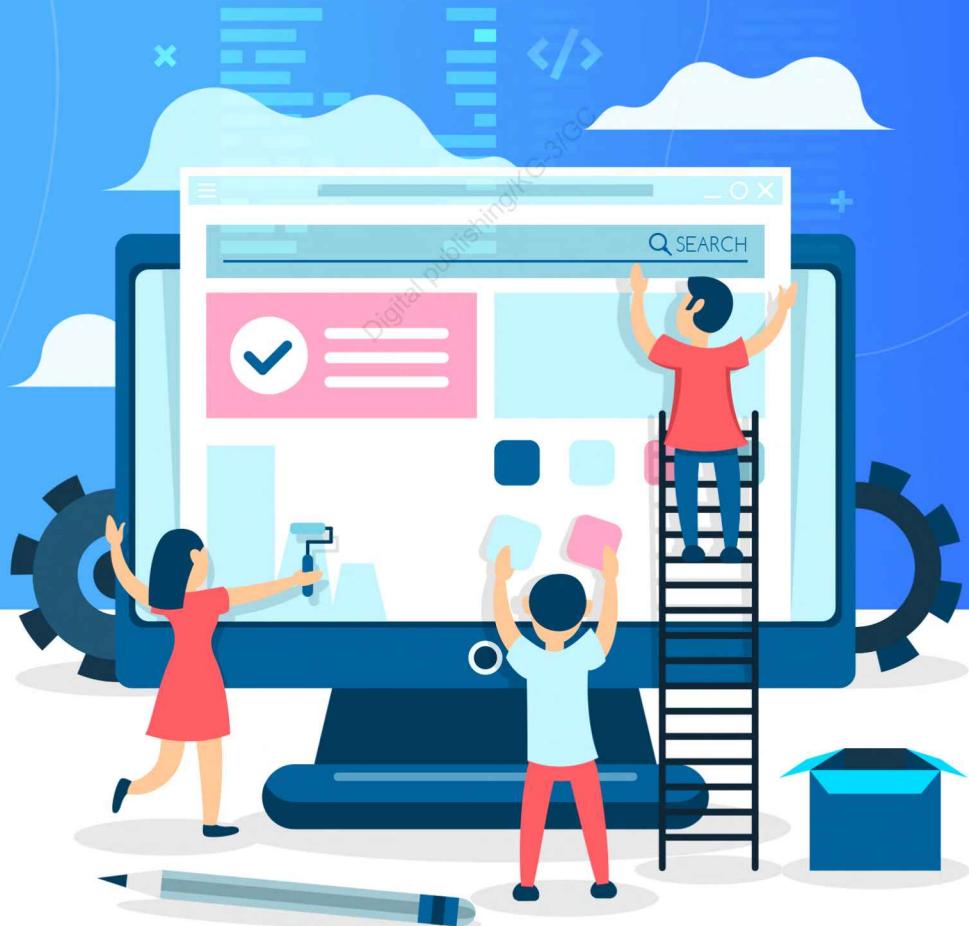


7 IN 1 PEMROGRAMAN WEB UNTUK PEMULA

Cara cepat dan efektif menjadi web programmer



Rohi Abdulloah

7 in 1 Pemrograman Web untuk Pemula

Sanksi Pelanggaran Pasal 113

Undang-Undang Nomor 28 Tahun 2014
tentang Hak Cipta

1. Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp100.000.000 (seratus juta rupiah).
2. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).
3. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/atau pidana denda paling banyak Rp1.000.000.000,00 (satu miliar rupiah).
4. Setiap Orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan/atau pidana denda paling banyak Rp4.000.000.000,00 (empat miliar rupiah).

7 in 1 Pemrograman Web untuk Pemula

Rohi Abdulloh

PENERBIT PT ELEX MEDIA KOMPUTINDO

 **KOMPAS GRAMEDIA**

7 in 1 Pemrograman Web untuk Pemula

Rohi Abdulloh

© 2018, PT Elex Media Komputindo, Jakarta

Hak cipta dilindungi undang-undang

Diterbitkan pertama kali oleh

Penerbit PT Elex Media Komputindo

Kelompok Gramedia, Anggota IKAPI, Jakarta 2018

718050xxx

ISBN: 978-602-04-xxxx-x

[eEp]

Dilarang keras menerjemahkan, memfotokopi, atau memperbanyak sebagian atau seluruh isi buku ini tanpa izin tertulis dari penerbit.

Dicetak oleh Percetakan PT Gramedia, Jakarta

Isi di luar tanggung jawab percetakan

KATA PENGANTAR

Puji syukur penulis sampaikan ke hadirat Allah SWT, karena berkat anugerah-Nya penulis dapat menyelesaikan buku ini hingga sampai ke tangan pembaca. Penulis berharap buku ini dapat memberi manfaat yang banyak bagi para pembaca.

Teknologi pemrograman web terus berkembang begitu cepat. Bagi pemula, tentu akan sangat tertinggal jika tidak cepat mengejar ketertinggalannya. Buku ini membahas 7 materi pemrograman web sekaligus yang menjadi materi utama dalam mempelajari pemrograman web. Dengan demikian, akan sangat membantu pemula yang ingin menguasai pemrograman web untuk cepat mengejar ketertinggalannya dan menjadi web programmer dalam waktu yang singkat.

Pembahasan dimulai dari pengetahuan dasar tentang pemrograman web, dilanjut dengan pembahasan 7 materi utama pemrograman web satu per satu disertai dengan contoh skrip beserta hasilnya. Disertai juga pembuatan aplikasi sederhana yang akan membantu pembaca menguasai pembuatan modul aplikasi. Untuk menunjang latihan pembaca, penulis juga menyertakan puluhan bonus skrip aplikatif, sehingga dengan menguasai buku ini diharapkan benar-benar menjadi web programmer yang siap dan mampu membuat aplikasi web secara mandiri. File pendukung buku yang terdiri dari file latihan dan bonus contoh skrip dapat diminta melalui email penulis dengan menyertakan bukti pembelian buku.

Akhir kata, penulis menyadari masih banyak terdapat kekurangan dalam buku ini. Pembaca dapat mengirimkan kritik, saran atau pertanyaan ke email penulis rohi.abdulloh@gmail.com.

Tegal, Jawa Tengah, April 2018

Rohi Abdulloh

DAFTAR ISI

KATA PENGANTAR.....	V
DAFTAR ISI	VII
BAB 1 DASAR PEMROGRAMAN WEB.....	1
1.1 Pengertian Website.....	1
1.2 Jenis-Jenis Website	1
1.2.1 Website Statis	1
1.2.2 Website Dinamis	2
1.2.3 Website Interaktif.....	2
1.3 Pengertian Pemrograman Web.....	2
1.4 Bahasa Pemrograman Web.....	2
1.5 Aplikasi Pengembang Web	3
1.5.1 Web Authoring	3
1.5.2 Text Editor	4
1.5.3 Web Brower.....	4
1.5.4 Web Server	4
1.6 Istilah-Istilah Pemrograman Web	5
BAB 2 HTML	7
2.1 Mengenal HTML.....	7
2.2 Unsur HTML.....	7
2.2.1 Tag	8
2.2.2 Atribut.....	8
2.2.3 Element	9
2.3 Struktur Dasar HTML	9
2.4 Pemformatan Paragraf dan Teks	10
2.4.1 Pemformatan Paragraf	10
2.4.2 Pemformatan Judul.....	11
2.4.3 Style Teks	12

2.4.4	Subscript dan Superscript	14
2.4.5	Simbol-Simbol Khusus.....	15
2.4.6	Teks Komentar	18
2.5	List.....	19
2.6	Tabel.....	20
2.7	Menambahkan Gambar	24
2.8	Menambahkan Video dan Audio	25
2.9	Link.....	27
2.10	Iframe	29
2.11	Formulir	30
2.11.1	Tag <label> ... </label>	31
2.11.2	Tag <input>	31
2.11.3	Tag <select> ... </select>	34
2.11.4	Tag <textarea> ... </textarea>	35
2.12	Semantic HTML 5 Template.....	36
2.13	Daftar Tag HTML	38
2.14	Global Attribute	42
BAB 3	CSS	45
3.1	Mengenal CSS.....	45
3.2	Dasar-Dasar CSS	45
3.2.1	Penulisan CSS.....	45
3.2.2	Selector.....	48
3.2.3	Daftar Selector CSS	50
3.2.4	Property dan Value	50
3.2.5	CSS-Shorthand.....	51
3.2.6	Pseudo-Class	52
3.2.7	Pseudo-Element.....	54
3.3	Unit CSS	55
3.3.1	Unit Relative	56
3.3.2	Unit Absolute	56
3.4	Warna CSS.....	57
3.5	Property CSS.....	58
3.5.1	Box Model	58
3.5.2	Font	68
3.5.3	Text.....	70
3.5.4	Background	71
3.5.5	List Style	73
3.6	Fitur Khusus CSS3.....	74
3.6.1	Rounded Corner.....	75
3.6.2	Border Image.....	75
3.6.3	Shadow.....	77

3.6.4	Gradient.....	78
3.6.5	Text Effect.....	79
3.6.6	Web Font.....	80
3.6.7	2D Transform	81
3.6.8	3D Transform	83
3.6.9	Transition.....	84
3.6.10	Animation	85
3.6.11	Flexbox Layout.....	87
3.6.12	Multiple Column.....	89
3.6.13	Media Query.....	90
3.7	Praktik Membuat Template.....	92
3.7.1	Desain Halaman Login.....	92
3.7.2	Desain Layout dan Menu Navigasi.....	94
3.7.3	Desain Tabel	97
3.7.4	Desain Form	99
3.8	Bonus File Contoh	101
BAB 4	MySQL	103
4.1	Pengenalan Database.....	103
4.1.1	Istilah-Istilah Dalam Database	103
4.1.2	Aturan Merancang Database	104
4.1.3	Penamaan Identifier	104
4.2	Tipe Data pada MySQL	105
4.2.1	Tipe Data Numerik.....	105
4.2.2	Tipe Data Waktu dan Tanggal.....	105
4.2.3	Tipe Data String.....	106
4.2.4	Tipe Data BLOB.....	106
4.2.5	Tipe Data Lainnya	107
4.3	Akses MySQL dengan Command Pompt.....	107
4.4	Manipulasi Database dan Tabel.....	108
4.4.1	Manipulasi Database	108
4.4.2	Manipulasi Tabel	109
4.5	Manipulasi Data.....	112
4.6	Beberapa Teknik Menampilkan Data	114
4.7	Function pada MySQL.....	116
4.7.1	Function untuk Operasi String	116
4.7.2	Function untuk Operasi Matematika.....	118
4.7.3	Function untuk Operasi Tanggal dan Waktu	119
4.7.4	Function untuk Keperluan Lain	121
4.8	Menampilkan Data dari Banyak Tabel	122
4.9	Menampilkan Data dengan Sub Select	124
4.10	Menampilkan Data dengan HAVING.....	124

4.11	Menyiapkan Database untuk Aplikasi.....	125
BAB 5	PHP	127
5.1	Mengenal PHP.....	127
5.1.1	Pengertian dan Aplikasi yang Diperlukan	127
5.1.2	Fungsi PHP	128
5.1.3	Penulisan Sintaks PHP	128
5.1.4	Menjalankan File PHP	129
5.1.5	Skrip HTML dalam PHP	130
5.1.6	Komentar.....	131
5.2	Dasar-Dasar PHP	131
5.2.1	Variabel.....	131
5.2.2	Tipe Data	133
5.2.3	Konstanta	134
5.2.4	Operator	135
5.2.5	Array	137
5.2.6	Statement Control.....	139
5.2.7	Modularitas	146
5.3	Function	147
5.3.1	Function Dasar	147
5.3.2	Function dengan Argumen	148
5.3.3	Function dengan Nilai Kembalian	149
5.3.4	Function untuk Operasi String	150
5.3.5	Function untuk Operasi Tanggal.....	152
5.3.6	Function untuk Operasi Matematika.....	155
5.3.7	Function untuk Penanganan File.....	157
5.4	Variabel Superglobal.....	159
5.4.1	Variabel \$GLOBAL	160
5.4.2	Variabel \$_SERVER	160
5.5	Session	162
5.6	Cookie.....	164
5.7	Penanganan Form.....	166
5.7.1	Variabel \$_GET	166
5.7.2	Variabel \$_POST	167
5.7.3	Variabel \$_FILES.....	168
5.8	Penanganan Database.....	169
5.9	Mengirim Email	172
5.10	Praktik Membuat Aplikasi	174
5.10.1	Membuat Koneksi ke Database	174
5.10.2	Membuat File Login dan Logout	175
5.10.3	Membuat Halaman Dashboard.....	176
5.10.4	Membuat Halaman Jabatan	179

5.10.5	Membuat Halaman Pegawai.....	184
5.11	Bonus File Contoh	192
BAB 6	JAVASCRIPT	193
6.1	Mengenal JavaScript	193
6.1.1	Pengertian.....	193
6.1.2	Meletakkan Skrip JavaScript.....	193
6.1.3	Mengatasi Error pada JavaScript	196
6.1.4	Menampilkan Hasil Program	197
6.1.5	Komentar.....	199
6.2	Dasar-Dasar JavaScript	200
6.2.1	Variabel.....	200
6.2.2	Tipe Data	202
6.2.3	Operator	202
6.2.4	Statement Control.....	207
6.2.5	Continue dan Break	214
6.3	Function	215
6.4	Event.....	216
6.5	Kotak Dialog	218
6.6	Objek String	219
6.7	Objek Window dan Dokumen.....	222
6.8	Method Tanggal dan Waktu	222
6.9	Memahami Array	223
6.10	Objek JavaScript	225
6.10.1	Membuat dan Mengakses Objek	226
6.10.2	Mengubah dan Menghapus Properti.....	227
6.10.3	Berbagai Nilai Properti.....	228
6.11	Perintah setTimeout() dan setInterval()	229
BAB 7	JQUERY	233
7.1	Mengenal jQuery	233
7.1.1	Pengertian jQuery.....	233
7.1.2	Kelebihan jQuery.....	233
7.1.3	Kemampuan jQuery	234
7.1.4	Mendapatkan Library jQuery.....	234
7.1.5	Cara Menggunakan jQuery.....	235
7.2	Memahami Selector	236
7.2.1	Basic Selector	236
7.2.2	Attribute Selector	237
7.2.3	Filter Selector.....	238
7.2.4	Form Selector	240
7.2.5	Hierarchy Selector.....	240

7.3	Memahami Effect	241
7.4	Event.....	243
7.4.1	Memahami Event	243
7.4.2	Berbagai Jenis Event.....	244
7.5	JQuery dan CSS.....	250
7.6	Manipulasi HTML dengan jQuery.....	252
7.7	Menggunakan Plugin jQuery.....	254
7.8	Berbagai Contoh Plugin jQuery	257
7.9	Membuat Plugin Sendiri	258
7.10	Bonus Skrip Contoh.....	260
BAB 8	BOOTSTRAP.....	261
8.1	Mengenal Bootstrap	261
8.1.1	Pengertian Bootstrap.....	261
8.1.2	Mendapatkan Bootstrap	261
8.1.3	Struktur File Bootstrap	262
8.1.4	Cara Menggunakan Bootstrap	263
8.2	Layout.....	263
8.3	CSS Utilities	267
8.4	Komponen Bootstrap	271
8.4.1	Button.....	271
8.4.2	Alert	273
8.4.3	Badge	274
8.4.4	Breadcrumb	275
8.4.5	Card.....	276
8.4.6	Jumbotron.....	278
8.4.7	List Group	279
8.4.8	Pagination.....	280
8.4.9	Media Object.....	281
8.4.10	Form.....	282
8.4.11	Input Group	285
8.4.12	Progress Bar	287
8.4.13	Tooltip	288
8.4.14	Popover.....	289
8.4.15	Dropdown	290
8.4.16	Collapse.....	292
8.4.17	Modal.....	293
8.4.18	Carousel.....	295
8.4.19	Nav	297
8.4.20	Navbar	299
8.4.21	Scrollspy	301
8.5	Penerapan JQuery dan Bootstrap pada Aplikasi	303

8.5.1	Mengatur Halaman Login.....	303
8.5.2	Mengatur Layout	305
8.5.3	Mengatur Halaman Dashboard.....	309
8.5.4	Mengatur Tampilan Tabel	310
8.5.5	Mengatur Tampilan Form	313
8.6	Bonus File Contoh	316
BAB 9	PENUTUP.....	317
KATA PENGANTAR.....		319

1 DASAR PEMROGRAMAN WEB

1.1 Pengertian Website

Website dapat diartikan sebagai kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara dan video atau gabungan dari semuanya yang disediakan melalui jalur koneksi internet sehingga dapat diakses dan dilihat oleh semua orang di seluruh dunia. Halaman website dibuat menggunakan bahasa standar yaitu HTML. Skrip HTML ini akan diterjemahkan oleh web browser sehingga dapat ditampilkan dalam bentuk informasi yang dapat dibaca oleh semua orang.

1.2 Jenis-Jenis Website

Secara umum, website dibagi menjadi 3 jenis, yaitu website statis, dinamis, dan interaktif.

1.2.1 Website Statis

Website statis yaitu jenis website yang isinya tidak diperbaharui secara berkala, sehingga isinya dari waktu ke waktu akan selalu tetap. Website jenis ini biasanya hanya digunakan untuk menampilkan profil dari pemilik website seperti profil perusahaan atau organisasi.

1.2.2 Website Dinamis

Website dinamis yaitu jenis website yang isinya terus diperbarui secara berkala oleh pengelola web atau pemilik website. Website jenis ini banyak dimiliki oleh perusahaan atau perorangan yang aktifitas bisnisnya memang berkaitan dengan internet. Contoh paling mudah dari website jenis ini yaitu web blog dan website berita.

1.2.3 Website Interaktif

Website interaktif pada dasarnya termasuk dalam kategori website dinamis, dimana isi informasinya selalu diperbarui dari waktu ke waktu. Hanya saja, isi informasi tidak hanya diubah oleh pengelola website tetapi lebih banyak dilakukan oleh pengguna website itu sendiri. Contoh website jenis ini yaitu website jejaring sosial seperti facebook dan twitter atau website marketplace seperti bukalapak, tokopedia, dan sebagainya.

1.3 Pengertian Pemrograman Web

Pemrograman web (web programming) terdiri dari kata pemrograman dan web. Permograman sendiri dapat diartikan sebagai proses atau cara pembuatan program menggunakan bahasa pemrograman. Adapun bahasa permograman merupakan bahasa yang digunakan untuk memberikan intruksi kepada komputer sehingga komputer dapat memproses data dan menampilkan informasi sesuai yang dikehendaki oleh pemrogram. Dengan demikian pemrograman web dapat diartikan sebagai kegiatan pembuatan program atau aplikasi berbasis web menggunakan bahasa pemrograman tertentu sehingga dapat memproses data dan menghasilkan informasi sesuai yang dikehendaki oleh pemilik website.

1.4 Bahasa Pemrograman Web

Bahasa pemrograman web terdiri dari beberapa unsur bahasa. Setidaknya terdapat 5 bahasa utama yang biasa digunakan dalam membuat website dinamis dimana masing-masing memiliki perannya sendiri-sendiri, yaitu sebagai berikut:

- **HTML**, berperan sebagai pembentuk struktur halaman website yang menempatkan setiap elemen website sesuai layout yang diinginkan.
- **CSS**, berperan sebagai pembentuk desain website dengan mengatur setiap elemen HTML agar tampil menarik pada browser.
- **PHP**, berperan sebagai pemroses data pada sisi server sesuai yang diminta oleh client menjadi informasi yang siap ditampilkan, juga sebagai penghubung aplikasi web dengan database. Selain PHP, dapat juga menggunakan bahasa pemrograman lain seperti ASP, Java, dan sebagainya.
- **SQL**, berperan sebagai bahasa yang mengatur transaksi data antara aplikasi dengan database sebagai tempat penyimpanan data. Database yang biasa digunakan diantaranya MySql, Oracle, SQL Server dan sebagainya.
- **JavaScript**, berperan sebagai bahasa yang memproses data pada sisi client serta dapat memanipulasi HTML dan CSS secara dinamis.

Dari 5 hal di atas, telah berkembang menjadi bentuk lain ada yang berupa library seperti **jQuery** yang merupakan library function JavaScript dan ada yang berupa framework seperti **Bootstrap** yang memudahkan programan dapat membuat desain website dengan lebih cepat. 2 hal tersebut juga akan dibahas pada buku ini, sehingga total ada 7 materi utama pemrograman web yang akan dibahas pada buku ini. Dengan menguasai 7 hal tersebut, sudah cukup digunakan untuk membuat aplikasi website dengan desain yang menarik dan responsive serta atraktif.

1.5 Aplikasi Pengembang Web

Untuk pengembangan aplikasi web tentu dibutuhkan berbagai macam aplikasi yang membantu web programer dalam merancang, menuliskan skrip program maupun menampilkan hasil program. Berikut ini berbagai jenis aplikasi yang digunakan dalam pengembangan aplikasi web:

1.5.1 Web Authoring

Web Authoring merupakan perangkat lunak yang biasanya bersifat "WYSIWYG" (What You See Is What You Get) yaitu apa yang dilihat merupakan apa yang akan dihasilkan. Aplikasi jenis ini sangat

memudahkan bagi pemula, karena setiap mengetikkan skrip program langsung terlihat hasilnya. Bahkan biasanya tersedia juga tool drag & drop yang memudahkan web designer dalam membuat elemen website. Yang termasuk jenis aplikasi ini diantaranya: Adobe Dreamweaver, Microsoft Expression Web, Nvu dan sebagainya.

1.5.2 Text Editor

Text editor memiliki fungsi yang sama dengan web authoring yaitu untuk mengetikkan skrip program, hanya saja tidak bersifat WYSIWYG, sehingga untuk menampilkan hasilnya memerlukan web browser. Namun, karena lebih ringan dan lisensinya yang kebanyakan open source banyak juga yang lebih memilih menggunakan text editor dibanding aplikasi sekelas Adobe Dreamweaver. Yang termasuk dalam aplikasi jenis ini yaitu Notepad, Notepad++, Sublime Text, Bracket, dan sebagainya. Dalam pembuatan buku ini, penulis lebih banyak menggunakan Sublime Text dan beberapa menggunakan Notepad++.

1.5.3 Web Browser

Web browser digunakan untuk menampilkan dan mengetes hasil program. Beberapa skrip CSS3 dan HTML5 ada yang hanya support pada web browser tertentu dan tidak support pada yang lain, sehingga menggunakan lebih dari satu web browser akan lebih baik. Walaupun untuk kebanyakan browser versi terbaru sudah support hampir semua fitur CSS3 dan HTML5.

1.5.4 Web Server

Web server merupakan perangkat lunak yang berfungsi untuk menerima permintaan (request) melalui protokol HTTP atau HTTPS dari client kemudian mengirimkan kembali dalam bentuk halaman-halaman web. Contoh yang termasuk web server adalah Apache. Dalam penggunaanya, biasanya sudah jadi satu paket dengan PHP dan MySQL. Contoh paket yang sudah berisi Apache, PHP dan MySQL diantaranya Xampp dan Appserv. Dalam buku ini, penulis menggunakan Xampp versi 5.6.30 yang di dalamnya berisi Apache 2.4.25 dan PHP 5.6.30.

1.6 Istilah-Istilah Pemrograman Web

Sebelum lebih jauh belajar pemrograman web, alangkah baiknya kenali dulu berbagai istilah yang berkaitan dengan pemrograman web. Sehingga, akan lebih mudah dalam memahami setiap keterangan yang menggunakan istilah-istilah tersebut. Beberapa istilah yang sering digunakan diantaranya:

- **Server**, merupakan komputer khusus dengan kecepatan dan kapasitas penyimpanan di atas komputer biasa yang digunakan sebagai tempat pemrosesan data dan juga sebagai tempat penyimpanan file-file website.
- **Client**, merupakan perangkat komputer biasa baik berupa komputer personal, laptop, perangkat mobile atau lainya yang dapat mengakses halaman website.
- **Domain**, merupakan nama unik yang mewakili alamat IP dari server website berupa kata yang mudah diingat sehingga lebih mudah dalam menemukan atau memanggil alamat sebuah website pada internet.
- **Hosting**, merupakan tempat menyimpan file-file website pada sebuah server di internet sehingga website dapat diakses dari mana saja melalui perangkat komputer atau mobile yang terhubung dengan internet.
- **URL**, merupakan singkatan dari *Uniform Resources Locator* yaitu nama unik yang mengidentifikasi letak sumber daya website berupa file HTML, gambar, video dan sebagainya yang biasanya akan tampil pada address bar browser ketika sumber daya tersebut diakses.
- **HTTP**, merupakan singkatan dari *Hypertext Transfer Protocol* yaitu protokol yang mengatur pengiriman data dari client ke server atau sebaliknya dan menejemahkanya menjadi informasi yang dapat dibaca oleh pengguna website.

2 HTML

2.1 Mengenal HTML

HTML merupakan singkatan dari *Hypertext Markup Language* yaitu bahasa standar web yang dikelola penggunaanya oleh W3C (*World Wide Web Consortium*) berupa tag-tag yang menyusun setiap elemen dari website. HTML berperan sebagai penyusun struktur halaman website yang menempatkan setiap elemen website sesuai layout yang diinginkan.

HTML biasanya disimpan dalam sebuah file berekstensi .html. Untuk mengetikkan skrip HTML dapat menggunakan text editor seperti Notepad sebagai bentuk paling sederhana atau text editor khusus yang dapat mengenali setiap unsur skrip HTML dan menampilkannya dengan warna yang berbeda sehingga mudah dibaca, seperti Notepad++, Sublime Text, dan masih banyak lagi aplikasi lain yang sejenis.

2.2 Unsur HTML

HTML terdiri atas unsur-unsur yang membentuk struktur skrip HTML, yaitu tag, atribut, dan elemen.

2.2.1 Tag

Tag adalah simbol khusus (markup) berupa dua karakter "<" dan ">" yang mengapit suatu teks sebagai nama tag. Contohnya tag <body> adalah tag dengan nama body. Sebagian besar tag memiliki tag pembuka dan tag penutup yang membatasi pengaruh dari tag tersebut. Namun, terdapat sebagian kecil tag yang tidak memiliki tag penutup. Penutup tag ditambah simbol "/" setelah tanda "<", misalnya </body> yang merupakan penutup tag <body>.

Berikut adalah karakteristik dari tag pada HTML:

- Tag boleh ditulis dengan huruf besar (kapital), kecil atau campuran, karena sifatnya yang *incase sensitive*. Misalnya: <BODY>, <body> atau <Body> semuanya memiliki hasil yang sama. Namun, untuk standarnya lebih baik menggunakan huruf kecil semua.
- Tag bisa berisi tag lain. Tag yang dibuka dahulu ditutup terakhir, sedangkan tag yang dibuka terakhir ditutup pertama. Namun, kesalahan dalam struktur ini tidak mengakibatkan error. Misalnya:

```
<p>I like <b>HTML</b></p>
```

2.2.2 Atribut

Atribut yaitu property yang mengatur bagaimana elemen dari suatu tag akan ditampilkan. Atribut ditulis di dalam simbol tag setelah nama tag dengan dipisahkan oleh spasi. Atribut ada yang memiliki nilai dan ada yang tidak memiliki nilai. Nilai suatu atribut ditulis di dalam tanda petik ganda ("..."), dipisahkan dengan simbol sama dengan (=) dari nama atribut. Misalnya: <p align="center">.

