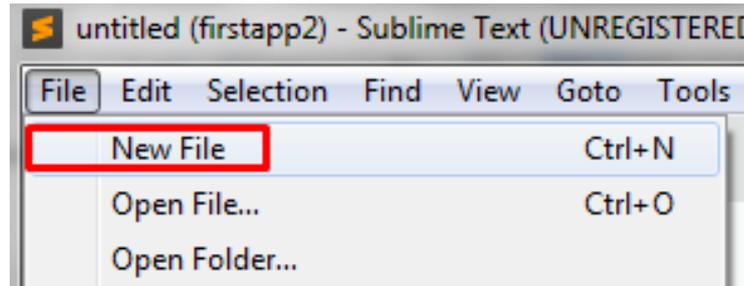
Gambar 2.5 Proses Memulai Instalasi

## 6. Implementasi Program

Pembahasan implementasi program pada modul ini menggunakan bahasa pemrograman JavaScript. JavaScript adalah bahasa program tingkat tinggi atau yang dapat mudah dimengerti oleh manusia. Pada awal mulanya JavaScript digunakan sebagai pengembangan halaman website berbasis *client side*, yaitu program akan dieksekusi disisi *client (browser)*.

Seiring berjalannya waktu, maka JavaScript berkembang sangat pesat, diantaranya adalah muncul berbagai macam varian *framework* yang dapat bertindak sebagai *server side* seperti Node.js. Bahkan sekarang JavaScript dapat digunakan untuk membangun aplikasi *mobile* baik *native* maupun *hybrid* seperti React native, ReactJS, Ionic Framework dan lain-lain. Tahapan yang dilakukan untuk membuat program menggunakan JavaScript adalah sebagai berikut:

- Jalankan editor Sublime Text yang telah berhasil dilakukan instalasi, dan pilih menu **File** → **New File** seperti pada gambar 2.6



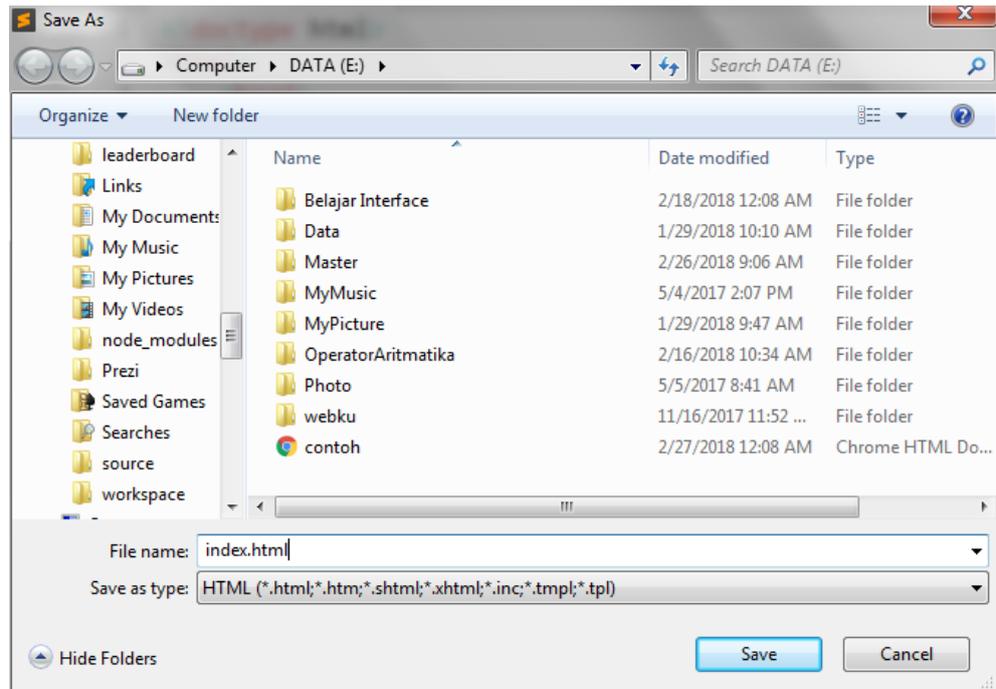
Gambar 2.6 *Menu File*

- b. Setelah itu *file* baru akan terbentuk yang digunakan untuk menuliskan program, ketikkan baris kode program JavaScript yang disisipkan dalam *file* html pada tag `<body>` seperti contoh *script 2.8*

```
<!doctype html>
<html>
  <head>
    <title>Pemrograman itu Mudah</title>
  </head>
  <body>
    <script>
      var bil1 = 100;
      var bil2 = 200;
      if(bil1 > bil2){
        document.write(bil1 + " Lebih Besar dari " + bil2);
      }
      else if(bil1 < bil2){
        document.write(bil1 + " Lebih Kecil dari " + bil2);
      }
      else if(bil1==bil2){
        document.write(bil1 + " Sama dengan " + bil2);
      }
    </script>
  </body>
</html>
```

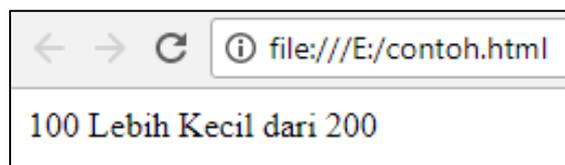
*Script 2.8* Contoh program JavaScript

- c. Langkah berikutnya adalah melakukan penyimpanan *file* dengan menekan tombol keyboard **Ctrl +S**, maka akan muncul kotak dialog pemilihan direktori. Simpanlah *file* tersebut menggunakan nama file berekstensi html, misal index.html seperti pada gambar 2.7



Gambar 2.7 Penyimpanan *File*

- d. Jalankan *file* yang telah disimpan pada proses sebelumnya dengan cara *double* klik pada file **index.html**. Jika penulisan telah memenuhi kaidah penulisan, maka akan muncul *output* seperti pada gambar 2.8



Gambar 2.8 Hasil *output* program

## B. Ketrampilan yang diperlukan dalam Menggunakan Tipe Data dan Control Program

1. Mengidentifikasi permasalahan yang akan dijadikan *control program*
2. Mengidentifikasi tipe data beserta variable yang digunakan
3. Menyusun percabangan dan perulangan pada sebuah *method*

### **C. Sikap kerja yang diperlukan dalam Menggunakan Tipe Data dan Control Program**

Harus bersikap secara:

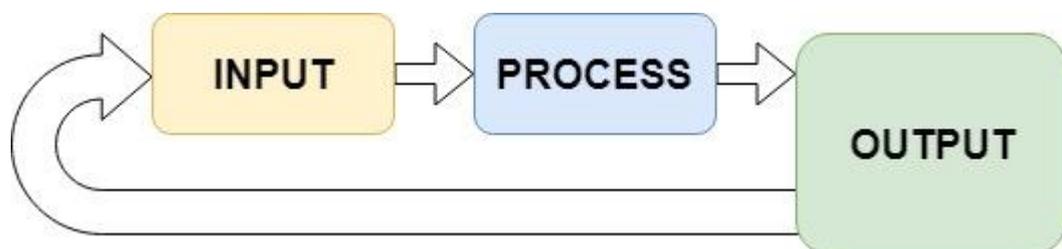
1. Cermat dan teliti dalam menganalisis data.
2. Tekun dalam proses pemahaman sintaks.
3. Sesuai dengan kaidah-kaidah bahasa pemrograman.
4. Berpikir analitis serta evaluatif ketika melakukan *trial and error*.

## BAB III MEMBUAT PROGRAM SEDERHANA

### A. PENGETAHUAN YANG DIPERLUKAN DALAM MEMBUAT PROGRAM SEDERHANA

#### 1. Input Output Program

Program adalah kumpulan dari beberapa set instruksi untuk mengerjakan fungsi yang spesifik pada komputer. Dalam program komputer terdiri beberapa tahapan penting yang tidak dapat dipisahkan. Tahapan pertama yaitu *input* dimana pengguna program akan dimungkinkan untuk memberikan nilai pada variabel melalui *input device* seperti *keyboard*, *mouse*, *scanner*, *joystick* dan lain-lain. Tahapan kedua adalah *process* yaitu tahapan dimana data pada *input device* akan dilakukan pemrosesan sesuai algoritma yang telah disusun. Jika proses tersebut berhubungan dengan perhitungan aritmatika maka proses tersebut akan dilakukan oleh ALU (Arithmetic Logical Unit). Tahap ketiga adalah *output*, yaitu tahapan untuk menampilkan hasil dari proses pada variabel ataupun *output device* seperti layar monitor, *printer*, *plotter* dan lain-lain. Hasil dari *output* tersebut dapat dijadikan sebagai *input*. Perhatikan gambar 3.1 untuk dapat memahami tahapan pada program komputer.



Gambar 3.1 I/O Proses

#### 2. Program Menggunakan JavaScript

JavaScript adalah bahasa *scripting* yang digunakan untuk membuat aplikasi berbasis web. Penulisan kode program menggunakan JavaScript dapat disisipkan langsung kedalam file html seperti pada *script* 3.1 berikut ini

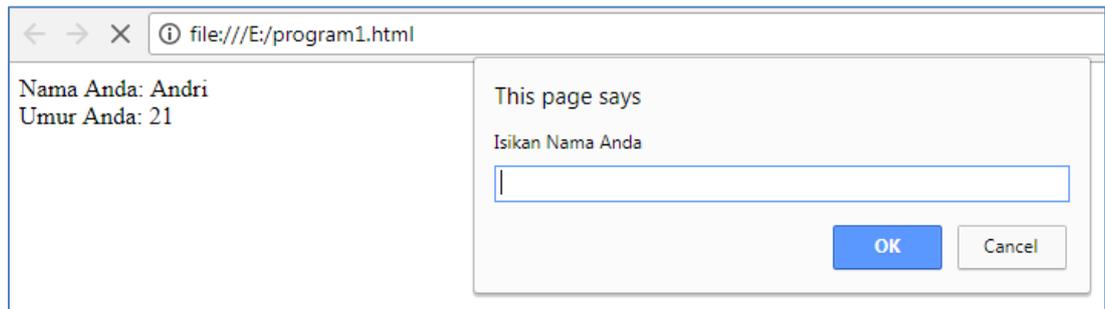
```
1  <!doctype html>
2  <html>
3      <head>
4          <title>Pemrograman itu Mudah</title>
5      </head>
6      <body>
7          <div id="keterangan"></div>
8          <script>
9              var nama = prompt("Isikan Nama Anda");
10             var umur = prompt("Isikan Umur Anda");
11             document.getElementById("keterangan")
12                 .innerHTML= "Nama Anda: " + nama + "<br>Umur Anda: " + umur;
13         </script>
14     </body>
15 </html>
```

*Script 3.1 Penyisipan file JavaScript*

**Keterangan:**

- Baris 1 adalah pengenalan pada browser bahwa *script* yang digunakan menggunakan dokumen HTML5.
- Baris 2, 15 adalah tag pembuka dan penutup HTML.
- Baris 3, 5 adalah tag pembuka dan penutup <head>, pada tag <head> dapat disisipkan tag <title> untuk penulisan judul program.
- Baris 4 adalah penulisan judul program menggunakan tag <title>
- Baris 6, 14 adalah awal dan akhir tag <body> yang digunakan untuk menuliskan konten web.
- Baris 7 adalah tag <div> yang digunakan sebagai tempat meletakkan hasil output.
- Baris 8, 13 adalah awal dan akhir tag <script> yang digunakan untuk meletakkan kode program dari JavaScript.
- Baris 9 adalah deklarasi variabel nama dengan nilai dari input pengguna berupa dialog prompt.
- Baris 10 adalah deklarasi variabel umur dengan nilai dari input pengguna berupa dialog prompt.
- Baris 11 adalah proses menampilkan *output* pada tag yang mempunyai elemen id=keterangan.

Lakukan penyimpanan pada *script* 3.1 dengan menekan tombol *keyboard* Ctrl+S atau dengan memilih menu pada editor Sublime Text **File** → **Save** , Isikan nama *file* **program1.html** atau nama lainnya yang berekstensi .html pada *window dialog* penyimpanan yang tampil. Kemudian jalankan program tersebut dengan melakukan *double* klik pada *file* yang telah tersimpan. Hasil dari program tersebut seperti pada gambar 3.2



Gambar 3.2 Hasil Program

### 3. Program dengan Struktur Kontrol

Setiap program yang baik dapat dipastikan tidak akan pernah terlepas dari struktur kontrol, baik itu kondisi percabangan maupun kondisi perulangan. Struktur kontrol digunakan untuk mengatur alur jalannya eksekusi baris program agar dapat berjalan dengan baik, efisien dalam penulisan kode dan dapat memecahkan sebuah permasalahan. Sebagai contoh program yang dapat melakukan pengecekan bilangan termasuk dalam kategori bilangan prima atau bukan prima seperti pada *script* 3.2. Bilangan prima adalah bilangan yang dapat dibagi dengan bilangan 1 dan bilangan dirinya sendiri atau bilangan yang mempunyai 2 faktor, sehingga alternatif rancangan algoritmanya adalah:

- Deklarasi variabel untuk bilangan.
- Deklarasi variabel untuk faktor dengan nilai awal sama dengan 0.
- Deklarasi variabel sisabagi.
- Deklarasi variabel hasilcek untuk menampung hasil keterangan Bilangan Prima atau Bukan Prima.
- Lakukan *input* pada variabel bilangan.
- Lakukan perulangan antara bilangan 1 sampai bilangan yang akan dicek.

- Lakukan pencarian sisa hasil bagi antara bilangan dengan indeks perulangan menggunakan operator modulus (%)
- Jika sisa hasil bagi sama dengan 0, maka jumlahkan dengan 1 nilai dari faktor
- Setelah proses perulangan selesai, lakukan pengecekan jika nilai faktor sama dengan 2 berarti bilangan prima, selain itu adalah bilangan bukan prima.

Implementasinya pada bahasa pemrograman JavaScript seperti pada *script* 3.2 berikut ini

```
<!doctype html>
<html>
  <head>
    <title>Pemrograman itu Mudah</title>
  </head>
  <body>
    <div id="keterangan"></div>
    <script>
      var bilangan;
      var faktor = 0;
      var sisabagi;
      var hasilcek;
      bilangan = prompt("Isikan Bilangan");

      for(var i=1; i<=bilangan;i++){
        sisabagi = bilangan % i;

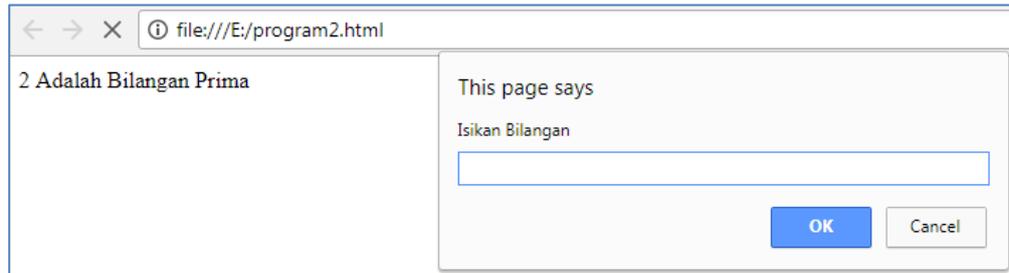
        if(sisabagi==0){
          faktor = faktor +1;
        }
      }

      if (faktor==2){
        hasilcek = "Bilangan Prima";
      }
      else{
        hasilcek = "Bilangan Bukan Prima";
      }

      document.getElementById("keterangan").innerHTML= bilangan + " Adalah " + hasilcek;
    </script>
  </body>
</html>
```

*Script* 3.2 Program pengkategorian bilangan prima

Jika program tersebut dijalankan maka akan tampil seperti pada gambar 3.3 berikut ini



Gambar 3.3 Hasil *output* program bilangan prima

## **B. Keterampilan yang diperlukan dalam Membuat Program Sederhana**

1. Mengidentifikasi permasalahan yang akan dijadikan *control program*
2. Mengidentifikasi tipe data beserta variabel yang digunakan
3. Menyusun rancangan algoritma pemecahan masalah.
4. Menerapkan kondisi percabangan dan perulangan.

## **C. Sikap kerja yang diperlukan dalam Membuat Program Sederhana**

Harus bersikap secara:

1. Cermat dan teliti dalam mengidentifikasi jenis-jenis tipe data
2. Sering melakukan percobaan dan analitis terhadap hasil program.
3. Tekun dan mau berusaha untuk mencoba berkali-kali

## BAB IV

### MEMBUAT PROGRAM MENGGUNAKAN PROSEDUR DAN FUNGSI

#### A. PENGETAHUAN YANG DIPERLUKAN DALAM MEMBUAT PROGRAM MENGGUNAKAN PROSEDUR DAN FUNGSI

##### 1. Prosedur dan Fungsi

Paradigma pemrograman terstruktur adalah proses implementasi dari langkah-langkah penyelesaian permasalahan pada program komputer berdasarkan algoritma yang sederhana, terurut dan sistematis. Ciri-ciri dari pemrograman terstruktur adalah baris-baris kode program akan dieksekusi secara berurutan dari atas kebawah tanpa ada pengulangan kecuali pada struktur kontrol perulangan. Proses penyelesaiannya adalah membagi-bagi instruksi program kedalam bagian kecil yang dikenal dengan istilah prosedur dan fungsi. Tujuan pembuatan fungsi dan prosedur adalah digunakan untuk membungkus program menjadi bagian-bagian kecil agar mudah dikelola. Logika program yang ada di dalam fungsi dan prosedur dapat digunakan kembali dengan cara memanggilnya, sehingga tidak perlu menulis ulang. Perbedaan mendasar antara prosedur dan fungsi terletak pada nilai balik (*return value*) untuk dijadikan *output* ataupun *input*. Jika sebuah program menggunakan fungsi maka akan memiliki nilai balik, sedangkan pada prosedur tidak memiliki nilai balik. Akan tetapi pada JavaScript tidak membedakan antara fungsi dan prosedur.