Berikut adalah karakteristik dari atribut pada HTML:

- Atribut hanya ditulis pada tag pembuka.
- Setiap tag memiliki atribut yang berbeda-beda yang dapat diterapkan pada tag tersebut.
- Beberapa atribut ada yang dapat diterapkan ke semua tag atau disebut global attribute, misalnya: class, id, style, dan sebagainya.
- Sebuah tag dapat ditulis dengan satu atribut, banyak atribut, atau tanpa atribut sama sekali.

- Jika tag ditulis dengan banyak atribut, masing-masing atribut dipisahkan oleh spasi.

2.2.3 Element

Element adalah bagian dari skrip HTML yang terdiri dari tag pembuka, isi element dan tag penutup. Jika sebuah element ditampilkan pada browser, maka hanya isi element yang tampil di browser. Misalnya:

```
<p> Aku senang belajar HTML</p>
```

Jika ditampilkan pada browser, maka hanya tulisan “**Aku senang belajar HTML**” akan tampil di browser.

2.3 Struktur Dasar HTML

HTML disimpan dengan ekstensi ***.html**, jika di dalamnya tidak mengandung skrip PHP. Jika di dalamnya mengandung skrip PHP, maka disimpan dengan ekstensi ***.php**.

Dalam penulisan HTML ada beberapa tag yang wajib dituliskan dengan struktur yang sudah ditentukan. Setiap file HTML harus menerapkan struktur tersebut, yaitu sebagai berikut:

File 2_2_struktur_dasar.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Judul Website</title>
  </head>
  <body>
    Isi Website
  </body>
</html>
```

Penjelasan dan fungsi dari masing-masing tag di atas sebagai berikut:

- **<!DOCTYPE HTML>**, adalah tag awal dari setiap dokumen HTML yang berfungsi untuk menginformasikan pada browser bahwa dokumen yang sedang dibuka adalah dokumen HTML. Tag ini perlu dicantumkan pada setiap awal penulisan HTML.
- **<html> ... </html>**, adalah tag yang menunjukkan pembuka dan penutup dokumen HTML.

- <head> ... </head>, adalah tag yang digunakan untuk menyimpan berbagai informasi tentang dokumen HTML. Apa yang terdapat di dalam tag ini tidak akan ditampilkan di browser.
- <title> ... </title>, adalah tag yang digunakan untuk membuat judul website yang akan muncul di title bar browser.
- <body> ... </body>, adalah tag yang menunjukkan bagian utama website. Semua yang akan ditampilkan pada halaman browser dituliskan di dalam tag ini.

2.4 Pemformatan Paragraf dan Teks

Ada beberapa jenis pemformatan text dalam HTML. Masing-masing akan dibahas satu per satu sebagai berikut:

2.4.1 Pemformatan Paragraf

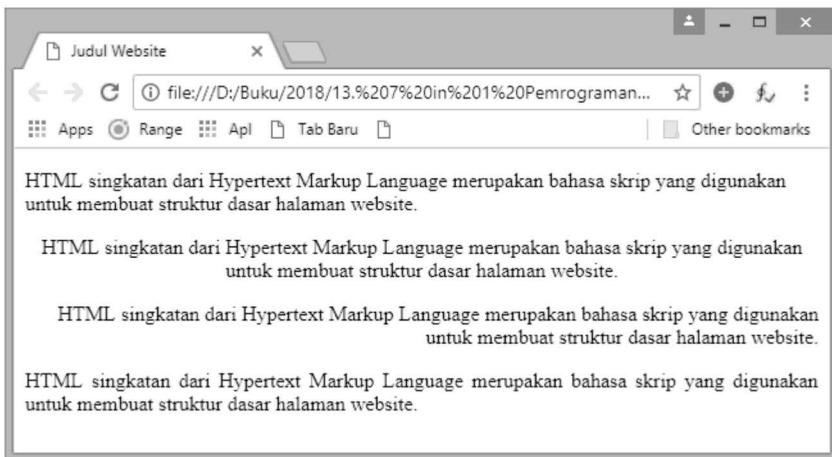
Paragraf dibuat dengan tag <p> ... </p>. Atribut yang sering digunakan dalam tag ini adalah **align**. Atribut ini digunakan untuk mengatur perataan paragraf. Nilai yang dapat diberikan pada atribut align antara lain *left*, *center*, *right* dan *justify*. Setiap penggunaan tag <p> akan berganti ke baris baru. Jika ingin pindah baris tanpa memulai dengan tag <p>, dapat menggunakan tag
 tanpa penutup.

Agar lebih jelas dalam memahami tag <p>, perhatikan contoh kode HTML berikut:

File 2_4_1_format_paragraf.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Judul Website</title>
  </head>
  <body>
    <p> HTML singkatan dari... </p>
    <p align="center"> HTML singkatan... </p>
    <p align="right"> HTML singkatan dari... </p>
    <p align="justify"> HTML singkatan... </p>
  </body>
</html>
```

Pada skrip di atas terdapat 4 paragraf dengan perataan yang berbeda. Hasilnya seperti gambar berikut:



Gambar 2.1 Hasil pemformatan paragraf

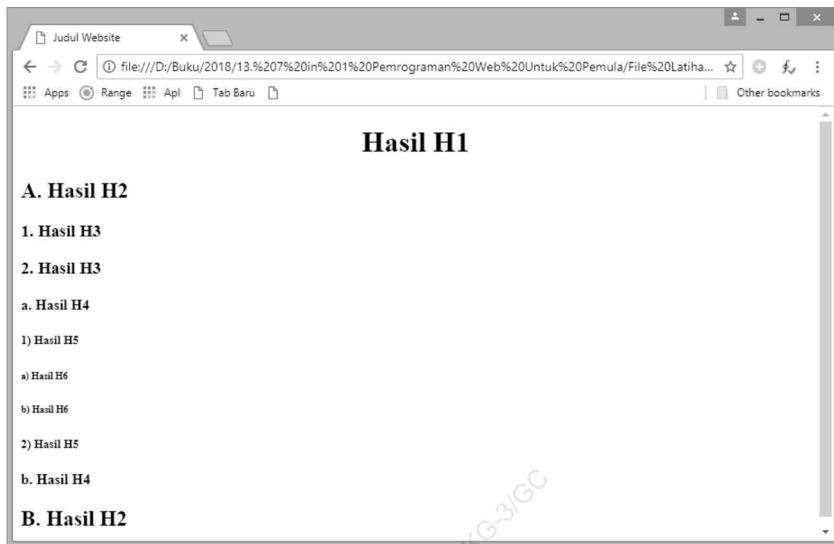
2.4.2 Pemformatan Judul

Ada 6 style judul dalam HTML yang dibuat menggunakan tag `<h1>` sampai `<h6>`. Tag `<h1>` menghasilkan judul dengan ukuran huruf paling besar, sedangkan `<h6>` akan menghasilkan judul dengan ukuran paling kecil.

File 2_4_2_format_judul.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Judul Website</title>
  </head>
  <body>
    <center><h1>Hasil H1</h1></center>
    <h2>A. Hasil H2</h2>
      <h3>1. Hasil H3</h3>
      <h3>2. Hasil H3</h3>
        <h4>a. Hasil H4</h4>
          <h5>1) Hasil H5</h5>
            <h6>a) Hasil H6</h6>
            <h6>b) Hasil H6</h6>
          <h5>2) Hasil H5</h5>
        <h4>b. Hasil H4</h4>
    <h2>B. Hasil H2</h2>
  </body>
</html>
```

Pada skrip di atas berisi susunan judul yang dibuat dengan <h1> hingga <h6>. Hasilnya seperti gambar berikut:



Gambar 2.2 *Hasil pemformatan judul*

2.4.3 Style Teks

Teks yang dibuat dengan HTML dapat dibuat tebal, miring, bergaris bawah, bergaris tengah dan sebagainya. Berikut tag yang dapat digunakan untuk mengatur style teks:

- Tag ... , untuk membuat teks tebal.
- Tag <i> ... </i>, untuk menjadikan teks miring.
- Tag <u> ... </u>, untuk membuat teks bergaris bawah.
- Tag <strike> ... </strike>, untuk membuat teks bergaris tengah.
- Tag ... , untuk menunjukkan teks penting dengan teks dicetak tebal.
- Tag ... , untuk menunjukkan teks penting dengan teks dicetak miring.

- Tag `<small> ... </small>`, untuk membuat teks lebih kecil dari umumnya.
- Tag `<mark> ... </mark>`, untuk membuat teks dengan background kuning, seperti teks yang ditandai.
- Tag ` ... `, untuk menunjukkan teks yang dihapus dengan teks dicoret tengah.
- Tag `<ins> ... </ins>`, untuk menunjukkan teks yang disisipkan dengan teks bergaris bawah.

File 2_4_3_style_teks.html

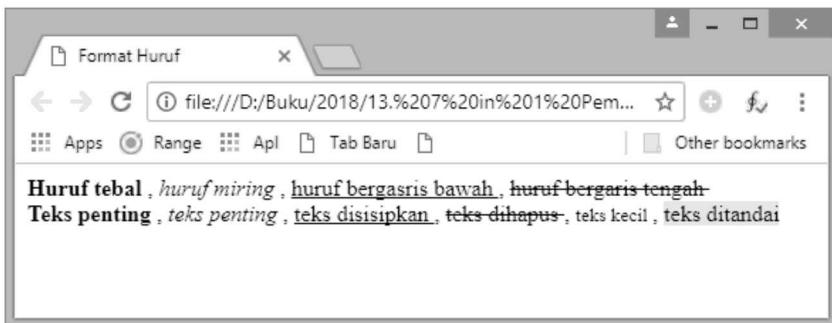
```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Format Huruf</title>
  </head>
  <body>
    <b> Huruf tebal </b>, <i> huruf miring </i>, <u> huruf
bergasris bawah </u>, <strike> huruf bergaris tengah </strike>
<br>

    <strong> Teks penting </strong>, <em> teks penting </em>,
<ins> teks disisipkan </ins>, <del> teks dihapus </del>, <small>
teks kecil </small>, <mark> teks ditandai </mark>

  </body>
</html>
```

Walaupun hasil yang didapat sama antara tag `` dan tag ``, tag `<i>` dan tag ``, tag `<u>` dan tag `<ins>`, serta tag `<strike>` dan tag `<delete>`, namun penggunaan tag tersebut memiliki tujuan yang berbeda. Tag `` dan `` lebih ditujukan untuk menandai teks yang penting.

Hasil dari skrip di atas seperti gambar berikut:



Gambar 2.3 Hasil style huruf

2.4.4 Subscript dan Superscript

Dalam penulisan rumus matematika, sering ditemui penulisan huruf yang ditulis lebih kecil dan posisinya lebih ke atas atau ke bawah yang biasanya disebut superscript dan subscript. Untuk membuat superscript menggunakan tag `<sup>` sedangkan untuk membuat subscript menggunakan tag `<sub>`.

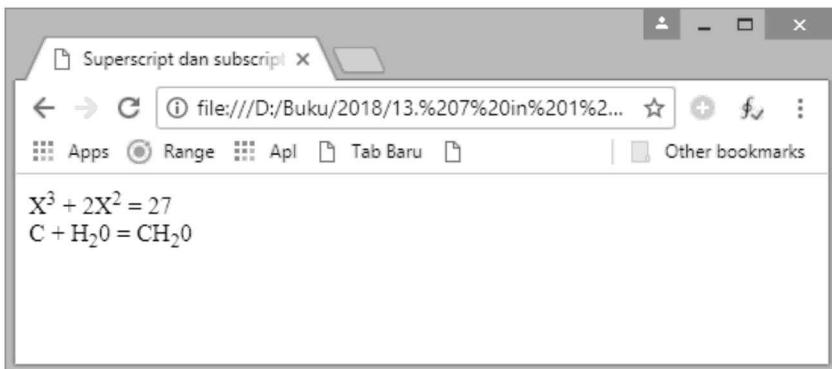
File 2_4_4_subscript_superscript.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Superscript dan subscript</title>
  </head>
  <body>

    X<sup>3</sup> + 2X<sup>2</sup> = 27 <br>
    C + H<sub>2</sub>0 = CH<sub>2</sub>0

  </body>
</html>
```

Skrip di atas berisi rumus matematika dan rumus kimia yang menggunakan superscript dan subscript. Hasilnya sebagai berikut:



Gambar 2.4 *Superscript dan subscript*

2.4.5 Simbol-Simbol Khusus

Setiap simbol memiliki kode nama dan kode nomor. Untuk menambahkan simbol-simbol tertentu pada HTML dapat dilakukan dengan menuliskan kode nama atau kode nomor dari simbol tersebut. Berikut daftar kode nama dan kode nomor simbol yang sering digunakan dalam HTML.

Simbol/ Karakter	Deskripsi	Kode Nomor HTML	Kode Nama HTML
'	Kutip satu	'	
-	Tanda kurang	-	
!	Tanda seru	!	
"	Kutip ganda	"	"
#	Simbol pagar	#	
\$	Simbol dollar	$	
%	Simbol persen	%	
&	Simbol dan	&	&
(Kurung buka	(
)	Kurung tutup)	
*	Tanda bintang	*	
,	Tanda koma	,	
.	Tanda titik	.	

Simbol/ Karakter	Deskripsi	Kode Nomor HTML	Kode Nama HTML
/	Garis miring	/	
:	Titik dua	:	
;	Titik koma	;	
?	Tanda tanya	?	
@	Simbol [at]	@	
[Kurung siku buka	[
\	Garis miring terbalik/backslash	\	
]	Kurung siku tutup]	
^	Tanda sisipan/sirkumfleksa	^	
_	Garis bawah	_	
`	Tanda petik	`	
{	Kurung kurawal buka	{	
	Garis tegak/pipe	|	
}	Kurung kurawal tutup	}	
~	Simbol ekuivalen	~	
i	Tanda seru terbalik	¡	¡
ı	Garis tegak terputus	¦	¦
..	Titik dua di atas huruf vokal	¨	¨
<	Panah kiri tunggal	‹	‘
>	Panah kanan tunggal	›	’
-	Garis atas	‾	‾
¢	Simbol sen	¢	¢
£	Simbol pondsterling	£	£
¤	Simbol mata uang	¤	¤
¥	Simbol yen	¥	¥
€	Simbol Euro	€	€
+	Tanda tambah	+	
*	Tanda bintang	∗	∗
<	Kurang dari	<	<
=	Sama dengan	=	
≠	Sama dengan dicoret	≠	≠
>	Lebih dari	>	>
±	Simbol lebih kurang	±	±

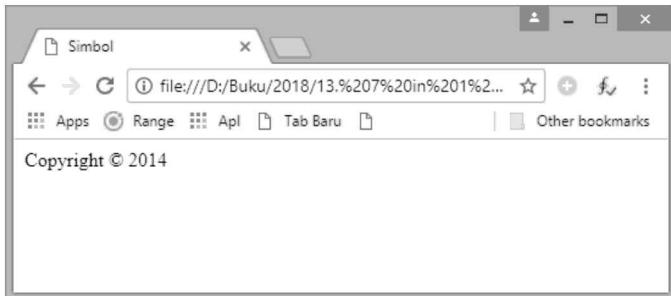
Simbol/ Karakter	Deskripsi	Kode Nomor HTML	Kode Nama HTML
«	Panah kiri ganda	«	&lquo;
»	Panah kanan ganda	»	&rquo;
×	Simbol perkalian	×	×
÷	Simbol pembagian	÷	÷
↑	Panah atas	↑	↑
→	Panah kanan	→	→
↓	Panah bawah	↓	↓
←	Panah kiri	←	←
↵	Simbol Enter/Carriage return	↵	↵
↔	Panah kiri kanan	↔	↔
§	Simbol bagian/seksi	§	§
©	Simbol hak cipta (Copyright)	©	©
®	Simbol merek dagang terdaftar (Registered trademark)	®	®
°	Simbol derajat	°	°
µ	Simbol mikro	µ	µ
¶	Simbol pilcrow, simbol paragraf	¶	¶
.	Titik tengah, koma Georgia	·	·
...	Ellipsis horizontal	…	…
¼	Satu per empat	¼	¼
½	Satu per dua, setengah	½	½
¾	Tiga per empat	¾	¾
¹	Pangkat satu	¹	¹
²	Pangkat dua, kuadrat, persegi	²	²
³	Pangkat tiga, kubik	³	³
α	alpha	α	α
β	beta	β	β
	Non-breaking space	 	

File 2_4_5_simbol.html

```
<!DOCTYPE HTML>
<html>
  <head>
```

```
<title>Simbol</title>
</head>
<body>
    Copyright &copy; 2014
</body>
</html>
```

Skrip di atas akan menghasilkan simbol ©, seperti gambar berikut:



Gambar 2.5 Simbol khusus pada HTML

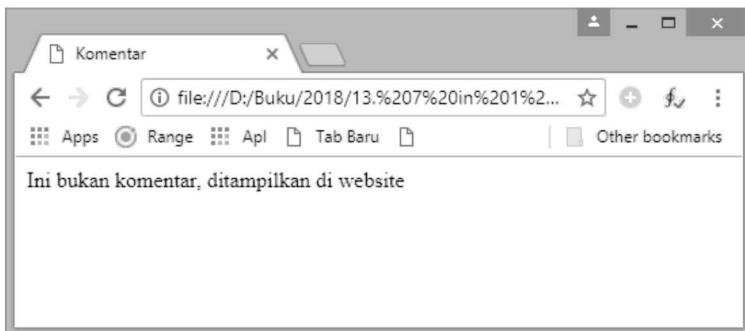
2.4.6 Teks Komentar

Komentar adalah baris text yang tidak ditampilkan dalam halaman website. Fungsinya untuk memberi keterangan pada skrip. Untuk menandai sebuah komentar menggunakan tag `<!-- ... -->`.

File 2_4_6_komentar.html

```
<!DOCTYPE HTML>
<html>
    <head>
        <title>Komentar</title>
    </head>
    <body>
        <!-- Ini komentar, tidak ditampilkan di website -->
        Ini bukan komentar, ditampilkan di website
    </body>
</html>
```

Pada skrip di atas, teks yang dicetak tebal adalah komentar. Tulisan tersebut tidak akan ditampilkan pada website seperti gambar berikut:



Gambar 2.6 Komentar pada HTML

2.5 List

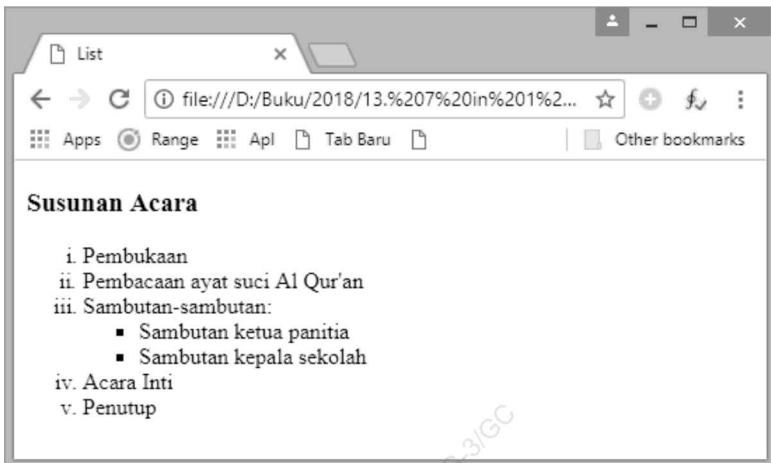
Ada dua jenis list yaitu list berurutan (*ordered list*) dan list tidak berurutan (*unordered list*). Untuk membuat list berurutan menggunakan tag ` ... `. Tag ini memiliki atribut `type` yang menunjukkan jenis penomoran dan dapat diberi nilai `1`, `a`, `A` atau `i`. Untuk membuat list tidak berurutan menggunakan tag ` ... `. Atribut ini juga memiliki atribut `type`, tetapi nilainya berbeda, yaitu `round`, `disc` dan `square`. Untuk membuat item list pada tiap baris list baik pada list berurutan maupun list tidak berurutan menggunakan tag ` ... `. Untuk lebih jelasnya, perhatikan contoh berikut:

File 2_5_list.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>List</title>
  </head>
  <body>
    <h3>Susunan Acara</h3>
    <ol type="i">
      <li>Pembukaan</li>
      <li>Pembacaan ayat suci Al Qur'an</li>
      <li>Sambutan-sambutan:
          <ul type="square">
            <li>Sambutan ketua panitia</li>
            <li>Sambutan kepala sekolah</li>
          </ul>
        </li>
        <li>Acara Inti</li>
        <li>Penutup</li>
    </ol>
```

```
</body>  
</html>
```

Skrip di atas berisi kombinasi list berurutan dan tidak berurutan. Hasilnya seperti gambar berikut:



Gambar 2.7 List pada HTML

2.6 Tabel

Tabel pada HTML dapat dibuat menggunakan tag `<table> ... </table>`. Isi dalam tabel dibagi menjadi dua bagian yaitu header tabel ditunjukkan dengan tag `<thead> ... </thead>` dan body tabel ditunjukkan dengan tag `<tbody> ... </tbody>`. Untuk menambah baris pada tabel menggunakan tag `<tr> ... </tr>`. Untuk menambahkan data, jika berupa data header menggunakan tag `<th> ... </th>` sedangkan pada body tabel menggunakan tag `<td> ... </td>`.

Format penulisan tabel secara umum adalah sebagai berikut:

```
<table>  
  <thead>  
    <tr>  
      <th> ..... </th>  
    </tr>  
  </thead>  
  <tbody>  
    <tr>
```

```
<td> ..... </td>
</tr>
</tbody>
</table>
```

Tag `<table>` memiliki beberapa atribut diantaranya:

- **Cellspacing**, untuk menentukan jarak antar sel.
- **Celpadding**, untuk menentukan jarak isi sel dengan bordernya.
- **Width**, untuk menentukan lebar tabel.
- **Height**, untuk menentukan tinggi tabel.
- **Bgcolor**, untuk memberi warna background pada tabel.
- **Align**, untuk menentukan perataan tabel.
- **Border**, untuk menentukan tebal border tabel.

Atribut juga dapat diberikan pada baris dan sel tabel yaitu pada tag `<tr>` atau `<td>`. Atribut yang dapat diberikan pada tag `<tr>` atau `<td>` diantaranya:

- **Align**, untuk menentukan perataan text pada isi sel.
- **Valign**, untuk menentukan perataan text secara vertikal.
- **Width**, untuk menentukan lebar sel.
- **Height**, untuk menentukan tinggi sel.
- **Bgcolor**, untuk memberi warna background pada sel.
- **Rowspan**, untuk menggabungkan sel baris berikutnya pada kolom yang sama.
- **Colspan**, untuk menggabungkan sel kolom berikutnya pada baris yang sama.

Sebagian besar atribut di atas sudah tidak support pada HTML5, kecuali colspan dan rowspan. Lebih jelasnya, perhatikan contoh berikut:

File 2_6_tabel_dasar.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Tabel Dasar</title>
  </head>
  <body>
    <h3> Jadwal Ujian </h3>
```

```

<table width="90%" border="1" cellspacing="0">
    <thead>
        <tr>
            <th>Hari</th>
            <th>Jam ke 1</th>
            <th>Jam ke 2</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>Senin</td>
            <td align="center">Bahasa Indonesia</td>
            <td align="center">Matematika</td>
        </tr>
        <tr>
            <td>selasa</td>
            <td align="center">Bahasa Inggris</td>
            <td align="center">Kejuruan</td>
        </tr>
    </tbody>
</table>
</body>
</html>

```

Skrip di atas berisi skrip membuat tabel dengan lebar 90% dari lebar layar browser. Hasilnya seperti gambar berikut:



Gambar 2.8 Tabel pada HTML

Dalam membuat tabel sering kali menggabungkan beberapa sel menjadi satu sel. Dalam HTML, penggabungan sel dikenal dengan istilah **span**. Untuk menggabungkan beberapa sel di dalam HTML menggunakan atribut **colspan** dan **rowspan**. Colspan digunakan untuk

menggabungkan beberapa kolom, sedangkan rowspan digunakan untuk menggabungkan beberapa baris. Nilainya diisi angka sesuai jumlah kolom atau baris yang digabungkan. Sebagai contoh, perhatikan skrip berikut:

File 2_6_tabel_span.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Tabel Span</title>
  </head>
  <body>
    <h3> Jadwal Ujian </h3>

    <table width="90%" border="1" cellspacing="0">
      <thead>
        <tr>
          <th rowspan="2">Hari</th>
          <th colspan="2">Jam</th>
        </tr>
        <tr>
          <th>pertama</th>
          <th>kedua</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>Senin</td>
          <td align="center">Bahasa Indonesia</td>
          <td align="center">Matematika</td>
        </tr>
        <tr>
          <td>selasa</td>
          <td align="center">Bahasa Inggris</td>
          <td align="center">Kejuruan</td>
        </tr>
      </tbody>
    </table>
  </body>
</html>
```

Pada skrip tabel di atas memiliki 3 kolom tabel. Pada baris pertama, karena ada penggabungan kolom kedua dan ketiga, maka hanya terdiri dari 2 kolom. Pada baris kedua, karena kolom pertama telah digabungkan dengan kolom pertama pada baris di atasnya, maka tinggal terdiri dari 2 kolom. Hasilnya seperti gambar berikut:

Jadwal Ujian

Hari	Jam	
	pertama	kedua
Senin	Bahasa Indonesia	Matematika
selasa	Bahasa Inggris	Kejuruan

Gambar 2.9 Tabel dengan span

2.7 Menambahkan Gambar

Tag yang digunakan untuk menambahkan gambar di dalam HTML adalah tag . Tag ini tidak memeliki tag penutup. Atribut yang dapat diberikan ke dalam tag ini diantaranya:

- **Src**, untuk menentukan sumber dan nama gambar
- **Width**, untuk menentukan lebar gambar
- **Height**, untuk menentukan tinggi gambar
- **Alt**, untuk memberi text alternatif yang ditampilkan jika gambar tidak dapat ditampilkan
- **Title**, untuk membuat text yang tampil ketika mouse di atas gambar

Jika gambar tidak ditentukan tinggi dan lebarnya, maka ukuran gambar sesuai dengan aslinya. Namun, jika gambar ditentukan tinggi dan lebarnya sekaligus, ukuran gambar menjadi tidak proporsional. Jika ukurannya ingin diubah, sebaiknya tentukan tinggi atau lebarnya saja.

Format gambar yang didukung oleh tag adalah **jpg**, **png** dan **gif**. Nama dan format gambar harus dituliskan dengan lengkap pada atribut **src**. Penulisan nama dan letak gambar ada beberapa kemungkinan yaitu sebagai berikut:

- Jika gambar satu folder dengan file HTML, maka penulisanya **src="nama_gambar.jpg"**.

- Jika gambar terletak di dalam folder yang berada di folder yang sama dengan file HTML, maka penulisannya menjadi **src="nama_folder/nama_gambar.jpg"**.
- Jika gambar terletak di folder yang berbeda dengan folder yang berisi file HTML, maka harus keluar dari folder yang beris file HTML dulu, sehingga penulisannya menjadi **src="..../nama_folder/nama_gambar.jpg"**. Tanda ../ artinya keluar dari folder yang berisi file HTML.

File 2_7_gambar.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Menambahkan Gambar</title>
  </head>
  <body>
    
    
    
  </body>
</html>
```

Skrip di atas berisi skrip memasukkan gambar, dimana semua gambar diletakkan di folder **gambar**.