##### 2. Pembuatan Fungsi pada JavaScript

Fungsi pada JavaScript dapat dibuat dengan menggunakan seperti *script* 4.1 berikut ini

```
function tambah(a,b){  
    hasil = a + b;  
    return hasil;  
}
```

*Script* 4.1 Fungsi pada JavaScript

Penulisan fungsi seperti ini dikenal dengan nama *function declaration* atau deklarasi fungsi. Fungsi tersebut mempunyai beberapa aturan yaitu:

- a. Kata kunci *function*, yang memberitahu Javascript bahwa terdapat sebuah fungsi pada kumpulan set instruksi program.
- b. Nama fungsi, dalam contoh adalah **tambah**. Dengan memberikan sebuah nama fungsi maka kita dapat merujuk ke fungsi tersebut dengan nama yang diberikan. Harus diingat bawa nama fungsi bersifat *opsional*, yang berarti **fungsi pada JavaScript tidak harus diberi nama**.
- c. Daftar parameter fungsi, yaitu a, b pada contoh di atas. Daftar parameter ini selalu dikelilingi oleh tanda kurung (()). Parameter boleh kosong, tetapi tanda kurung wajib tetap dituliskan. Parameter fungsi akan secara otomatis didefinisikan menjadi variabel yang hanya bisa dipakai di dalam fungsi. Variabel pada parameter ini diisi dengan nilai yang dikirimkan kepada fungsi secara otomatis.
- d. Sekumpulan perintah yang ada di dalam kurung kurawal ({}). Perintah-perintah ini dikenal dengan nama badan fungsi. Badan fungsi dieksekusi secara berurut ketika fungsi dijalankan.

Penulisan deklarasi fungsi (*function declaration*) seperti di atas merupakan cara penulisan fungsi yang umumnya kita gunakan pada bahasa pemrograman imperatif dan berorientasi objek. Tetapi selain deklarasi fungsi, Javascript juga mendukung cara penulisan fungsi lain, yaitu dengan memanfaatkan ekspresi fungsi (*function expression*). Ekspresi fungsi merupakan cara pembuatan fungsi yang memperbolehkan kita melewatkan nama fungsi. Fungsi yang dibuat tanpa nama dikenal dengan sebutan fungsi anonim atau fungsi lambda. *Script 4.2* berikut ini adalah contoh ekspresi fungsi:

```
var tambah = function(a,b){  
    hasil = a + b;  
    return hasil;  
}
```

*Script 4.2* Ekspresi fungsi

Terdapat hanya sedikit perbedaan antara ekspresi fungsi dan deklarasi fungsi:

- a. Penamaan fungsi. Pada deklarasi fungsi, langsung memberikan nama fungsi sesuai dengan sintaks yang disediakan Javascript. Menggunakan ekspresi fungsi, pada dasarnya adalah menyimpan sebuah fungsi anonim ke dalam variabel, dan nama fungsi adalah nama variabel yang kita buat. Perlu diingat juga bahwa pada dasarnya ekspresi fungsi *adalah* fungsi anonim. Penyimpanan ke dalam variabel hanya diperlukan karena akan memanggil fungsi nantinya.
- b. Ekspresi fungsi dapat dipandang sebagai sebuah ekspresi atau perintah standar bagi Javascript, sama seperti ketika ada penulisan kode `var i = 0;`. Deklarasi fungsi merupakan konstruksi khusus untuk membuat fungsi. Hal ini berarti pada akhir dari ekspresi fungsi harus menambahkan `;`, sementara pada deklarasi fungsi hal tersebut tidak penting.

Karena pada JavaScript sebuah fungsi juga adalah sekaligus sebuah objek. Setiap kali penciptaan fungsi, pada dasarnya adalah membuat sebuah objek *function* baru, dengan pemberian sebuah nama yang akan menyederhanakan kode program, dan pada akhirnya akan mempermudah pemahaman tentang kode program.

Aturan pembuatan fungsi, baik ekspresi fungsi maupun deklarasi fungsi, sama dengan aturan penulisan ekspresi. Di mana diperbolehkan penulisan ekspresi dan definisi fungsi. Karena aturan ini, maka terdapat definisi fungsi di dalam fungsi lainnya. Fungsi yang berada di dalam fungsi lainnya memiliki akses terhadap semua variabel yang ada pada fungsi penampungnya. Keterhubungan fungsi di dalam fungsi ini dikenal dengan nama *closure*.

### 3. Implementasi Fungsi dan Prosedur

Sebagai contoh untuk pembuatan program pada bagian ini adalah pembuatan fungsi dan prosedur yang digunakan untuk menghitung total

belanja dari beberapa barang dengan *input* berupa komponen *input form* dari HTML. Adapun langkah-langkahnya adalah:

- a. Jalankan *editor software* Sublime Text seperti pada pembahasan sebelumnya.
- b. Buat *file* dengan nama **belanja.html** dan jangan lupa untuk selalu melakukan penyimpanan data setiap ada perubahan kode program.
- c. Buat kode program untuk tampilan *input form* seperti pada *script 4.3*

```
1 <!doctype html>
2 <html>
3   <head>
4     <title>Program Belanja</title>
5   </head>
6   <body>
7
8   <div>
9     Barang :
10    <select id="listbarang">
11      <option value="Buku" selected>Buku</option>
12      <option value="Tas">Tas</option>
13      <option value="Pensil">Pensil</option>
14      <option value="Penghapus">Penghapus</option>
15    </select>
16
17    <button id="tambah" onClick="cek_barang()">(+)</button>
18  </div>
19  <div>
20    Jumlah :
21    <input type="text" id="jumlah">
22  </div>
23  <div id="barang"></div>
24  <div id="total"></div>
25
26  </body>
27 </html>
```

*Script 4.3 Input Form*

Penjelasan:

- Baris 1 adalah pengenalan pada browser bahwa *script* yang digunakan menggunakan dokumen HTML5.
- Baris 2, 27 adalah tag pembuka dan penutup HTML.
- Baris 3, 5 adalah tag pembuka dan penutup `<head>`, pada tag `<head>` dapat disisipkan tag `<title>` untuk penulisan judul program.
- Baris 4 adalah penulisan judul program menggunakan tag `<title>`
- Baris 6, 26 adalah awal dan akhir tag `<body>` yang digunakan untuk menuliskan konten web.
- Baris 8 adalah tag **div** yang digunakan untuk membagi tampilan konten *web*
- Baris 9 adalah label sebagai keterangan tulisan **Barang :**
- Baris 10, 15 adalah awal dan akhir tag **select** yang digunakan untuk membuat komponen *form* berupa *drop down* menu atau pilihan.
- Baris 17 adalah komponen *form* berupa tombol (*button*) dimana saat diklik, maka akan memanggil fungsi dengan nama **cek\_barang**
- Baris 19, 22 adalah tag `<div>` yang digunakan untuk membagi tampilan konten web
- Baris 20 adalah label sebagai keterangan tulisan **Jumlah :**
- Baris 21 adalah komponen *form* berupa *textbox* yang digunakan untuk melakukan *input* jumlah yang akan dibeli.
- Baris 23, 24 adalah tag **div** yang digunakan untuk mencetak keterangan belanja berupa nama barang, harga barang dan total belanja.

d. Buat sebuah fungsi dengan nama **cek\_barang** seperti pada *script* 4.4, dimana fungsi tersebut akan dipanggil ketika tombol **(+)** diklik, letakkan fungsi tersebut didalam tag **body**.

```
1 <script>
2     var harga;
3     var nama_brg;
4     var ket = "";
5     var jumlah=0;
6     var total=0;
7     var subtotal=0;
8
9     function cek_barang(){
10        var nama = document.getElementById("listbarang").value;
11        switch(nama){
12            case 'Buku':
13                harga = 10000;
14                nama_brg = "Buku";
15                break;
16            case 'Tas':
17                harga = 50000;
18                nama_brg = "Tas";
19                break;
20            case 'Pensil':
21                harga = 5000;
22                nama_brg = "Pensil";
23                break;
24            case 'Penghapus':
25                harga = 15000;
26                nama_brg = "Penghapus";
27                break;
28        }
29
30        jumlah = parseInt(document.getElementById("jumlah").value);
31        subtotal = jumlah * harga;
32        total = total + subtotal;
33        ket = ket + nama_brg + ":" + harga + "*" + jumlah + "=" + subtotal + "<br>";
34        document.getElementById("barang").innerHTML = ket;
35        document.getElementById("total").innerHTML = "Total Belanja = " + total;
36    }
37 </script>
```

Script 4.4 Fungsi cek\_barang

Keterangan :

- Baris 1 adalah tag script yang digunakan untuk memperkenalkan kepada *browser* bahwa didalamnya adalah terdapat JavaScript.
- Baris 2 sampai 7 adalah berurutan deklarasi variabel global untuk harga, nama\_brg, ket, jumlah, total dan subtotal.
- Baris 9 adalah awal dimulainya sebuah fungsi.
- Baris 10 adalah deklarasi variabel nama yang diisi dengan nilai komponen yang mempunyai nama id adalah **listbarang**. Jadi ketika menu pilihan diisi dengan pilihan Buku, maka nilai dari variabel nama adalah Buku.
- Baris 11 sampai 28 adalah implementasi dari kondisi percabangan menggunakan **switch**.

- Baris 30 adalah variabel jumlah diisi nilainya dari komponen yang mempunyai nama id adalah **jumlah**. Jadi ketika *textbox* jumlah diisi dengan nilai 2, maka nilai dari variabel jumlah adalah 2. **parseInt** digunakan untuk mengubah data dari string menjadi integer atau numerik, dengan tujuan agar dapat dilakukan operasi aritmatik seperti perkalian.
  - Baris 31 adalah nilai variabel subtotal didapat dari jumlah yang dibeli dikali dengan harga barang.
  - Baris 32 adalah nilai variabel total didapat dari nilai total ditambah dengan subtotal.
  - Baris 33 adalah variabel ket untuk menampung keterangan transaksi.
  - Baris 34 dan 35 adalah proses untuk menampilkan hasil *output* pada tag div yang mempunyai id barang dan total.
- e. Buatlah komentar pada fungsi dengan tujuan agar ketika program semakin kompleks maka kode tersebut dapat dengan mudah dan identifikasi ketika ada perbaikan atau penambahan fitur. Pemberian komentar dapat diberikan pada 1 (satu) baris ataupun lebih seperti pada contoh *script* 4.5

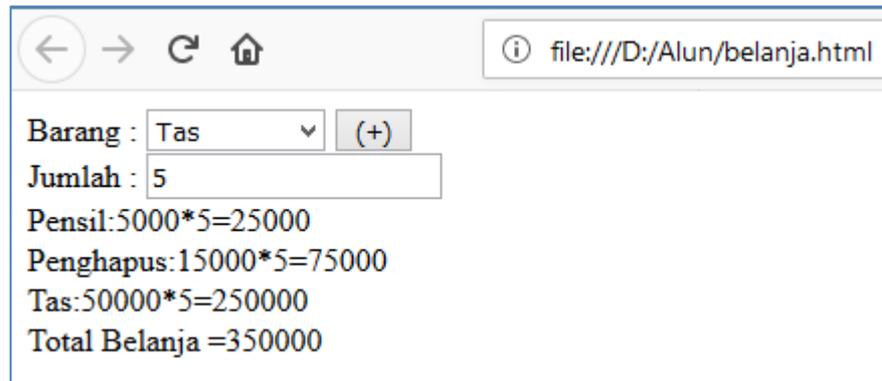
```

1 <script>
2 // Fungsi cek_barang untuk mengetahui harga barang
3 function cek_barang(){
4     var nama = document.getElementById("listbarang").value;
5     switch(nama){
6         case 'Buku':
7             harga = 10000;
8             nama_brg = "Buku";
9             break;
10        case 'Tas':
11            harga = 50000;
12            nama_brg = "Tas";
13            break;
14        case 'Pensil':
15            harga = 5000;
16            nama_brg = "Pensil";
17            break;
18        case 'Penghapus':
19            harga = 15000;
20            nama_brg = "Penghapus";
21            break;
22    }
23
24    /* Komentar banyak baris
25    semua kode yang ditulis diantara
26    komentar, maka tidak akan dieksekusi */
27    jumlah = parseInt(document.getElementById("jumlah").value);
28    subtotal = jumlah * harga;
29    total = total + subtotal;
30    ket = ket + nama_brg + ":" + harga + "*" + jumlah + "=" + subtotal + "<br>";
31    document.getElementById("barang").innerHTML = ket;
32    document.getElementById("total").innerHTML = "Total Belanja =" + total;
33 }
34
35 </script>

```

*Script* 4.5 Pemberian komentar pada program

- f. Jalankan program belanja tersebut dengan mengisikan jumlah barang yang akan dibeli, kemudian lakukan klik pada tombol **(+)**, hasilnya akan tampil seperti pada gambar 4.1



Gambar 4.1 Hasil *output* program

## **B. Keterampilan yang diperlukan dalam Membuat Program Menggunakan Prosedur dan Fungsi**

1. Membuat program dengan prosedur.
2. Membuat program dengan fungsi.
3. Membuat fungsi dan prosedur secara bersamaan.
4. Memberikan komentar pada masing-masing fungsi.

## **C. Sikap kerja yang diperlukan dalam Membuat Program Menggunakan Prosedur dan Fungsi**

Harus bersikap secara:

1. Cermat dan teliti dalam mengidentifikasi bentuk fungsi dan prosedur.
2. Sering melakukan percobaan dan analitis terhadap hasil program.
3. Tekun dan mau berusaha untuk mencoba berkali-kali

## **BAB V**

### **MEMBUAT PROGRAM MENGGUNAKAN ARRAY**

#### **A. Pengetahuan yang diperlukan dalam Membuat Program Menggunakan Array**

##### **1. Pengertian Array**

Array adalah tipe data yang berisi kumpulan dari beberapa nilai. Nilai yang terdapat dalam array disebut dengan elemen, dan pada setiap elemen memiliki nomor pengenal yang dikenal dengan istilah indeks. Penomoran indeks di dalam array dimulai dari angka 0, sehingga elemen pertama berada di index 0, elemen kedua berada di indeks 1, sehingga dapat disimpulkan bahwa elemen yang ke  $n$  berada pada indeks  $n-1$ . Indeks maksimum yang dapat digunakan dalam array pada JavaScript adalah  $4.294.967.294 (2^{23} - 2)$ , dengan jumlah elemen maksimum adalah  $4.294.967.295$ . Jenis-jenis array yang sering digunakan dalam pemrograman adalah:

a. Array 1 (satu) dimensi

- Array satu dimensi adalah struktur data yang statis mempunyai satu nama tetapi memiliki banyak tempat.
- Array satu dimensi atau dalam variabel beberapa literatur disebut sebagai larik, adalah suatu tipe variabel yang berisi kumpulan data dengan setiap element datanya bertipe sama, kecuali jika diimplementasikan menggunakan JavaScript.
- Array satu dimensi merupakan sebuah variabel yang menyimpan sekumpulan data yang memiliki tipe sama dan elemen yang diakses hanya melalui 1 indeks atau subskrip.

Gambar 5.1 merupakan ilustrasi array 1 (satu) dimensi, yaitu variabel array yang mempunyai nama **DATA**, didalam variabel tersebut indeks ke 0 mempunyai nilai 20, ke 1 mempunyai nilai 1, ke 2 mempunyai nilai 19, ke 3 mempunyai nilai 8 dan ke 4 mempunyai nilai 9.

DATA				
20	1	19	8	9
indeks 0	indeks 1	indeks 2	indeks 3	indeks 4
5				

Gambar 5.1 Ilustrasi variabel array

b. Array Multi dimensi

- Array multi dimensi merupakan sebuah variabel yang menyimpan sekumpulan data yang memiliki tipe sama dan elemen yang akan diakses melalui banyak indeks atau subskrip.
- Array multi-dimensi merupakan array yang mempunyai ukuran lebih dari satu.

Gambar 5.2 merupakan ilustrasi array 2 (dua) dimensi.

	Variabel Array				
Baris 0	data[0,0]	data[0,1]	data[0,2]	.....	data[0,M-1]
Baris 1	data[1,0]	data[1,1]	data[1,2]	.....	data[1,M-1]
Baris 2	data[2,0]	data[2,1]	data[2,2]	.....	data[2,M-1]
.....	.....	.....	.....	.....	.....
Baris N-1	data[N-1,0]	data[N-1,1]	data[N-1,2]	.....	data[N-1,M-1]
	Kolom 0	Kolom 1	Kolom 2	.....	Kolom M-1

Gambar 5.2 Ilustrasi array 2(dua) dimensi

Array di dalam JavaScript tidak bertipe (untyped array). Elemen dari array bisa bertipe data string, number dan boolean dalam sebuah array yang sama, bahkan elemen dari array bisa berupa objek atau array yang lain. Array di dalam JavaScript bersifat dinamis, sehingga tidak perlu mendefinisikan

berapa ukuran array pada saat membuat variabel. Jumlah elemen dapat ditambah dan dikurang setiap saat. Indeks array di dalam JavaScript juga tidak harus berurutan, JavaScript membolehkan elemen dari array tidak terurut seperti pengisian hanya pada index 1, 7, dan 11 saja di dalam array.