Gambar 2.10 Hasil menambahkan gambar

2.8 Menambahkan Video dan Audio

Pada HTML 5 ada tag tambahan yang dapat menampilkan video atau audio di dalam browser. Tag yang digunakan untuk menampilkan video

yaitu `<video> ... </video>`. Adapun tag yang digunakan untuk menampilkan audio yaitu `<audio> ... </audio>`. Kedua tag ini memiliki kesamaan dalam atributnya yaitu sebagai berikut:

- **Src**, untuk menentukan nama file dan sumber/letak video atau audio
- **Width**, untuk menentukan lebar video atau audio control
- **Height**, untuk menentukan lebar video, hanya dipakai pada tag `<video>`
- **Controls**, pilihan untuk menampilkan tombol kontrol atau tidak.
- **Autoplay**, pilihan untuk langsung dimainkan atau tidak

Format video yang didukung yaitu **ogg**, **mp4** dan **webm**. Sedangkan untuk audio, format yang didukung yaitu **ogg**, **mp3** dan **wav**. Aturan penulisan nama file dan letak file sama dengan aturan untuk menampilkan gambar.

File 2_8_audio_video.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Audio dan Video</title>
  </head>
  <body>

    <video src="video/video.mp4" width="320" height="240"
controls> </video>

    <audio src="audio/Indonesia Raya.mp3" width="320" controls
autoplay> </audio>

  </body>
</html>
```

Skrip di atas berisi skrip untuk menampilkan audio dan video. Pada tag `<audio>` diatur langsung dimainkan dengan atribut `autoplay`, sedangkan pada tag `<video>` tidak menggunakan atribut `autoplay`. Hasilnya seperti berikut:



Gambar 2.11 Menampilkan video dan audio

2.9 Link

Link merupakan pintasan untuk menuju suatu URL website, baik halaman lain pada suatu website maupun halaman pada website lain. Link sering digunakan untuk membuat menu website. Tag yang digunakan untuk membuat link adalah tag `<a> ... `. Tag ini memiliki beberapa atribut diantaranya:

- **Href**, menentukan tujuan dari link ketika diklik.
- **Target**, menentukan target yang dituju saat diklik apakah di tab yang sama atau tab lain.
- **Title**, menentukan tulisan yang tampil saat mouse memasuki area link

Penulisan nilai atribut href memiliki beberapa cara tergantung dari fungsi link yaitu sebagai berikut:

- Jika link digunakan untuk membuka file cara penulisannya `href="galeri.html"`. Jika format file dapat dibuka di browser seperti html, txt, jpg, png atau gif, maka file akan ditampilkan pada halaman browser. Jika format file tidak dapat dibuka oleh browser seperti docx, xlsx, zip, rar, dan sebagainya, maka file akan didownload.
- Jika link digunakan untuk menuju bagian lain dalam satu halaman website maka penulisannya menjadi `href="#bawah"`. Agar link

tersebut berfungsi, maka harus ada tag `<a>` yang diberi atribut `id="bawah"` sebagai target yang dituju ketika link diklik.

- Jika link digunakan untuk menuju halaman website lain, maka penulisanya menjadi `href="http://www.nama_website.com"`.
- Jika link digunakan untuk mengirim email maka penulisanya menjadi `href="mailto: nama_email"`.

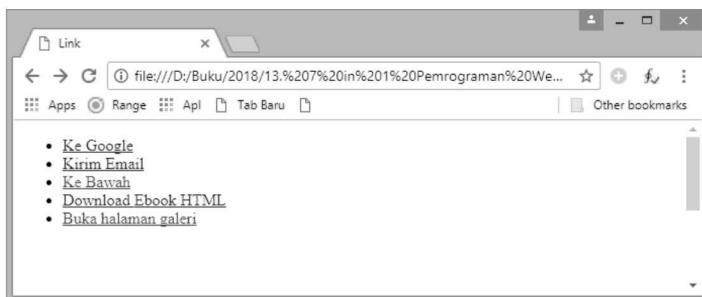
Untuk lebih jelasnya perhatikan contoh berikut:

File 2_9_link.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Link</title>
  </head>
  <body>
    <ul>
      <li> <a href="www://www.google.com" title="ke google"
target="blank"> Ke Google </a> </li>
      <li> <a href="mailto: rohi.abdulloh@gmail.com"> Kirim
Email </a> </li>
      <li> <a href="#bawah"> Ke Bawah </a> </li>
      <li> <a href="file/ebook_html.zip"> Download Ebook
HTML</a> </li>
      <li> <a href="11_gambar.html"> Buka halaman galeri </a>
</li>
    </ul>

    <br><br><br><br><br><br><br><br><br>
    <a id="bawah">Bagian Bawah</a>
  </body>
</html>
```

Skrip di atas berisi 5 link dengan fungsi yang berbeda-beda. Untuk mengetahui fungsinya, silakan klik masing-masing link.



Gambar 2.12 Berbagai jenis link

2.10 Iframe

Iframe digunakan untuk membuat area yang dapat menampilkan konten dari website lain. Tag yang digunakan yaitu `<iframe> ... </iframe>`. Tag ini sering digunakan untuk membuat widget pada blog. Atribut yang bisa digunakan yaitu:

- **Src**, menentukan sumber website yang ingin ditampilkan.
- **Width**, menentukan lebar area.
- **Height**, menentukan tinggi area.

Lebih jelasnya, perhatikan contoh berikut:

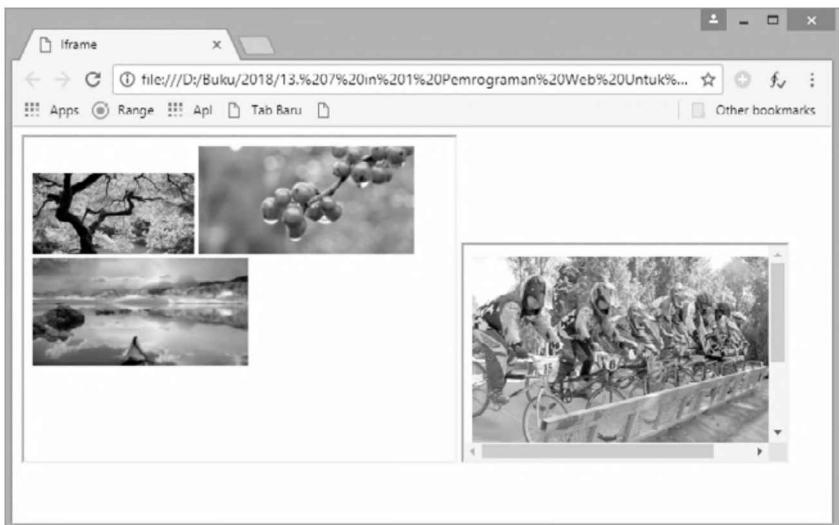
File 2_10_iframe.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Iframe</title>
  </head>
  <body>

    <iframe src="7_gambar.html" width="400"
height="300"></iframe>
    <iframe src="8_audio_video.html" width="300"
height="200"></iframe>

  </body>
</html>
```

Skrip di atas akan menampilkan hasil file yang telah dibuat sebelumnya, sesuai yang ditulis pada atribut src.



Gambar 2.13 Menampilkan file html dengan *iframe*

2.11 Formulir

Formulir digunakan untuk menginputkan data oleh pengguna pada website. Data yang diinputkan biasanya akan disimpan pada database. Tag yang digunakan untuk membuat formulir yaitu `<form> ... </form>`. Tag ini memiliki beberapa atribut yaitu:

- **Action**, menentukan nama file yang akan memproses data yang dikirim.
- **Method**, mengatur metode pengiriman data, bisa diisi *get* atau *post*.
- **Name**, mengatur nama form.

Perbedaan antara method *get* dan *post* pada form yaitu sebagai berikut:

Method GET:

- Hanya dapat mengirim data dalam ukuran kecil.
- Data yang dikirim akan ditampilkan pada address bar browser.

Method POST:

- Dapat mengirim data dalam ukuran besar.
- Data tidak ditampilkan pada address bar.

Di dalam tag `<form> ... </form>` ada beberapa tag yang digunakan untuk membuat alat input data, yaitu:

2.11.1 Tag `<label> ... </label>`

Tag `<label>` digunakan untuk membuat label dari sebuah alat input. Tag ini biasanya ditulis sebelum tag `<input>`. Atribut yang biasanya digunakan dalam tag ini adalah `for` yang diisi dengan nama `id` dari alat input yang diberi label.

2.11.2 Tag `<input>`

Tag `<input>` digunakan untuk membuat alat input dengan berbagai macam tipe. Atribut yang dapat digunakan pada tag ini yaitu:

- **Name**, memberi nama sebagai identitas data yang akan dikirim.
- **Value**, memberi nilai/isi data yang akan dikirim.
- **Size**, mengatur lebar alat input.
- **Placeholder**, mengatur tulisan yang tampil ketika nilainya kosong.
- **Disabled**, membuat input tidak dapat diubah isinya.
- **Readonly**, seperti disabled tetapi masih bisa diubah nilainya dengan skrip.
- **Type**, menentukan tipe alat input.
- **Id**, memberi identitas data yang akan dikirim.

Khusus tipe radio dan checkbox memiliki atribut `checked`. Adapun macam-macam tipe input yang dapat diberikan pada atribut `type`, yaitu:

- **Text**, untuk memasukkan text biasa.
- **Hidden**, untuk menyembunyikan input.
- **Password**, untuk memasukkan password.

- **Radio**, untuk memberi pilihan beberapa nilai yang dapat dipilih dengan cara diklik.
- **Checkbox**, untuk memberi pilihan dengan cara dicentang.
- **Submit**, untuk membuat tombol yang akan mengirim data ketika diklik.
- **Reset**, untuk membuat tombol yang akan mengosongkan formulir ketika diklik.
- **Button**, untuk membuat tombol biasa yang memiliki fungsi lain.

Pada HTML 5 ada penambahan beberapa tipe input. Tipe input yang ditambahkan pada HTML 5 adalah sebagai berikut:

- **Color**, untuk memilih warna.
- **Time**, untuk memasukkan waktu.
- **Number**, untuk memasukkan khusus angka.
- **Range**, untuk memilih angka pada batas tertentu dengan cara menggeser slider.
- **Date**, untuk memasukkan tanggal.
- **Email**, untuk memasukkan email.
- **Url**, untuk memasukkan URL (alamat website).

Beberapa tipe input di atas memiliki tampilan yang sama, namun ketika dimasuki nilai yang tidak sesuai dengan formatnya akan menampilkan pesan kesalahan. Sebagai contoh, perhatikan skrip berikut:

```
File 2_11_form_input.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Form</title>
  </head>
  <body>
    <form method="post" action="proses.php">
      <label for="nama">Nama</label>
      <input type="text" id="nama"> <br>

      <label for="password">Password</label>
      <input type="password" id="password"><br>

      <label for="jk">Jenis Kelamin</label>
```

```

        <input type="radio" id="jk" value="L" checked> Laki -
laki
        <input type="radio" id="jk" value="P" checked>
Perempuan <br>

        <label for="status">Status</label>
        <input type="checkbox" value="Y" id="status"> Menikah
<br>

        <label for="tanggal">Tanggal Lahir</label>
        <input type="date" id="tanggal"> <br>

        <label for="warna">Warna Favorit</label>
        <input type="color" id="warna"> <br>

        <label for="hp">No. HP</label>
        <input type="number" id="hp"> <br>

        <label for="berat">Berat Badan</label>
        <input type="range" id="berat"> <br>

        <label for="email">Email</label>
        <input type="email" id="email"> <br>

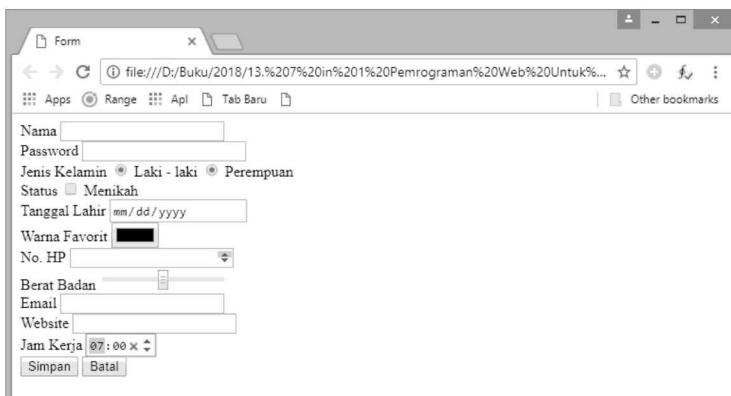
        <label for="website">Website</label>
        <input type="url" id="website"> <br>

        <label for="jam">Jam Kerja</label>
        <input type="time" id="jam"> <br>

        <input type="submit" value="Simpan">
        <input type="reset" value="Batal">
    </form>
</body>
</html>

```

Hasil dari skrip di atas seperti gambar berikut:



Gambar 2.14 Macam-macam tipe input

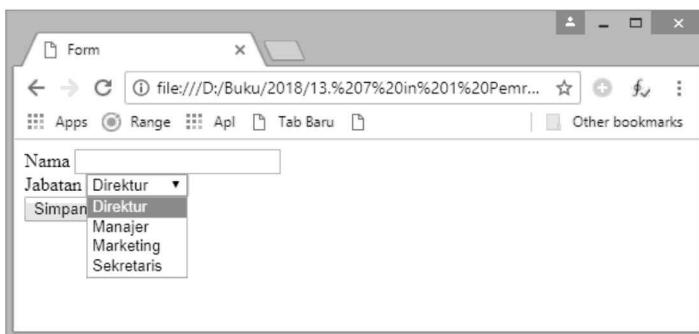
2.11.3 Tag <select> ... </select>

Tag <select> ... </select> digunakan untuk membuat input berupa pilihan dalam bentuk dropdown. Atribut yang bisa digunakan sama dengan atribut tag <input>. Yang membedakan pada tag ini dapat diberi atribut **multiple** yang memungkinkan pengguna dapat memilih lebih dari satu pilihan. Untuk menambahkan pilihan menggunakan tag <option> ... <option> yang ditulis di dalam tag <select> ... </select>. Lebih jelasnya, perhatikan contoh berikut:

File 2_11_form_select.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Form</title>
  </head>
  <body>
    <form method="post" action="proses.php">
      <label for="nama">Nama</label> <input type="text" id="nama"> <br>
      <label for="jabatan">Jabatan</label>
      <select id="jabatan">
        <option value="direktur"> Direktur </option>
        <option value="manajer"> Manajer </option>
        <option value="marketing"> Marketing </option>
        <option value="sekretaris"> Sekretaris </option>
      </select><br>
      <input type="submit" value="Simpan">
      <input type="reset" value="Batal">
    </form>
  </body>
</html>
```

Hasil skrip di atas seperti gambar berikut:



Gambar 2.15 Pilihan pada form

2.11.4 Tag <textarea> ... </textarea>

Tag <textarea> ... </textarea> digunakan untuk membuat input yang dapat diisi dengan text dalam jumlah banyak dan lebih dari satu baris, misalnya untuk input pesan, komentar, berita, dll. Atribut yang dapat digunakan juga sama, namun ada beberapa tambahan sebagai berikut:

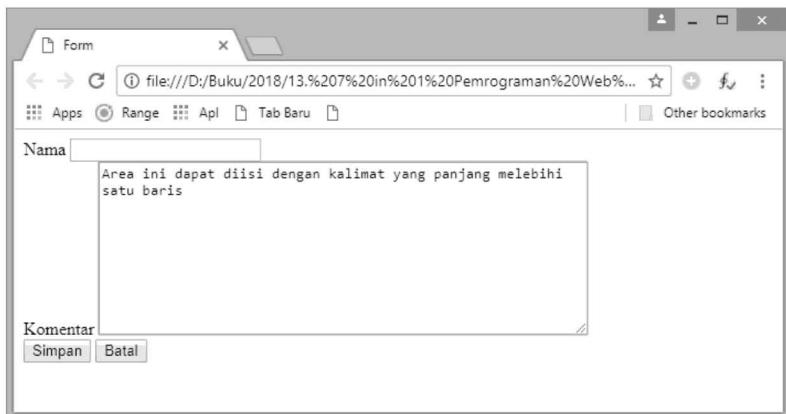
- **Cols**, untuk mengatur jumlah huruf yang bisa diinputkan dalam 1 baris.
- **Rows**, untuk mengatur jumlah baris yang dapat ditampilkan tanpa scroll.

Contoh penggunaanya perhatikan skrip berikut:

File 2_11_form_textarea.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Form</title>
  </head>
  <body>
    <form method="post" action="proses.php">
      <label for="nama">Nama</label>
      <input type="text" id="nama"> <br>
      <label for="komentar">Komentar</label>
      <textarea cols="60" rows="10" id="komentar">
</textarea> <br>
      <input type="submit" value="Simpan">
      <input type="reset" value="Batal">
    </form>
  </body>
</html>
```

Textarea pada skrip di atas diatur dengan lebar 60 huruf dan tinggi 10 baris teks. Hasilnya seperti gambar berikut:



Gambar 2.16 Tampilan textarea

2.12 Semantic HTML 5 Template

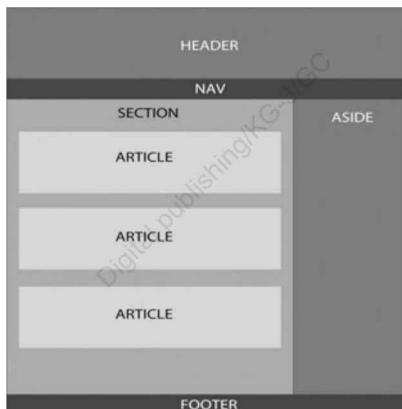
Sebelum ada HTML 5, biasanya template HTML dibuat dengan tag <div> yang diberi id atau class header, nav, menu, content, title, dan footer. Pada HTML 5, dikenalkan tag-tag baru yang akan mewakili bagian-bagian website, dengan tujuan untuk menyampaikan tujuan dari elemen-elemen tersebut kepada pengembang, browser dan algoritma mesin pencari.

Tag-tag yang ditambahkan tersebut yaitu sebagai berikut:

- Tag <header> ... </header>, digunakan untuk menunjukkan bagian yang berfungsi sebagai header website. Tag ini dapat muncul beberapa kali sesuai dengan kebutuhan. Bahkan di dalam sebuah konten memungkinkan terdapat header lagi.
- Tag <nav> ... </nav>, digunakan untuk menunjukkan bagian yang berfungsi sebagai navigasi.
- Tag <section> ... </section>, digunakan untuk memisahkan bagian-bagian pada konten website. Tag ini dapat digunakan berkali-kali dalam satu halaman website.
- Tag <main> ... </main>, digunakan untuk menunjukkan bagian paling penting dalam satu halaman website. Berbeda dengan tag

- <section>, tag ini biasanya hanya dimunculkan satu kali yaitu pada bagian paling penting saja, seperti konten/artikel utama.
- Tag <article> ... </article>, digunakan untuk menampung konten yang berupa artikel. Umumnya diletakkan di dalam tag <section> atau <main>.
 - Tag <aside> ... </aside>, digunakan untuk menunjukkan bagian yang bukan merupakan konten utam tetapi memiliki kaitan dengan konten yang sedang ditampilkan. Biasanya digunakan untuk menandai sidebar.
 - Tag <footer> ... </footer>, digunakan untuk menunjukkan bagian footer.

Dari penjelasan di atas, maka struktur pada sebuah template HTML dapat dibuat seperti gambar berikut:



Gambar 2.17 Contoh struktur template HTML

Struktur pada gambar di atas hanya contoh. Pada praktiknya, setruktur template bisa lebih kompleks dari gambar di atas. Bisa saja sebuah layout website memiliki beberapa header atau bahkan beberapa navigasi.

Jika layout pada gambar di atas diterjemahkan kedalam skrip HTML, maka menjadi seperti berikut:

File 2_12_template_semantic.html

```
<!DOCTYPE HTML>
<html>
```

```

<head>
  <title>Semantic Template</title>
</head>
<body>
  <header></header>
  <nav></nav>
  <section>
    <article></article>
    <article></article>
  </section>
  <aside></aside>
  <footer></footer>
</body>
</html>

```

Skrip di atas tentu belum menghasilkan apa-apa. Agar menghasilkan layout template sesuai gambar sebelumnya harus diatur menggunakan CSS.

2.13 Daftar Tag HTML

Walaupun tag-tag utama HTML yang sering digunakan telah dibahas satu per satu, masih banyak tag yang dimiliki HTML yang tidak memungkinkan untuk dibahas satu per satu. Sebagai referensi, tag-tag lain yang belum dibahas akan dirangkum dalam tabel berikut:

Nama Tag	Fungsi
<abbr>	<p>Menunjukkan akronim. Kepanjangan dari akronim ditulis pada atribut title.</p> <p>Saya sedang belajar <abbr title="Hypertext Markup Language">HTML</abbr></p>
<address>	<p>Menunjukkan informasi kontak author dokumen.</p> <pre> <address> Rohi Abdulloh Slawi, Tegal Jawa Tengah </address> </pre>
<area>	<p>Menunjukkan area di dalam image-map.</p> <pre> <map name="planetmap"> <area shape="rect" coords="0,0,82,126"> </pre>

	<pre>href="sun.htm" alt="Sun">> </map></pre>
<base>	Menentukan base URL untuk semua relative URL pada dokumen, digunakan di dalam tag <head>. <base href="https://www.rohiabdulloh.com" target="_blank">
<bdo>	Untuk mengubah arah teks yang berbeda dari arah teks yang sedang digunakan. <bdo dir="rtl"> Arah teks ini dari kanan ke kiri </bdo>
<bdi>	Untuk mengisolasi teks yang mungkin berbeda arah dari teks di sekitarnya. User <bdi>jdoe</bdi>: 80 points User <bdi>ابن</bdi>: 90 points
<blockquote>	Menunjukkan kutipan dari suatu sumber. <blockquote cite="http://www.rohiabdulloh.com"> Teks kutipan </blockquote>
<button>	Membuat tombol yang dapat diklik. <button type="button">Klik di sini </button>
<canvas>	Menunjukkan area canvas untuk menggambar grafik dengan skrip JavaScript. <canvas id="myCanvas"></canvas> <script> var canvas = document.getElementById("myCanvas"); var ctx = canvas.getContext("2d"); ctx.fillStyle = "#FF0000"; ctx.fillRect(80,80,0,0) </script>
<caption>	Membuat judul tabel. <table> <caption>Judul Tabel</caption> </table>
<cite>	Menunjukkan judul sebuah pekerjaan.

<colgroup>	Membuat grup satu atau lebih kolom pada tabel. <pre><table> <colgroup> <col span="2" style="background-color:red"> </colgroup> <tr> ... </tr> </table></pre>
<col>	Menentukan property masing-masing kolom pada tag <colgroup>.
<code>	Menunjukkan kode komputer.
<datalist>	Untuk membuat list data yang dapat dimasukkan pada tag <input> dengan cara id dari tag ini dimasukkan ke atribut list pada tag <input>. List data dibuat dengan tag <option> <pre><input list="browsers"> <datalist id="browsers"> <option value="Internet Explorer"> <option value="Firefox"> <option value="Chrome"> </datalist></pre>
<dfn>	Menunjukkan kata yang didefinisikan. <p><p><dfn>HTML</dfn> adalah bahasa standar untuk membuat halaman website.</p></p>
<div>	Membuat area pada dokumen HTML yang memenuhi lebar container, sehingga elemen berikutnya otomatis berada di bawahnya. <pre><div class="container"> <div class="header"> ... </div> <div class="content"> ... </div> <div class="footer"> ... </div> </div></pre>
<dl>	Membuat list deskripsi. <pre><dl> <dt>Kopi</dt> <dd>Minuman panas hitam</dd> <dt>Susu</dt> <dd>Minuman dingin putih</dd> </dl></pre>
<dt>	Menunjukkan item yang dideskripsikan dari tag <dl>.
<dd>	Menunjukkan deskripsi dari kata yang ditunjukkan tag <dt>.
<embed>	Membuat area untuk memasukkan aplikasi eksternal (non-HTML). <pre><embed src="helloworld.swf"></pre>
<fieldset>	Mengelompokkan beberapa elemen pada form.

	<pre><form> <fieldset> <legend>Profil:</legend> Nama: <input type="text">
 Email: <input type="text">
 </fieldset> </form></pre>
<kbd>	<p>Menunjukkan input keyboard.</p> <p>Silakan tekan <kbd>Ctrl+D</kbd></p>
<legend>	Membuat judul pada tag <fieldset>.
<map>	Membuat image-map.
<menu>	<p>Membuat daftar menu yang dapat diakses dengan cara klik kanan.</p> <pre><menu label="Bagikan ke..."> <menuitem label="Twitter" icon="twitter.png" ==""></menuitem> <menuitem label="Facebook" icon="facebook.png"></menuitem> </menu></pre>
<menuitem>	Membuat daftar menu yang dapat diklik dalam tag <menu>.
<noscript>	<p>Menunjukkan konten alternatif yang tidak support client-side script.</p> <pre><script> document.write("Hello World!") </script> <noscript>Browser tidak mendukung JavaScript!</noscript></pre>
<object>	<p>Untuk embed objek.</p> <pre><object data="horse.wav"> <param name="autoplay" value="true"> </object></pre>
<param>	Menunjukkan parameter objek.
<output>	Menunjukkan hasil sebuah perhitungan.
<pre>	<p>Menunjukkan teks yang tidak diformat, dapat digunakan untuk menulis kode program.</p> <pre><pre> Teks ini Akan ditampilkan Apa adanya </pre></pre>
	Elemen yang digunakan untuk membuat elemen yang akan diberi class tertentu pada sebagian elemen tanpa ganti baris.

	<p>Adikku memiliki mata biru.</p>
<style>	Untuk menuliskan kode CSS di dalamnya. <pre><style> h1 {color:red;} p {color:blue;} </style></pre>
<time>	Menunjukkan tanggal atau waktu.
<var>	Menunjukkan sebuah variabel.

2.14 Global Attribute

Beberapa atribut yang berhubungan dengan suatu tag tertentu telah dijelaskan bersamaan dengan pembahasan tag yang bersangkutan. Namun, ada beberapa atribut yang dapat digunakan untuk semua tag HTML, tidak terkait dengan suatu tag tertentu. Atribut tersebut akan dirangkum pada tabel berikut:

Nama Tag	Fungsi
Accesskey	Menentukan shortcut key untuk mengaktifkan elemen, diisi huruf yang menjadi shortcut.
Class	Menentukan sebuah kelas atau lebih yang akan menjadi referensi dalam pengaturan desain dengan CSS.
Contenteditable	Menentukan apakah sebuah elemen dapat diedit atau tidak, nilainya true atau false.
Contextmenu	Menentukan context menu pada sebuah elemen yang akan tampil ketika melakukan klik kanan pada elemen tersebut, diisi id dari tag <menu> yang akan menjadi menu.
Data-*	Untuk menyimpan data khusus dari sebuah aplikasi web. Tanda bintang dapat diganti kata apa saja yang menjadi identitas data.
Dir	Menentukan arah teks pada sebuah elemen, dapat diisi ltr, rtl atau auto.
Draggable	Menentukan apakah sebuah elemen dapat di-drag atau tidak, nilainya true, false atau auto.
Dropzone	Menentukan perlakuan terhadap data yang di-drag ke dalamnya, dapat diisi copy, move atau link.
Hidden	Menentukan sebuah elemen disembunyikan atau tidak, digunakan tanpa memberi nilai.

Id	Menentukan identitas dari elemen yang membedakan dari elemen lainnya.
Lang	Menentukan bahasa yang digunakan pada elemen, diisi dengan kode bahasa.
Spellcheck	Menentukan apakah elemen menerapkan pengecekan pengejaan kata atau tidak, nilainya true atau false.
Style	Menentukan style elemen menggunakan kode CSS.
Tabindex	Menentukan urutan aktif ketika menggunakan tombol tab sebagai navigasi, diisi dengan angka mulai dari 1.
Title	Menentukan informasi tambahan dari sebuah elemen, yang akan tampil ketika pointer mouse menunjuk elemen.
Translate	Menunjukkan apakah sebuah elemen diterjemahkan atau tidak, dapat diisi yes atau no.

3 CSS

3.1 Mengenal CSS

CSS adalah singkatan dari *Cascading Style Sheet* yaitu dokumen web yang berfungsi mengatur elemen HTML dengan berbagai property yang tersedia sehingga dapat tampil dengan berbagai gaya yang diinginkan. Sebagian orang menganggap CSS bukan termasuk salah satu bahasa pemrograman kerena memang strukturnya yang sederhana, hanya berupa kumpulan-kumpulan aturan yang mengatur style elemen HTML.

Cara kerja CSS dalam memodifikasi HTML dengan memilih elemen HTML yang akan diatur kemudian memberikan property yang sesuai dengan tampilan yang diinginkan. Dalam memberikan aturan pada elemen HTML, skrip CSS terdiri atas 3 bagian yaitu **selector** untuk memilih elemen yang akan diberi aturan, **property** yang merupakan aturan yang diberikan dan **value** sebagai nilai dari aturan yang diberikan.

3.2 Dasar-Dasar CSS

3.2.1 Penulisan CSS

Cara penulisan CSS dibedakan menjadi 3 macam yaitu **inline**, **internal** dan **external**. Ketiganya dapat digunakan sesuai dengan kebutuhan. Berikut penjelasan masing-masing metode dalam penulisan CSS:

Inline CSS

Inline CSS yaitu menuliskan CSS dengan menggunakan atribut `style` yang langsung dituliskan di dalam tag HTML. Cara ini cocok digunakan jika CSS memiliki satu atau dua property saja dan bersifat permanen. Salah satu kekurangan penggunaan cara ini yaitu susahnya ketika terjadi perbaikan desain. Perhatikan contoh berikut:

File 3_2_1_inline.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Inline</title>
  </head>
  <body>
    <p style="color: red"> Tulisan ini berwarna merah </p>
    <p style="font-weight: bold"> Tulisan ini dicetak tebal
  </p>
  </body>
</html>
```

Dari skrip di atas, dapat dilihat bahwa penggunaan CSS dengan cara ini membuat skrip HTML menjadi panjang, apalagi jika terdiri dari beberapa property. Tentu membuat tampilan skrip menjadi tidak rapi.