## 2. Array pada JavaScript

### a. Cara Penulisan Array

Terdapat 2 (dua) cara penulisan tipe data array di dalam **JavaScript**, yaitu dengan penulisan menggunakan kurung siku [ dan ], dan menggunakan *keyword* **new Array()**. Perintah **new** ini akan membuat **objek array**. *Script* 5.1 dan 5.2 berikut ini adalah pembuatan **array** di dalam JavaScript:

```
<script>
  var arrKosong = []; //array kosong tanpa data
  var arrKota = ["Malang","Surabaya","Medan"]; //Array dengan data string
  var arrAngka = [10,3,67,43,56]; //Array dengan data numerik
  var arrMixed = [1,45,6,"Andi","Nia"]; //Array dengan data gabungan

  //Deklarasi Array dengan penciptaan objek
  var arrObject = new Array();
  var arrObjKota = new Array("Magelang", "Semarang", "Banyuwangi");
  var arrObjAngka = new Array(20,567,32,44);
  var arrObjMixed = new Array(76,32,"Roy","Fahmi");

  //Tampilkan Array
  alert(arrKosong.toString());
  alert(arrKota.toString());
  alert(arrAngka.toString());
  alert(arrMixed.toString());
  alert(arrObject.toString());
  alert(arrObjKota.toString());
  alert(arrObjAngka.toString());
  alert(arrObjMixed.toString());
</script>
```

*Script* 5.1 Deklarasi array 1(satu) dimensi

```
<script>
  var warna = [
    ["putih","merah","hitam"],
    ["hijau","abu abu","biru"],
  ];

  //Tampilkan Array
  alert(warna[1][2].toString()); //Biru
  alert(warna[0][1].toString()); //Merah
</script>
```

*Script* 5.2 Deklarasi array 2(dua) dimensi

Pada *script* 5.1 adalah contoh deklarasi variabel array dengan menggunakan kurung siku `[]` dan *keyword* **new Array()**. Proses untuk menampilkan data tersebut dengan cara memanggil *method* **toString()**. yaitu mengubah setiap array yang berupa objek menjadi **string**.

### b. Pengambilan Nilai Array.

Nilai dari elemen array dapat bersifat tidak tetap, dan dapat mengambil dari nilai variabel lain seperti pada *script* 5.3

```
<script>
  var nilai = 0;
  var arrData = [nilai + 1, nilai -10, 2,56,"Malang"];

  //Tampilkan Array
  alert(arrData.toString());
  //1,-10,2,56,Malang
</script>
```

*Script* 5.3 Pemberian nilai array dari variabel lain

### c. Pemberian dan Pengambilan Nilai Array

Cara untuk mengambil nilai dari elemen array yaitu menggunakan sintaks

```
nama_array[indeks]
```

Sedangkan untuk memberi nilai pada elemen array yaitu menggunakan sintaks

```
nama_array[indeks] = value
```

Jika pemberian nilai dilakukan pada elemen indeks yang sudah ada nilainya, maka nilai yang lama akan ditimpa dengan nilai yang baru seperti *script* 5.4

```
<script>
  var arrNegara = ["Malaysia","Inggris","Jerman","Jepang"];

  //Tampilkan Array ke 0
  alert("Nilai indeks ke 0 : "+arrNegara[0].toString());

  //Pemberian nilai pada array 0
  arrNegara[0] = "Indonesia";

  //Tampilkan Array ke 0
  alert("Nilai indeks ke 0 : "+ arrNegara[0].toString());
</script>
```

*Script* 5.4 Pemberian nilai pada variabel array

#### d. Jumlah Elemen Array

Mengetahui jumlah elemen array yaitu dengan mencari tahu panjang array tersebut. JavaScript menyediakan sebuah *method* yaitu array *length of property*. Hasil dari *property array length* adalah berapa banyak elemen di dalam array tersebut, bukan indeks tertinggi. Karena indeks dimulai dari 0, maka nilai **length** akan selalu lebih besar daripada nilai **indeks** tertinggi. Untuk **array** dengan **indeks** yang terurut dari 0, maka *indeks* tertinggi adalah nilai **length-1**. Cara penggunaannya seperti pada *script 5.5*

```
<script>
  var arrNegara = ["Indonesia", "Inggris", "Jerman", "Jepang"];

  alert(arrNegara.toString()); //Indonesia, Inggris, Jerman, Jepang
  alert(arrNegara.length); //4
</script>
```

*Script 5.5 Penggunaan length of property*

#### e. Menghapus Elemen Array

Menghapus elemen array dapat menggunakan operator *delete*, seperti pada *script 5.6*

```
<script>
  var arrNegara = ["Indonesia", "Inggris", "Jerman", "Jepang"];

  alert(arrNegara.toString()); //Indonesia, Inggris, Jerman, Jepang
  delete arrNegara[1]; //Inggris
  alert(arrNegara.toString()); //4
</script>
```

*Script 5.6 Menghapus array*

Operator **delete** hanya menghapus nilai dari elemen, bukan menghapus elemen array sama sekali. Operator ini seolah-olah memberikan nilai **undefined** kedalam elemen array yang akan dihapus. Nilai dari *property length* juga tidak dipengaruhi oleh operator **delete** ini.

#### f. Mengakses Array Menggunakan Perulangan

Sebuah program terkadang memiliki array yang sangat panjang, misal array yang digunakan untuk menampung nilai piksel dari gambar. Jika mengakses nilainya menggunakan baris perbaris, maka kode program menjadi tidak efisien, mempunyai ukuran *file* yang lebih besar. Permasalahan tersebut dapat diselesaikan menggunakan konsep perulangan seperti yang telah dibahas pada bagian sebelumnya. *Script 5.7* berikut ini adalah contoh penggunaan perulangan untuk mengakses nilai array.

```
<script>
    var arrNegara = [
        "Indonesia",
        "Inggris",
        "Jerman",
        "Jepang",
        "Arab Saudi",
        "Afrika Selatan"
    ];

    for(var i=0;i<=arrNegara.length-1;i++){
        alert(arrNegara[i].toString());
    }
</script>
```

*Script 5.7* Penggunaan perulangan untuk mengakses array

### 3. Program *Sorting* Menggunakan Array

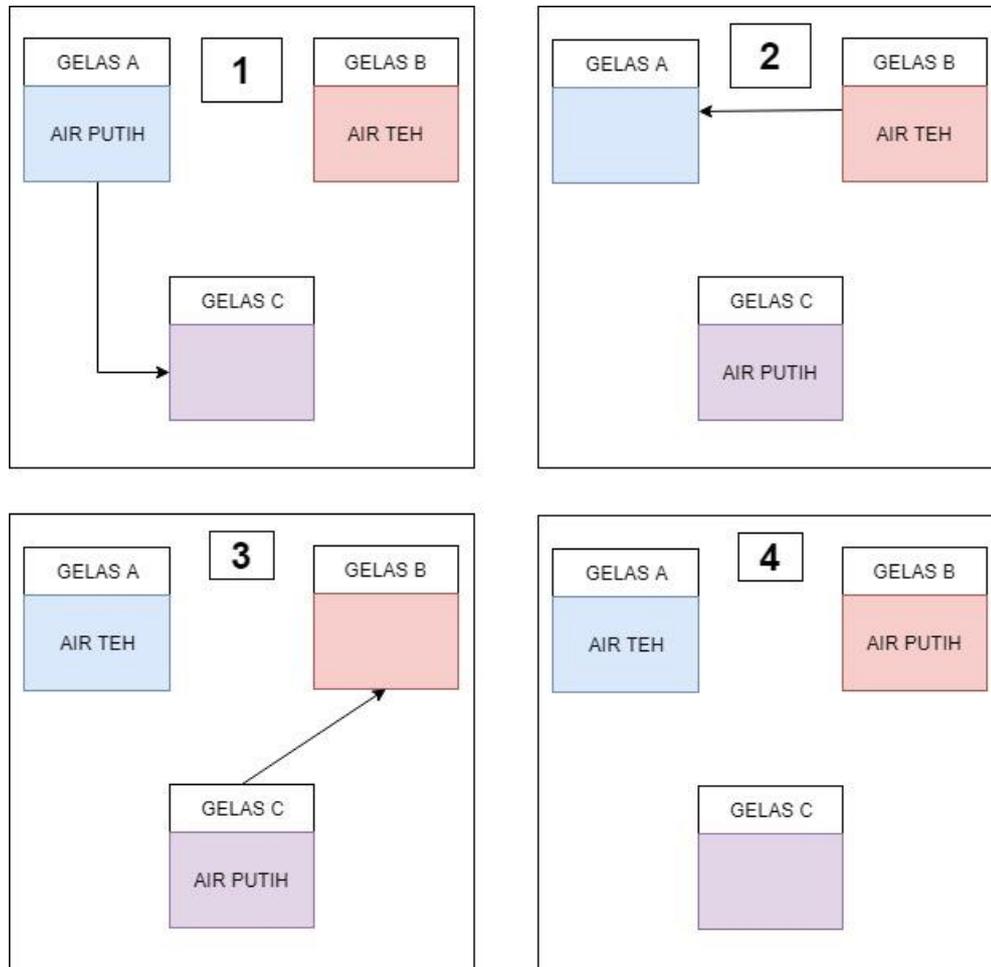
Tipe data array dapat dimanfaatkan salah satunya adalah untuk melakukan pengurutan data (*sorting*). *Sorting* adalah proses pengaturan sekumpulan data sesuai dengan susunan tertentu. Susunan dari data yang digunakan dapat diatur dari data terkecil sampai data terbesar (*ascending*) atau sebaliknya dari data terbesar sampai data terkecil (*descending*). Misal terdapat tipe data array dengan kumpulan nilai 5,1,4,2, maka dengan memanfaatkan algoritma sortir *ascending* nilai tersebut menjadi 1,2,4,5. Salah satu algoritma untuk menyelesaikannya adalah membandingkan nilainya satu persatu atau yang dikenal dengan *selection sort* seperti pada gambar 5.3



Gambar 5.3 Ilustrasi *selection sort*

Proses sortir menggunakan perulangan untuk membandingkan antar nilai-nilai yang terdapat dalam *array*, proses perbandingannya tergantung pada susunan urutannya. Jika disusun secara *ascending*, maka menggunakan operator aritmatika lebih besar(>) dan sebaliknya jika secara *descending*

menggunakan lebih kecil (<). Proses pertukaran atau *swap* menggunakan variabel temporary atau variabel bantuan (*dummy*). Seperti pada contoh kehidupan nyata, bagaimana cara untuk menukar segelas air putih dengan segelas air teh, maka dibutuhkan gelas bantuan (*dummy*) seperti pada ilustrasi gambar 5.4



Gambar 5.4 Ilustrasi pertukaran data

Secara garis besar tahapan implementasi sortir menggunakan JavaScript adalah:

a. Tentukan dimensi *array*

*Array* yang digunakan adalah 1(satu) dimensi yaitu menggunakan satu kurung siku, contoh:

```
var dataArray = [];
```

b. Tentukan tipe datanya.

Menggunakan tipe data numerik (integer) karena yang data yang dilakukan pengurutan berupa sebuah angka, yaitu dengan langsung memberikan nilai pada variabel *array* yang telah dideklarasikan, contoh:

```
dataArray = [23,100,54,97,30];
```

c. Tentukan Panjang array

Panjang array didapat dengan menggunakan *method* *length*, contoh:

```
var pjpg_array = dataArray.length;
```

d. Lakukan sortir

Proses sortir yang dilakukan sesuai dengan konsep sortir yang telah dibahas sebelumnya, dimana panjang array digunakan sebagai batas akhir perulangan seperti pada *script* 5.8

```
<html>
  <head>
  </head>
  <body>
    <script>
      var dataArray = [];
      dataArray = [23,100,54,97,30];
      var pjpg_array = dataArray.length;
      var dummy;

      alert("Data Array Awal "+ dataArray.toString());

      for(var i=0;i<pjpg_array;i++){
        for(var j=0;j<pjpg_array;j++){
          if (dataArray[j] > dataArray[i]){
            dummy = dataArray[j];
            dataArray[j] = dataArray[i];
            dataArray[i] = dummy;
          }
        }
      }
      alert("Data Array setelah di sortir "+dataArray.toString());
    </script>
  </body>
</html>
```

*Script* 5.8 Selection sort

Jika ingin melihat hasil sortir tiap langkah, maka pindahkan *script alert* kedalam blok kode perulangan seperti pada *script 5.9*

```
<html>
  <head>
  </head>
  <body>
    <script>
      var dataArray = [];
      dataArray = [23,100,54,97,30];
      var pjg_array = dataArray.length;
      var dummy;

      alert("Data Array Awal "+ dataArray.toString());

      for(var i=0;i<pjg_array;i++){
        for(var j=0;j<pjg_array;j++){
          if (dataArray[j] > dataArray[i]){
            dummy = dataArray[j];
            dataArray[j] = dataArray[i];
            dataArray[i] = dummy;
          }
        }
        alert("Data Array setelah di sortir "+dataArray.toString());
      }
    </script>
  </body>
</html>
```

*Script 5.9* Sortir dengan *output* tiap langkah

Lakukan lagi pemindahan **alert** kedalam perulangan di atasnya. Lakukan Analisa proses apa yang terjadi.

## **B. Keterampilan yang diperlukan dalam Membuat Program Menggunakan Array**

1. Membuat menentukan dimensi *array*.
2. Membuat menentukan tipe data *array*.
3. Membuat menentukan Panjang *array*
4. Memberikan menggunakan pengurutan

## **C. Sikap kerja yang diperlukan dalam Membuat Program Menggunakan Array**

Harus bersikap secara:

1. Cermat dan teliti pada penggunaan perulangan.

2. Sering melakukan percobaan dan analitis terhadap hasil program.
3. Tekun dan mau berusaha untuk mencoba berkali-kali

## BAB VI

### MEMBUAT PROGRAM UNTUK AKSES FILE

#### A. Pengetahuan yang diperlukan dalam Membuat Program Untuk Akses File

##### 1. Akses File

Pada sebuah aplikasi, terkadang dibutuhkan proses untuk membaca *file* dari media penyimpanan ataupun menulis *file* pada media penyimpanan. *File* yang digunakan untuk akses *file* sangat beragam, dapat berbentuk *file teks*, gambar dan juga audio video. Adapun media penyimpanan dapat berupa *local storage* seperti *harddisk*, *USB Flash disk*, *Compact Disk (CD)*, *Digital Versatile Disk (DVD)* ataupun berupa *database* yang tersimpan dalam jaringan *internet* yang lebih dikenal dengan istilah *cloud storage*.

##### 2. Akses File Menggunakan JavaScript

Salah satu cara untuk mengakses *file* adalah dengan menggunakan library API (*Application Programming Interface*) yaitu **FileReader**. API adalah sekumpulan fungsi yang telah disediakan oleh JavaScript, sehingga programmer dapat langsung menggunakannya tanpa membuat dari awal. Objek **FileReader** memungkinkan aplikasi web secara *asynchronous* membaca isi *file* (atau *buffer* data mentah) yang tersimpan di komputer pengguna, menggunakan objek *file* atau *Binary Large Object (BLOB)* untuk menentukan *file* atau data yang akan dibaca. Tabel 6.1 merupakan beberapa *property* dan *method* dari **FileReader** yang sering digunakan.

Tabel 6.1 API FileReader

NO	Fungsi	Keterangan
1.	FileReader.error	Memberikan <i>output</i> keterangan <i>error</i> ketika terjadi kesalahan dalam pengaksesan <i>file</i>
2.	FileReader.readyState	Menghasilkan angka yang mengindikasikan keadaan ketika pembacaan <i>file</i> .

		<ul style="list-style-type: none"> <li>• 0 berarti tidak ada data yang dibaca</li> <li>• 1 berarti data sedang diproses</li> <li>• 2 berarti berarti data telah selesai dibaca</li> </ul>
3.	FileReader.onabort	Mengambil alih ketika terdapat <i>abort event</i> . Melakukan trigger ketika keluar dari program
4.	FileReader.onprogress	Dilakukan ketika pembacaan konten BLOB seperti gambar. <i>Menghandle</i> ketika pembacaan file masih dalam proses
5.	FileReader.readAsArrayBuffer()	Mulai membaca isi BLOB yang ditentukan, setelah proses selesai, atribut hasil berisi ArrayBuffer yang mewakili data file.
6.	FileReader.readAsBinaryString()	Mulai membaca isi BLOB yang ditentukan, setelah selesai, atribut hasil berisi data biner mentah dari file sebagai string.
7.	FileReader.readAsDataURL ()	Mulai membaca isi Blob yang ditentukan, setelah selesai, atribut hasil berisi data: URL yang mewakili data file
8.	FileReader.readAsText()	Mulai membaca isi Blob yang ditentukan, setelah selesai, atribut hasil berisi isi file sebagai string teks.

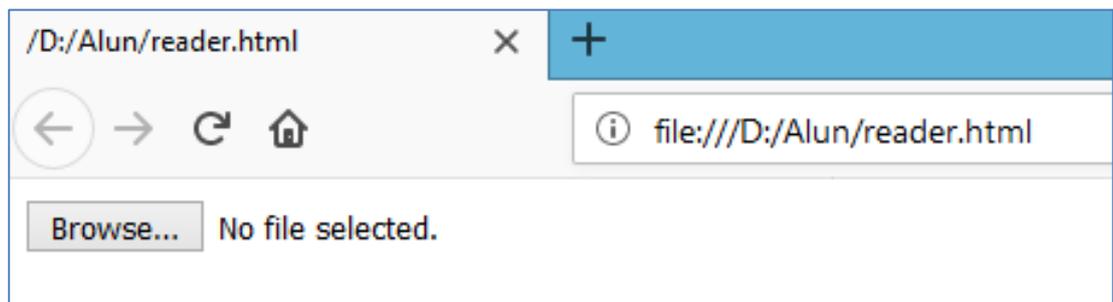
Buatlah *file* berektensi html dengan menggunakan komponen *form* dengan tipe input berupa *file* dengan event **onchange** yaitu proses untuk menampilkan *file dialog* untuk memilih *file* mana yang akan dibaca. Adapun contoh kode programnya seperti pada *script* 6.1 yang digunakan untuk membaca *file* berupa gambar.

```
<!doctype html>
<head>
  <title>Data Reading</title>
</head>
<body>
  <input type='file' accept='image/*' onchange='BukaFile(event)'><br>
  <img id='output'>
  <script>
    var BukaFile = function(event) {
      var input = event.target;

      var reader = new FileReader();
      reader.onload = function(){
        var dataURL = reader.result;
        var output = document.getElementById('output');
        output.src = dataURL;
      };
      reader.readAsDataURL(input.files[0]);
    };
  </script>
</body>
</html>
```

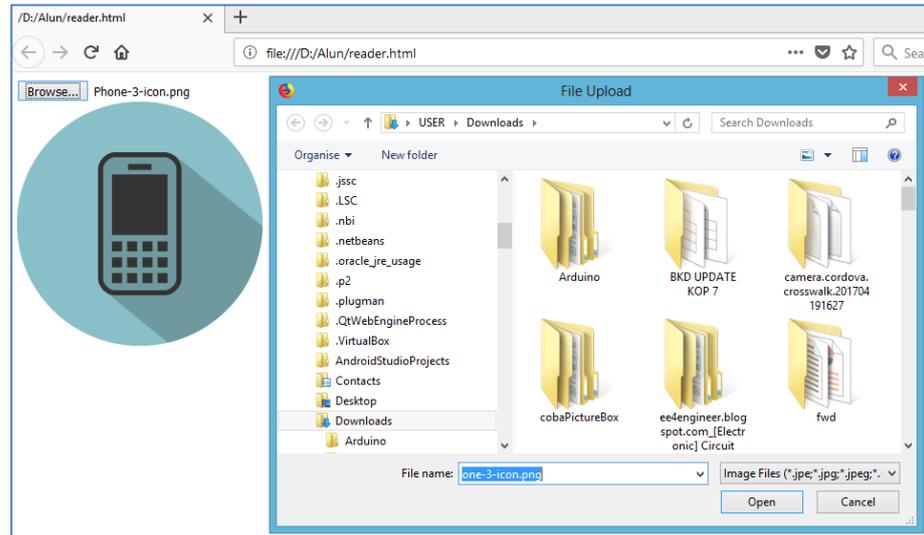
Script 6.1 Membuka *file* gambar

Jalankan *script* 6.1 maka akan tampil hasil program seperti pada gambar 6.1 berikut ini:



Gambar 6.1 Tampilan hasil program

Klik pada tombol **Browse** sehingga akan muncul *file dialog*, kemudian pilihlah salah satu gambar yang tersimpan pada harddisk, maka hasil program akan tampil seperti pada gambar 6.2



Gambar 6.2 File Dialog pemilihan gambar

Sedangkan untuk pembacaan file teks seperti pada *script 6.2*

```
<!doctype html>
<head>
  <title>Data Reading</title>
</head>
<body>
  <div id="page-wrapper">
    <h1>Text File Reader</h1>
    <div>
      Select a text file:
      <input type="file" id="fileInput">
    </div>
    <pre id="fileDisplayArea"><pre>
  </div>

  <script>
    window.onload = function() {
      var fileInput = document.getElementById('fileInput');
      var fileDisplayArea = document.getElementById('fileDisplayArea');

      fileInput.addEventListener('change', function(e) {
        var file = fileInput.files[0];
        var fileType = /text.*/;

        if (file.type.match(fileType)) {
          var reader = new FileReader();

          reader.onload = function(e) {
            fileDisplayArea.innerText = reader.result;
          }

          reader.readAsText(file);
        } else {
          fileDisplayArea.innerText = "File not supported!"
        }
      });
    }
  </script>
</body>
</html>
```

Script 6.2 Pembacaan file teks

Jalankan dan Analisa hasil *output* nya. Ujilah program tersebut dengan menggunakan parameter yang berbeda.

**B. Keterampilan yang diperlukan dalam Membuat Program Untuk Akses File**

1. Menggunakan sintaks bahasa pemrograman JavaScript
2. Menggunakan *Library Application Programming Interface*
3. Membaca data berupa teks dan gambar

**C. Sikap kerja yang diperlukan dalam Membuat Program Untuk Akses File**

Harus bersikap secara:

1. Cermat dan teliti.
2. Sering melakukan percobaan dan analitis terhadap hasil program.
3. Tekun dan mau berusaha untuk mencoba berkali-kali

## **BAB VII**

### **MENGGOMPILASI PROGRAM**

#### **A. PENGETAHUAN YANG DIPERLUKAN DALAM MENGGOMPILASI PROGRAM**

##### **1. Perbaiki Kesalahan**

Sebuah program yang muncul pesan kesalahan atau *error* adalah hal yang wajar pada dunia pemrograman. Semakin kompleks dan besar skala program maka akan semakin banyak kemungkinan terjadinya *error*. Hampir tidak ada satupun program yang tidak memiliki *error*, semua dikarenakan keterbatasan manusia sebagai pembuatnya. Melihat kenyataan tersebut yang dapat dilakukan oleh seorang *programmer* adalah berusaha untuk mencari kesalahan-kesalahan tersebut baik kesalahan besar ataupun kecil untuk diperbaiki terus menerus, melakukan *update* program dengan tujuan untuk meminimalisir sebuah *error* pada program. Bayangkan jika kesalahan menuliskan hasil *output* 1000 menjadi 100, sudah pasti hal tersebut akan mengganggu kinerja dari penggunanya. Untuk mengenali kesalahan-kesalahan tersebut maka setidaknya seorang *programmer* mengetahui jenis-jenis kesalahan program. Secara garis besar *error* pada program dibagi menjadi 3 (tiga) yaitu

##### a. *Syntax Error*

Kesalahan yang paling sering ditemukan pada saat membuat program adalah kesalahan sintaks atau *Syntax Error*, dimana perintah atau *statement* yang diketikkan menyalahi aturan pengkodean yang dimiliki oleh bahasa pemrograman yang digunakan.

Sebuah bahasa pemrograman memiliki aturan pengkodean tersendiri yang harus dipatuhi, sebagai contoh pada bahasa pemrograman JavaScript, setiap *statement* diwajibkan diakhiri dengan tanda titik koma (;). Jika tidak dituliskan, maka program akan menampilkan pesan *Syntax Error* pada saat dijalankan. Setiap bahasa pemrograman memiliki *keyword*, yaitu perintah-perintah baku yang digunakan. Sebagai contoh, *keyword* yang umum adalah kondisi *if*, perulangan *for* atau *while*, penulisan fungsi dan lambang aritmatika seperti modulus, pangkat, dan lain-lain. Kesalahan penulisan *keyword* juga merupakan *Syntax Error*.

Kesalahan penulisan parameter pada sebuah *function* atau *procedure* juga termasuk dalam kategori *Syntax Error*, misalnya jika *function* yang digunakan memerlukan parameter sementara *programmer* lupa menuliskan parameter tersebut. *Syntax Error* merupakan jenis kesalahan yang paling sering ditemui, tetapi juga pada umumnya paling mudah untuk ditanggulangi. *Syntax Error* cukup mudah diketahui karena setiap bahasa pemrograman yang digunakan akan menampilkan pesan kesalahan dan menunjukkan baris kesalahan dengan tepat. Pada beberapa bahasa pemrograman, disediakan fasilitas *Auto Syntax Check*, dimana akan muncul sebuah pesan peringatan ketika mengetikkan sintaks yang salah.

b. *Run-time Error*

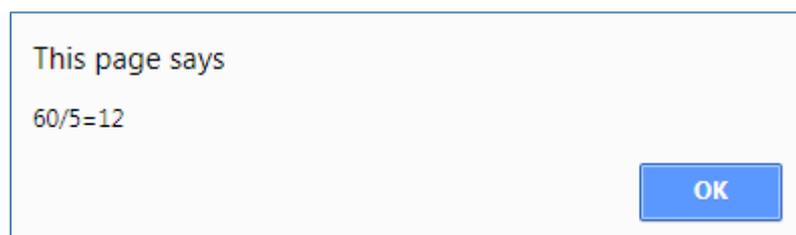
Jenis kesalahan *Run-time Error* terjadi ketika kode program melakukan sesuatu yang tidak dimungkinkan. Contohnya pada saat sebuah aplikasi mencoba mengakses *file* yang tidak terdapat dalam media penyimpanan, atau terjadi kesalahan alokasi ruang memori. Terkadang *Run-time Error* terjadi karena berbagai aspek dan tidak selalu karena kesalahan pemrograman, sebagai contoh jika *programmer* sengaja menghapus beberapa *file* penting yang digunakan oleh suatu aplikasi, maka terdapat kemungkinan akan terjadi *Run-time Error* pada saat aplikasi tersebut dijalankan. *Run-time Error* menyebabkan sebuah program akan langsung berhenti. Walaupun demikian, pencegahan semaksimal mungkin dengan memberikan validasi dan pesan yang *user-friendly* saat terjadi kesalahan pada aplikasi, akan sangat membantu untuk mengetahui mengapa aplikasi tidak berjalan dengan semestinya. Contoh penggunaanya seperti pada *script* program 7.1 berikut ini

```
<html>
  <head>
    <title>Error Programming</title>
  </head>
  <body>
    <script>
      var bilangan1 = 10;
      var bilangan2 = 0;

      alert(bilangan1+bilangan2);
      alert(bilangan1-bilangan2);
      alert(bilangan1*bilangan2);
      alert(bilangan1/bilangan2);
    </script>
  </body>
</html>
```

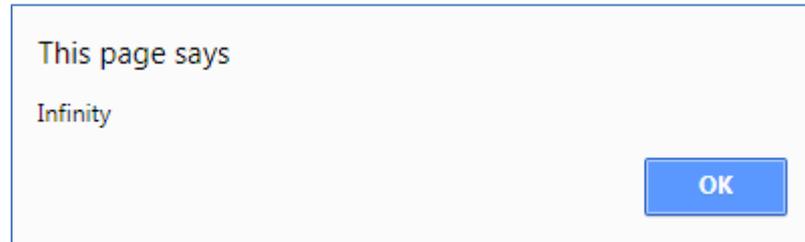
Script 7.1 Contoh program sederhana

Contoh program sederhana pada *Script 7.1* adalah berfungsi untuk menghitung pembagian 2(dua) bilangan. Program tersebut tidak mengalami kesalahan sintaks, karena terbukti bahwa program tersebut dapat dikompilasi dan dijalankan, selain itu juga tidak terdapat kesalahan logika (*logical error*), karena ketika program diisikan bilangan1 dengan nilai 60 dan bilangan2 dengan nilai 5 maka hasil *output* yang tampil adalah 12 seperti pada gambar 6.1 berikut ini



Gambar 7.1 Hasil *output* program pembagian

Akan tetapi program tersebut mengalami *Run-time Error* ketika nilai pada bilangan2 diisikan dengan angka 0(nol), seperti pada gambar 6.2, karena dalam operasi aritmatika tidak diijinkan melakukan pembagian dengan 0(nol)



Gambar 7.2 Output hasil pembagian dengan bilangan 0

### c. *Logical Error*

*Logical Error* merupakan jenis kesalahan yang cukup sulit untuk ditemukan penyebabnya. Karena aplikasi yang mengandung *Logical Error* berjalan tanpa pesan kesalahan, tetapi mengeluarkan hasil yang tidak diharapkan, misalnya jika aplikasi program menghasilkan perhitungan yang salah. *Logical Error* baru dapat diketahui setelah dilakukan *testing* dan meninjau hasilnya. *Logical Error* dapat diperbaiki dengan memeriksa alur program dan nilai variabel yang dihasilkan.

## 2. Program *Debugging*

Sebuah *error* pada aplikasi disebut dengan istilah *bug*, atau dalam Bahasa Inggris berarti kutu atau binatang kecil. Konon istilah *bug* muncul karena ditemukannya binatang kecil yang menyebabkan kerusakan pada sebuah komputer tabung pada tahun 1945. *Bug* aplikasi terdapat pada kode program, yang dapat mengganggu kenyamanan pengguna aplikasi. Pada tingkat tertentu, keberadaan *bug* cukup memberikan efek yang besar.

Mungkin belum melupakan saat dimana orang ramai membicarakan "**Y2K Bug**" atau *bug* tahun 2000, atau munculnya istilah "**Blue screen of Death**" pada sistem operasi Windows, atau "**Kernel Panic**" pada sistem operasi Linux, semua contoh tersebut menunjukkan sebuah *bug* serius dapat mengakibatkan dampak negatif yang cukup besar.

Proses mencari penyebab *bug* disebut dengan *debugging*, yang juga merupakan tugas programmer setelah menerima laporan *bug*. Walaupun demikian, jangan menjadikan pekerjaan sebagai pencari *bug*, untuk itu hanya ada satu cara, minimalkan *bug* pada aplikasi yang dibuat. Apa yang harus dilakukan untuk

menghindari jenis-jenis kesalahan yang telah disebutkan di atas. Bisa jadi tidak ada program yang sempurna, tetapi selalu ada cara untuk mengatasi atau menghindari kesalahan semaksimal mungkin.

g. Selalu Deklarasikan Variabel

*Syntax Error*, bahkan *Logical Error*, mungkin terjadi jika terdapat penulisan variabel yang salah. Sebaiknya mendeklarasikan variabel yang digunakan walaupun bisa jadi bahasa pemrograman yang digunakan mengizinkan untuk tidak melakukan deklarasi variabel. Visual Basic merupakan salah satu bahasa pemrograman yang mengizinkan penggunaan variabel tanpa deklarasi, walaupun demikian disarankan tetap menggunakan deklarasi variabel. Hal tersebut akan memperkecil kesalahan penulisan variabel. Masih dengan contoh Visual Basic yang dapat menambahkan perintah **Option Explicit** pada program untuk mencegah kesalahan tulis pada variabel, jika terdapat variabel yang belum dideklarasikan, maka Visual Basic akan menampilkan pesan kesalahan. Sebaiknya memiliki suatu skema standard untuk pemberian nama variabel dan konsisten dengan penggunaannya. Contohnya berikan nama variabel diawali dengan huruf s jika bertipe data string, misalnya sResult, sTemp, dan lain-lain. Pada Visual Basic maupun beberapa bahasa pemrograman lain yang berorientasi object, kita dapat mendeklarasikan variabel dengan tipe data object. Terdapat berbagai jenis macam object yang dikenal, dan sebaiknya menuliskannya dengan lengkap object yang dimaksud. Misalnya object ListBox, Label, dan lain-lain.

h. Gunakan Variabel Lokal

Sangat disarankan agar selalu menggunakan variabel lokal. Salah satu manfaatnya adalah jika terjadi kesalahan program (terutama *Logical Error*), maka penyebab kesalahan dan solusinya akan lebih mudah ditemukan. Hal ini dikarenakan variabel lokal memiliki ruang lingkup penggunaan yang lebih kecil dibandingkan variabel global, yang dapat diakses oleh procedure yang mana saja. Penggunaan variabel global, sering menimbulkan kerancuan dan memperbesar kemungkinan terjadinya kesalahan tanpa disadari.

i. Kenali Jenis *Bug*

*Bug* yang timbul pada sebuah aplikasi memiliki karakteristik. Karena itu selalu baca dan perhatikan baik-baik pesan kesalahan yang timbul. Beberapa jenis *bug* berdasarkan karakteristiknya adalah sebagai berikut:

i. *Divide By Zero.*

Jika pada sebuah pembagian, pembagi bernilai 0, maka program akan terhenti dan mengalami error.

ii. *Infinite Loop.*

Pengertian loop adalah perulangan, yang sering digunakan dalam teknik pemrograman. Penggunaan loop yang salah dapat mengakibatkan program menjalankan sebuah procedure tanpa akhir.

iii. *Arithmetic overflow or Underflow.*

*Overflow* terjadi saat sebuah perhitungan menghasilkan nilai yang lebih besar daripada nilai yang dapat ditampung oleh media/variabel penyimpan. Sementara *underflow* merupakan kebalikannya. Pada perhitungan aritmatik, hal ini sering ditemukan dan menjadi masalah.

iv. *Exceeding Array Bounds.*

*Array* merupakan variabel berdimensi yang memiliki indeks. Saat program mengakses indeks diluar *array* yang ditentukan, maka akan mengakibatkan *error*.

v. *Access Violation.*

Hal yang terjadi saat sebuah proses mencoba melewati batas yang diijinkan oleh sistem. Misalnya menulis sebuah nilai pada alamat *memory*, segmen, atau media yang diproteksi.

vi. *Memory Leak.*

Penggunaan *memory* yang tidak diinginkan, dapat terjadi karena program gagal melepaskan *memory* yang sudah tidak digunakan.

vii. *Stack Overflow or Underflow.*

*Stack* merupakan struktur data dengan prinsip LIFO (*Last In First Out*), pada program dapat mengimplementasikan logika *stack* untuk suatu tujuan. Tetapi jika *stack* melebihi atau dibawah nilai yang diijinkan oleh program, maka akan timbul kesalahan *Stack Overflow/Underflow*.

viii. *Buffer Overflow.*

*Buffer* merupakan tempat penyimpanan sementara dalam teknik pemrograman. *Buffer Overflow* terjadi jika menyimpan terlalu banyak data yang tidak dapat ditampung oleh *buffer* yang disediakan.

ix. *Deadlock.*

Merupakan suatu kondisi dimana dua atau lebih proses saling menunggu satu sama lain untuk menyelesaikan prosesnya, dan tidak satupun dari proses tersebut yang selesai. *Problem deadlock* sering ditemukan pada *multiprocessing*.

x. *Off By One Error.*

Merupakan istilah untuk menggambarkan perulangan yang terlalu banyak atau terlalu sedikit. Misalnya perulangan yang dikehendaki adalah lima kali, tetapi kenyataan yang terjadi aplikasi mengulang proses tersebut sebanyak empat kali atau enam kali. Kesalahan ini pada umumnya terjadi karena kesalahan logika penulisan kode pada proses perulangan.

xi. Berikan Komentar

Jangan khawatir kode program dipenuhi oleh komentar. Karena akan lebih mudah bagi *programmer* untuk mempelajari lagi kode-kode program yang pernah dibuat dengan membaca komentar. Dengan mengerti kode program dengan baik, maka akan menjadi lebih mudah jika pada suatu saat terdapat Logical Error yang membutuhkan analisa ulang kode program.

xii. Gunakan *Log File*

Informasi proses yang dijalankan aplikasi dan tersimpan pada sebuah *log file* (dapat berupa file text atau table) dapat menjadi informasi yang sangat berguna untuk menganalisa *bug* yang mungkin terjadi. Terutama informasi yang menjelaskan apa yang terjadi sebelum, selama, dan sesudah *bug* terjadi. Untuk menghindari *log file* yang terlalu besar, yaitu dapat memisahkan *log file* terbagi menjadi *log* untuk komponen-komponen utama pada aplikasi. Jangan lupa untuk selalu mencatat waktu (*timestamp*) pada setiap *record*, dan pastikan

dapat menghapus atau melakukan backup pada log file secara periodik.

xiii. Validasi

Tidak semua orang mematuhi aturan yang diterapkan pada aplikasi, karena itu *programmer* harus melakukan validasi untuk data yang dimasukkan oleh pengguna. Misalnya pada suatu form pendaftaran, sebaiknya melakukan validasi untuk *input* yang tidak boleh kosong (*mandatory/required fields*), melakukan pembatasan karakter, dan validasi huruf/angka yang diperlukan.

xiv. Menenal Environment

Saat mengetikkan kode program, menjalankannya, atau melakukan *debug* pada program, maka terdapat *environment* yang berbeda-beda. Terdapat 3 *environment* yang umum dikenal, yaitu:

a. *Design Time*.