Internal CSS

Internal CSS yaitu penulisan CSS menggunakan tag `<style> ... </style>` yang ditulis di dalam elemen `<head> ... </head>`. Cara ini cocok digunakan jika skrip CSS hanya digunakan pada 1 halaman HTML dan baris CSS hanya terdiri dari sedikit selector. Cara ini masih memiliki kekurangan, yaitu skrip CSS tidak dapat digunakan untuk mengatur file HTML lain. Contoh skripnya sebagai berikut:

File 3_2_1_internal.css

```
<!DOCTYPE HTML>
<html>
  <head>
    <style type="text/css">
      p{
        color: red;
        font-weight: bold;
      }
    </style>
  </head>
  <body>
    <p> Tulisan ini berwarna merah </p>
    <p> Tulisan ini dicetak tebal </p>
  </body>
```

```
</html>
```

External CSS

External CSS yaitu penulisan CSS dimana skrip CSS disimpan dalam file tersendiri dengan ekstensi .css dan terpisah dengan file HTML. Untuk menghubungkan file HTML dan file CSS, file CSS dipanggil di dalam elemen `<head> ... </head>` pada file HTML seperti berikut:

```
<link rel="stylesheet" href="nama_file.css">
```

Jika file CSS terdapat pada suatu folder, maka nama folder disertakan pada atribut href. Aturanya sama seperti penulisan nilai src pada tag `` yang dibahas pada bab sebelumnya. Sebagai contoh, perhatikan contoh file HTML dan CSS berikut:

File 3_2_1_external.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Internal</title>
    <link rel="stylesheet" href="3_2_1_external.css">
  </head>
  <body>
    <p> Tulisan ini berwarna merah </p>
    <p> Tulisan ini dicetak tebal </p>
  </body>
</html>
```

File 3_2_1_external.css

```
p{
  color: red;
  font-weight: bold;
  text-transform: uppercase;
}
```

Cara penulisan external css adalah cara penulisan yang paling disarankan. Kelebihan dari cara penulisan ini adalah, file CSS dapat digunakan oleh banyak file HTML. Tinggal panggil file CSS dari file HTML yang akan menggunakaninya.

Dalam banyak contoh penggunaan CSS pada bab ini, penulis akan menggunakan cara internal, tujuannya agar skrip CSS dan HTML ditampilkan bersamaan sehingga diharapkan lebih mudah dipahami.

3.2.2 Selector

Selector adalah sebuah kata yang digunakan untuk memilih elemen HTML tertentu yang akan dimodifikasi dengan skrip CSS. Secara garis besar ada tiga macam selector dalam penulisan CSS yaitu selector tag, selector class, dan selector id. Penjelasanya masing-masing sebagai berikut:

Selector tag

Selector tag yaitu selector dengan menyebutkan secara langsung nama tag HTML yang ingin diatur dengan CSS. Penyebutan tag HTML tanpa disertai tanda <>, cukup ditulis nama tagnya saja. Contohnya sebagai berikut:

```
File 3_2_2_selector_tag.html

<!DOCTYPE HTML>
<html>
  <head>
    <style type="text/css">
      h2{
        color: red;
      }
      p{
        text-decoration: underline;
      }
    </style>
  </head>
  <body>
    <h2> Judul Artikel</h2>
    <p> Isi Artikel </p>
  </body>
</html>
```

Pada skrip di atas jelas sekali, selector pada CSS menggunakan nama tag **p** dan **h2** secara langsung tanpa simbol pembuka dan penutup tag.

Selector class

Selector class yaitu selector dengan menyebutkan nama class dari suatu tag yang ingin dimodifikasi dengan CSS. Penyebutan nama class disertai titik (.) sebelum nama class. Selector ini lebih spesifik daripada selector tag. Contohnya skrip berikut:

```
File 3_2_2_selector_class.html
```

```

<!DOCTYPE HTML>
<html>
  <head>
    <style type="text/css">
      h2{
        color: red;
      }
      .biru{
        color: blue;
      }
    </style>
  </head>
  <body>
    <h2> Judul Artikel</h2>
    <p> Isi Artikel pertama </p>
    <p class="biru"> Isi Artikel kedua</p>
    <p class="biru"> Isi Artikel ketiga</p>
  </body>
</html>

```

Selector id

Selector id Yaitu penulisan selector dengan menuliskan nama id dari elemen HTML yang akan dimodifikasi dengan CSS. Penulisan nama id disertai dengan tanda hastag (#) sebelum nama id. Satu nama id hanya bisa digunakan oleh satu elemen HTML saja. Selector ini adalah yang paling spesifik dari dua selector yang lain. Contohnya sebagai berikut:

File 3_2_2_selector_id.html

```

<!DOCTYPE HTML>
<html>
  <head>
    <style type="text/css">
      h2{
        color: red;
      }
      .biru{
        color: blue;
      }
      #tebal{
        font-weight: bold;
        font-size: 16px;
      }
    </style>
  </head>

  <body>
    <h2> Judul Artikel</h2>
    <p> Isi Artikel pertama </p>
    <p class="biru"> Isi Artikel kedua</p>
    <p class="biru" id="tebal"> Isi Artikel ketiga</p>
  </body>
</html>

```

3.2.3 Daftar Selector CSS

Walaupun ada 3 macam selector dasar, namun pengembangan dari 3 macam tersebut dapat memunculkan berbagai kombinasi selector yang dirangkum pada tabel berikut:

Selector	Contoh	Keterangan
*	*	Memilih semua elemen
elemen, elemen	div, p	Memilih tag <div> dan tag <p> sekaligus
elemen elemen	div p	Memilih tag <p> yang berada di dalam <div>
elemen>elemen	div>p	Memilih tag <p> yang berada di dalam tag <div> secara langsung
elemen+elemen	div+p	Memilih tag <p> yang berada langsung setelah tag <div>
elemen~elemen	div~p	Memilih tag <p> yang memiliki parent (container) sama dengan tag <div>
[attr]	[disabled]	Memilih semua elemen yang memiliki atribut disabled
[attr=value]	[type=text]	Memilih semua elemen yang memiliki atribut type=text
[attr~=value]	[title~=indah]	Memilih elemen yang memiliki nilai atribut class mengandung kata "indah", contoh: "pantai indah"
[attr =value]	[class =my]	Memilih elemen yang memiliki nilai atribut class dimulai dengan kata "my", contoh: my-name, my-address
[attr^=value]	[class^=ti]	Memilih elemen yang memiliki nilai atribut class diawali dengan kata "ti", contoh: tinggi, tipis
[attr\$=value]	[class\$=ar]	Memilih elemen yang memiliki nilai atribut class diakhiri dengan kata "ar", contoh: lebar, besar
[attr*=value]	[class*=in]	Memilih elemen yang nilai atribut class mengandung kata potongan kata "in", contoh: indah, miring

3.2.4 Property dan Value

Property merupakan sifat-sifat atau aturan yang ingin diberikan kepada elemen HTML seperti warna, background, jenis huruf, dan sebagainya. Nilai dari sebuah property disebut **value**. Property dari sebuah selector

terletak di antara tanda { dan }. Property dan value dipisah dengan tanda ":" dan diakhiri dengan tanda ";".

Value dari setiap property berbeda-beda, ada yang berupa angka dan ada yang berupa salah satu pilihan dari beberapa value yang disediakan untuk property tertentu.

Hampir setiap property pada CSS memiliki turunan, misalnya property ***background*** memiliki turunan berupa property ***background-image***, ***background-repeat***, ***background-position*** dan sebagainya. Penjelasan lebih lengkap masing-masing property dan value akan dibahas secara khusus dalam subbab tersendiri.

3.2.5 CSS-Shorthand

Seperti dijelaskan sebelumnya, setiap property memiliki beberapa turunan misalnya ***padding***, memiliki property turunan ***padding-top***, ***padding-right***, ***padding-bottom*** dan ***padding-left***. Daripada menuliskannya satu-satu, CSS menyediakan fitur yang disebut **css-shorthand**, yaitu menggabungkan beberapa value dari beberapa property turunan yang diturunkan dari property yang sama. Caranya dengan menggunakan property asal dan masing-masing value dari property turunan diurutkan dengan urutan yang telah ditentukan.

Sebagai contoh, misal terdapat property sebagai berikut:

```
div{  
    padding-top: 15px;  
    padding-right: 20px;  
    padding-bottom: 15px;  
    padding-left: 20px;  
}
```

Penulisan CSS di atas dapat disederhanakan dengan property ***padding***, dengan urutan value searah jarum jam dimulai dari ***padding-top***, yaitu sebagai berikut:

```
div{  
    padding: 15px 20px 15px 20px;  
}
```

Krip di atas, masih dapat disingkat lagi yaitu dengan menggabungkan ***top*** dengan ***bottom*** dan menggabungkan ***right*** dengan ***left***, dengan syarat keduanya memiliki nilai yang sama dan urutannya dimulai dari ***top*** dan ***bottom*** baru ***right*** dan ***left***, sehingga menjadi seperti berikut:

```
div{  
    padding: 15px 20px;  
}
```

Jika keempatnya memiliki nilai yang sama, maka dapat dituliskan hanya dalam satu value saja. Misalnya angka 15 pada skrip di atas diganti 20 juga, maka penulisannya dapat dipersingkat seperti berikut:

```
div{  
    padding: 20px;  
}
```

Pada skrip di atas berarti seluruh padding baik top, right, bottom dan left memiliki nilai 20px.

Penyederhanaan penulisan property seperti contoh di atas disebut CSS-shorthand. Bentuk-bentuk CSS-shorthand pada property yang lain akan dibahas secara bertahap.

3.2.6 Pseudo-Class

Pseudo-class digunakan untuk menunjukkan keadaan khusus pada selector yang dipilih. Misalnya elemen yang ditunjuk pointer mouse, elemen yang fokus, elemen aktif dan sebagainya. Ciri-ciri pseudo-class yaitu diawali tanda titik dua (:). Pseudo-class ditulis setelah selector tanpa dipisahkan spasi.

Beberapa contoh pseudo-class pada CSS dapat dilihat pada tabel berikut:

Selector	Contoh	Keterangan
:active	a:active	Memilih link yang aktif
:checked	input:checked	Memilih tag <input> yang dicentang, biasanya untuk tipe radio dan checkbox
:disabled	input:disabled	Memilih tag <input> yang dinonaktifkan
:empty	p:empty	Memilih tag <p> yang tidak memiliki elemen anak
:enable	input:enable	Memilih tag <input> yang tidak dinonaktifkan (enable)
:first-child	p:first-child	Memilih tag <p> pertama dalam satu container
:focus	input:focus	Memilih tag <input> yang sedang fokus

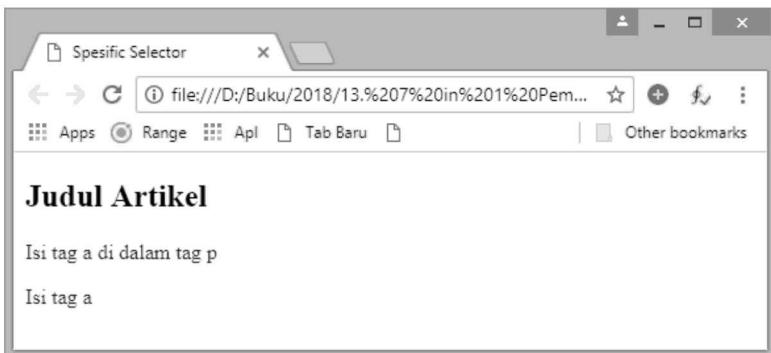
:hover	a:hover	Memilih link yang sedang ditunjuk mouse
:last-child	p:last-child	Memilih tag <p> terakhir dalam satu container
:link	a:link	Memilih link yang belum dikunjungi
:not(selector)	:not(p)	Memilih selain tag <p>
:nth-child(n)	p:nth-child(2)	Memilih tag <p> pada urutan kedua
:nth-last-child(n)	p:nth-last-child(2)	Memilih tag <p> pada urutan kedua dari akhir
:read-only	input:read-only	Memilih tag <input> yang memiliki status read only
:visited	a:visited	Memilih link yang telah dikunjungi
:required	input:required	Memilih tag <input> yang harus diisi
:valid	input:valid	Memilih tag <input> yang memiliki nilai valid
:target	#tab:target	Memilih elemen dengan id tab yang aktif (menjadi target klik dari suatu link)

Sebagai contoh, perhatikan skrip berikut:

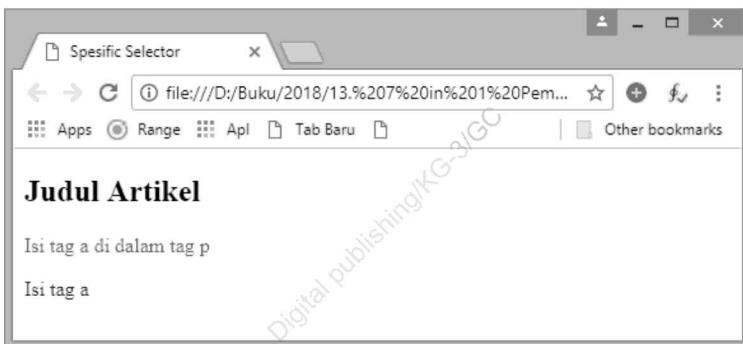
File 3_2_6_pseudo-class.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <style type="text/css">
      .biru{
        color: blue;
      }
      .biru:hover{
        color: red;
        text-decoration: none;
      }
    </style>
  </head>
  <body>
    <h2> Judul Artikel</h2>
    <p><a class="biru"> Isi tag a di dalam tag p </a> </p>
      <a class="biru"> Isi tag a</a>
    </body>
  </html>
```

Hasilnya seperti pada gambar berikut:



Gambar 3.1 Sebelum ditunjuk mouse teks berwarna biru



Gambar 3.2 Setelah ditunjuk mouse tulisan teks berwarna merah

3.2.7 Pseudo-Element

Pseudo-element digunakan untuk memilih bagian dari suatu elemen. Ciri-ciri pseudo-element yaitu diawali dengan tanda titik dua dobel (::). Beberapa pseudo-element yang dapat digunakan pada CSS dapat dilihat pada tabel berikut:

Selector	Contoh	Keterangan
::after	p::after	Menyisipkan sesuatu setelah konten pada tag <p>
::before	p::before	Menyisipkan sesuatu sebelum konten pada tag <p>
::first-letter	p::first-letter	Memilih huruf pertama pada tag <p>

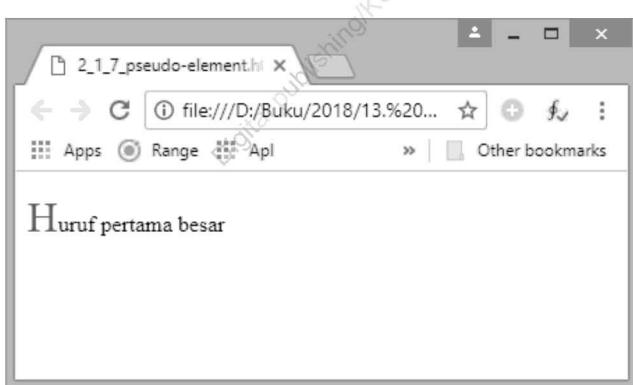
::first-line	p::first-line	Memilih baris pertama pada tag <p>
::selection	p::selection	Memilih bagian elemen <p> yang dipilih (diblok) oleh pengguna.

Sebagai contoh, perhatikan skrip berikut:

File 3_2_7_pseudo-element.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <style type="text/css">
      p::first-letter{
        color: red;
        font-size: 30px;
      }
    </style>
  </head>
  <body>
    <p>Huruf pertama besar</p>
  </body>
</html>
```

Hasilnya seperti gambar berikut:



Gambar 3.3 Penggunaan psoudo-element

3.3 Unit CSS

Banyak dari property di CSS yang nilainya berupa angka. Angka-angka tersebut biasanya memiliki satuan atau unit yang berbeda-beda

tergantung dari property yang digunakan. Unit pada CSS dibagi menjadi 2, yaitu relative dan absolute.

3.3.1 Unit Relative

Unit relative yaitu unit yang menunjukkan ukuran yang berubah-ubah menyesuaikan property lain, serta akan berubah ketika melakukan zoom pada browser. Yang termasuk dalam kelompok ini yaitu:

- **em**, relative terhadap ukuran font-size dari elemen, 2em berarti 2 kali dari ukuran font yang digunakan saat ini.
- **ex**, relative terhadap x-height dari font yang digunakan saat ini.
- **ch**, relative terhadap lebar “0” (nol).
- **rem**, relative terhadap font-size dari root elemen.
- **vw**, relative terhadap 1% dari lebar viewport, akan berubah ketika lebar browser diubah.
- **vh**, relative terhadap 1% dari tinggi viewport, akan berubah ketika tinggi browser diubah.
- **vmin**, relative terhadap 1% dari ukuran viewport yang lebih kecil antara lebar dan tinggi.
- **vmax**, relative terhadap 1% dari ukuran viewport yang lebih besar antara lebar dan tinggi.
- **%**, relative terhadap ukuran container, biasanya digunakan pada property width dan height.

Unit yang paling umum digunakan biasanya em dan %.

3.3.2 Unit Absolute

Unit absolute yaitu unit yang ukurannya sudah pasti, tidak berubah dan tidak terpengaruh oleh elemen manapun. Yang termasuk dalam kelompok ini yaitu **cm** (centimeter), **mm** (millimeter), **in** (inci), **px** (pixel), **pt** (point) dan **pc** (pica). Perbandingan dari masing-masing ukuran yaitu sebagai berikut:

- $1\text{in} = 96\text{px} = 2,54\text{cm}$
- $1\text{px} = 1/96\text{in}$

- 1pt = 1/72in
- 1pc = 12pt

3.4 Warna CSS

CSS memiliki dukungan terhadap berbagai jenis warna, artinya setiap property yang menunjukkan warna seperti color, background-color, border-color dan sebagainya, dapat diberi nilai sesuai jenis warna yang dipilih. Berikut jenis-jenis warna yang dapat digunakan pada CSS:

- **Color name**, yaitu nama-nama warna dalam bahasa inggris, seperti: red, blue, green, dan sebagainya. CSS support lebih dari 100 nama warna. Contoh: `background: red`.
- **Heksadesimal color**, yaitu kode warna dengan format #RRGGBB, dimana RR merupakan nilai heksadesimal dari red (merah), GG nilai heksadesimal dari green (hijau), dan BB nilai heksadesimal dari blue (biru). Angka yang dapat diberikan pada masing-masing warna antara 00 sampai FF. Jika semuanya 00 menghasilkan warna hitam dan jika semuanya FF menghasilkan warna putih. Misal `#23a4b1` berarti warna merah 23, warna hijau a4 dan warna biru b1. Contoh: `background: #ffe2c4`.
- **RGB color**, yaitu nilai warna menggunakan fungsi `rgb()` dengan format `rgb(merah, hijau, biru)`. Nilai masing-masing warna dapat diisi antara 0 sampai 255 atau 0% sampai 100%. Contoh: `background: rgb(255, 200, 1)`.
- **RGBA color**, yaitu nilai warna menggunakan fungsi `rgba()` dengan format `rgba(merah, hijau, biru, alpha)`. Nilai alpha antara 0.0 hingga 1.0. Nilai alpha semakin rendah, maka warnanya semakin transparan. Jenis warna ini dapat digunakan untuk membuat background transparan dengan mengatur nilai alpha kurang dari 1. Contoh: `background: rgba(255, 200, 1, 0.5)`.
- **HSL color**, yaitu nilai warna menggunakan fungsi `hsl()` dengan format `hsl(hue, saturation, lightness)`. Hue merupakan nilai derajat dalam lingkaran warna, nilainya antara 0 sampai 360. 0 atau 360 berarti merah, 120 berarti hijau dan 240 berarti biru. Nilai saturation antara 0% sampai 100%, dimana semakin kecil semakin mendekati abu-abu. Nilai lightness antara 0% sampai 100%, semakin kecil

semakin mendekati hitam dan semakin besar semakin mendekati putih. Contoh: **background: hsl(290, 60%, 70%)**.

- **HSLA color**, yaitu nilai warna menggunakan fungsi **hsla()** dengan format **hsla(hue, saturation, lightness, alpha)**. Nilai alpha sama dengan RGBA, artinya jenis warna ini juga dapat digunakan untuk membuat background transparan. Contoh: **background(290, 60%, 70%, 0.4)**.

3.5 Property CSS

3.5.1 Box Model

Box model merupakan sebuah konsep dimana setiap elemen pada halaman website dianggap layaknya sebagai sebuah box (kotak). Box ini yang kemudian membangun struktur desain dan layout dari sebuah website. Konsep box ini melekat pada setiap elemen HTML.

Height dan Width

Property **height** menentukan tinggi box, sedangkan **width** menentukan lebar box. Jika memberi background pada tag <div> dengan CSS, biasanya background tidak akan tampil jika belum mengatur lebar dan tinggi box dengan property ini. Nilai dari property ini berupa angka dengan satuan biasanya px atau %.

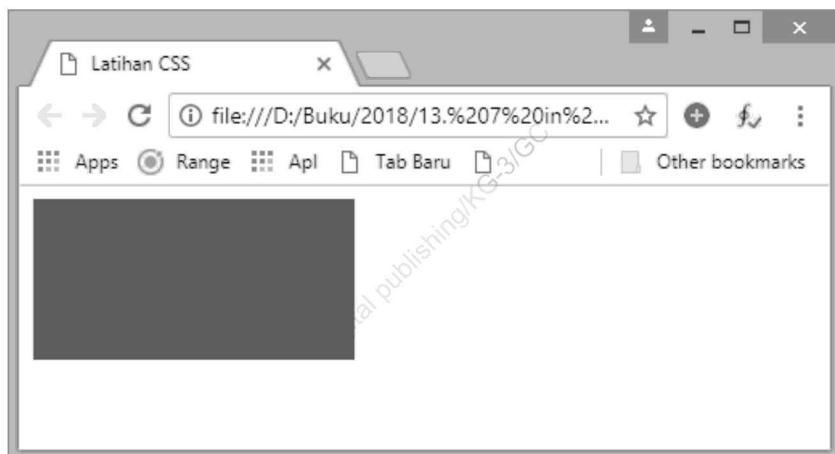
Property lain yang berhubungan dengan width dan height yaitu sebagai berikut:

- **Max-width**, mengatur lebar menyesuaikan konten tetapi tidak melebihi batas yang ditentukan.
- **Max-height**, mengatur tinggi menyesuaikan konten tetapi tidak melebihi batas yang ditentukan.
- **Min-width**, mengatur lebar menyesuaikan konten, tetapi tidak kurang dari batas yang ditentukan.
- **Min-height**, mengatur tinggi menyesuaikan konten, tetapi tidak kurang dari batas yang ditentukan.

Contoh penggunaan width dan height, perhatikan skrip berikut:

File 3_5_1_width_height.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Latihan CSS</title>
    <style type="text/css">
      .kotak{
        background: red;
        width: 200px;
        height: 100px;
      }
    </style>
  </head>
  <body>
    <div class="kotak"></div>
  </body>
</html>
```



Gambar 3.4 Property *width* dan *height*

Margin dan Padding

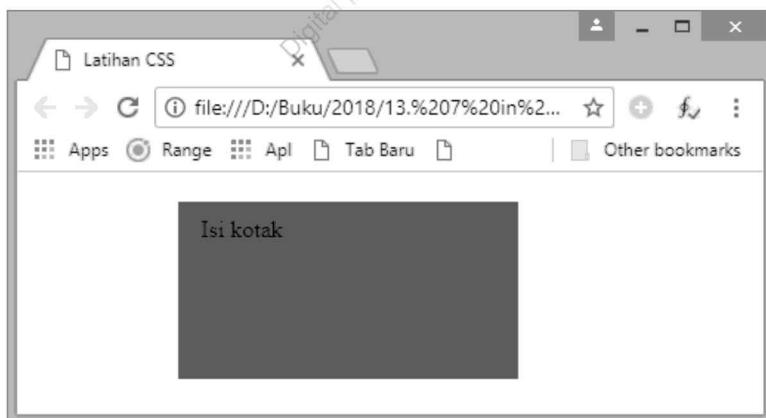
Property margin mengatur jarak box dari elemen di sekitar box atau dengan elemen pembungkus box (container). Sedangkan padding mengatur jarak box dengan konten yang berada di dalamnya. Property margin memiliki turunan *margin-top*, *margin-right*, *margin-bottom*, dan *margin-left*. Adapun padding terdiri dari *padding-top*, *padding-right*, *padding-bottom*, dan *padding-left*. Keempat property turunan baik margin maupun padding dapat disederhanakan menjadi *margin: [top] [right] [bottom] [left]* dan *padding: [top] [right] [bottom] [left]*. Nilai dari kedua property ini berupa angka dengan satuan yang umum

digunakan yaitu **px**. Nilai lain yang dapat diberikan yaitu *inherit* dan *auto*. Jika diisi *auto*, browser akan membuat nilainya sama pada setiap sisi, sehingga dapat membuat box rata tengah.