Aplikasi yang dikerjakan dilakukan pada saat design time.

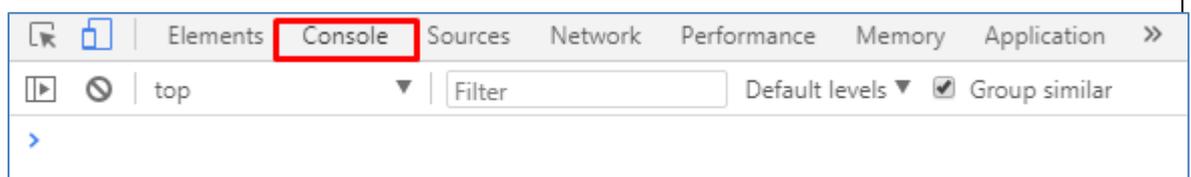
b. *Run Time*.

Saat menjalankan aplikasi.

c. *Break Mode*.

Environment saat melakukan proses debug atau melihat kode program saat program tersebut dijalankan, dapat melihat alur program dan perubahan nilai pada variabel, sehingga dapat menelusuri kesalahan yang terjadi. *Break Mode* terletak diantara *Design Time* dan *Run Time*.

Untuk melakukan *debugging* pada pemrograman JavaScript dikenal dengan istilah *inspect* yaitu dengan membuka jendela console dari masing-masing *browser*, jika menggunakan *browser* Google Chrome dapat dilakukan menekan tombol keyboard **Ctrl + Shift + I**, kemudian memilih tab console seperti pada gambar 7.3 berikut ini



Gambar 7.3 Proses *debugging*

## **B. Keterampilan yang diperlukan dalam Mengkompilasi Program**

1. Identifikasi jenis-jenis kesalahan pada program.
2. Mengoperasikan beberapa jenis *browser*
3. Menguasai sintaks bahasa pemrograman JavaScript.

## **C. Sikap kerja yang diperlukan dalam Mengkompilasi Program**

Harus bersikap secara:

1. Cermat dan teliti dalam menganalisis kode program.
2. Tekun dalam proses pemahaman sintaks.
3. Sesuai dengan kaidah-kaidah bahasa pemrograman.
4. Berpikir analitis serta evaluatif ketika melakukan *trial and error*.
5. Giat dalam mencari *bugs* dan memperbaikinya.

## DAFTAR PUSTAKA

### A. Buku Referensi

- a. Antani, Timms dan Mantyla, JavaScript: Functional Programming for JavaScript Developers, New York, August 2016

### B. Referensi Lainnya

- a. JavaScript Tutorial, alamat url : <https://www.w3schools.com/js>, diakses pada tanggal 5 Maret 2018

## DAFTAR ALAT DAN BAHAN

### A. Daftar Peralatan/Mesin

No.	Nama Peralatan/Mesin	Keterangan
1.	Laptop, infocus, laserpointer	Untuk di ruang teori
2.	Laptop	Untuk setiap peserta
3.	Software Browser Client	Untuk setiap peserta
4.	IDE Sublime Text 3	Untuk setiap peserta

### B. Daftar Bahan

No.	Nama Bahan	Keterangan
1.	Kertas A4	Setiap peserta
2.	Alat Tulis	Setiap peserta

### DAFTAR PENYUSUN

<b>No.</b>	<b>Nama</b>	<b>Profesi</b>
1.	Alun Sujjada, S.Kom.,M.T	1. Dosen STT Atlas Nusantara Malang Prodi Teknik Informatika 2. Asesor LSP STT Atlas Nusantara Malang 3. Anggota APTIKOM (Asosiasi Perguruan Tinggi Komputer)



2

# LAMPIRAN BUKU KERJA



## PENJELASAN UMUM

Pengembangan Keprofesian Berkelanjutan berbasis kompetensi mengharuskan proses pelatihan memenuhi unit kompetensi secara utuh yang terdiri atas pengetahuan, keterampilan, dan sikap kerja. Dalam buku informasi **Mengimplementasikan Pemrograman Terstruktur** telah disampaikan informasi apa saja yang diperlukan sebagai pengetahuan yang harus dimiliki untuk melakukan praktik/keterampilan terhadap unit kompetensi tersebut. Setelah memperoleh pengetahuan dilanjutkan dengan latihan-latihan guna mengaplikasikan pengetahuan yang telah dimiliki tersebut. Untuk itu diperlukan buku kerja **Mengimplementasikan Pemrograman Terstruktur** ini sebagai media praktik dan sekaligus mengaplikasikan sikap kerja yang telah ditetapkan karena sikap kerja melekat pada keterampilan. Adapun tujuan dibuatnya buku kerja ini adalah:

1. Prinsip pelatihan berbasis kompetensi dapat dilakukan sesuai dengan konsep yang telah digariskan, yaitu pelatihan ditempuh elemen kompetensi per elemen kompetensi, baik secara teori maupun praktik;
2. Prinsip praktik *dapat dilakukan setelah dinyatakan kompeten teorinya* dapat dilakukan secara jelas dan tegas;
3. Pengukuran unjuk kerja dapat dilakukan dengan jelas dan pasti.

Ruang lingkup buku kerja ini meliputi pengerjaan tugas-tugas teori dan praktik per elemen kompetensi dan kriteria unjuk kerja berdasarkan SKKNI Subgolongan Pemrograman. Ruang lingkup buku kerja ini meliputi pengerjaan tugas-tugas teori dan praktik per elemen kompetensi dan kriteria unjuk kerja berdasarkan SKKNI Subgolongan Pemrograman.

## DAFTAR ISI

PENJELASAN UMUM .....	2
DAFTAR ISI .....	3
BAB I TUGAS TEORI DAN PRAKTIK .....	4
A. Menggunakan Tipe Data dan Control Program .....	4
1. Tugas Teori I .....	4
2. Tugas Praktik I .....	8
B. Membuat Program Sederhana .....	12
1. Tugas Teori II .....	12
2. Tugas Praktik II .....	15
C. Membuat Program Menggunakan Prosedur dan Fungsi .....	19
1. Tugas Teori III .....	19
2. Tugas Praktik III .....	22
D. Membuat Program Menggunakan Array .....	25
1. Tugas Teori IV .....	25
2. Tugas Praktik IV .....	28
E. Membuat Program Untuk Akses File .....	31
1. Tugas Teori V .....	31
2. Tugas Praktik V .....	33
F. Mengkompilasi Program .....	36
1. Tugas Teori VI .....	36
2. Tugas Praktik VI .....	39
BAB II CEK LIST TUGAS .....	43

## **BAB I**

### **TUGAS TEORI DAN PRAKTIK**

#### **A. MENGGUNAKAN TIPE DATA DAN CONTROL PROGRAM**

##### **1. Tugas Teori I**

Perintah : Jawablah soal di bawah ini

Waktu Penyelesaian : 60 menit

Soal :

1. Sebutkan dan jelaskan tipe-tipe data dalam JavaScript!
2. Jelaskan apa yang dimaksud dengan *control* program!
3. Jelaskan perbedaan penggunaan perulangan *while* dan *do while*!
4. Sebuah kampus "XYZ" mempunyai standar konversi penilaian dari nilai berbentuk angka menjadi huruf. Adapun ketentuannya adalah sebagai berikut:
  - Nilai 100 – 80 mempunyai nilai huruf A
  - Nilai 79 – 75 mempunyai nilai huruf B+
  - Nilai 74 – 70 mempunyai nilai huruf B
  - Nilai 69 – 65 mempunyai nilai huruf C+
  - Nilai 64 – 60 mempunyai nilai huruf C
  - Nilai 59 – 55 mempunyai nilai huruf D
  - Nilai 54 – 0 mempunyai nilai huruf ESelesaikan permasalahan tersebut menggunakan kondisi percabangan!
5. Tuliskan perulangan pada JavaScript yang digunakan untuk menampilkan data kota yang tersimpan dalam tipe data array!

Jawaban:

1. Tipe data JavaScript adalah:

- Numerik
- Boolean
- String
- Array
- Objek

2. *Control* program adalah baris blok kode program yang digunakan untuk mengatur eksekusi program. Pada dasarnya *control* program terdiri dari kondisi percabangan dan kondisi perulangan.

3. Perulangan *while* diproses dengan pengecekan kondisinya terlebih dahulu, kemudian jika sesuai maka blok pernyataan akan dieksekusi berulang-ulang sampai kondisi persyaratannya tidak terpenuhi. Dengan menggunakan *while* maka blok pernyataan akan dijalankan minimal sebanyak 0x. Sedangkan penggunaan perulangan *do while* adalah kebalikannya yaitu blok pernyataan akan dieksekusi terlebih dahulu baru kemudian dilakukan pengecekan kondisi persyaratan, sehingga blok pernyataan akan dieksekusi minimal 1x.

```
4. if (nilai >=80){  
    huruf = "A";  
}  
else if (nilai >=75){  
    huruf = "B+";  
}  
else if (nilai >=70){  
    huruf = "B";  
}  
else if (nilai >=65){  
    huruf = "C+";  
}  
else if (nilai >=60){  
    huruf = "C";  
}  
else if (nilai >=55){  
    huruf = "D";  
}  
else{  
    huruf = "E"  
}
```

```
5. <script>
    var kota =["Jakarta","Surabaya","Padang","Medan","Semarang"];
    for(var i=0;i<=kota.length-1;i++){
        document.write(kota[i] + "<br>");
    }
</script>
```

## Lembar Evaluasi Tugas Teori Menggunakan Tipe Data dan Control Program

Semua kesalahan harus diperbaiki terlebih dahulu sebelum ditandatangani.

No.	Benar	Salah
1.		
2.		
3.		
4.		
5.		

Apakah semua pertanyaan Tugas Teori **Menggunakan Tipe Data dan Control Program** dijawab dengan benar dengan waktu yang telah ditentukan?

YA

TIDAK

	NAMA	TANDA TANGAN
PESERTA	.....	.....
PENILAI	.....	.....

Catatan Penilai:

## 2. Tugas Praktik I

- a. Elemen Kompetensi : Menggunakan Tipe Data dan *Control* Program
- b. Waktu Penyelesaian : 60 menit
- c. Capaian Unjuk Kerja :

Setelah menyelesaikan tugas Menggunakan Tipe Data dan *Control* Program, maka peserta mampu:

- 1) Menentukan tipe data yang sesuai standar
- 2) Menguasai sintaks program yang digunakan sesuai standar
- 3) Menggunakan struktur kontrol program yang dikuasai sesuai standar.

- d. Daftar Alat/Mesin dan Bahan:

NO	NAMA BARANG	SPESIFIKASI	KETERANGAN
A.	ALAT		
1.	Komputer/Laptop	Standar processor Intel Pentium I3	Setiap peserta
2.	Kertas dan alat tulis	Standar	
B.	BAHAN		
1.	Web Browser	Google Chrome, Mozilla Firefox	Setiap peserta
2.	Sublime Text Editor	Versi 2 atau 3	

- e. Indikator Unjuk Kerja (IUK):

- 1) Mampu menentukan tipe data yang sesuai standar
- 2) Mampu menguasai sintaks program yang digunakan sesuai standar
- 3) Mampu menggunakan struktur kontrol program yang dikuasai sesuai standar.

- f. Keselamatan dan Kesehatan Kerja

Keselamatan dan kesehatan kerja yang perlu dilakukan pada waktu melakukan praktik kerja ini adalah:

- 1) Memposisikan sikap badan tegak, mengatur jarak pandang terhadap layar monitor
- 2) Waktu menggunakan komputer, printer, dan alat lainnya mengikuti petunjuknya masing-masing yang sudah ditetapkan.

- g. Standar Kinerja

- 1) Dikerjakan selesai tepat waktu, waktu yang digunakan tidak lebih dari yang ditetapkan.

2) Toleransi kesalahan 5% dari hasil yang harus dicapai, tetapi bukan pada kesalahan kegiatan kritis.

#### h. Tugas

##### Abstraksi Tugas Praktik I

1. Analisa permasalahan berikut kemudian buatlah program menggunakan JavaScript untuk menentukan Total gaji seorang pegawai, berdasarkan Presensi dan Lembur jam kerja. Dengan ketentuan sebagai berikut:
  - a. Jam berupa angka 1-24.
  - b. Jam kerja normal karyawan mulai masuk jam 7 s/d pulang jam 16, jika lebih dianggap lembur
  - c. Kode Supervisor = SPV, HRD = HRD, Teknisi = TKN, Karyawan = KRY
  - d. Perincian komponen dan Perhitungan Gaji:
    - i. Supervisor
      1. Gaji Pokok = 10.000.000
      2. Transport = 1.000.000
      3. Lembur / jam = 10.000
      4. Tunjangan Istri = 300.000
      5. Tunjangan 1 Anak = 100.000, > 1 = 200.000
    - ii. HRD
      1. Gaji Pokok = 5.000.000
      2. Transport = 750.000.000
      3. Lembur / jam = 7.500
      4. Tunjangan Istri = 250.000
      5. Tunjangan 1 Anak = 90.000, > 1 = 190.000
    - iii. Teknisi
      1. Gaji Pokok = 3.000.000
      2. Transport = 500.000
      3. Lembur / jam = 5.000
      4. Tunjangan Istri = 200.000
      5. Tunjangan 1 Anak = 75.000, > 1 = 150.000
    - iv. Karyawan
      1. Gaji Pokok = 1.500.000

- 2. Transport = 250.000
- 3. Lembur / jam = 5.000
- 4. Tunjangan Istri = 200.000
- 5. Tunjangan 1 Anak = 60.000, > 1 = 130.000

e. Jika Presensi Masuk dilakukan lebih dari jam 7, maka gaji pokok dikurangi 100.000 \* jam keterlambatan.

f. Jika Presensi Keluar dilakukan sebelum jam 16, maka gaji pokok dikurangi 100.000 \* jam kekurangan.

#### i. Instruksi Kerja

Setelah membaca abstraksi nomor **h** selanjutnya ikuti instruksi kerja sebagai berikut:

- 1) Siapkan peralatan tulis dan kertas.
- 2) Buatlah tabel perancangan penggajian.
- 3) Buatlah *input form* menggunakan *file* html
- 4) Buatlah penyisipan pada *file* html berupa struktur *control program* menggunakan sintaks JavaScript.
- 5) Jalankan *file* html

j. Daftar Cek Unjuk Kerja Tugas I

NO	DAFTAR TUGAS/INSTRUKSI	POIN YANG DICEK	PENCAPAIAN		PENILAIAN	
			YA	TIDAK	K	BK
1.	Siapkan peralatan tulis dan kertas.	Pemahaman fungsi alat tulis dan kertas				
2.	Buatlah tabel perancangan penggajian.	Kesesuaian dengan permasalahan				
3.	Buatlah <i>input form</i> menggunakan <i>file</i> html	Ketepatan penggunaan jenis komponen form				
4.	Buatlah penyisipan pada <i>file</i> html berupa struktur <i>control program</i> menggunakan sintaks JavaScript.	Penyelesaian permasalahan				
5.	Jalankan file html	Cara menjalankan program				

Apakah semua instruksi kerja tugas praktik Menggunakan tipe data dan control program dilaksanakan dengan benar dengan waktu yang telah ditentukan?

YA

TIDAK

	NAMA	TANDA TANGAN
PESERTA	.....	.....
PENILAI	.....	.....

Catatan Penilai:

## B. MEMBUAT PROGRAM SEDERHANA

### 1. Tugas Teori II

Perintah : Jawablah soal dibawah ini

Waktu Penyelesaian : 45 menit

Soal :

1. Jelaskan definisi dari program!
2. Sebutkan perangkat keras (*hardware*) yang berfungsi sebagai *input* !
3. Sebutkan perangkat keras (*hardware*) yang berfungsi sebagai *output* !
4. Apakah fungsi dari ALU (*Arithmetic Logic Unit*) ?
5. Lengkapilah *function* dari potongan *script* berikut ini, agar program dapat berfungsi untuk melakukan operasi aritmatika 2 bilangan!

```
<html>
  <head>
    <title>Error Programming</title>
  </head>
  <body>
    Bilangan 1 <input type="text" id="bil1"> <br>
    Operator
    <select id="operator">
      <option value="+">+</option>
      <option value="-">-</option>
      <option value="*">*</option>
      <option value="/">/</option>
    </select> <br>
    Bilangan 2 <input type="text" id="bil2"> <br>
    <button value="HITUNG" onclick="hitung()">HITUNG</button> <br>
    <div id="hasil"></div>
    <script>
      function hitung(){
        .....
        .....
        .....
      }
    </script>
  </body>
</html>
```

Jawaban:

1. Program adalah sekumpulan set instruksi yang disusun untuk menyelesaikan permasalahan dan fungsi spesifik pada komputer.
2. Keyboard, joystick, mouse, scanner
3. Monitor, printer, plotter
4. ALU berfungsi untuk mengerjakan proses yang berkaitan dengan operasi perhitungan aritmatika dan logika.

5. 

```
function hitung(){
    var bil1 = parseInt(document.getElementById("bil1").value);
    var bil2 = parseInt(document.getElementById("bil2").value);
    var opr = document.getElementById("operator").value;
    var hasil="";
    if(opr=="+"){
        hasil = bil1 + bil2;
    }
    else if(opr=="-"){
        hasil = bil1 - bil2;
    }
    else if(opr=="*"){
        hasil = bil1 * bil2;
    }
    else if(opr=="/"){
        if(bil2 != 0){
            hasil = bil1 / bil2;
        }
        else{
            alert("Pembagian dengan 0 tidak diperbolehkan");
        }
    }
    document.getElementById("hasil").innerHTML = hasil;
}
```

### Lembar Evaluasi Tugas Teori Membuat Program Sederhana

Semua kesalahan harus diperbaiki terlebih dahulu sebelum ditandatangani.

No.	Benar	Salah
1.		
2.		
3.		
4.		
5.		

Apakah semua pertanyaan Tugas Teori **Membuat Program Sederhana** dijawab dengan benar dengan waktu yang telah ditentukan?

YA

TIDAK

	NAMA	TANDA TANGAN
PESERTA	.....	.....
PENILAI	.....	.....

Catatan Penilai:

## 2. Tugas Praktik II

a. Elemen Kompetensi : **Membuat Program Sederhana**

b. Waktu Penyelesaian : 60 menit

c. Capaian Unjuk Kerja :

Setelah menyelesaikan tugas **Membuat Program Sederhana**, maka peserta mampu:

- 1) Membuat program baca dan tulis dari keyboard ke monitor.
- 2) Menggunakan kontrol percabangan dan pengulangan pada program

d. Daftar Alat/Mesin dan Bahan:

NO	NAMA BARANG	SPESIFIKASI	KETERANGAN
A.	ALAT		
1.	Komputer/Laptop	Standar processor Intel Pentium I3	Setiap peserta
2.	Kertas dan alat tulis	Standar	
B.	BAHAN		
1.	Sublime Text Editor	Versi 2 atau di atasnya	Setiap peserta
2.	Web Browser	Google Chrome, Mozilla, Opera dan lain-lain	

e. Indikator Unjuk Kerja (IUK):

- i. Mampu membuat program baca tulis untuk memasukkan data dari keyboard dan menampilkan ke layar monitor termasuk variasinya sesuai standar masukan/keluaran.
- ii. Mampu menggunakan struktur kontrol percabangan dan perulangan dalam membuat program Mampu menguasai *control* program

f. Keselamatan dan Kesehatan Kerja

Keselamatan dan kesehatan kerja yang perlu dilakukan pada waktu melakukan praktik kerja ini adalah:

- i. Memposisikan sikap badan tegak, mengatur jarak pandang terhadap layar monitor
- ii. Bekerja dalam ruang dengan cahaya penerangan yang baik.
- iii. Waktu menggunakan komputer, printer, dan alat lainnya mengikuti petunjuknya masing-masing yang sudah ditetapkan.

g. Standar Kinerja

- i. Dikerjakan selesai tepat waktu, waktu yang digunakan tidak lebih dari yang ditetapkan.

ii. Toleransi kesalahan 5% dari hasil yang harus dicapai, tetapi bukan pada kesalahan kegiatan kritis.

h. Tugas

Abstraksi Tugas Praktik I

Buatlah program menggunakan JavaScript dengan menerapkan kondisi percabangan dan pengulangan untuk menghitung nilai rata-rata siswa, seperti pada ilustrasi berikut ini:

NAMA	ROMI
NILAI RPL	90
NILAI MAT	60
NILAI IPS	80
NILAI IPA	75
<input type="button" value="HITUNG"/>	
NAMA : ROMI NILAI TOTAL : 305 NILAI RATA-RATA : 76 KETERANGAN : BAIK	

i. Instruksi Kerja

Setelah membaca abstraksi nomor **h** selanjutnya ikuti instruksi kerja sebagai berikut:

- 1) Siapkan peralatan komputer atau laptop masing-masing
- 2) Jalankan *software* Sublime Text Editor
- 3) Buatlah *input form* komponen menggunakan html
- 4) Buatlah *file* JavaScript meliputi
  - Variabel nama
  - Variabel nilai dengan tipe data array
  - Variabel total
  - Variabel rata-rata
  - Variabel keterangan
- 5) Gunakan perulangan untuk menghitung total nilai siswa dalam array.
- 6) Hitung nilai rata-rata siswa

7) Gunakan kondisi percabangan untuk mengkategorikan nilai siswa termasuk sangat baik, baik, cukup dan kurang

j. Daftar Cek Unjuk Kerja Tugas I

NO	DAFTAR TUGAS/INSTRUKSI	POIN YANG DICEK	PENCAPAIAN		PENILAIAN	
			YA	TIDAK	K	BK
1.	Siapkan peralatan komputer atau laptop masing-masing	Kesesuaian dengan langkah-langkah keselamatan kerja				
2.	Jalankan software Sublime Text Editor	Kemampuan mengoperasikan				
3.	Buatlah input form komponen menggunakan html	Kesesuaian pemilihan komponen form dengan permasalahan				
4.	Buatlah file JavaScript	Kesesuaian variabel nama, nilai, total, rata-rata, keterangan				
5.	Gunakan perulangan untuk menghitung total nilai siswa dalam array.	Hasil nilai total menggunakan perulangan				
6.	Hitung nilai rata-rata siswa	Hasil nilai rata-rata siswa				
7.	Gunakan kondisi percabangan untuk mengkategorikan nilai siswa termasuk sangat baik, baik, cukup dan kurang	Penggunaan kondisi percabangan untuk kategori nilai.				

Apakah semua instruksi kerja tugas praktik Menggunakan **Membuat Program Sederhana** dilaksanakan dengan benar dengan waktu yang telah ditentukan?

YA

TIDAK

	NAMA	TANDA TANGAN
PESERTA	.....	.....
PENILAI	.....	.....

Catatan Penilai:

## C. MEMBUAT PROGRAM MENGGUNAKAN PROSEDUR DAN FUNGSI

### 1. Tugas Teori III

Perintah : Jawablah soal dibawah ini

Waktu Penyelesaian : 30 menit

Soal :

1. Jelaskan konsep paradigma pemrograman terstruktur!
2. Sebutkan ciri-ciri pemrograman terstruktur!
3. Jelaskan perbedaan antara prosedur dan fungsi!
4. Lengkapi kode program berikut ini agar fungsi dapat digunakan untuk menghitung luas lingkaran!

```
<script>
    function luasLingkaran(){
        var luas = 3.14 * jari * jari;
        return
    }
</script>
```

5. Berdasarkan kode program berikut ini, jika parameter **a** diisi dengan nilai 5 dan parameter **b** diisi dengan nilai 3, berapakah hasil *output* yang tampil? Jelaskan kegunaan dari fungsi tersebut!

```
<script>
    function hitung(a,b){
        var hasil=1;
        for(i =1; i<=b;i++){
            hasil = hasil * a;
        }
        return hasil;
    }
    alert(hitung(3,3));
</script>
```

Jawaban:

1. Pemrograman terstruktur adalah penyelesaian permasalahan pada program komputer yang disusun menjadi bagian-bagian kecil.
2. Eksekusi baris kode program yang berurutan dari atas ke bawah, membagi instruksi kedalam fungsi dan prosedur.
3. Penggunaan prosedur tidak menghasilkan nilai balik, sedangkan fungsi menghasilkan nilai balik yang dapat diproses lagi sesuai kebutuhan program.

4. 

```
<script>
    function luasLingkaran(jari){
        var luas = 3.14 * jari * jari;
        return luas
    }
</script>
```

5. 125, fungsi yang digunakan untuk menghitung nilai pangkat, parameter a menyatakan bilangan dan parameter b menyatakan pangkat, jadi jika a=5 dan b = 3 maka hasil yang tampil adalah  $5^3 = 125$

## Lembar Evaluasi Tugas Teori Membuat Program Menggunakan Prosedur dan Fungsi

Semua kesalahan harus diperbaiki terlebih dahulu sebelum ditandatangani.

No.	Benar	Salah
1.		
2.		
3.		
4.		
5.		

Apakah semua pertanyaan Tugas Teori **Membuat Program Menggunakan Prosedur dan Fungsi** dijawab dengan benar dengan waktu yang telah ditentukan?

YA

TIDAK

	NAMA	TANDA TANGAN
PESERTA	.....	.....
PENILAI	.....	.....

Catatan Penilai:

## 2. Tugas Praktek III

a. Elemen Kompetensi : **Membuat Program Menggunakan Prosedur dan Fungsi**

b. Waktu Penyelesaian : 60 menit

c. Capaian Unjuk Kerja :

Setelah menyelesaikan tugas Membuat Program Menggunakan Prosedur dan Fungsi, maka peserta mampu

- 1) Membuat program menggunakan prosedur
- 2) Membuat program menggunakan fungsi
- 3) Membuat program menggunakan prosedur dan fungsi secara bersamaan
- 4) Memberikan keterangan pada program.

d. Daftar Alat/Mesin dan Bahan:

NO	NAMA BARANG	SPEKIFIKASI	KETERANGAN
A.	ALAT		
1.	Komputer/Laptop	Standar processor Intel Pentium I3	Setiap peserta
2.	Kertas dan alat tulis	Standar	
B.	BAHAN		
1.	Sublime Text Editor	Versi 2 atau 3	Setiap peserta
2.	Web Browser	Google Chrome, Mozilla Firefox	

e. Indikator Unjuk Kerja (IUK):

- 1) Mampu membuat program menggunakan prosedur sesuai aturan penulisan program
- 2) Mampu membuat program menggunakan fungsi sesuai aturan penulisan program
- 3) Mampu membuat program menggunakan prosedur dan fungsi secara bersamaan sesuai aturan penulisan program
- 4) Mampu memberikan keterangan pada setiap prosedur dan fungsi yang dibuat.

f. Keselamatan dan Kesehatan Kerja

Keselamatan dan kesehatan kerja yang perlu dilakukan pada waktu melakukan praktik kerja ini adalah:

- 1) Memosisikan sikap badan tegak, mengatur jarak pandang terhadap layar monitor

2) Waktu menggunakan komputer, printer, dan alat lainnya mengikuti petunjuknya masing-masing yang sudah ditetapkan.

g. Standar Kinerja

- 1) Dikerjakan selesai tepat waktu, waktu yang digunakan tidak lebih dari yang ditetapkan.
- 2) Toleransi kesalahan 5% dari hasil yang harus dicapai, tetapi bukan pada kesalahan kegiatan kritis.

h. Tugas

Abstraksi Tugas Praktik I

Buatlah program kasir untuk restoran makanan yang digunakan untuk menghitung total pemesanan konsumen!

i. Instruksi Kerja

- 1) Tentukan rancangan menu beserta harganya
- 2) Buatlah *input form* pemesanan menu
- 3) Tentukan prosedur dan fungsi yang akan digunakan.
- 4) Implementasikan prosedur dan fungsi pada *file* JavaScript.
- 5) Buatlah *event handling* pada *input form*.
- 6) Panggil fungsi dan prosedur

j. Daftar Cek Unjuk Kerja Tugas I

NO	DAFTAR TUGAS/INSTRUKSI	POIN YANG DICEK	PENCAPAIAN		PENILAIAN	
			YA	TIDAK	K	BK
1.	Tentukan rancangan menu beserta harganya	Ketepatan pengambilan contoh permasalahan				
2.	Buatlah <i>input form</i> pemesanan menu	Kesesuaian dengan kebutuhan rancangan				
3.	Tentukan prosedur dan fungsi yang akan digunakan.	Desain prosedur dan fungsi				
4.	Implementasikan prosedur dan fungsi pada <i>file</i> JavaScript.	Penggunaan prosedur dan fungsi sesuai sintaks dan permasalahan				
5.	Buatlah <i>event handling</i> pada <i>input form</i> .	Penempatan dan jenis <i>event handling</i>				
6.	Panggil fungsi dan prosedur	Output dari fungsi dan prosedur				

Apakah semua instruksi kerja tugas praktik **Membuat Program Menggunakan Prosedur dan Fungsi** dilaksanakan dengan benar dengan waktu yang telah ditentukan?

YA

TIDAK

	NAMA	TANDA TANGAN
PESERTA	.....	.....
PENILAI	.....	.....

Catatan Penilai:

## D. MEMBUAT PROGRAM MENGGUNAKAN ARRAY

### 1. Tugas Teori IV

Perintah : Jawablah soal dibawah ini

Waktu Penyelesaian : 45 menit

Soal :

1. Jelaskan pengertian dari tipe data Array!
2. Sebutkan dan jelaskan jenis tipe data Array!
3. Tambahkan script pada titik-titik kode program berikut ini, agar dapat berfungsi untuk menjumlah total keseluruhan nilai array!

```
<script>
    var bilangan = [2, 3, 8, 4, 4];
    .....
    for(i=0;i<bilangan.length;i++){
        .....
    }
    alert(total);
</script>
```

4. Selesaikanlah permasalahan pada soal nomor 3 dengan menggunakan perulangan while

Jawaban:

1. Tipe data array adalah kumpulan dari beberapa nilai yang masing-masing mempunyai indeks untuk digunakan sebagai pengakses elemen
2. Tipe data array terbagi menjadi array 1 (satu) dimensi dan multi dimensi. Array 1 (satu) dimensi adalah kumpulan dari beberapa nilai yang pada umumnya adalah sejenis, yang hanya dapat diakses menggunakan 1 (satu) indeks. Array multi dimensi adalah kumpulan dari beberapa nilai yang pada umumnya sejenis, yang dapat diakses menggunakan banyak indeks.

3. 

```
<script>
    var bilangan = [2, 3, 8, 4, 4];
    var total = 0;
    for(i=0;i<bilangan.length;i++){
        total = total + bilangan[i];
    }
    alert(total);
</script>
```

4. 

```
<script>
    var bilangan = [2, 3, 8, 4, 4];
    var total = 0;
    var i = 0;

    while(i < bilangan.length){
        total = total + bilangan[i];
        i++;
    }
    alert(total);
</script>
```

### Lembar Evaluasi Tugas Teori Membuat Program Menggunakan Array

Semua kesalahan harus diperbaiki terlebih dahulu sebelum ditandatangani.

No.	Benar	Salah
1.		
2.		
3.		
4.		

Apakah semua pertanyaan Tugas Teori **Membuat Program Menggunakan Array** dijawab dengan benar dengan waktu yang telah ditentukan?

YA

TIDAK

	NAMA	TANDA TANGAN
PESERTA	.....	.....
PENILAI	.....	.....

Catatan Penilai:

## 2. Tugas Praktek IV

a. Elemen Kompetensi : Membuat Program Menggunakan Array

b. Waktu Penyelesaian : 60 menit

c. Capaian Unjuk Kerja :

Setelah menyelesaikan tugas Membuat Program Menggunakan Array, maka peserta mampu:

- 1) Menentukan dimensi array
- 2) Menentukan tipe data array
- 3) Menentukan panjang array
- 4) Menggunakan pengurutan array

d. Daftar Alat/Mesin dan Bahan :

NO	NAMA BARANG	SPESIFIKASI	KETERANGAN
A.	ALAT		
1.	Komputer/Laptop	Standar processor Intel Pentium I3	Setiap peserta
2.	Kertas dan alat tulis	Standar	
B.	BAHAN		
1.	Sublime Text Editor	Versi 2 atau 3	Setiap peserta
2.	Web Browser	Google Chrome, Mozilla Firefox	

e. Indikator Unjuk Kerja (IUK):

- i. Mampu menentukan dimensi array
- ii. Mampu menentukan tipe data array
- iii. Mampu menentukan Panjang array
- iv. Mampu menggunakan pengurutan array

f. Keselamatan dan Kesehatan Kerja

Keselamatan dan kesehatan kerja yang perlu dilakukan pada waktu melakukan praktik kerja ini adalah:

- i. Memposisikan sikap badan tegak, mengatur jarak pandang terhadap layar monitor
- ii. Waktu menggunakan komputer, printer, dan alat lainnya mengikuti petunjuknya masing-masing yang sudah ditetapkan.

g. Standar Kinerja

- i. Dikerjakan selesai tepat waktu, waktu yang digunakan tidak lebih dari yang ditetapkan.

- ii. Toleransi kesalahan 5% dari hasil yang harus dicapai, tetapi bukan pada kesalahan kegiatan kritis.

h. Tugas

Abstraksi Tugas I

Buatlah program untuk menjumlahkan nilai dari 2 variabel tipe data array dengan ketentuan sebagai berikut

- Penjumlahan dilakukan pada masing-masing indeks yang sama
- Array A dilakukan sortir ascending terlebih dahulu
- Array B dilakukan sortir descending terlebih dahulu

Perhatikan gambar ilustrasi berikut ini:

Array A					ASCENDING	Array A				
10	24	53	5	20		5	10	20	24	53
										
Array B					DESCENDING	Array B				
2	1	32	15	9		32	15	9	2	1
										
						Array Hasil				
						37	25	29	26	4

i. Instruksi Kerja

Setelah membaca abstraksi nomor **h** selanjutnya ikuti instruksi kerja sebagai berikut:

- 1) Buatlah 2 buah tipe data array dengan nama A dan B sesuai dimensi
- 2) Lakukan pengurutan ascending pada array A
- 3) Lakukan pengurutan descending pada array B
- 4) Lakukan penjumlahan masing-masing nilai pada indeks yang sama menggunakan perulangan.
- 5) Tampilkan hasil penjumlahannya.

j. Daftar Cek Unjuk Kerja Tugas I

NO	DAFTAR TUGAS/INSTRUKSI	POIN YANG DICEK	PENCAPAIAN		PENILAIAN	
			YA	TIDAK	K	BK
1.	Buatlah 2 buah tipe data array dengan nama A dan B sesuai dimensi	Kesesuaian dengan sintaks variabel array				
2.	Lakukan pengurutan ascending pada array A	Hasil urutan nilai array secara ascending				
3.	Lakukan pengurutan descending pada array B	Hasil urutan nilai array secara descending				
4.	Lakukan penjumlahan masing-masing nilai pada indeks yang sama menggunakan perulangan	Proses penjumlahan variabel array				
5.	Tampilkan hasil penjumlahannya	Hasil penjumlahan array				

Apakah semua instruksi kerja tugas praktik **Membuat Program Menggunakan Array** dilaksanakan dengan benar dengan waktu yang telah ditentukan?