Khusus property padding, property ini akan mempengaruhi lebar atau tinggi box secara keseluruhan. Jadi, misalnya ada box memiliki lebar 60px dan diberi padding 10px, maka lebarnya menjadi 80px karena ditambah kanan 10px dan kiri 10px.

File 3_5_1_margin_padding.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Latihan CSS</title>
    <style type="text/css">
      .kotak{
        background: red;
        width: 200px;
        height: 100px;
        margin: 20px 0 0 100px;
        padding: 10px 15px;
      }
    </style>
  </head>
  <body>
    <div class="kotak"> Isi kotak </div>
  </body>
</html>
```



Gambar 3.5 Property margin dan padding

Pada gambar di atas terlihat jelas, terdapat jarak di sebelah kanan dan atas box sesuai dengan pengaturan margin, serta jarak dengan teks di dalamnya sesuai dengan pengaturan padding.

Border dan Outline

Property border memberikan bingkai pada box. Property border memiliki turunan yang dibagi menjadi 2. Bagian pertama menunjukkan sisi border, terdiri dari **border-top**, **border-right**, **border-bottom** dan **border-left**. Bagian kedua mengatur tampilan border, terdiri dari **border-width** (tebal bingkai), **border-style** (sifat bingkai) dan **border-color** (warna bingkai). Kedua bagian tersebut masih dapat digabung lagi menjadi: **border-top-style**, **border-left-style**, **border-top-color**, **border-left-color**, **border-top-size**, dan sebagainya.

Jika turunan bagian pertama digabungkan, urutannya sama dengan margin dan padding. Sedangkan jika bagian kedua digabungkan urutanya sebagai berikut:

```
border: [border-width] [border-style] [border-color];
```

Nilai dari border-style yang dapat diberikan yaitu: *dotted*, *dashed*, *solid*, *double*, *groove*, *ridge*, *inset*, *outset*, *none*, *hidden*, dan *mix*.

Seperti padding, border juga menambah total lebar dari sebuah box. Jika lebar box 20px, dan border 2px, maka total lebar menjadi 24px (kanan 2px dan kiri 2px). Jika ditambah padding 5px, maka lebar totalnya menjadi 34px.

Ada property yang sejenis dengan border, yaitu **outline**. Turunan dan aturan penggunaan property ini sama persis dengan border termasuk nilai property yang dapat diberikan. Bedanya, outline tidak termasuk dari bagian element, sehingga tidak mempengaruhi lebar box. Bahkan, outline bisa saja menutupi konten lain. Ada satu turunan property outline yang tidak dimiliki border yaitu **outline-offset** yang mengatur jarak outline dengan elemen box.

File 3_5_1_border.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <style type="text/css">
      .kotak{
        background: red;
        width: 200px;
```

```

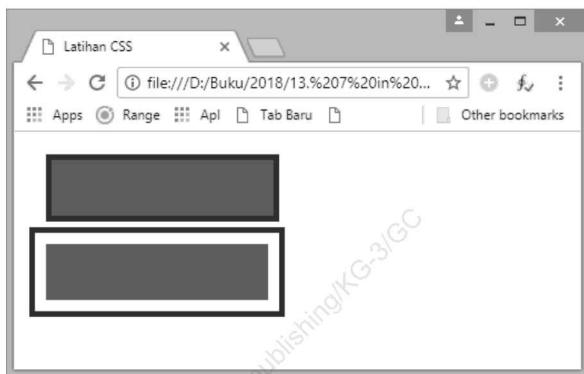
        height: 50px;
        margin: 20px;
    }
    .border{ border: 5px solid blue; }
    .outline{
        outline: 5px solid blue;
        outline-offset: 10px;
    }

```

```

</style>
</head>
<body>
    <div class="kotak border"></div>
    <div class="kotak outline"></div>
</body>
</html>>

```



Gambar 3.6 Property border dan outline

Position

Property position menentukan posisi suatu box terhadap objek lain. Nilai yang dapat diberikan pada property ini sebagai berikut:

- **Static**, posisinya tidak terpengaruh oleh property top, bottom, left dan right.
- **Relative**, posisinya terpengaruh oleh property top, bottom, left dan right.
- **Fixed**, posisinya tidak berubah walaupun melakukan scroll pada layar.
- **Absolute**, posisinya dapat diatur secara bebas terhadap container yang diberi posisi relative.

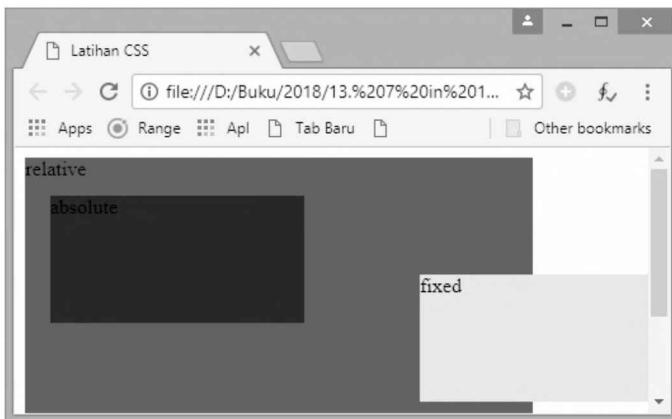
- **Sticky**, menggabungkan posisi relative dan fixed tergantung posisi scroll.

Ada beberapa property yang penggunaanya tergantung dari property position, dimana property ini tidak terpengaruh jika nilai position static. Property tersebut yaitu sebagai berikut:

- **Top**, mengatur jarak elemen dari atas.
- **Right**, mengatur jarak elemen dari kanan.
- **Bottom**, mengatur jarak elemen dari bawah.
- **Left**, mengatur jarak elemen dari kiri.
- **Z-index**, mengatur urutan elemen yang saling menumpuk, dimana yang nilainya paling tinggi akan menutupi yang nilainya rendah.

File 3_5_1_position.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Latihan CSS</title>
    <style type="text/css">
      .relative{
        background: red; width: 400px; height: 300px;
        position: relative;
      }
      .absolute{
        background: blue; width: 200px; height: 100px;
        position: absolute; left: 20px; top: 30px;
      }
      .fixed{
        background: yellow; width: 200px; height: 100px;
        position: fixed; right: -20px; top: 100px;
      }
    </style>
  </head>
  <body>
    <div class="relative"> relative
      <div class="absolute"> absolute</div>
      <div class="fixed"> fixed</div>
    </div>
  </body>
</html>
```



Gambar 3.7 Property position

Overflow

Property overflow mengatur sifat box ketika isinya melampaui ukuran box. Jika diisi **hidden**, maka konten yang melampaui akan disembunyikan dengan syarat tinggi atau lebar box ditentukan dengan property width atau height. Jika lebar atau tinggi tidak ditentukan, maka lebar atau tinggi akan mengikuti lebar atau tinggi konten. Jika diisi **scroll** maka akan muncul scrollbar pada box baik isinya melampaui box atau tidak. Jika diisi auto, maka akan muncul scroll bar hanya jika isinya melampaui saja. Nilai default-nya **visible**, yaitu isi yang melampaui akan tetap ditampilkan.

Property overflow memiliki turunan **overflow-x** untuk mengatur secara horizontal saja dan **overflow-y** untuk mengatur secara vertikal.

File 3_5_1_overflow.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Latihan CSS</title>
    <style type="text/css">
      .box1{
        width: 150px;
        height: 150px;
        overflow: scroll;
        background: red;
        float: left;
      }
      .box2{
        width: 150px;
        height: 150px;
```

```

        overflow: hidden;
        background: red;
    }

```

```
</head>
```

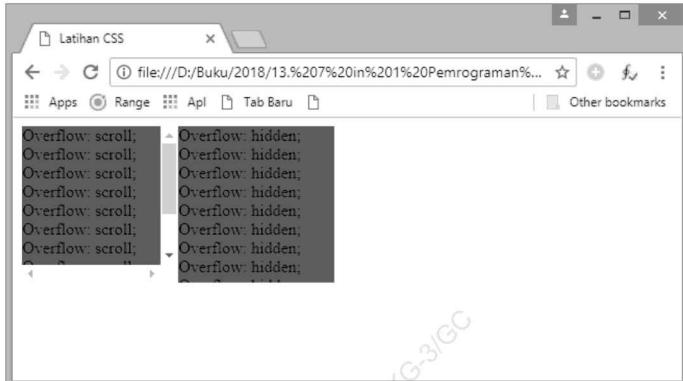
```
<body>
```

```
    <div class="box1">Overflow: scroll; ... </div>
```

```
    <div class="box2">Overflow: hidden ... </div>
```

```
</body>
```

```
</html>
```



Gambar 3.8 Property overflow

Float dan clear

Property **float** mengatur posisi box berada di kiri atau kanan teks di sekitarnya, dapat diisi **right** atau **left**. Jika diisi **left**, maka box akan ke kiri dan menyediakan ruang kosong di sebelah kanan untuk dapat diisi elemen lain. Begitu pula sebaliknya.

Penggunaan property ini sering membuat tampilan jadi berantakan. Untuk menormalkan agar element lain tidak terpengaruh, dapat menggunakan property **clear** yang dapat diberi nilai **none**, **left**, **right**, **both** (**right** & **left**) dan **inherit** (mengikuti nilai elemen induk).

File 3_5_1.float.html

```

<!DOCTYPE HTML>
<html>
    <head>
        <title>Latihan CSS</title>
        <style type="text/css">
            .kiri{
                width: 200px;
                float: left;
                margin-right: 20px;
            }
        </style>
    </head>
    <body>
        <div>Content 1</div>
        <div>Content 2</div>
        <div>Content 3</div>
    </body>
</html>

```

```

    }
.kanan{
    width: 200px;
    float: right;
    margin-left: 20px;
}
</style>
</head>
<body>
    <p>Lorem ipsum dolor
sit amet, ...</p>

    <p>Lorem ipsum
dolor sit amet, ...</p>
</body>
</html>

```



Gambar 3.9 Property float

Display dan Visibility

Tag HTML terdiri dari block element dan inline element. Ciri-ciri block element yaitu lebar area akan memenuhi lebar container, contohnya tag `<p>`, `<h1>` - `<h6>`, `<div>`, `<header>`, dan sebagainya. Block element dapat diberi property width, height dan padding. Sedangkan inline element lebarnya mengikuti konten, tidak memenuhi container, contohnya tag ``, `<a>`, ``, dan sebagainya. Sifat block dan inline tersebut dapat diatur ulang menggunakan property display. Nilai yang dapat diberikan sebagai berikut:

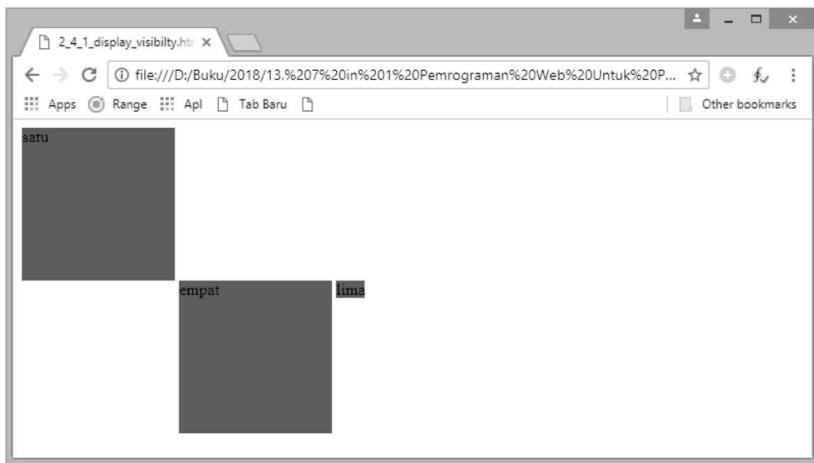
- **Inline**, elemen tidak dapat diberi property width dan height dan lebarnya mengikuti konten.

- **Block**, lebar element memenuhi container sebelum diberi property width dan height.
- **Inline-block**, lebar elemen mengikuti konten seperti inline, tetapi dapat diberi property width dan height.
- **None**, elemen akan disembunyikan.

Property display dapat digunakan untuk menyembunyikan sebuah elemen, tetapi elemen akan benar-benar dihapus dari layout dan akan mempengaruhi elemen lain. Alternatif lain dapat menggunakan property **visibility: hidden** untuk menyembunyikan elemen tanpa menghapus ruang sehingga tidak mempengaruhi elemen lain. Untuk menampilkan kembali dapat menggunakan **visibility: visible**.

```
File 3_5_1_display_visibility.html

<!DOCTYPE HTML>
<html>
  <head>
    <style type="text/css">
      div{
        width: 150px; height: 150px;
        background: red;
      }
      .satu{ display: block }
      .dua{ display: none }
      .tiga{
        display: inline-block;
        visibility: hidden;
      }
      .empat{ display: inline-block }
      .lima{ display: inline }
    </style>
  </head>
  <body>
    <div class="satu"> satu </div>
    <div class="dua"> dua </div>
    <div class="tiga"> tiga </div>
    <div class="empat"> empat </div>
    <div class="lima"> lima </div>
  </body>
</html>
```



Gambar 3.10 *Property display dan visibility*

3.5.2 Font

Property font mengatur segala sesuatu yang berhubungan dengan penggunaan font seperti jenis font, ukuran font, dan sebagainya. Beberapa property yang berhubungan dengan font yaitu:

- **Font-family**, mengatur jenis font yang digunakan, dapat diisi dengan nama grup font yang terdiri dari **serif**, **sans-serif** dan **monospace**, atau diisi nama font seperti **arial**, **calibri**, **verdana** dan sebagainya. Biasanya font yang masukkan lebih dari satu dengan dipisahkan tanda koma. Maksudnya, jika font pertama tidak ada pada komputer, akan digunakan font kedua dan seterusnya. Jika nama font lebih dari 1 suku kata digunakan tanda petik. Contohnya:

```
font-family: "Times New Roman", Times, serif;
```

- **Font-style**, mengatur teks miring atau tidak. Nilai yang dapat dimasukkan yaitu *normal*, *italic* dan *oblique*. *Italic* dan *oblique* sama-sama teks miring, hanya *oblique* kurang dukungan.
- **Font-size**, mengatur ukuran huruf. Nilai yang dapat diberikan berupa angka dengan satuan px atau em.
- **Font-weight**, mengatur teks tebal atau tidak. Nilai yang dapat diberikan *normal* dan *bold*.

- **Font-variant**, mengatur teks ditampilkan dalam small-caps atau tidak. Nilai yang dapat diberikan *normal* dan *small-caps*. Jika diisi *small-caps*, semua teks kecil (lowercase) akan dikonversi ke huruf kapital tapi dengan ukuran lebih kecil dibanding huruf kapital yang sesungguhnya.
- **Font-stretch**, mengatur font tampil lebih lebar (*expanded*) atau sempit (*condensed*). Nilai yang dapat diberikan yaitu *ultra-condensed*, *extra-condensed*, *condensed*, *semi-condensed*, *normal*, *semi-expanded*, *expanded*, *extra-expanded*, dan *ultra-expanded*.

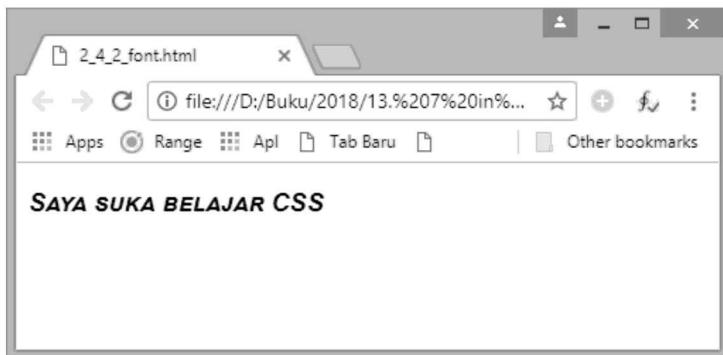
Beberapa property di atas dapat digabungkan dalam properti **font**, dengan format sebagai berikut:

```
font: [font-style] [font-variant] [font-weight] [font-size]  
[font-family]
```

Tidak semua property harus dicantumkan, kecuali font-size dan font-family yang wajib ada.

Sebagai contoh penggunaan property font, perhatikan skrip berikut:

```
File 3_5_2_font.html  
  
<!DOCTYPE HTML>  
<html>  
  <head>  
    <style type="text/css">  
      p{  
        font-family: arial, verdana, helvetica;  
        font-size: 18px;  
        font-weight: bold;  
        font-style: italic;  
        font-variant: small-caps;  
      }  
    </style>  
  </head>  
  <body>  
    <p> Saya suka belajar CSS</p>  
  </body>  
</html>
```



Gambar 3.11 *Property font dan turunannya*

3.5.3 Text

Ada banyak property yang dapat diberikan pada CSS berkaitan dengan tampilan teks. Property yang berhubungan dengan teks yaitu:

- **Color**, mengatur warna teks.
- **Direction**, mengatur arah teks *rtl* (*right to left*) atau *ltr* (*left to right*).
- **Text-align**, mengatur perataan teks secara horizontal. Dapat diisi *center*, *left*, *justify* atau *right*.
- **Text-decoration**, mengatur dekorasi teks. Dapat diisi *none*, *underline*, *overline*, *line-through*, *initial* atau *inherit*.
- **Text-indent**, mengatur alinea pada paragraf yang dapat diisi dengan nilai indentation dengan satuan **px**. Dapat juga diisi *initial* atau *inherit*.
- **Text-transform**, mengatur penggunaan huruf kapital pada teks. Dapat diisi *none*, *capitalize*, *uppercase*, *lowercase*, *initial* atau *inherit*.
- **Letter-spacing**, mengatur jarak antar huruf ditulis dengan angka dalam **px**.
- **Word-spacing**, mengatur jarak antar kata.
- **Line-height**, mengatur tinggi baris teks.
- **Vertical-align**, mengatur perataan secara vertical.

Contoh penggunaan property yang berhubungan dengan teks dapat dilihat pada skrip berikut:

```
File 3_5_3_text.html

<!DOCTYPE HTML>
<html>
  <head>
    <style type="text/css">
      p{
        font-family: arial, verdana, helvetica;
        font-size: 18px;
        font-weight: bold;
        text-transform: uppercase;
        text-decoration: underline;
        text-align: center;
      }
    </style>
  </head>
  <body>
    <p> Saya suka belajar CSS</p>
  </body>
</html>
```



Gambar 3.12 *Property berkatian dengan text*

3.5.4 Background

Property background digunakan untuk pengaturan background warna maupun background gambar pada suatu elemen. Beberapa property yang berkaitan dengan background yaitu:

- **Background-color**, mengatur warna latar belakang elemen.
- **Background-image**, mengatur gambar yang akan dijadikan latar belakang. Isinya berupa path dari sebuah gambar.

- **Background-attachment**, mengatur sifat background terhadap perlakuan scroll. Dapat diisi *scroll (default)*, *fixed*, *local*, *initial* dan *inherit*. Jika diisi *fixed*, background tidak akan ikut scroll bersama halaman website. Jika diisi *local*, background akan ikut scroll bersama konten elemen.
- **Background-position**, mengatur posisi background. Dapat diisi dengan 2 buah angka, angka pertama menunjukkan posisi dari kanan, sedangkan angka kedua menunjukkan posisi dari atas.
- **Background-repeat**, mengatur perulangan background. Dapat diisi *repeat*, *repeat-x*, *repeat-y*, *no-repeat*, *space*, *round*, *initial* dan *inherit*.
- **Background-origin**, mengatur penempatan background pada box-model. Dapat diisi *padding-box*, *border-box*, *content-box*, *initial* dan *inherit*.
- **Background-size**, mengatur ukuran background. Dapat diisi *auto*, *cover*, *contain*, *initial* dan *inherit*. Juga dapat diisi dengan angka atau persentase.

Nilai dari beberapa property di atas dapat digabungkan dalam property **background**, dengan format sebagai berikut:

```
background: [bg-color] [bg-image] [bg-position] [bg-repeat] [bg-origin] [bg-clip] [bg-attachment];
```

Property **background** dapat juga diisi dengan nilai **background-color** atau **background-image** saja. Contoh penggunaan property **background** dapat dilihat pada skrip berikut:

File 3_5_4_background.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <style type="text/css">
      div{
        width: 150px; height: 150px;
        background: red;
        float: left;
        margin: 10px;
        background-image: url('gambar/bg.jpg');
      }
      .dua{
        background-repeat: no-repeat;
        background-position: 30px 100px;
      }
      .tiga{
        background-repeat: repeat-x;
```

```

    }
    .empat{
        background-repeat: repeat-y;
    }

```

</style>

</head>

<body>

<div class="satu"></div>

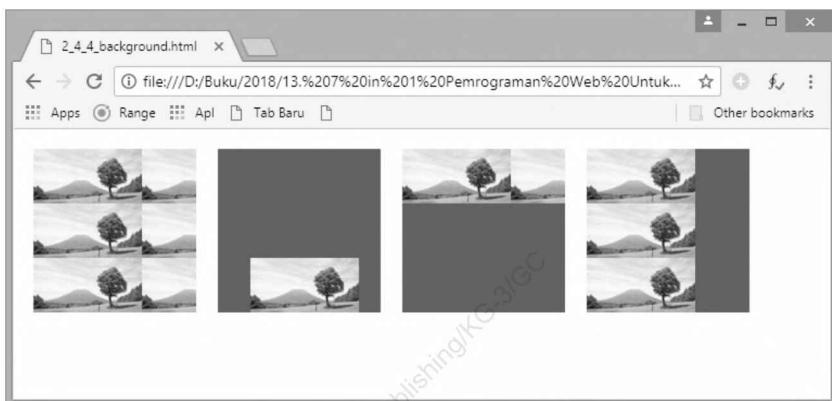
<div class="dua"></div>

<div class="tiga"></div>

<div class="empat"></div>

</body>

</html>



Gambar 3.13 Property background

3.5.5 List Style

Property `list-style` digunakan untuk mengatur tampilan list pada tag `` dan ``. Property yang berkaitan dengan `list-style` yaitu:

- **List-style-type**, mengatur simbol atau angka yang digunakan sebagai list. Banyak sekali nilai yang dapat diberikan, namun yang umum digunakan yaitu *disc*, *circle*, *square*, *decimal* dan *none* (tidak ada simbol).
- **List-style-position**, mengatur posisi simbol atau angka di luar area atau di dalam area list. Nilai yang dapat diberikan *outside* atau *inside*.
- **List-style-image**, mengatur gambar yang digunakan sebagai simbol list. Diisi dengan path gambar.

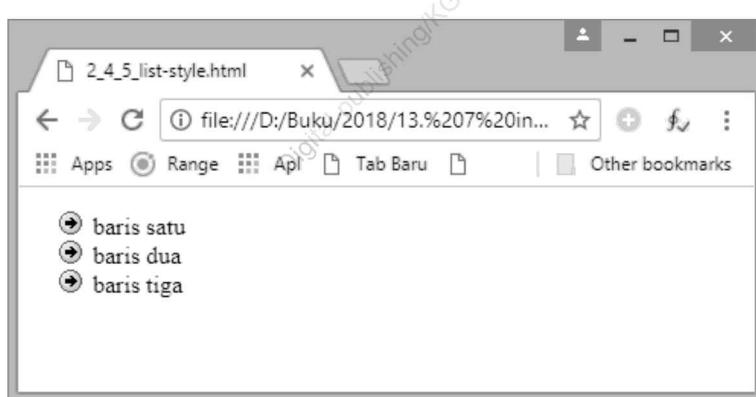
Nilai property di atas dapat digabungkan dalam property **list-style** dengan format sebagai berikut:

```
list-style: [list-style-type] [list-style-position] [list-style-image];
```

Sebagai contoh, perhatikan skrip berikut:

File 3_5_5_list-style.html

```
<!DOCTYPE HTML>
<html>
<head>
<style type="text/css">
    ul{ list-style: url('gambar/arrow.gif'); }
</style>
</head>
<body>
<ul>
    <li>baris satu</li>
    <li>baris dua</li>
    <li>baris tiga</li>
</ul>
</body>
</html>
```



Gambar 3.14 Property list-style

3.6 Fitur Khusus CSS3

Selain property yang telah dijelaskan pada subbab sebelumnya, CSS masih memiliki beberapa properti penting yang baru ada pada CSS3.

Pada subbab ini akan dibahas berbagai fitur unggulan dari CSS3 yang memiliki fungsi penting dalam desain web.

Contoh skrip CSS pada subbab ini tidak ditunjukkan secara lengkap, untuk melihat versi lengkapnya bisa dilihat pada file pendukung.

3.6.1 Rounded Corner

CSS3 mampu membuat sudut pada box menjadi melengkung (rounded) dengan menggunakan property **border-radius**. Bahkan dengan property ini dapat digunakan untuk membuat elemen berbentuk melingkar. Contohnya sebagai berikut:

File 3_6_1_rounded_corner.html

```
...
.box1{ border-radius: 20px; }
.box2{ border-radius: 50%; }
...
```



Gambar 3.15 Rounded corner

3.6.2 Border Image

Pada CSS3, border dapat menggunakan gambar dengan property **border-image**. Property ini dibagi menjadi beberapa property turunan sebagai berikut:

- **Border-image-source**, mengatur path gambar.
- **Border-image-slice**, mengatur bagaimana border akan dipotong.
- **Border-image-width**, mengatur lebar border.
- **Border-image-outset**, mengatur jarak border di luar box.
- **Border-image-repeat**, mengatur bagaimana border akan diulang, dapat diisi *stretch*, *repeat*, *round* dan *space*.

Property di atas dapat digabungkan dalam property **border-image** dengan format sebagai berikut:

```
border-image: [border-source] [border-slice] [border-width]
[border-outset] [border-repeat];
```

Pada contoh di bawah ini, akan digunakan gambar berikut:



Gambar 3.16 Gambar border

Dari gambar di atas, akan dihasilkan border sebagai berikut:

File 3_6_2_border_image.html

```
...
.box1{
  border: 10px solid transparent;
  border-image: url('gambar/border.png') 50 round;
}
.box2{
  border: 10px solid transparent;
  border-image: url('gambar/border.png') 20% round;
}
.box3{
  border: 10px solid transparent;
  border-image: url('gambar/border.png') 30 stretch;
```



Gambar 3.17 Hasil property border-image

3.6.3 Shadow

CSS3 memiliki kemampuan membuat bayangan pada teks maupun box. Untuk membuat bayangan pada text menggunakan **text-shadow**, sedangkan pada box menggunakan **box-shadow**. Kedua property ini memiliki beberapa nilai dengan urutan sebagai berikut:

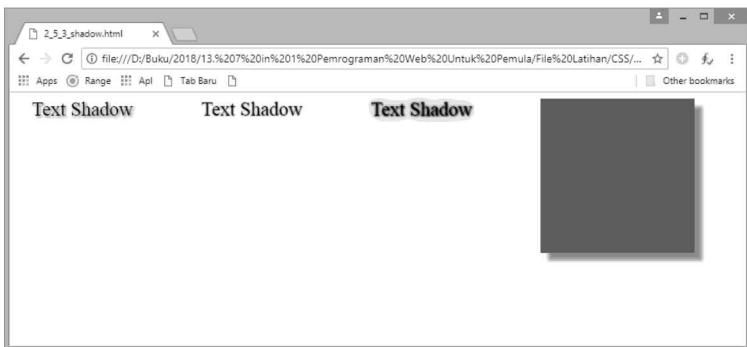
- **H-offset**, yaitu jarak bayangan secara horizontal.
- **V-offset**, yaitu jarak bayangan secara vertikal.
- **Blur**, yaitu tingkat blur bayangan, semakin tinggi nilainya semakin blur.
- **Spread**, yaitu jarak penyebaran bayangan, semakin tinggi nilainya semakin menyebar jauh.
- **Color**, yaitu warna bayangan.
- **Inset**, untuk membuat bayangan ke dalam jika nilai ini dicantumkan.

Property pertama dan kedua di atas wajib dicantumkan, sedangkan yang selanjutnya bersifat opsional. Text atau box dapat memiliki bayangan lebih dari satu, dengan dipisahkan menggunakan koma.

Contohnya sebagai berikut:

File 3_6_3_shadow.html

```
.text1{ text-shadow: 2px 2px 5px red; }  
.text2{ text-shadow: 0 0 3px #FF0000; }  
.text3{ text-shadow: 1px 1px 2px black, 0 0 25px blue, 0 0 5px  
darkblue; }  
.box{ background: red; box-shadow: 10px 10px 5px grey; }
```



Gambar 3.18 Text shadow dan box shadow

3.6.4 Gradient

Salah satu fitur CSS3 yang menarik adalah kemampuan membuat gradasi warna. Fitur ini sering diterapkan pada desain tombol. Ada dua jenis gradient yaitu **linear-gradient** dan **radial-gradient**.

Linear-gradient digunakan untuk membuat gradasi warna lurus. Format penggunaanya sebagai berikut:

```
background: linear-gradient(direction, color-stop1, color-stop2);
```

Direction dapat diisi dengan arah dalam bahasa Inggris misalnya: *to right*, *to left*, *to bottom*, *to bottom right*, *to top left*, dan sebagainya. Dapat juga diisi dengan sudut misalnya: *90deg*, *240deg*, *-90deg*, dan sebagainya.

Color-stop diisi dengan berbagai jenis warna termasuk RGBA untuk membuat gradasi warna ke transparan. Color-stop tidak terbatas 2 warna saja, dapat diisi sampai berapapun dengan dipisahkan tanda koma.

Setiap color-stop dapat diatur posisinya dengan menggunakan persen yang dipisahkan spasi dengan warna.

Radial-gradient digunakan untuk membuat gradasi warna berbentuk lingkaran atau elips. Format penggunaanya sebagai berikut:

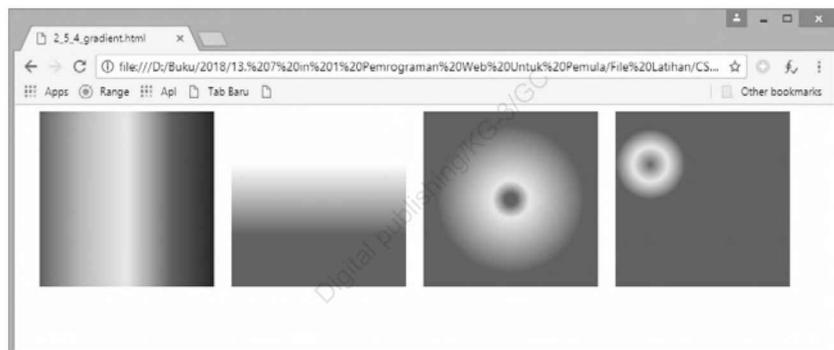
```
background: radial-gradient(shape size at position, start-color, ... , last-color);
```

Shape dapat diisi *elips* (default) atau *circle*. Size dapat diisi *closest-side*, *farthest-side*, *closest-corner* dan *farthest-corner*. Position diisi dengan posisi x dan posisi y dalam satuan persen. Start-color adalah warna paling tengah dan last-color adalah warna paling luar. Aturan start-color dan stop-color sama dengan color-stop pada linear-gradient.

Contoh penggunaanya sebagai berikut:

File 3_6_4_gradient.html

```
.box1{ background: linear-gradient(to right,  
red,orange,yellow,green,blue) }  
.box2{ background: linear-gradient(to bottom, rgba(255,0,0,0)  
30%, rgba(255,0,0,1) 70%) }  
.box3{ background: radial-gradient(red 5%, yellow 15%, green  
60%) }  
.box4{ background: radial-gradient(circle closest-side at 20%  
30%, red, yellow, green); }
```



Gambar 3.19 Linear gradient dan radiel gradient

3.6.5 Text Effect

Pada CSS3 ada 3 property tambahan untuk pengaturan teks yang cukup bermanfaat dalam desain web. Ketiga property tersebut yaitu:

- **text-overflow**, mengatur bagian akhir teks ketika melewati batas konten. Nilai yang dapat diberikan *clip*, *ellipsis* dan *string*.
- **Word-break**, mengatur pemenggalan kata pada setiap baris teks. Nilai yang dapat diberikan *normal*, *break-all* dan *keep-all*.

- **Word-wrap**, mengatur dibolehkannya pemenggalan kata pada kata yang panjang atau tidak. Dapat diisi *normal* atau *break-word*.

Contoh penggunaanya sebagai berikut:

File 3_6_5_text_effect.html

```
.box1{  
    overflow: hidden;  
    white-space: nowrap;  
    text-overflow: ellipsis;  
}  
.box2{ word-break: break-all; }  
.box3{ word-wrap: break-word; }
```



Gambar 3.20 Text effect

3.6.6 Web Font

CSS3 mengijinkan web designer menggunakan jenis font yang tidak terinstal pada komputer client dengan cara menyimpan file font pada server. Format font yang didukung meliputi: *.ttf, *.otf, *.woff, svg, dan *.eot. Setiap format dapat disertakan semua, sehingga ketika salah satu format tidak didukung oleh browser akan dialihkan ke format yang didukung. Cara menggunakannya yaitu dengan selector **@web-font**. Property yang dapat digunakan pada selector tersebut yaitu:

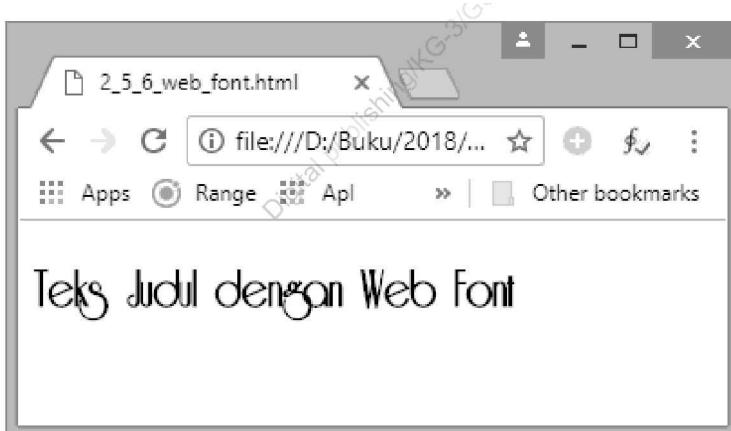
- **Font-family**, menentukan nama font ketika digunakan oleh suatu elemen.

- **Src**, menentukan letak file font.
- **Font-stretch**, menentukan gaya stretch font.
- **Font-weight**, menentukan ketebalan font.
- **Font-style**, membuat font miring.
- **Unicode-range**, menentukan Unicode karakter font yang didukung.

Property yang wajib disertakan hanya font-family dan src, sedangkan yang lain bersifat opsional. Nilai yang dapat diberikan pada font-stretch, font-widht dan font-style sama dengan property font yang dibahas sebelumnya. Contoh penggunaanya, perhatikan skrip berikut:

File 3_6_6_web_font.html

```
@font-face{
  font-family: riesling;
  src: url('font/riesling.ttf');
}
h1{ font-family: riesling; }
```



Gambar 3.21 Penerapan web font pada H1

3.6.7 2D Transform

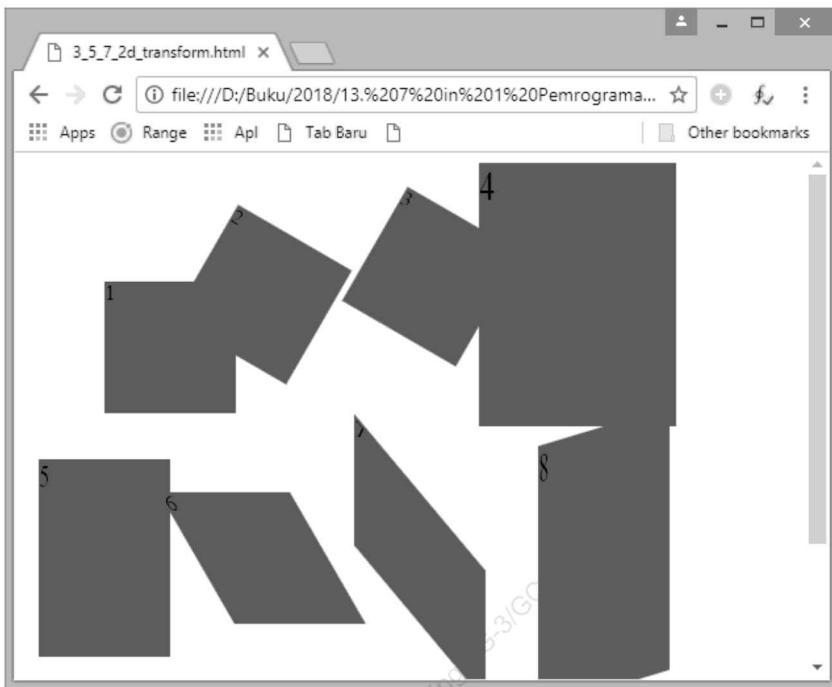
2D transform yaitu melakukan transformasi terhadap elemen menggunakan property **transform**. Ada beberapa metode transformasi yang dapat dilakukan yaitu:

- **Translate(x, y)**, yaitu menggeser elemen dengan jarak x secara horizontal dan jarak y secara vertikal. Untuk 1 arah saja dapat menggunakan translateX() atau translateY().
- **Rotate(deg)**, yaitu memutar elemen dengan sudut tertentu.
- **Scale(width, height)**, yaitu mengubah ukuran width dan height objek. Untuk 1 arah saja dapat menggunakan scaleX() atau scaleY().
- **Skew(x-deg, y-deg)**, yaitu memiringkan elemen dengan sudut tertentu ke arah horizontal maupun vertikal. Untuk 1 arah saja dapat menggunakan skewX() atau skewY().
- **Matrix(scaleX, skewX, skewY, scaleY, translateX, translate)**, yaitu melakukan transformasi dengan berbagai metode transformasi sekaligus.

Untuk menggeser posisi titik tumpuan transformasi dapat menggunakan property **transform-origin**. Sebagai contoh perhatikan skrip berikut:

File 2_6_7_2d_transform.html

```
.box1{ transform: translate(50px, 40px) }  
.box2{ transform: rotate(30deg); }  
.box3{ transform: rotate(30deg); transform-origin: 80% 60%; }  
.box4{ transform: scale(1.5, 2); }  
.box5{ transform: scaleY(1.5); }  
.box6{ transform: skewX(30deg); }  
.box7{ transform: skewY(50deg); }  
.box8{ transform: matrix(1, -0.3, 0, 2, 20, 0); }
```



Gambar 3.22 2d transform

3.6.8 3D Transform

Berbeda dengan 2D transform yang berlaku untuk berbagai metode transformasi, 3D transform hanya berlaku untuk rotasi yang dibagi menjadi 3, yaitu:

- **rotateX(deg)**, yaitu rotasi elemen dengan tumpuan pada sumbu X.
- **rotateY(deg)**, yaitu rotasi elemen dengan tumpuan pada sumbu Y.
- **rotateZ(deg)**, yaitu rotasi elemen dengan tumpuan pada sumbu Z.

Ada beberapa property yang dapat dipakai bersamaan dengan penggunaan 3D transform, yaitu:

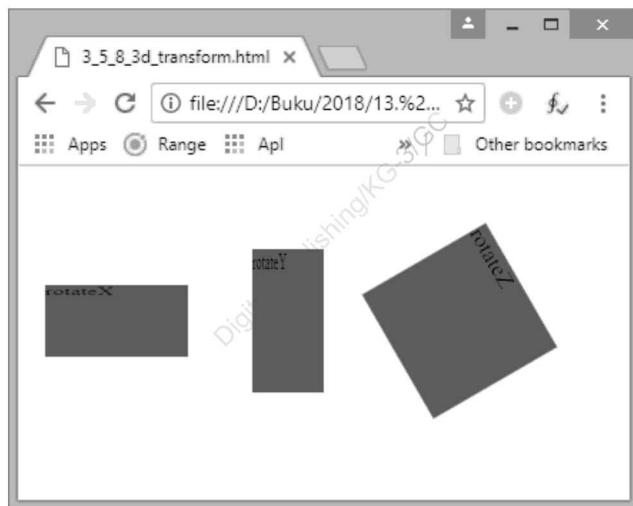
- **Perspective**, untuk membuat efek perspective.
- **Perspective-origin**, untuk menentukan base perspective berdasarkan sumbu X dan Y.

- Backface-visibility, untuk menentukan apakah bagian belakang elemen terlihat atau tidak, dapat diisi visible atau hidden.
- Transform-style, untuk menentukan bagaimana elemen di dalamnya diberlakukan, dapat diisi flat atau preserve-3d.
- Transform-origin, untuk menentukan posisi base transform berdasarkan sumbu X, Y dan Z.

Sebagai contoh, perhatikan skrip berikut:

File 2_6_8_3d_transform.html

```
.box1{ transform: rotateX(60deg); }
.box2{ transform: rotateY(60deg); }
.box3{ transform: rotateZ(60deg); }
```



Gambar 3.23 3D transform

3.6.9 Transition

Transition digunakan untuk membuat animasi perubahan dari suatu kondisi ke kondisi yang lain, misalnya animasi perubahan ketika tombol ditunjuk pointer mouse. Property yang dianimasikan meliputi sebagian besar property yang bernilai angka dan warna. Beberapa property yang digunakan yaitu:

- **Transition-delay**, menunjukkan jarak waktu sebelum terjadi transisi dalam satuan s (second).
- **Transition-property**, menunjukkan property apa yang akan dianimasikan. Jika tidak ditentukan, maka semua property akan dianimasikan.
- **Transition-duration**, menunjukkan waktu terjadinya transisi dalam satuan s.
- **Transition-timing-function**, menunjukkan perubahan kecepatan animasi, yang dapat diisi *ease*, *linear*, *ease-in*, *ease-out*, *ease-in-out*, *step-start*, *step-end*, *steps*, dan *cubic-bazier*.

Nilai property di atas dapat digabungkan dalam property **transition** dengan format sebagai berikut:

```
Transition: [property] [duration] [timing-function] [delay];
```

Sebagai contoh, perhatikan skrip berikut:

```
File 3_6_9_transition.html  
div{  
background: red;  
width: 100px;  
height: 100px;  
}  
  
div:hover{  
transition: 2s ease;  
background: green;  
width: 200px;  
height: 150px;  
}
```



Coba dijalankan pada browser, lalu tunjuk kotak dengan mouse, maka akan ada animasi dari background merah ke hijau serta perubahan ukuran kotak.

3.6.10 Animation

Sedikit berbeda dengan transition yang terjadinya animasi hanya ketika ada perubahan kondisi. Animation digunakan untuk membuat animasi saat halaman di-load. Animasi dibuat dengan selector **@keyframes** dan digunakan dengan property **animation**. Beberapa property yang berhubungan dengan animasi yaitu:

- **Animation-delay**, menunjukkan lama waktu sebelum terjadinya animasi.
- **Animation-duration**, menunjukkan lama waktu terjadinya animasi.
- **Animation-direction**, menunjukkan arah terjadinya animasi, dapat diisi *normal*, *reverse*, *alternate*, dan *alternate-reverse*.
- **Animation-fill-mode**, menunjukkan bagaimana style elemen ketika animasi tidak berjalan, baik sebelum animasi maupun setelahnya, dapat diisi *none*, *forwards*, *backwards* dan *both*.
- **Animation-iteration-count**, menunjukkan berapa kali animasi akan dijalankan.
- **Animation-name**, menunjukkan nama animasi yang akan dijalankan.
- **Animation-play-state**, menunjukkan animasi berjalan atau pause, dapat diisi *running* atau *paused*.
- **Animation-timing-function**, menunjukkan perubahan kecepatan animasi, isinya sama seperti pada transition.

Beberapa property di atas dapat digabungkan dalam property **animation** dengan format sebagai berikut:

```
animation: [name] [duration] [timing-function] [delay]
[iteration-count] [direction] [fill-mode] [play-state];
```

Sebagai contoh, perhatikan skrip berikut:

```
File 2_6_10_animation.html

.box1{
  position: relative;
  animation: gerak 2s ease;
}

@keyframes gerak{
  from{left: 20px; }
  30%{left: 800px; }
  to{left: 800px; }
}
```

From dapat diganti dengan 0% dan to dapat diganti dengan 100%. Untuk menambahkan keyframe tambahkan baris baru diawali dengan nilai persentase animasi diikuti perubahan property, misal: 60%{left: 400px}. Jalankan pada browser untuk melihat hasilnya.

3.6.11 Flexbox Layout

Flexbox Layout digunakan untuk pengaturan struktur layout dengan lebih mudah tanpa menggunakan float maupun position. Pada flexbox layout terdiri dari parent element sebagai pembungkus dan child element sebagai elemen yang akan diatur layoutnya.

Parent element harus diset **display: flex**. Property yang digunakan pada parent element yaitu:

- **Flex-direction**, mengatur bagaimana child element akan disusun, dapat diisi row, row-reverse, column dan column-reverse.
- **Flex-wrap**, mengatur bagaimana child element akan disusun jika lebar total melebihi lebar parent element, dapat diisi nowrap, wrap dan wrap-reverse.
- **Flex-flow**, merupakan shorthand untuk mengatur flex-direction dan flex-wrap sekaligus.
- **Justify-content**, mengatur align item secara horizontal, dapat diisi center, flex-start, flex-end, space-around dan space-between.
- **Align-items**, mengatur align item secara vertikal, dapat diisi center, flex-start, flex-end, stretch, baseline dan baseline.
- **Align-content**, mengatur align konten, dapat diisi space-between, space-around, stretch, center, flex-start dan flex-end.

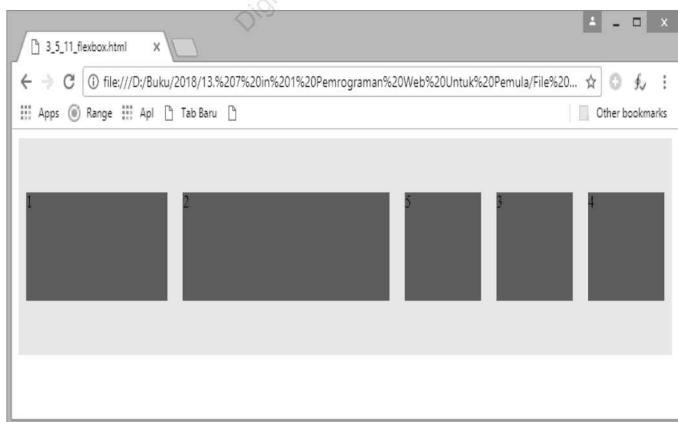
Child element memiliki property tersendiri yang dapat diterapkan, yaitu:

- **Order**, mengatur urutan susunan item.
- **Flex-grow**, menambah ukuran item berdasarkan space kosong yang akan diambil.
- **Flex-shrink**, mengurangi ukuran item jika space yang ada tidak muat.
- **Flex-basis**, mengatur lebar item dengan satuan px.
- **Align-self**, mengubah perataan suatu item menggantikan perataan yang ditentukan pada parent element.
- **Flex**, shorthand untuk mengatur flex-grow, flex-shrink dan flex-basis sekaligus.

Sebagai contoh, perhatikan skrip berikut:

```
File 2_6_11_flexbox.html

<!DOCTYPE HTML>
<html>
  <head>
    <style type="text/css">
      .row{
        display: flex;
        background: yellow;
        height: 200px;
        align-items: center;
      }
      .col{
        background: red;
        width: 100px;
        height: 100px;
        margin: 10px;
        flex-wrap: wrap;
      }
    </style>
  </head>
  <body>
    <div class="row">
      <div class="col" style="flex-grow: 2">1</div>
      <div class="col" style="flex-grow: 4">2</div>
      <div class="col" style="order: 4">3</div>
      <div class="col" style="order: 5">4</div>
      <div class="col" style="order: 3">5</div>
    </div>
  </body>
</html>
```



Gambar 3.24 Flexbox layout

3.6.12 Multiple Column

Pada CSS3, sebuah paragraf dapat dijadikan 2 kolom bahkan lebih. Property yang digunakan untuk membuat kolom yaitu sebagai berikut:

- **Column-width**, menentukan jumlah kolom yang akan dibuat.
- **Column-fill**, menentukan bagaimana teks mengisi kolom apakah diseimbangkan atau tidak, dapat diisi *balance* atau *auto*.
- **Column-gap**, menentukan jarak antar kolom.
- **Column-rule-color**, menentukan warna garis pemisah kolom.
- **Column-rule-style**, menentukan tipe garis pemisah kolom, diisinya sama dengan border-style.
- **Column-rule-width**, menentukan tebal garis pemisah kolom.
- **Column-rule**, shorthand untuk menentukan nilai column-rule-width, column-rule-style dan column-rule-color.
- **Column-span**, menentukan jumlah kolom yang dilalui dari sebuah elemen yang melintasi beberapa kolom. Jika melintasi semua kolom diisi *all*.
- **Column-width**, menentukan lebar tiap kolom.

Properti-properti di atas dapat digabungkan menggunakan property **columns** dengan format sebagai berikut:

```
columns: [column-width] [column-count]
```

Contoh penggunaanya sebagai berikut:

```
File 3_6_12_multiple_column.html

<!DOCTYPE HTML>
<html>
  <head>
    <style type="text/css">
      div{
        column-count: 3;
        column-rule: 1px dotted #ccc;
        column-gap: 30px;
      }
      h2{
        column-span: all;
      }
    </style>
  </head>
  <body>
```

```

<div>
    <h2>Lorem ipsum</h2>
    dolor sit amet ...
</div>
</body>
</html>

```



Gambar 3.25 Multiple column

3.6.13 Media Query

Media query digunakan untuk memberikan aturan yang berbeda pada media atau ukuran layar yang berbeda. Format penulisan media query sebagai berikut;

```

@media not|only mediatype and (expressions) {
    CSS-Code;
}

```

Kata **not** digunakan untuk pengecualian pada media tertentu, sedangkan **only** digunakan untuk pengkhususan pada media tertentu. Media type dapat diisi *all*, *print* (untuk printer), *screen* (untuk layar) dan *speech* (untuk screen reader). Adapun expressions umumnya diisi batasan screen menggunakan **min-width** dan **max-width**. Format di atas tidak harus semuanya dicantumkan. Selain cara di atas juga dapat ditentukan pada HTML dengan skrip sebagai berikut:

```

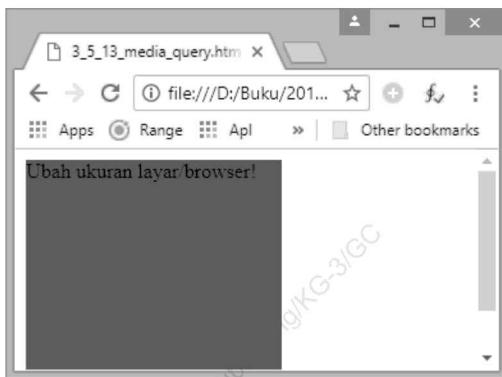
<link rel="stylesheet" media="mediatype and|not|only
(expressions)" href="print.css">

```

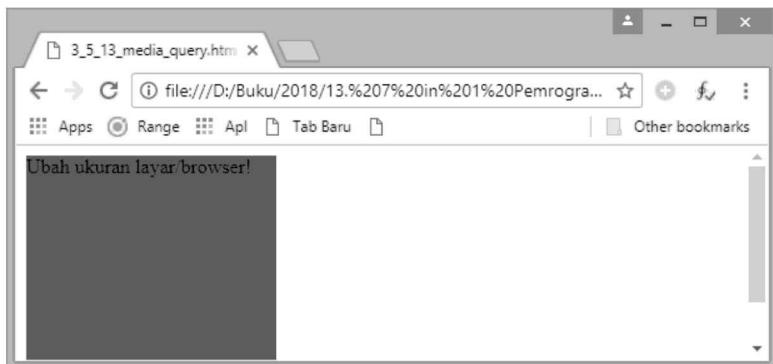
Contoh penggunaanya sebagai berikut:

File 3_6_13_media_query.html

```
@media screen and (max-width: 480px){  
    div{ background: red; }  
}  
@media screen and (min-width: 480px) and (max-width: 960px){  
    div{ background: green; }  
}  
@media screen and (min-width: 960px){  
    div{ background: blue; }  
}
```



Gambar 3.26 Background merah jika ukuran browser kurang dari 480px



Gambar 3.27 Background hijau jika ukuran browser lebih dari 480px dan kurang dari 960px

3.7 Praktik Membuat Template

Menguasai teori HTML dan CSS tidak ada jaminan telah mampu membuat desain web, sebelum praktik langsung. Untuk itu, pada subbab ini akan dibahas contoh desain template aplikasi berbasis web menggunakan HTML dan CSS. Template yang akan dibahas, nantinya akan digunakan pada praktik-praktik di bab selanjutnya.

Sebagai langkah awal buat dulu folder **template** sebagai tempat menyimpan file template, dan buat di dalamnya folder **css** dan **images**. Folder css untuk menyimpan file style.css dan folder images untuk menyimpan file-file gambar.

3.7.1 Desain Halaman Login

Untuk membuat halaman login, buat file **login.html** dengan isi skrip sebagai berikut:

```
File login.html
<!DOCTYPE HTML>

<html>
    <head>
        <title>Login Aplikasi</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width,
initial-scale=1">
        <link rel="stylesheet" href="css/style.css">
    </head>
    <body>
        <div class="container">
            <section class="login-box">

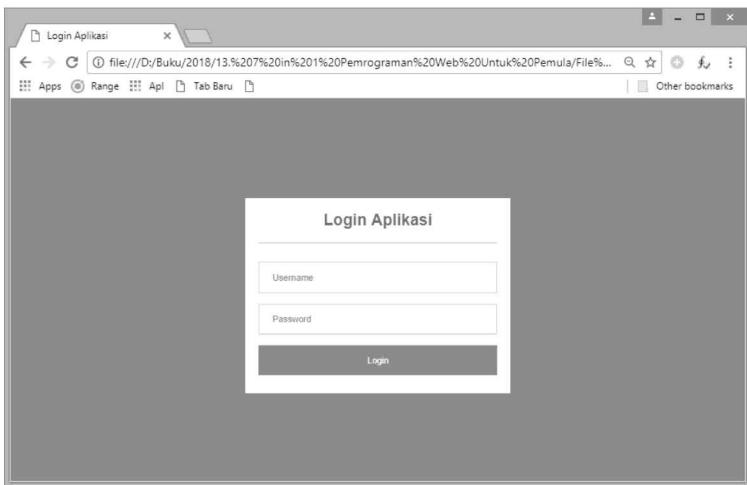
<h2>Login Aplikasi</h2>
<form method="post" action="ceklogin.php">
    <input type="text" placeholder="Username" id="username">
    <input type="password" placeholder="Password"
id="password">
    <input type="submit" value="Login">
</form>
</div>
        </div>
    </body>
</html>
```

Agar layout responsive, pada skrip di atas ditambah tag **<meta name="viewport">** pada bagian head. Untuk mengatur desainya, buat file **style.css** pada folder css dengan skrip sebagai berikut:

File style.css

```
*{  
    box-sizing: border-box;  
}  
  
/* Mengatur font yang dipakai dan background halaman */  
body {  
    font-family: Arial;  
    margin: 0;  
    background: #038ade;  
}  
  
/* Membuat container menjadi layout flexbox dan kotak login di  
tengah */  
.container{  
    display: flex;  
    flex-wrap: wrap;  
    justify-content: center;  
}  
  
/* Mengatur kotak form login lebar 400px dan background putih*/  
.login-box{  
    margin-top: 150px;  
    flex-basis: 400px;  
    background: #fff;  
    padding: 20px;  
}  
  
/* Mengatur judul form dengan border bawah dan posisi di tengah  
*/  
.login-box h2{  
    border-bottom: 1px solid #ccc;  
    color: #555;  
    text-align: center;  
    padding-bottom: 20px;  
    margin: 0 0 20px;  
}  
  
/* Mengatur agar tag input dan select memenuhi container dan  
tampilan lebih bagus */  
input, select{  
    width: 100%;  
    padding: 15px 20px;  
    margin: 8px 0;  
    border: 1px solid #ccc;  
}  
  
/* Mengatur tampilan tombol submit */  
input[type=submit] {  
    background-color: #038ade;  
    border: none;  
    color: #fff;  
    text-decoration: none;  
    cursor: pointer;  
}
```

Jalankan pada browser maka hasilnya seperti gambar berikut:



Gambar 3.28 Desain halaman login

3.7.2 Desain Layout dan Menu Navigasi

Selanjutnya membuat halaman dashboard untuk praktik membuat layout dan menu navigasi. Buat file **dashboard.html** dengan skrip sebagai berikut:

File **dashboard.html**

```
<!DOCTYPE HTML>

<html>
  <head>
    <title>Dashboard</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
    <header>
      Aplikasi Manajemen Pegawai
    </header>
    <div class="container">
      <aside>
```



```
<ul class="menu">
  <li> <a href="dashboard.html" class="aktif">Dashboard</a>
</li>
  <li> <a href="tabel.html">Data Pegawai</a> </li>
  <li> <a href="tabel.html">Data Jabatan</a> </li>
```

```

<li> <a href="login.html">Keluar</a> </li>
</ul>

</aside>
<section class="main">

<h1>Selamat Datang di Aplkasi Manajemen Pegawai</h1>
<h3>Anda login sebagai Administrator</h3>

</section>
</div>
<footer>
    Copyright © Rohi Abdulloh
</footer>
</body>
</html>

```

Pada skrip di atas, menu navigasi dibuat dengan tag ``. Selanjutnya atur desainya dengan menambahkan skrip CSS pada file style.css sebagai berikut:

File style.css

```

...
/* Mengatur desain header */
header{
    background: #038ade;
    color: #fff;
    padding: 15px 20px;
    font-size: 20px;
}

/* Mengatur lebar menu 20% dengan background hitam */
aside{
    flex: 20%;
    background: #222;
}

/* Mengatur lebah konten utama 80% dengan background putih */
section.main{
    flex: 80%;
    background: #fff;
    min-height: 530px;
    padding: 15px 20px;
}

/* Mengatur footer dengan background abu-abu dan teks di tengah */
footer{
    background: #ccc;
    color: #555;
    padding: 20px;
    text-align: center;
}

/* Mengatur layout pada layar kecil (responsive) */
@media (max-width: 700px) {
    .container {
        flex-direction: column;
}

```

```

        }

/* Navigasi */
/* Membuang padding tag <ul> */
.menu{
    padding: 0;
}

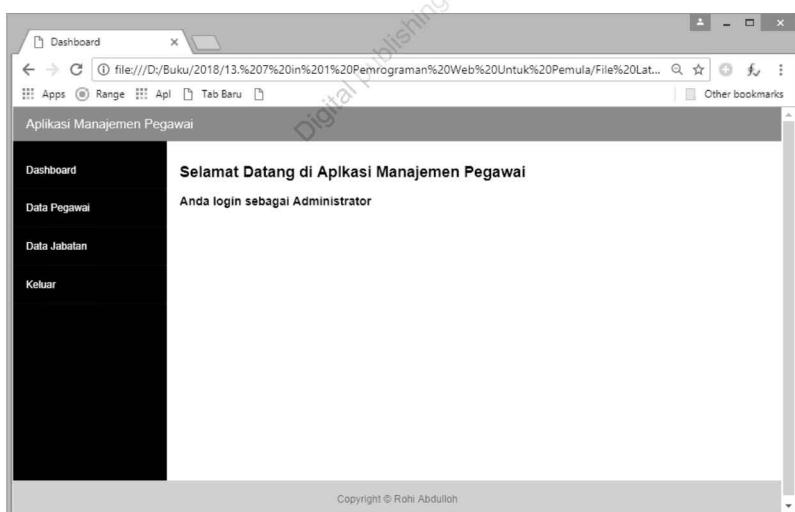
/* Menambah border bawah tag <li> */
.menu li{
    border-bottom: 1px dotted #444;
}

/* Mengatur desain menu yang dibuat dengan tag a */
.menu a{
    display: block;
    padding: 20px 20px;
    list-style: none;
    color: #fff;
    text-decoration: none;
}

/* Mengganti background menu ketika ditunjuk mouse */
.menu a:hover{
    background: #000;
}

```

Jika file dashboard.html dijalankan, hasilnya seperti gambar berikut:



Gambar 3.29 Desain halaman dashboard

3.7.3 Desain Tabel

Untuk mengatur desain pada halaman yang berisi tabel, buat file **tabel.html** dengan skrip sama seperti dashboard.html. Kemudian ubah isi tag <section> menjadi seperti berikut:

```
File tabel.html
...
<section class="main">
    <h2 class="judul">Data Pegawai</h2>
    <a class="tombol" href="form.html">Tambah</a>

    <table class="table">
        <thead>
            <tr>
                <th>No</th>
                <th>Foto</th>
                <th>Nama</th>
                <th>Jenis Kelamin</th>
                <th>Tanggal Lahir</th>
                <th>Jabatan</th>
                <th>Keterangan</th>
                <th>Aksi</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>1</td>
                <td></td>
                <td>Daffa' Shidqi</td>
                <td>L</td>
                <td>9 Juli 2015</td>
                <td>Direktur</td>
                <td>--</td>
                <td>
                    <a class="tombol edit" href="#"> Edit </a>
                    <a class="tombol hapus" href="#"> Hapus </a>
                </td>
            </tr>
            ...
        </tbody>
    </table>
</section>
...
```

Atur desainnya dengan menambah skrip CSS pada file style.css sebagai berikut:

```
File style.css
...
```

```

/* Mengatur desain judul halaman dengan border di bawah */
.judul{
    color: #555;
    border-bottom: 1px solid #ccc;
    padding-bottom: 15px;
}

/* Mengatur agar lebar tabel penuh dan antar border tidak
memiliki jarak */
.table{
    width: 100%;
    margin-top: 20px;
    border-collapse: collapse;
}

/* Menambahkan border dan padding pada sel tabel */
.table th, .table td{
    border: 1px solid #ccc;
    padding: 10px;
}

/* Mengatur background bagian judul tabel */
.table th{
    background: #eee;
}

/* Mengatur desain tombol yang dibuat dengan tag <a> */
.tombol{
    display: inline-block;
    max-width: 150px;
    padding: 15px 30px;
    color: #fff;
    text-decoration: none;
    background: #038ade;
    cursor: pointer;
}

/* Desain tombol edit dan hapus agar lebih kecil */
.tombol.edit, .tombol.hapus{
    padding: 10px 15px;
    font-size: 12px;
}
.tombol.hapus{
    background: #fc4c4c;
}

```

Jalankan file tabel.html, maka hasilnya seperti gambar berikut:

No	Foto	Nama	Jenis Kelamin	Tanggal Lahir	Jabatan	Keterangan	Aksi
1		Daffa' Shidqi	L	9 Juli 2015	Direktur	-	<button>Edit</button> <button>Hapus</button>
2		Annisa Nabil	P	31 Juli 2011	Manajer	-	<button>Edit</button> <button>Hapus</button>
3		Kaesang Ayu	P	9 Juli 2016	sekretaris	-	<button>Edit</button> <button>Hapus</button>

Gambar 3.30 Desain tabel

3.7.4 Desain Form

Untuk mengatur desain pada halaman yang berisi form, buat file **form.html** dengan skrip sama seperti **dashboard.html**. Kemudian ubah isi tag **<section>** menjadi seperti berikut:

File **form.html**

```

<h2 class="judul">Tambah Pegawai</h2>
<form method="post" action="proses.php">
  <div class="form-group">
    <label for="foto">Foto</label>
    <div class="input"><input type="file" id="foto"></div>
  </div>

  <div class="form-group">
    <label for="nama">Nama</label>
    <div class="input"><input type="text" id="nama"></div>
  </div>

  <div class="form-group">
    <label for="jk">Jenis Kelamin</label>
    <input type="radio" id="jk" name="jk" value="L"> Laki-laki
    <input type="radio" id="jk" name="jk" value="P"> Perempuan
  </div>

  <div class="form-group">
    <label for="tanggal">Tanggal</label>
  
```

```

<div class="input"><input type="date" id="tanggal"></div>
</div>

<div class="form-group">
    <label for="jabatan">Jabatan</label>
    <div class="input">
        <select id="jabatan">
            <option value="direktur"> Direktur </option>
            <option value="manajer"> Manajer </option>
            <option value="marketing"> Marketing </option>
            <option value="sekretaris"> Sekretaris </option>
        </select>
    </div>
</div>

<div class="form-group">
    <input type="submit" value="Simpan" class="tombol simpan">
    <input type="reset" value="Batal" class="tombol reset">
</div>
</form>

```

Pada skrip di atas, setiap label dan input diletakkan di dalam tag <div> dengan class .form-group untuk memudahkan pengaturan desain. Langsung saja atur desain form dengan menambah skrip CSS pada file style.css sebagai berikut:

File style.css

```

...
/* Mengatur form-group agar tidak terpengaruhi float */
.form-group{
    clear: both;
}

/* Mengatur lebar label 30% dengan teks bold */
.form-group label{
    display: block;
    padding: 15px 0;
    float: left;
    width: 30%;
    font-weight: bold;
}

/* Mengatur lebar .input 50% dan sejajar dengan label */
.form-group .input{
    display: block;
    width: 50%;
    float: left;
}

/* Mengatur input radio, tombol submit dan tombol reset */
input[type=radio]{
    width: 50px
}
.tombol.simpan{
    background: #038ade;
}
.tombol.reset{

```

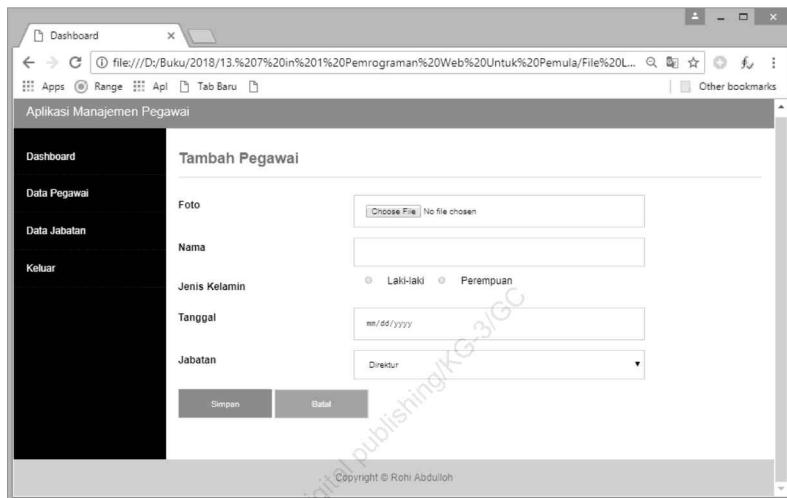
```

background: #fb8307;
}

/* Membuat tampilan form menumpuk pada layar kecil */
@media (max-width: 700px) {
    .form-group label, .form-group .input {
        width: 100%;
    }
}

```

Jalankan file form.html, maka hasilnya seperti gambar berikut:



Gambar 3.31 Desain form

3.8 Bonus File Contoh

Untuk menunjang agar lebih menguasai CSS, penulis sertakan bonus file contoh CSS yang terletak pada folder CSS/BONUS pada file pendukung buku. Folder tersebut berisi berbagai contoh desain dengan CSS meliputi responsive layout, responsive galery, berbagai style image, berbagai style form, berbagai style tombol, tooltip, dan sebagainya.

File pendukung buku yang terdiri dari file latihan dan bonus contoh skrip dapat diminta melalui email penulis rohi.abdulloh@gmail.com dengan menyertakan bukti pembelian buku.

4 MySQL

4.1 Pengenalan Database

4.1.1 Istilah-Istilah Dalam Database

Sebelum lebih jauh membahas database, ada baiknya kenali dulu berbagai istilah yang sering digunakan pada pembahasan database, yaitu sebagai berikut:

- **DBMS**, yaitu sistem perangkat lunak yang menyediakan layanan bagi user untuk membuat, mengontrol dan mengakses database.
- **Database** atau **basis data**, adalah kumpulan informasi yang disimpan dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi.
- **Table**, merupakan kumpulan data yang diorganisasikan dalam baris (record) dan kolom (field). Dalam satu database biasanya terdiri dari beberapa tabel.
- **Field**, merupakan kolom pada tabel yang memiliki ukuran dan tipe data tertentu.
- **Record**, merupakan baris data yang memiliki nilai untuk masing-masing field pada tabel. Satu tabel biasanya terdiri dari banyak record.

- **Key**, merupakan suatu field (kolom) yang dapat dijadikan kunci dalam operasi tabel.
- **SQL**, merupakan singkatan dari *Structured Query Language* yaitu bahasa yang digunakan untuk mengakses dan memanipulasi database.

4.1.2 Aturan Merancang Database

Dalam pembuatan aplikasi perlu memperhatikan rancangan database agar aplikasi yang dibuat dapat berjalan sesuai konsep yang direncanakan. Perancangan database yang baik, hendaknya mengikuti aturan seperti berikut:

- Tidak boleh ada dua data atau lebih yang sama atau redundancy data.
- Setiap tabel harus memiliki kolom yang unik yang disebut primary key, dimana kolom tersebut tidak dapat berisi data yang sama.
- Besar dan ukuran tabel dibuat seminimal mungkin dengan pengaturan tipe data yang tepat.
- Pembuatan tabel dan kolom pada tabel hendaknya disesuaikan dengan kebutuhan aplikasi.

4.1.3 Penamaan Identifier

Identifier yaitu kata yang digunakan untuk penamaan database, tabel maupun kolom pada tabel. Memilih identifier tidak bebas, tetapi harus memenuhi aturan sebagai berikut:

- Identifier bersifat case sensitive, artinya huruf besar dan kecil dibedakan. Usahakan konsisten dalam menamai tabel atau kolom, bisa dengan format lower-case, UPPER-CASE atau camelCase, dan sebagainya.
- Identifier maksimal 64 karakter.
- Jangan menggunakan simbol atau karakter khusus kecuali strip (-) atau underscore (_).
- Identifier jangan diawali dengan angka.
- Jangan menggunakan kata yang termasuk dalam perintah SQL.

- Usahakan penamaan tabel dan kolom mencerminkan data yang akan disimpan pada tabel atau kolom tersebut.

4.2 Tipe Data pada MySQL

Dalam pembuatan tabel pada database hendaknya setiap kolom dibuat dengan tipe data yang tepat sesuai dengan data yang akan disimpan. Tipe data pada MySQL dibagi menjadi beberapa kelompok.

4.2.1 Tipe Data Numerik

Tipe data numerik digunakan untuk menyimpan data angka. Yang termasuk kelompok tipe data numerik yaitu sebagai berikut:

Tipe Data	Keterangan
Tinyint	Menyimpan bilangan bulat dari -128 hingga 127. Ukuran 1 byte.
Smallint	Menyimpan bilangan bulat dari -32768 hingga 32767. Ukuran 2 byte.
Int	Menyimpan bilangan bulat dari -2147483648 hingga 2147483647. Ukuran 4 byte.
Bigint	Menyimpan bilangan bulat dari $-9,22 \times 10^{18}$ hingga $9,22 \times 10^{18}$. Ukuran 8 byte.
Float	Menyimpan bilangan pecahan dari $-3.402823466E+38$ hingga $1.175494351E-38$, 0, dan $1.175494351E-38$ hingga $3.402823466E+38$. Ukuran 4 byte.
Double	Menyimpan bilangan pecahan dari $-1.79...E+308$ hingga $-2.22...E-308$, 0, dan $2.22...E-308$ hingga $1.79...E+308$. Ukuran 8 byte.
Real	Alias dari double.
Decimal	Menyimpan bilangan desimal.
Numeric	Alias dari decimal.

4.2.2 Tipe Data Waktu dan Tanggal

Untuk penyimpanan data waktu atau tanggal memiliki pilihan tipe tersendiri. Tipe data yang termasuk dalam kelompok waktu dan tanggal dapat dilihat pada tabel berikut:

Tipe Data	Keterangan
Date	Menyimpan data tanggal dari 1000-01-01 hingga 9999-12-31. Ukuran 3 byte.
Time	Menyimpan data waktu dengan jangkauan -838:59:59 hingga 838:59:59. Ukuran 3 byte.
Datetime	Menyimpan data tanggal dan waktu sekaligus dengan jangkauan 1000-01-01 00:00:00' s/d '9999-12-31 23:59:59. Ukuran 8 byte.
Year	Menyimpan data tahun dari sebuah tanggal dengan jangkauan 1900 hingga 2155. Ukuran 1 byte.

4.2.3 Tipe Data String

Tipe data dalam kelompok ini digunakan untuk menyimpan data berupa teks. Adapun macam-macam tipe datanya sebagai berikut:

Tipe Data	Keterangan
Char	Menyimpan data string dengan ukuran tetap dengan jangkauan 0 sampai 255 karakter.
Varchar	Menyimpan data string dengan ukuran dinamis. Jangkauannya 0 sampai 255.
Tinytext	Menyimpan data text dengan jangkauan 0 sampai 255 karakter.
Text	Menyimpan data text dengan jangkauan 0 sampai 65535 karakter.

4.2.4 Tipe Data BLOB

Tipe data blob digunakan untuk menyimpan data berupa kode biner. Yang termasuk kelompok tipe data ini yaitu:

Tipe Data	Keterangan
Bit	Menyimpan data biner hingga 64 digit biner.
Tinyblob	Menyimpan data biner hingga 255 digit biner.
Blob	Menyimpan data biner dari 2^{24} dengan jangkauan 6^{16} sampai 6^{16} sampai 1.
Mediumblob	Menyimpan data biner dari 2^{24} sampai 1 byte.
Longblob	Menyimpan data biner dari 1^{32} sampai 1.

4.2.5 Tipe Data Lainnya

Ada 2 tipe data lagi yang tidak dapat dikelompokkan ke tipe data sebelumnya, yaitu Enum dan Set.

Tipe Data	Keterangan
Enum	Kumpulan data (enumerasi), dapat menampung hingga 65535 string.
Set	Himpunan data (kombinasi), dapat menampung hingga 355 string.

4.3 Akses MySQL dengan Command Pompt

Karena MySQL yang digunakan satu paket dengan Xampp, maka aplikasi MySQL dapat ditemukan pada folder **xampp/mysql/bin**. Untuk membukanya melalui Command Prompt, ikuti langkah berikut:

1. Pastikan service MySQL telah dijalankan melalui Xampp Control Panel.
2. Buka command prompt, lalu masuk ke partisi tempat instal xampp.
3. Masuk ke folder bin dengan **cd xampp/mysql/bin**, lalu tekan enter.
4. Jalankan koneksi ke MySQL dengan skrip **mysql -u root**
5. Jika telah masuk ke MySQL, maka akan bertuliskan **mysql>**, tandanya siap untuk menjalankan perintah dengan skrip MySQL.

Lebih jelasnya perhatikan gambar berikut:

```
Command Prompt - mysql - u root
Microsoft Windows [Version 6.2.9200]
(C) 2012 Microsoft Corporation. All rights reserved.

C:\Users\Admin>D:
D:\>cd xampp\mysql\bin\>mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.5.5-10.1.21-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Gambar 4.1 Koneksi ke mysql pada Command Prompt

4.4 Manipulasi Database dan Tabel

Perintah-perintah yang digunakan untuk manipulasi database dan table pada MySQL biasa disebut **DDL** (*Data Definition Language*).

4.4.1 Manipulasi Database

Setiap membuat aplikasi pastinya membuat database terlebih dahulu. Adakalanya juga perlu menghapus database yang diperlukan, untuk itu perlu mengenali perintah-perintah yang digunakan untuk keperluan tersebut. Perintah-perintah yang digunakan untuk manipulasi database yaitu:

- **CREATE DATABASE**, digunakan untuk membuat database dengan format sebagai berikut:

```
CREATE DATABASE [IF NOT EXISTS] nama_database;
```

IF NOT EXISTS digunakan untuk memastikan database yang dibuat belum ada, namun tidak wajib dituliskan.

- **SHOW DATABASES**, digunakan untuk menampilkan daftar database yang telah dibuat.
- **USE**, digunakan untuk membuka database sehingga dapat memanipulasi tabel yang ada di dalamnya. Formatnya sebagai berikut:

```
USE nama_database;
```

- **DROP DATABASE**, digunakan untuk menghapus database dengan format sebagai berikut:

```
DROP DATABASE [IF EXISTS] nama_database;
```

Sebagai praktik pembuatan database, silakan ketikkan beberapa baris skrip yang dicetak tebal berikut:

```
mysql > CREATE DATABASE IF NOT EXISTS pekerja;  
Query OK, 1 row affected (0.00 sec)
```

```
mysql > CREATE DATABASE pegawai;  
Query OK, 1 row affected (0.06 sec)
```

```
mysql > SHOW DATABASES;  
+-----+
```

```
| Database          |
+-----+-----+
| mysql           |
| pegawai         |
| pekerja         |
+-----+
10 rows in set (0.09 sec)
```

```
mysql > DROP DATABASE IF EXISTS pekerja;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql > USE pegawai;
Database changed
mysql >
```

4.4.2 Manipulasi Tabel

Untuk dapat memanipulasi tabel, pastikan telah masuk ke salah satu database menggunakan perintah **USE**. Perintah-perintah yang dapat digunakan yaitu:

- **CREATE TABLE**, digunakan untuk membuat tabel baru. Formatnya sebagai berikut:

```
CREATE TABLE nama_tabel (
    kolom1 tipe(panjang) option1 option2 ... ,
    kolom2 tipe(panjang) option1 option2 ... ,
    ...
    kolomn tipe(panjang) option1 option2 ... ,
    PRIMARY KEY (kolom_key)
);
```

Banyaknya baris tergantung jumlah kolom yang dibuat pada tabel tersebut. Option diisi pengaturan tambahan pada kolom tersebut yang akan dibahas tersendiri. Kolom_key diisi nama kolom yang menjadi primary key, biasanya yang ditulis pertama.

- **DESC** atau **DESCRIBE**, digunakan untuk melihat struktur dari tabel yang telah dibuat. Formatnya sebagai berikut:

```
DESC nama_tabel;
```

- **SHOW TABLES**, digunakan untuk menampilkan tabel yang telah dibuat.
- **RENAME TABLE**, digunakan untuk mengubah nama tabel dengan format sebagai berikut:

```
RENAME TABLE nama_lama TO nama_baru;
```

- **DROP TABLE**, digunakan untuk menghapus tabel yang telah dibuat dengan format sebagai berikut:

```
DROP TABLE nama_tabel ;
```

- **ALTER TABLE**, digunakan untuk mengubah struktur tabel dengan format sebagai berikut:

```
ALTER TABLE nama_tabel alter_options;
```

Pada perintah **ALTER TABLE**, **alter_options** dapat diisi dengan salah satu option sesuai dengan perubahan struktur tabel yang diinginkan. Berikut format option yang dapat diberikan:

- **ADD** *nama_kolom tipe.panjang option_lain*, digunakan untuk menambahkan kolom baru dengan format penulisan nama kolom seperti saat membuat tabel.
- **ADD INDEX** *nama_index*, digunakan untuk menambah index pada tabel.
- **ADD PRIMARY KEY** (*kolom_kunci*), digunakan untuk menambahkan primary key.
- **CHANGE** *nama_kolom ...* digunakan untuk mengubah kolom menjadi kolom baru.
- **MODIFY ...** digunakan untuk mengubah definisi kolom dengan definisi yang baru.
- **DROP** *nama_kolom*, digunakan untuk menghapus suatu kolom.
- **RENAME TO** *nama_tabel_baru*, digunakan untuk mengubah nama tabel.