YA

TIDAK

	NAMA	TANDA TANGAN
PESERTA	.....	.....
PENILAI	.....	.....

Catatan Penilai:

## E. MEMBUAT PROGRAM UNTUK AKSES FILE

### 1. Tugas Teori V

Perintah : Jawablah soal dibawah ini

Waktu Penyelesaian : 30 menit

Soal :

1. Sebutkan dan jelaskan jenis-jenis *file* dan media penyimpanan pada komputer!
2. Sebutkan 3 (tiga) *property* dan *method* dari *library* API FileReader !
3. Sebutkan dan jelaskan indikasi keadaan ketika pembacaan *file* !

Jawaban:

1.
  - File teks berekstensi txt
  - File audio berekstensi wma (windows media audio), advance audio coding, wav (wave audio), mp3, Audio Interchange File Format (AIFF) dan lain-lain.
  - File video berekstensi mpeg, mp4, avi dan lain-lain.
  - File gambar berekstensi jpg, bmp, png, tiff dan lain-lain.
2.
  - FileReader.error yaitu menghasilkan *output* ketika terjadi kesalahan dalam mengakses *file*.
  - FileReader.readAsText() yaitu pembacaan *file* BLOB dan atribut hasil berisi string teks.
  - FileReader.onabort yaitu melakukan *trigger* ketika keluar dari program.
3.
  - 0 mengindikasikan tidak ada data yang dibaca
  - 1 mengindikasikan data sedang diproses
  - 2 mengindikasikan data telah selesai dibaca

### Lembar Evaluasi Tugas Teori Membuat Program untuk Akses File

Semua kesalahan harus diperbaiki terlebih dahulu sebelum ditandatangani.

No.	Benar	Salah
1.		
2.		
3.		

Apakah semua pertanyaan Tugas Teori **Membuat Program untuk Akses File** dijawab dengan benar dengan waktu yang telah ditentukan?

YA

TIDAK

	NAMA	TANDA TANGAN
PESERTA	.....	.....
PENILAI	.....	.....

Catatan Penilai:

## 2. Tugas Praktek V

- a. Elemen Kompetensi : Membuat Program untuk Akses File
- b. Waktu Penyelesaian : 60 menit
- c. Capaian Unjuk Kerja :

Setelah menyelesaikan tugas Mengkompilasi Program, maka peserta mampu

- 1) Membuat program untuk menulis data dalam media penyimpanan.
- 2) Membuat program untuk membaca data dari media penyimpanan.

### d. Daftar Alat/Mesin dan Bahan

NO	NAMA BARANG	SPEKIFIKASI	KETERANGAN
A.	ALAT		
1.	Komputer/Laptop	Standar processor Intel Pentium I3	Setiap peserta
2.	Kertas dan alat tulis	Standar	
B.	BAHAN		
1.	Sublime Text Editor	Versi 2 atau 3	Setiap peserta
2.	Web Browser	Google Chrome, Mozilla Firefox	

### e. Indikator Unjuk Kerja (IUK)

- 1) Mampu membuat program untuk menulis data dalam media penyimpanan.
- 2) Mampu membuat program untuk membaca data dari media penyimpanan.

### f. Keselamatan dan Kesehatan Kerja

Keselamatan dan kesehatan kerja yang perlu dilakukan pada waktu melakukan praktik kerja ini adalah:

- 1) Memposisikan sikap badan tegak, mengatur jarak pandang terhadap layar monitor
- 2) Waktu menggunakan komputer, printer, dan alat lainnya mengikuti petunjuknya masing-masing yang sudah ditetapkan.

### g. Standar Kinerja

- 1) Dikerjakan selesai tepat waktu, waktu yang digunakan tidak lebih dari yang ditetapkan.
- 2) Toleransi kesalahan 5% dari hasil yang harus dicapai, tetapi bukan pada kesalahan kegiatan kritis.

h. Tugas

Abstraksi Tugas Praktik I

Buatlah aplikasi untuk mengakses file teks atau gambar dalam 1 (satu) program.

i. Instruksi Kerja

- 1) Buat *input form* untuk pemilihan file yang akan diakses
- 2) Tampilkan *file* yang dipilih sesuai dengan jenis tipe *file*.
- 3) Tampilkan nama *file*
- 4) Tampilkan ukuran *file*

j. Daftar Cek Unjuk Kerja Tugas I

NO	DAFTAR TUGAS/INSTRUKSI	POIN YANG DICEK	PENCAPAIAN		PENILAIAN	
			YA	TIDAK	K	BK
1.	Buat <i>input form</i> untuk pemilihan file yang akan diakses	Kesesuaian <i>input form</i> dengan permasalahan				
2.	Tampilkan <i>file</i> yang dipilih sesuai dengan jenis tipe <i>file</i> .	<ul style="list-style-type: none"> <li>• Pengecekan jenis tipe file</li> <li>• Penggunaan kondisi percabangan untuk menyeleksi teks atau gambar</li> </ul>				
3.	Tampilkan nama file	Hasil berupa nama <i>file</i>				
4.	Tampilkan ukuran file	Hasil ukuran <i>file</i>				

Apakah semua instruksi kerja tugas praktik **Membuat Program untuk Akses File** dilaksanakan dengan benar dengan waktu yang telah ditentukan?

YA

TIDAK

	NAMA	TANDA TANGAN
PESERTA	.....	.....
PENILAI	.....	.....

Catatan Penilai:

## F. MENGKOMPILASI PROGRAM

### 1. Tugas Teori VI

Perintah : Jawablah soal dibawah ini

Waktu Penyelesaian : 30 menit

Soal :

1. Sebutkan dan jelaskan jenis-jenis kesalahan program!
2. Sebutkan cara-cara untuk meminimalisir kesalahan!
3. Kesalahan apakah, jika terdapat pesan kesalahan seperti gambar berikut ini ?

```
▲ SyntaxError: missing ) after argument list [Learn More]
▲ The character encoding of the HTML document was not declared.
  outside the US-ASCII range. The character encoding of the page
```

4. Kesalahan apakah, jika terdapat pesan kesalahan seperti gambar berikut ini ?

```
▲ ReferenceError: echo is not defined [Learn More]
▲ The character encoding of the HTML document was not declared.
  outside the US-ASCII range. The character encoding of the page
```

Jawaban:

1.	<p>a. <i>Syntax Error</i></p> <p>Kesalahan yang paling sering ditemukan pada saat membuat program adalah kesalahan sintaks atau <i>Syntax Error</i>, dimana perintah atau <i>statement</i> yang diketikkan menyalahi aturan pengkodean yang dimiliki oleh bahasa pemrograman yang digunakan</p> <p>b. <i>Logical Error</i></p> <p><i>Error</i> yang disebabkan karena kesalahan logika saat membuat program</p> <p>c. <i>Run-time Error</i></p> <p><i>Error</i> yang terjadi ketika kode program melakukan sesuatu yang tidak dimungkinkan</p>
2.	Selalu mendeklarasikan variabel, menggunakan variabel lokal, mengenali jenis <i>bug</i> ,
3.	<ul style="list-style-type: none"><li>• Terdapat pemanggilan fungsi yang tidak sesuai parameternya.</li></ul>

	<ul style="list-style-type: none"><li>• Terdapat kode program yang jumlah kurung buka tidak sama dengan kurung tutup.</li><li>• Kesalahan penulisan string dalam fungsi</li></ul>
4.	<ul style="list-style-type: none"><li>• Kesalahan <i>scope</i> atau ruang lingkup.</li><li>• Variabel tidak dideklarasikan</li></ul>

### Lembar Evaluasi Tugas Teori Mengkompilasi Program

Semua kesalahan harus diperbaiki terlebih dahulu sebelum ditandatangani.

No.	Benar	Salah
1.		
2.		
3.		
4.		

Apakah semua pertanyaan Tugas Teori **Mengkompilasi Program** dijawab dengan benar dengan waktu yang telah ditentukan?

YA

TIDAK

	NAMA	TANDA TANGAN
PESERTA	.....	.....
PENILAI	.....	.....

Catatan Penilai:

## 2. Tugas Praktek VI

k. Elemen Kompetensi : Mengkompilasi Program

l. Waktu Penyelesaian : 60 menit

m. Capaian Unjuk Kerja :

Setelah menyelesaikan tugas Mengkompilasi Program, maka peserta mampu

- 1) Mengoreksi kesalahan program
- 2) Membebaskan kesalahan syntax dalam program

n. Daftar Alat/Mesin dan Bahan

NO	NAMA BARANG	SPESIFIKASI	KETERANGAN
A.	ALAT		
1.	Komputer/Laptop	Standar processor Intel Pentium I3	Setiap peserta
2.	Kertas dan alat tulis	Standar	
B.	BAHAN		
1.	Sublime Text Editor	Versi 2 atau 3	Setiap peserta
2.	Web Browser	Google Chrome, Mozilla Firefox	

o. Indikator Unjuk Kerja (IUK)

- 1) Mampu mengoreksi kesalahan program
- 2) Mampu membebaskan kesalahan syntax dalam program

p. Keselamatan dan Kesehatan Kerja

Keselamatan dan kesehatan kerja yang perlu dilakukan pada waktu melakukan praktik kerja ini adalah:

- 1) Memosisikan sikap badan tegak, mengatur jarak pandang terhadap layar monitor
- 2) Waktu menggunakan komputer, printer, dan alat lainnya mengikuti petunjuknya masing-masing yang sudah ditetapkan.

q. Standar Kinerja

- 1) Dikerjakan selesai tepat waktu, waktu yang digunakan tidak lebih dari yang ditetapkan.
- 2) Toleransi kesalahan 5% dari hasil yang harus dicapai, tetapi bukan pada kesalahan kegiatan kritis.

r. Tugas

Abstraksi Tugas Praktik I

Buatlah *file* html seperti berikut ini

```
<!doctype html>
<html>
  <body>
    <script>
      var keterangan = function(score) {
        if (score >= 90)
          return 'Sangat Baik!';
        }

        elseif (score >= 70) {
          return 'Baik!';
        }

        elseif (score >= 60) {
          return 'Cukup!';
        }

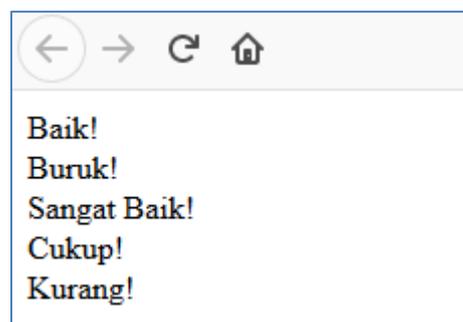
        elseif (score >= 50) {
          return 'Kurang!';
        }

        elseif (score >= 0) {
          return 'Buruk!';
        }
      }

      var nilai =[80,35,100,65,50];

      for(i=1;i<=nilai.length;i++){
        document.write(keterangan(nilai[i])+"<br>");
      }
    </script>
  </body>
</html>
```

Kemudian lakukan *debugging* untuk memperbaiki kesalahannya, sehingga ketika program dijalankan akan menampilkan hasil *output* seperti berikut ini



s. Instruksi Kerja

- 1) Buat *file* html yang didalamnya terdapat *script* dari JavaScript
- 2) Jalankan *file* tersebut menggunakan *web browser*
- 3) Lakukan *debugging* dengan membuka jendela *console* pada web browser
- 4) Analisa jenis kesalahan yang terdapat pada jendela *console*.
- 5) Perbaiki kesalahannya
- 6) Program bebas dari kesalahan sintaks dapat dihasilkan

t. Daftar Cek Unjuk Kerja Tugas I

NO	DAFTAR TUGAS/INSTRUKSI	POIN YANG DICEK	PENCAPAIAN		PENILAIAN	
			YA	TIDAK	K	BK
1.	Buat <i>file</i> html yang didalamnya terdapat <i>script</i> dari JavaScript	Cara membuat <i>file</i>				
2.	Jalankan file tersebut menggunakan web browser	Cara menjalankan program JavaScript				
3.	Lakukan debugging dengan membuka jendela console pada web browser	Jendela <i>console</i> terbuka				
4.	Analisa jenis kesalahan yang terdapat pada jendela <i>console</i> .	Analisa jenis kesalahan				
5.	Perbaiki kesalahannya	Perbaikan kesalahan				
6.	Program bebas dari kesalahan sintaks dapat dihasilkan	Program dapat dijalankan dan sesuai permasalahan				

Apakah semua instruksi kerja tugas praktik **Mengkompilasi Program** dilaksanakan dengan benar dengan waktu yang telah ditentukan?

YA

TIDAK

	NAMA	TANDA TANGAN
PESERTA	.....	.....
PENILAI	.....	.....

Catatan Penilai:

**BAB II**  
**CEK LIS TUGAS**

NO	TUGAS UNJUK KERJA	PENILAIAN		TANGGAL
		K	BK	
1.	Menggunakan tipe data dan control program			
2.	Membuat program sederhana			
3.	Membuat program menggunakan prosedur dan fungsi			
4.	Membuat program menggunakan array			
5.	Membuat program untuk akses file			
6.	Mengkompilasi Program			

Apakah semua tugas unjuk kerja **Mengimplementasikan Pemrograman Terstruktur** telah dilaksanakan dengan benar dan dalam waktu yang telah ditentukan?

YA

TIDAK

	NAMA	TANDA TANGAN
PESERTA	.....	.....
PENILAI	.....	.....

Catatan Penilai:



**3**

**LAMPIRAN  
BUKU PENILAIAN**



## PENJELASAN UMUM

Buku penilaian untuk unit kompetensi Mengimplementasikan Pemrograman Terstruktur dibuat sebagai konsekuensi logis dalam pelatihan berbasis kompetensi yang telah menempuh tahapan penerimaan pengetahuan, keterampilan, dan sikap kerja melalui buku informasi dan buku kerja. Setelah latihan-latihan (*exercise*) dilakukan berdasarkan buku kerja maka untuk mengetahui sejauh mana kompetensi yang dimilikinya perlu dilakukan uji komprehensif secara utuh per unit kompetensi dan materi uji komprehensif itu ada dalam buku penilaian ini.

Adapun tujuan dibuatnya buku penilaian ini, yaitu untuk menguji kompetensi peserta pelatihan setelah selesai menempuh buku informasi dan buku kerja secara komprehensif dan berdasarkan hasil uji inilah peserta akan dinyatakan kompeten atau belum kompeten terhadap unit kompetensi Mengimplementasikan Pemrograman Terstruktur. Metoda Penilaian yang dilakukan meliputi penilaian dengan opsi sebagai berikut:

1. Metoda Penilaian Pengetahuan

- a. Tes Tertulis

Untuk menilai pengetahuan yang telah disampaikan selama proses pelatihan terlebih dahulu dilakukan tes tertulis melalui pemberian materi tes dalam bentuk tertulis yang dijawab secara tertulis juga. Untuk menilai pengetahuan dalam proses pelatihan materi tes disampaikan lebih dominan dalam bentuk obyektif tes, dalam hal ini jawaban singkat, menjodohkan, benar-salah, dan pilihan ganda. Tes essay bisa diberikan selama tes essay tersebut tes essay tertutup, tidak essay terbuka, hal ini dimaksudkan untuk mengurangi faktor subyektif penilai.

- b. Tes Wawancara

Tes wawancara dilakukan untuk menggali atau memastikan hasil tes tertulis sejauh itu diperlukan. Tes wawancara ini dilakukan secara perseorangan antara penilai dengan peserta uji/peserta pelatihan. Penilai sebaiknya lebih dari satu orang.

## 2. Metoda Penilaian Keterampilan

### a. Tes Simulasi

Tes simulasi ini digunakan untuk menilai keterampilan dengan menggunakan media bukan yang sebenarnya, misalnya menggunakan tempat kerja tiruan (bukan tempat kerja yang sebenarnya), obyek pekerjaan disediakan atau hasil rekayasa sendiri, bukan obyek kerja yang sebenarnya.

### b. Aktivitas Praktik

Penilaian dilakukan secara sebenarnya, di tempat kerja sebenarnya dengan menggunakan obyek kerja sebenarnya.

## 3. Metoda Penilaian Sikap Kerja

### a. Observasi

Untuk melakukan penilaian sikap kerja digunakan metoda observasi terstruktur, artinya pengamatan yang dilakukan menggunakan lembar penilaian yang sudah disiapkan sehingga pengamatan yang dilakukan mengikuti petunjuk penilaian yang dituntut oleh lembar penilaian tersebut. Pengamatan dilakukan pada waktu peserta uji/peserta pelatihan melakukan keterampilan kompetensi yang dinilai karena sikap kerja melekat pada keterampilan tersebut.

## DAFTAR ISI

PENJELASAN UMUM .....	2
DAFTAR ISI .....	4
BAB I PENILAIAN TEORI .....	5
A. Lembar Penilaian Teori .....	5
B. Ceklis Penilaian Teori.....	12
BAB II PENILAIAN PRAKTIK .....	13
A. Lembar Penilaian Praktik .....	13
B. Ceklis Aktivitas Praktik .....	14
BAB III PENILAIAN SIKAP KERJA .....	15
A. Ceklis Penilaian Sikap Kerja .....	15
LAMPIRAN .....	16
Lampiran 1. Kunci Jawaban .....	17

## BAB I PENILAIAN TEORI

### A. Lembar Penilaian Teori

Unit Kompetensi : Mengimplementasikan Pemrograman Terstruktur  
Diklat :  
Waktu : 60 menit

#### PETUNJUK UMUM

1. Jawablah materi tes ini pada lembar jawaban/kertas yang sudah disediakan.
2. Modul terkait dengan unit kompetensi agar disimpan.
3. Bacalah materi tes secara cermat dan teliti.

#### Isian

Jawablah pertanyaan dibawah ini dengan singkat dan tepat!

1. Sebutkan jenis-jenis tipe data yang digunakan dalam pemrograman!
2. Tuliskan sintaks penulisan **if** untuk 1 (satu) kondisi!
3. Perhatikan gambar berikut ini

```
<script>
    var nilai= 13 % 7;
    var keterangan;

    if(nilai==6){
        keterangan=" Sekarang Hari Sabtu";
    }

    if(nilai > 0){
        keterangan=" Nilai Lebih besar dari 0";
    }

    if(nilai > 1){
        keterangan=" Nilai Lebih besar dari 1";
    }

    if(nilai > 2){
        keterangan=" Nilai Lebih besar dari 2";
    }

    if(nilai > 3){
        keterangan=" Nilai Lebih besar dari 3";
    }

    document.write(keterangan);
</script>
```

Berdasarkan *listing* program tersebut, *output* apakah yang akan tampil pada *web browser*?

4. Berdasarkan *listing* program berikut ini, *output* apakah yang akan tampil pada *web browser*?

```
<script>
var propinsi = ["Jawa Timur","Jawa Tengah","Sulawesi Selatan","Sumatera Utara"];
var posisi = 1;
var ibukota;
switch(propinsi[posisi+1]){
    case "Jawa Timur" :
        ibukota = "Surabaya";
        break;
    case "Jawa Tengah" :
        ibukota = "Semarang";
        break;
    case "Sulawesi Selatan" :
        ibukota = "Makassar";
        break;
    case "Sumatera Utara" :
        ibukota = "Medan";
        break;
}

document.write("Ibu kota propinsi " + propinsi[posisi+1] + " adalah " + ibukota);
</script>
```

5. *Output* apakah yang akan ditampilkan pada *listing* program berikut ini

```
<script>
var propinsi = ["Jawa Timur","Jawa Tengah","Sulawesi Selatan","Sumatera Utara"];

for (i=1;i<=propinsi.length-1;i++){
    document.write(propinsi[i] + "<br>");
}
</script>
```

6. Perhatikan *script* program berikut ini

```
<script>
var propinsi = ["Jawa Timur","Jawa Tengah","Sulawesi Selatan","Sumatera Utara"];
var a=0;
do{
    document.write(propinsi[a]);
    a = a+1;
}while(a > propinsi.length);
</script>
```

*output* apakah yang akan tampil pada *web browser*?

7. Terletak pada apakah perbedaan dasar antara prosedur dan fungsi?  
8. Berapakah panjang array jika terdapat *script* **var arr = [0,1,2,3,4,5] ?**  
9. JavaScript mempunyai library API untuk mengakses file yaitu ...  
10. Termasuk error apakah kode program berikut ini

```
<script>
var propinsi = ["Jawa Timur","Jawa Tengah","Sulawesi Selatan","Sumatera Utara"];
var a=0;
do while(a > propinsi.length){
    document.write(propinsi[a]);
    a = a+1;
}
</script>
```

### Benar-Salah

Nyatakan pernyataan di bawah ini benar atau salah dengan cara menulis huruf B jika Benar dan huruf S jika Salah.

- B S 1. Penulisan nilai variabel diawali dengan 0x adalah heksadesimal.
- B S 2. Tanda % adalah operator aritmatika untuk sisa bagi.
- B S 3. Fungsi *else* adalah menjalankan blok kode program jika kondisi pertama benar.
- B S 4. Perulangan *while* akan dijalankan minimal 1x.
- B S 5. Pembuatan array dengan *keywords new* adalah `var data = new [1,2,3];`
- B S 6. Fungsi pada JavaScript tidak harus diberi nama fungsi.
- B S 7. Indeks pada array dimulai dari ke 0.
- B S 8. Angka 2 pada pemanggilan `FileReader.readyState` adalah data sedang diproses.
- B S 9. Kesalahan yang sulit dideteksi adalah *logic error*.
- B S 10. Proses *debugging* pada JavaScript menggunakan jendela *console* pada *web browser*.

### Pilihan Ganda

Jawablah pertanyaan/ Pernyataan di bawah ini dengan cara memilih pilihan jawaban yang tepat dan menuliskan huruf A/B/C/D yang sesuai dengan pilihan tersebut.

1.	Manakah diantara penulisan variabel berikut ini yang tepat?			
	a.	var _A = 50;	b.	var switch = 90;
	c.	var -nama = 'Andi';	d.	var _a = 100
2.	<i>Script</i> manakah yang menghasilkan output <b>Selasa</b>			
	a.	<pre>&lt;script&gt;     var hari = 2;     var ket;     if (hari==2){         ket="Selasa";     }      if (hari&gt;1){         ket="Rabu";     }     alert(ket); &lt;/script&gt;</pre>	b.	<pre>&lt;script&gt;     var hari = 5;     var ket;     if (hari==1){         ket="Senin";     }      else{         ket="Selasa";     }     alert(ket); &lt;/script&gt;</pre>
	c.	<pre>&lt;script&gt;     var hari = 2;     var ket;     if (hari==2){         ket="Selasa";         if(hari&gt;1){             ket = "Rabu";         }     }     alert(ket); &lt;/script&gt;</pre>	d.	<pre>&lt;script&gt;     var hari = 2;     var ket;     if (hari==2){         ket="Selasa";         ket="";     }      alert(ket); &lt;/script&gt;</pre>
3.	Hasil <i>output</i> apakah yang tampil pada <i>script</i> berikut ini?			
	<pre>&lt;script&gt; var t=1; for(i=1;i&lt;=10;i++){     document.write(i * t + ',');     t = t * -1; } &lt;/script&gt;</pre>			
	a.	-1,-2,-3,-4,-5,-6,-7,-8,-9,-10	b.	-1,2,-3,4,-5,6,-7,8,-9,10
	c.	10,9,8,7,6,5,4,3,2,1	d.	1,-2,3,-4,5,-6,7,-8,9,-10

4.	Hasil <i>output</i> apakah yang tampil pada <i>script</i> berikut ini? <pre>                 &lt;script&gt;                 var data=10;                 do{                 data=data+1;                 document.write(data);                 }while(data&lt;10);                 &lt;/script&gt;                 </pre>		
	a. null	b. 11	
	c. 10	d. 11,12	
5.	Pada <i>script</i> berikut ini, baris manakah yang termasuk dalam <i>input</i> ? <pre>                 1 &lt;script&gt;                 2   var nilai;                 3   nilai = document.getElementById("angka").value;                 4   nilaiAngka = parseInt(nilai);                 5   var hasil = nilaiAngka * 10;                 6   alert(hasil);                 7 &lt;/script&gt;                 </pre>		
	a. Baris 2	b. Baris 3, 4, 5	
	c. Baris 3	d. Baris 6	
6.	Manakah diantara pernyataan berikut ini yang benar?		
	a. Pada JavaScript sebuah fungsi juga merupakan objek.	b. Parameter pada fungsi tidak boleh kosong	
	c. Pada JavaScript sebuah fungsi harus diberi nama	d. JavaScript membedakan antara fungsi dan prosedur	
7.	Apakah <i>output</i> yang akan ditampilkan pada <i>script</i> berikut ini? <pre>                 &lt;script&gt;                 var nilai=[80,100,30,20];                 var total=0;                 for(i=0;i&lt;3;i++){                 delete nilai[1];                 if(nilai[i]!=null)                 total = total + nilai[i];                 }                 alert(total);                 &lt;/script&gt;                 </pre>		
	a. 130	b. 30	
	c. 110	d. 150	

8. Berdasarkan *script* berikut ini

```
<script>
  var a=[70,55,63,21];
  var t=0;
  for(j=0;j<4;j++){
    .....
    .....
  }
  alert(rata2);
</script>
```

Manakah yang tepat untuk ditambahkan pada titik-titik agar program berfungsi untuk menghitung nilai rata-rata dari *array*?

a.	<code>t = t + a[j]; var rata2 = t / a.length-1;</code>	b.	<code>t = t + a[j-1]; var rata2 = t / a.length;</code>
c.	<code>t = t + a[j]; var rata2 = t / a.length;</code>	d.	<code>a[j] = t + a[j]; var rata2 = t / a.length-1;</code>

9. Sebuah jenis *error* yang terjadi ketika perintah atau *statement* yang diketikkan menyalahi aturan pengkodean yang dimiliki oleh bahasa pemrograman

a.	<i>Syntax Error</i>	b.	<i>Logic Error</i>
c.	<i>Class Error</i>	d.	<i>Run-time Error</i>

10. Jika menemui pesan kesalahan seperti berikut ini

```
▲ ▶ TypeError: [70, 55, 63, 21] is not a constructor [Learn More]
▲ The character encoding of the HTML document was not declared. T
  outside the US-ASCII range. The character encoding of the page
```

Adalah termasuk jenis error .....

a.	<i>Syntax Error</i>	b.	<i>Constructor Error</i>
c.	<i>Type Error</i>	d.	<i>Run-time Error</i>

## Essay

Jawablah pertanyaan-pertanyaan di bawah ini dengan jelas dan benar!

1. Buatlah fungsi untuk membuat matriks identitas, yaitu matriks diagonal yang semua elemen pada diagonal utamanya adalah 1 (satu) dan lainnya 0 (nol).

Contoh: 
$$\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

2. Buatlah program menggunakan JavaScript untuk konversi suhu dari celcius ke fahrenheit dan sebaliknya!

### B. Ceklis Penilaian Teori

NO. KUK	NO. SOAL	KUNCI JAWABAN	JAWABAN PESERTA	PENILAIAN		KETERANGAN
				K	BK	
	<b>Isian</b>					
1.1	1.	String, Boolean, float, integer, numerik, character				
1.3	2.	If(condition){ statement }				
2.2	3.	Nilai lebih besar dari 3				
4.1	4.	Ibu kota propinsi Sulawesi Selatan adalah Makassar				
4.2	5.	Jawa Tengah Sulawesi Selatan Sumatera Utara				
4.3	6.	Jawa Timur				
3.1	7.	Nilai Balika tau return value				
4.3	8.	6				
5.1	9.	FileReader				
6.1	10.	Syntax Error				
	<b>B-S</b>					
1.1	1.	B				
1.3	2.	B				
2.1	3.	S				
2.2	4.	S				
2.2	5.	S				
2.2	6.	B				
3.3	7.	B				
4.3	8.	S				
5.2	9.	B				
6.1	10.	B				
	<b>PG</b>					
1.1	1.	A				
1.3	2.	B				
1.3	3.	D				
2.2	4.	B				
2.1	5.	B				
3.1	6.	A				
4.1	7.	C				
4.2	8.	C				
6.1	9.	A				
6.2	10.	A				
	<b>Essay</b>					
	1.	Terlampir				
	2.	Terlampir				

## BAB II PENILAIAN PRAKTIK

### A. Lembar Penilaian Praktik

Tugas Unjuk Kerja Mengimplementasikan Pemrograman Terstruktur

1. Waktu : 120 menit
2. Alat : laptop/PC, alat tulis, Kertas
3. Bahan : Sublime Text Editor, Web Browser
4. Indikator Unjuk Kerja
  - a. Mampu menggunakan tipe data dan control program.
  - b. Mampu membuat program sederhana.
  - c. Mampu membuat program menggunakan prosedur dan fungsi.
  - d. Mampu membuat program menggunakan array.
  - e. Mampu membuat program untuk akses file.
  - f. Mampu mengkompilasi program.
5. Standar Kinerja
  - a. Selesai dikerjakan tidak melebihi waktu yang telah ditetapkan.
  - b. Toleransi kesalahan 5% (lima persen), tetapi tidak pada aspek kritis.
6. Instruksi Kerja

Abstraksi tugas:

Buatlah sebuah file html dan JavaScript dengan beberapa fungsi atau prosedur yang digunakan untuk rental mobil.

Untuk menyelesaikan tugas ini, ikuti instruksi selanjutnya di bawah ini.

  - a. Tentukan variabel dan nilai untuk rental mobil.
  - b. Buatlah kondisi percabangan untuk harga dan merk.
  - c. Tentukan fungsi dan prosedur yang digunakan
  - d. Simpanlah data mobil menggunakan *file* teks
  - e. Akseslah data mobil pada *file* teks kedalam variabel array.
  - f. Gunakan pengulangan untuk mengakses data mobil.
  - g. Buatlah *input form* menggunakan *file* html.
  - h. Implementasikan rancangan pada *file* JavaScript
  - i. Lakukan *debugging* jika terdapat kesalahan.

**B. Ceklis Aktivitas Praktik**

Kode Unit Kompetensi : J.620100.017.02

Judul Unit Kompetensi : Mengimplementasikan Pemrograman Terstruktur

Nama Peserta/Asesi : .....

INDIKATOR UNJUK KERJA	TUGAS	HAL-HAL YANG DIAMATI	PENILAIAN	
			K	BK
1. Menggunakan tipe data dan control program	1.1 Tentukan variabel dan nilai untuk rental mobil. 1.2 Buatlah kondisi percabangan untuk harga dan merk.	<ul style="list-style-type: none"> <li>Variabel dan nilai yang digunakan</li> <li>Penggunaan kondisi percabangan</li> </ul>		
2. Membuat program sederhana	2.1 Buatlah <i>input form</i> menggunakan <i>file</i> html. 2.2 Implementasikan rancangan pada <i>file</i> JavaScript	<ul style="list-style-type: none"> <li>Pemilihan <i>input form</i> sesuai permasalahan</li> <li>Implementasi pada JavaScript sesuai sintaks.</li> </ul>		
3. Membuat program menggunakan prosedur dan fungsi	3.1 Tentukan fungsi dan prosedur yang digunakan	Penggunaan fungsi dan prosedur Bersama-sama		
4. Membuat program menggunakan array	4.1 Gunakan pengulangan untuk mengakses data mobil.	Ketepatan penggunaan pengulangan.		
5. Membuat program untuk akses file	5.1 Akseslah data mobil pada <i>file</i> teks kedalam variabel array.	Pembuatan <i>file</i> teks		
6. Mengkompilasi Program	6.1 Lakukan <i>debugging</i> jika terdapat kesalahan.	Cara melakukan <i>debugging</i> dan hasil kode program tanpa kesalahan.		

Catatan:

.....

.....

.....

.....

Tanda Tangan Peserta Pelatihan : .....

Tanda Tangan Instruktur : .....

**BAB III**  
**CEKLIS PENILAIAN SIKAP KERJA**

**A. CEKLIS PENILAIAN SIKAP KERJA**

**Mengimplementasikan Pemrograman Terstruktur**

INDIKATOR UNJUK KERJA	NO. KUK	K	BK	KETERANGAN
1. Harus benar dan sesuai dengan kaidah bahasa pemrograman	1.1			
2. Harus sesuai dengan kebutuhan permasalahan	1.2			
3. Harus sesuai dengan kebutuhan permasalahan	1.3			
4. Harus efisien	2.1			
5. Cermat, tekun dan teliti	2.2			
6. Cermat, tekun dan teliti	2.3			
7. Harus berfikir analitis dan sesuai kaidah Bahasa pemrograman	3.1			
8. Harus berfikir analitis dan sesuai kaidah Bahasa pemrograman	3.2			
9. Harus berfikir analitis dan sesuai kaidah Bahasa pemrograman	3.3			
10. Sesuai kaidah Bahasa pemrograman	4.1			
11. Cermat, tekun dan teliti	5.1			
12. Cermat, tekun dan teliti	5.2			

Catatan:

.....

.....

.....

.....

.....

.....

Tanda Tangan Peserta : .....

Tanda Tangan Instruktur : .....

## LAMPIRAN-LAMPIRAN

## LAMPIRAN 1

### Kunci Jawaban Penilaian Teori

NO. KUK	NO. SOAL	KUNCI JAWABAN
	<b>Isian</b>	
1.1	1.	String, Boolean, float, integer, numerik, character
1.3	2.	If(condition){ statement }
2.2	3.	Nilai lebih besar dari 3
4.1	4.	Ibu kota propinsi Sulawesi Selatan adalah Makassar
4.2	5.	Jawa Tengah Sulawesi Selatan Sumatera Utara
4.3	6.	Jawa Timur
3.1	7.	Nilai Balik tau return value
4.3	8.	6
5.1	9.	FileReader
6.1	10.	Syntax Error
	<b>B-S</b>	
1.1	1.	B
1.3	2.	B
2.1	3.	S
2.2	4.	S
2.2	5.	S
2.2	6.	B
3.3	7.	B
4.3	8.	S
5.2	9.	B
6.1	10.	B
	<b>PG</b>	
1.1	1.	A
1.3	2.	B
1.3	3.	D
2.2	4.	B
2.1	5.	B
3.1	6.	A
4.1	7.	C
4.2	8.	C
6.1	9.	A
6.2	10.	A
	<b>Essay</b>	
	1.	Terlampir
	2.	Terlampir

## Jawaban Soal Essay

1.

```
<html>
<body>
<script>
  var matriksIdentitas = function(baris,kolom){
    for(i=0;i<baris;i++){
      for(j=0;j<kolom;j++){
        if(i==j){
          document.write("1");
        }
        else{
          document.write("0");
        }
      }
      document.write("<br>");
    }
  }

  matriksIdentitas(5,5);
</script>

</body>
</html>
```

2.

```
<html>
<body>
<script>
  var konversiSuhu = function(dari,ke,suhu){
    var hasil=0;
    if(dari=='C' && ke=='F'){
      hasil = ((9/5) * suhu) + 32;
    }
    else if(dari=='F' && ke=='C'){
      hasil = (5/9) * (suhu-32);
    }
    else{
      hasil = "Terjadi kesalahan parameter";
    }

    return hasil;
  }

  document.write(konversiSuhu('C','F',190));
</script>
</body>
</html>
```



[kptk.or.id](http://kptk.or.id)



[instagram.com/lp3tk](https://www.instagram.com/lp3tk)



[facebook.com/lp3tk](https://www.facebook.com/lp3tk)



[twitter.com/lp3tk](https://twitter.com/lp3tk)



[youtube.com/lp3tk](https://www.youtube.com/lp3tk)