Adapun pada perintah **CREATE**, option yang dapat diberikan pada masing-masing kolom setelah penulisan tipe data dan ukurannya yaitu sebagai berikut:

- **NOT NULL**, digunakan untuk mewajibkan suatu kolom harus diisi, tidak boleh dikosongkan.
- **AUTO_INCREMENT**, digunakan untuk membuat angka urutan secara otomatis pada suatu kolom, biasanya yang dijadikan primary key.

- **DEFAULT** ‘nilai’, digunakan untuk membuat nilai default, artinya jika nilai suatu kolom tidak diisi ketika menginputkan data, maka nilai default yang akan jadi nilainya.

Berikut ini contoh skrip untuk latihan manipulasi tabel, silakan diketikkan yang dicetak tebal.

```
mysql > CREATE TABLE pekerja(
    -> id_pegawai int(5) NOT NULL AUTO_INCREMENT,
    -> nama_pegawai varchar(30) NOT NULL,
    -> tgl_lahir date,
    -> foto varchar(50),
    -> keterangan text,
    -> id_jabatan int(5),
    -> PRIMARY KEY (id_pegawai)
    -> );
Query OK, 0 rows affected (0.19 sec)

mysql > CREATE TABLE jabatan(
    -> id_jabatan int(5) NOT NULL,
    -> nama_jabatan varchar(40)
    -> );
Query OK, 0 rows affected (0.08 sec)

mysql > RENAME TABLE pekerja TO pegawai;
Query OK, 0 rows affected (0.33 sec)

mysql > DESC jabatan;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id_jabatan | int(5) | NO | | NULL | |
| nama_jabatan | varchar(40) | YES | | NULL | |
+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)

mysql > ALTER TABLE jabatan DROP nama_jabatan;
Query OK, 0 rows affected (0.27 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql > ALTER TABLE jabatan ADD nama_jabatan varchar(20);
Query OK, 0 rows affected (0.23 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql > ALTER TABLE jabatan ADD PRIMARY KEY (id_jabatan);
Query OK, 0 rows affected (0.16 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql > ALTER TABLE jabatan CHANGE id_jabatan id_jabatan int(5)
NOT NULL AUTO_INCREMENT;
Query OK, 0 rows affected (0.11 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

4.5 Manipulasi Data

Perintah-perintah yang digunakan untuk memanipulasi data biasa disebut DML (Data Manipulation Language). Perintah ini terdiri dari 4 perintah utama yaitu INSERT, UPDATE, DELETE dan SELECT. Berikut penjelasan dari keempat perintah tersebut:

- **INSERT**, digunakan untuk menyisipkan data dengan format yang biasa digunakan yaitu:

```
INSERT INTO nama_tabel VALUES ('nilai 1', 'nilai 2', ...);
```

Jika data yang diinput tidak semua kolom, dapat menggunakan format berikut:

```
INSERT INTO nama_tabel (kolom1, kolom2, ...) VALUES ('nilai 1', 'nilai 2', ...);
```

Selain dengan cara di atas, dapat juga menggunakan format berikut:

```
INSERT INTO nama_tabel SET kolom1='nilai 1', kolom2='nilai 2', ...;
```

- **UPDATE**, digunakan untuk mengubah data yang telah diinput dengan format:

```
UPDATE nama_tabel SET kolom1='nilai 1', kolom2='nilai 2', ... [WHERE kondisi];
```

- **DELETE**, digunakan untuk menghapus data yang telah diinput dengan format:

```
DELETE FROM nama_tabel [WHERE kondisi];
```

- **SELECT**, digunakan untuk menampilkan data yang telah diinput dengan format:

```
SELECT [kolom | *] FROM nama_tabel [WHERE kondisi];
```

Perintah **WHERE** baik pada UPDATE, DELETE maupun SELECT tidak wajib disertakan, namun jika tidak disertakan maka akan berpengaruh pada seluruh data yang ada. Umumnya pada perintah UPDATE dan DELETE selalu disertai WHERE untuk melakukan perubahan atau menghapus data dengan kondisi tertentu saja.

Pada perintah SELECT, jika diikuti tanda *, maka akan ditampilkan semua kolom. Jika ingin menampilkan beberapa kolom saja, maka tanda tersebut diganti nama kolom yang ingin ditampilkan dipisahkan dengan koma (,).

Sebagai latihan, ketikkan perintah-perintah yang dicetak tebal berikut:

```
mysql > INSERT INTO pegawai (nama_pegawai, tgl_lahir, id_jabatan)
VALUES ('Afaf Firdaus', '1998-08-16', '2');
Query OK, 1 row affected (0.08 sec)
```

```
mysql > INSERT INTO jabatan SET nama_jabatan='Direktur' ;
Query OK, 1 row affected (0.50 sec)
```

```
mysql > INSERT INTO jabatan VALUES (NULL, 'Manajer') ;
Query OK, 1 row affected (0.33 sec)
```

```
mysql > SELECT * FROM pegawai ;
+-----+-----+-----+-----+
| nama_pegawai | tgl_lahir | foto | keterangan | id_jabatan |
+-----+-----+-----+-----+
| Afaf Firdaus | 1998-08-16 | NULL | NULL | 2 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
mysql > UPDATE pegawai SET tgl_lahir='1997-08-17' WHERE id_pegawai='1' ;
Query OK, 1 row affected (0.13 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql > SELECT nama_pegawai, tgl_lahir FROM pegawai ;
+-----+-----+
| nama_pegawai | tgl_lahir |
+-----+-----+
| Afaf Firdaus | 1997-08-17 |
+-----+-----+
1 row in set (0.00 sec)
```

```
mysql > SELECT * FROM jabatan;
+-----+-----+
| id_jabatan | nama_jabatan |
+-----+-----+
| 1 | Direktur |
| 2 | Manajer |
+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql > DELETE FROM jabatan WHERE id_jabatan='1' ;
Query OK, 1 row affected (0.08 sec)
```

```
mysql > SELECT * FROM jabatan;
+-----+-----+
| id_jabatan | nama_jabatan |
+-----+-----+
```

```

|      2 | Manaj er      |
+-----+
1 row in set (0.00 sec)

mysql > INSERT INTO jabatan SET nama_jabatan='Sekretaris';
Query OK, 1 row affected (0.06 sec)

mysql > SELECT id_jabatan, nama_jabatan FROM jabatan;
+-----+-----+
| id_jabatan | nama_jabatan |
+-----+-----+
|      2 | Manaj er      |
|      3 | Sekretaris     |
+-----+-----+
2 rows in set (0.00 sec)

mysql > SELECT * FROM jabatan WHERE id_jabatan='3';
+-----+-----+
| id_jabatan | nama_jabatan |
+-----+-----+
|      3 | Sekretaris     |
+-----+-----+
1 row in set (0.00 sec)

```

4.6 Beberapa Teknik Menampilkan Data

Dalam menampilkan data menggunakan perintah SELECT banyak sekali variasi yang dapat digunakan untuk menampilkan data sesuai dengan kriteria yang diinginkan. Beberapa variasi yang dapat digunakan yaitu sebagai berikut:

- Menggunakan operator perbandingan pada kondisi, operator yang dapat digunakan yaitu = (sama dengan), <> atau != (tidak sama dengan), > (lebih dari), >= (lebih dari atau sama dengan), < (kurang dari), dan <= (kurang dari atau sama dengan).
- Menggunakan **LIKE** untuk menampilkan data dengan kolom mengandung kata tertentu. Biasanya dikombinasikan dengan tanda % di awal atau akhir kata.
- Menggunakan operator penghubung jika kondisi lebih dari satu, yang dapat digunakan yaitu **AND** atau **&&** (bernilai true jika keduanya benar), **OR** atau **||** (bernilai true jika minimal salah satu benar) dan **!** (membalikkan nilai kondisi antara true dan false).
- Menggunakan **ORDER BY** diikuti nama kolom untuk menampilkan data berdasarkan urutan pada kolom tertentu. Diikuti **DESC** setelah nama kolom jika diurutkan secara descending (dari besar ke kecil).

Jika ingin diurutkan secara acak, maka nama kolom setelah ORDER BY dapat diganti dengan RAND().

- Mengelompokkan data dengan **GROUP BY** diikuti nama kolom yang akan dijadikan acuan pengelompokan data.
- Menggunakan **LIMIT** untuk membatasi jumlah data yang tampil dengan format:

LIMIT awal , jumlah record

Perintah LIMIT juga dapat digunakan tanpa menuliskan awalnya, langsung menyebutkan jumlah record saja.

- Menggunakan function untuk menampilkan hasil operasi data pada tabel seperti menampilkan jumlah data menggunakan **count()**, menampilkan nilai tertinggi dengan **max()**, dan sebagainya. Penjelasan mengenai function akan dibahas pada subbab tersendiri.

Sebagai latihan, ketikkan skrip yang dicetak tebal sehingga tampil seperti berikut:

```
mysql > INSERT INTO jabatan set nama_jabatan='Bendahara';
Query OK, 1 row affected (0.31 sec)
```

```
mysql > SELECT * FROM jabatan ORDER BY id_jabatan DESC;
+-----+-----+
| id_jabatan | nama_jabatan |
+-----+-----+
| 4 | Bendahara |
| 3 | Sekretaris |
| 2 | Manajer |
+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql > SELECT * FROM jabatan ORDER BY nama_jabatan LIMIT 2;
+-----+-----+
| id_jabatan | nama_jabatan |
+-----+-----+
| 4 | Bendahara |
| 2 | Manajer |
+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql > SELECT * FROM jabatan WHERE nama_jabatan LIKE '%ar%';
+-----+-----+
| id_jabatan | nama_jabatan |
+-----+-----+
| 3 | Sekretaris |
| 4 | Bendahara |
+-----+-----+
2 rows in set (0.00 sec)
```

```

mysql > SELECT * FROM jabatan WHERE id_jabatan>2 AND
id_jabatan<4;
+-----+-----+
| id_jabatan | nama_jabatan |
+-----+-----+
|       3 | Sekretaris   |
+-----+-----+
1 row in set (0.00 sec)

mysql > SELECT count(id_jabatan) FROM jabatan;
+-----+
| count (id_jabatan) |
+-----+
|           3 |
+-----+
1 row in set (0.00 sec)

```

4.7 Function pada MySQL

Function pada MySQL ditulis setelah perintah SELECT. Function pada MySQL hampir sama dengan bahasa pemrograman lainnya. Untuk penjelasan masing-masing function akan dibagi menjadi beberapa kelompok.

4.7.1 Function untuk Operasi String

Function-function untuk operasi string pada MySQL beserta keterangannya dapat dilihat pada tabel berikut:

Function	Keterangan
CONCAT(string1, string2, ...)	Untuk menggabungkan beberapa string menjadi satu string.
CONCAT_WS(separator, string1, string2, ...)	Untuk menggabungkan beberapa string menjadi satu string dengan separator (pemisah) diantara masing-masing string.
SUBSTR(string, awal, panjang)	Untuk mengambil sebagian string pada argumen pertama dari posisi awal pada argumen kedua, dan panjang karakter pada argumen ketiga. Perintah ini memiliki fungsi sama dengan function MID().
LENGTH(string)	Untuk menghitung panjang karakter dari sebuah string.
LEFT(string, panjang)	Untuk mengambil sebagian string dari posisi kiri sebanyak jumlah karakter yang ditentukan pada argumen kedua.

RIGHT(string, panjang)	Untuk mengambil sebagian string dari posisi kanan sebanyak jumlah karakter yang ditentukan pada argumen kedua.
LTRIM(string)	Untuk menghilangkan spasi di awal string.
RTRIM(string)	Untuk menghilangkan spasi di akhir string.
TRIM(string)	Untuk menghilangkan spasi di awal dan akhir string.
REPLACE(string, kata_awal, kata_baru)	Untuk mengubah suatu kata yang disebutkan pada argumen kedua dari string pada argumen pertama dengan kata yang disebutkan pada argumen ketiga.
REPEAT(string, jumlah)	Untuk mengulang string sebanyak jumlah yang disebutkan pada argumen kedua.
REVERSE(string)	Untuk membalik string.
LCASE(string)	Untuk mengubah string menjadi huruf kecil semua. Sama fungsinya dengan LOWER().
UCASE(string)	Untuk mengubah string menjadi huruf kapital semua. Sama fungsinya dengan UPPER().

Sebagai latihan, silakan ketikkan skrip yang dicetak tebal dan lihat hasilnya di bawahnya.

```
mysql > SELECT SUBSTR(nama_jabatan, 3, 4) FROM jabatan;
+-----+
| SUBSTR(nama_jabatan, 3, 4) |
+-----+
| naje |
| kret |
| ndah |
+-----+
3 rows in set (0.00 sec)

mysql > SELECT LCASE(nama_jabatan) FROM jabatan;
+-----+
| LCASE(nama_jabatan) |
+-----+
| manajer |
| sekretaris |
| bendahara |
+-----+
3 rows in set (0.07 sec)

mysql > SELECT UCASE(nama_jabatan) FROM jabatan;
+-----+
| UCASE(nama_jabatan) |
+-----+
| MANAJER |
| SEKRETARIS |
| BENDAHARA |
+-----+
3 rows in set (0.02 sec)
```

```
mysql > SELECT LEFT(nama_jabatan, 3) FROM jabatan;
+-----+
| LEFT(nama_jabatan, 3) |
+-----+
| Man          |
| Sek          |
| Ben          |
+-----+
3 rows in set (0.00 sec)
```

4.7.2 Function untuk Operasi Matematika

Untuk operasi angka terdapat function khusus yang berbeda dengan operasi string. Berikut daftar function untuk operasi angka beserta keterangannya.

Function	Keterangan
ABS(angka_negatif)	Untuk mengambil nilai positif dari sebuah angka.
MOD(angka1, angka2)	Untuk mengambil sisa dari angka1 dibagi angka2.
FLOOR(angka_desimal)	Untuk membulatkan angka desimal ke angka di bawahnya.
CEIL(angka_desimal)	Untuk membulatkan angka desimal ke angka di atasnya.
ROUND(angka_desimal, angka)	Untuk membulatkan angka desimal sehingga angka di belakang koma sejumlah angka yang disebutkan pada argumen kedua.
POW(angka, pangkat)	Untuk memangkatkan angka pada argumen pertama dengan angka pada argumen kedua.
MOD(angka1, angka2)	Untuk mengambil sisa dari angka1 dibagi angka2.
RAND()	Untuk mengambil angka acak antara 0 sampai 10.
GREATEST(angka1, angka2, ...)	Untuk mengambil angka terbesar dari kumpulan angka.
COUNT(kolom)	Untuk menghitung jumlah baris kolom tertentu pada sebuah tabel.
MAX(kolom)	Untuk mengambil nilai tertinggi dari sebuah kolom pada tabel.
MIN(kolom)	Untuk mengambil nilai terendah dari sebuah kolom pada tabel.
SUM(kolom)	Untuk mengambil jumlah total nilai dari suatu kolom pada tabel.
AVG(kolom)	Untuk mengambil nilai rata-rata dari sebuah kolom pada tabel.

Selain menggunakan function matematika, dapat juga menggunakan operasi matematika pada query dengan operator + (penjumlahan), - (pengurangan), * (perkalian), / (pembagian) dan % (modulus).

Sebagai latihan, ketikkan perintah-perintah berikut:

```
mysql > SELECT 5 * 5;
+-----+
| 5 * 5 |
+-----+
|      25 |
+-----+
1 row in set (0.07 sec)

mysql > SELECT ABS(-20);
+-----+
| ABS(-20) |
+-----+
|      20 |
+-----+
1 row in set (0.00 sec)

mysql > SELECT POW(5, 2);
+-----+
| POW(5, 2) |
+-----+
|      25 |
+-----+
1 row in set (0.08 sec)

mysql > SELECT ROUND(2.1780821, 2);
+-----+
| ROUND(2.1780821, 2) |
+-----+
|      2.18 |
+-----+
1 row in set (0.04 sec)

mysql > SELECT MAX(id_jabatan) FROM jabatan;
+-----+
| MAX(id_jabatan) |
+-----+
|          4 |
+-----+
1 row in set (0.06 sec)
```

4.7.3 Function untuk Operasi Tanggal dan Waktu

Untuk operasi tanggal dan waktu, berikut function yang dapat digunakan:

Function	Keterangan
NOW()	Untuk mendapatkan waktu saat ini.

MONTH(tanggal)	Untuk mendapatkan bulan dari suatu tanggal dalam bentuk angka yang dimulai dari 1 untuk bulan Januari.
WEEK(tanggal)	Untuk mendapatkan urutan minggu dalam setahun dari suatu tanggal.
YEAR(tanggal)	Untuk mendapatkan tahun dari suatu tanggal.
HOUR(waktu)	Untuk mendapatkan nilai jam dari suatu waktu.
MINUTE(waktu)	Untuk mendapatkan nilai menit dari suatu waktu.
SECOND(waktu)	Untuk mendapatkan nilai detik dari suatu waktu.
DATE_FORMAT(tanggal, format)	Untuk memformat tampilan tanggal.
TIME_FORMAT(waktu, format)	Untuk memformat tampilan waktu.

Untuk perintah DATE_FORMAT() dan TIME_FORMAT(), format yang dapat digunakan yaitu sebagai berikut:

- %M : Nama bulan (January – December)
- %W : Nama hari dalam seminggu (Sunday – Saturday)
- %D : Urutan hari dalam sebulan
- %Y : Tahun dalam 4 digit
- %y : Tahun dalam 2 digit
- %a : Singkatan nama hari dalam seminggu (Sun – Sat)
- %H : Jam dalam format 24 jam
- %i : Menit antara 00 – 59
- %s : Detik antara 00 – 99

Sebagai contoh, perhatikan baris kode berikut:

```
mysql > SELECT YEAR(tgl_akhir) FROM pegawai ;
+-----+
| YEAR(tgl_akhir) |
+-----+
| 1997 |
+-----+
1 row in set (0.08 sec)
```

```
mysql > SELECT YEAR(tgl_akhir) AS tahun FROM pegawai ;
+-----+
| tahun |
+-----+
| 1997 |
+-----+
1 row in set (0.03 sec)
```

```

mysql > SELECT DATE_FORMAT(tgl_lahir, '%W, %M %D, %Y') FROM pegawai;
+-----+
| DATE_FORMAT(tgl_lahir, '%W, %M %D, %Y') |
+-----+
| Sunday, August 17th, 1997                |
+-----+
1 row in set (0.03 sec)

```

4.7.4 Function untuk Keperluan Lain

Selain untuk operasi string, tanggal dan operasi matematika, masih ada beberapa function yang digunakan untuk keperluan lain. Berikut daftar dan keterangannya:

Function	Keterangan
DATABASE()	Untuk mengambil nama database yang sedang dibuka.
USER()	Untuk mengambil user yang sedang aktif.
PASSWORD(string)	Untuk melakukan enkripsi suatu string.
ENCODE(string, password)	Untuk melakukan enkripsi suatu string dengan kata kunci tertentu.
DECODE(string, password)	Untuk melakukan deskripsi string yang telah dienkripsi dengan ENCODE() menggunakan kata kunci tertentu.
MD5(string)	Untuk mengenkripsi string menggunakan metode MD5.
LAST_INSERT_ID()	Untuk mengambil id terakhir dalam proses insert jika kolomnya menggunakan AUTO_INCREMENT.
VERSION()	Untuk mengecek versi MySQL yang sedang digunakan.

Sebagai contoh, ketikkan skrip berikut dan lihat hasilnya di bawahnya.

```

mysql > SELECT PASSWORD('admin');
+-----+
| PASSWORD('admin') |
+-----+
| *4ACFE3202A5FF5CF467898FC58AAB1D615029441 |
+-----+
1 row in set (0.00 sec)

mysql > SELECT MD5('admin');
+-----+
| MD5('admin') |
+-----+

```

```
| 21232f297a57a5a743894a0e4a801fc3 |
+-----+
1 row in set (0.06 sec)

mysql > SELECT ENCODE('admin', 'pass');
+-----+
| ENCODE('admin', 'pass') |
+-----+
| ••KÜ• |
+-----+
1 row in set (0.30 sec)
```

4.8 Menampilkan Data dari Banyak Tabel

Dalam sebuah database biasanya ada beberapa tabel yang saling berhubungan satu sama lain. Dalam mengakses database tersebut, adakalanya harus menampilkan data dari kedua tabel atau lebih tersebut yang saling berhubungan secara bersamaan. Untuk keperluan ini ada beberapa perintah yang dapat digunakan yaitu:

- **INNER JOIN**, digunakan untuk menghubungkan 2 tabel secara dua arah, sehingga jika pada salah satu tabel ada data yang NULL, data tidak ditampilkan. Format penulisannya sebagai berikut:

```
SELECT tabel 1.* , tabel 2.* FROM tabel 1 INNER JOIN tabel 2 ON
tabel 1.primary_key=tabel 2.foreign_key;
```

Khusus INNER JOIN, selain dengan format di atas juga dapat menggunakan format sederhana dengan WHERE sebagai berikut:

```
SELECT tabel 1.* , tabel 2.* FROM tabel 1, tabel 2 WHERE
tabel 1.primary_key=tabel 2.foreign_key;
```

- **LEFT JOIN**, digunakan untuk menghubungkan 2 tabel dimana data pada tabel sebelah kiri akan ditampilkan semua walaupun pada tabel sebelah kanan NULL. Format penulisannya sebagai berikut:

```
SELECT tabel 1.* , tabel 2.* FROM tabel 1 LEFT JOIN tabel 2 ON
tabel 1.primary_key=tabel 2.foreign_key;
```

- **RIGHT JOIN**, digunakan untuk menghubungkan 2 tabel dimana data pada tabel sebelah kanan akan ditampilkan semua walaupun pada tabel sebelah kiri NULL. Format penulisannya sebagai berikut:

```
SELECT tabel 1.* , tabel 2.* FROM tabel 1 RIGHT JOIN tabel 2 ON
tabel 1.primary_key=tabel 2.foreign_key;
```

Pada penggunaan semua perintah di atas, masing-masing tabel dapat diberi alias menggunakan **AS**, sehingga memudahkan dalam penulisanya. Misalnya pada INNER JOIN, formatnya dapat dibuat sebagai berikut:

```
SELECT t1.*, t2.* FROM tabel1 AS t1 RIGHT JOIN tabel2 AS t2 ON  
t1.primary_key=t2.foreign_key;
```

Sebagai latihan, silakan ketikkan skrip berikut:

```
mysql > SELECT * FROM pegawai, jabatan WHERE pegawai.id_jabatan  
= jabatan.id_jabatan;  
+-----+-----+-----+-----+  
| nama_pegawai | tgl_lahir | foto | keterangan | id_jabatan |  
i_d_pegawai | i_d_jabatan | nama_jabatan |  
+-----+-----+-----+-----+  
| Afaf Firdaus | 1997-08-17 | NULL | NULL | 2 |  
1 | 2 | Manajer |  
+-----+-----+-----+-----+  
1 row in set (0.14 sec)
```

```
mysql > SELECT pegawai.*, jabatan.* FROM pegawai INNER JOIN  
jabatan ON pegawai.id_jabatan=jabatan.id_jabatan;  
+-----+-----+-----+-----+  
| nama_pegawai | tgl_lahir | foto | keterangan | id_jabatan |  
i_d_pegawai | i_d_jabatan | nama_jabatan |  
+-----+-----+-----+-----+  
| Afaf Firdaus | 1997-08-17 | NULL | NULL | 2 |  
1 | 2 | Manajer |  
+-----+-----+-----+-----+  
1 row in set (0.31 sec)
```

```
mysql > SELECT t1.i_d_pegawai, t1.nama_pegawai, t2.nama_jabatan  
FROM pegawai AS t1 LEFT JOIN jabatan AS t2 ON t1.i_d_jabatan=  
t2.i_d_jabatan;  
+-----+-----+-----+  
| i_d_pegawai | nama_pegawai | nama_jabatan |  
+-----+-----+-----+  
| 1 | Afaf Firdaus | Manajer |  
+-----+-----+-----+  
1 row in set (0.00 sec)
```

```
mysql > SELECT t1.i_d_pegawai, t1.nama_pegawai, t2.nama_jabatan  
FROM pegawai AS t1 RIGHT JOIN jabatan AS t2 ON t1.i_d_jabatan=  
t2.i_d_jabatan;  
+-----+-----+-----+  
| i_d_pegawai | nama_pegawai | nama_jabatan |  
+-----+-----+-----+  
| 1 | Afaf Firdaus | Manajer |  
| NULL | NULL | Sekretaris |  
| NULL | NULL | Bendahara |  
+-----+-----+-----+
```

```
3 rows in set (0.00 sec)
```

Pada hasil perintah RIGHT JOIN di atas tampak semua data jabatan (tabel kanan) ditampilkan walaupun belum ada data pegawai (tabel kiri) yang memiliki jabatan tersebut (data NULL).

4.9 Menampilkan Data dengan Sub Select

MySQL telah mendukung adanya sub select yaitu adanya perintah SELECT lain di dalam perintah SELECT. Cara penulisannya dapat menggunakan salah satu format berikut:

```
SELECT .... WHERE kolom = [ANY|ALL] (SELECT ....);
SELECT .... WHERE kolom [NOT] IN (SELECT ....);
SELECT ROW(nilai 1, nilai 2, ...) = [ANY] (SELECT kolom1, kolom2,
....);
SELECT .... WHERE kolom = [NOT] EXISTS (SELECT ....);
SELECT .... FROM (SELECT ....) AS alias WHERE ....;
```

Contoh penggunaan sub select dapat dilihat pada skrip berikut:

```
mysql > SELECT * FROM jabatan WHERE id_jabatan IN (SELECT
```

```
id_jabatan FROM pegawai);
```

id_jabatan	nama_jabatan
2	Manajer

```
1 row in set (0.03 sec)
```

```
mysql > SELECT * FROM jabatan WHERE id_jabatan=(SELECT
```

```
max(id_jabatan) FROM jabatan);
```

id_jabatan	nama_jabatan
4	Bendahara

```
1 row in set (0.00 sec)
```

4.10 Menampilkan Data dengan HAVING

Hasil dari penggunaan function sering kali ditampilkan dengan alias. Sayangnya, alias tersebut tidak efektif digunakan untuk pengontrolan

kondisi dengan WHERE. Sebagai gantinya dapat menggunakan HAVING. Contohnya pada skrip berikut:

```
mysql > SELECT t1.id_pegawai, t1.nama_pegawai,  
LEFT(t2.nama_jabatan, 3) AS singkatan_jabatan FROM pegawai AS t1,  
jabatan AS t2 WHERE t1.id_jabatan=t2.id_jabatan HAVING  
singkatan_jabatan='Dir';  
+-----+-----+-----+  
| id_pegawai | nama_pegawai | singkatan_jabatan |  
+-----+-----+-----+  
| 2 | Daf fa Shidqi | Dir |  
+-----+-----+-----+  
1 row in set (0.00 sec)
```

4.11 Menyiapkan Database untuk Aplikasi

Pada bab selanjutnya, akan dibahas contoh pembuatan aplikasi sederhana yang menggunakan database yang telah dibuat pada bab ini yaitu database **pegawai**. Karena struktur tabel maupun nama tabel mungkin telah mengalami banyak perubahan dari hasil percobaan, pastikan kembali database pegawai telah memiliki 2 tabel yaitu **pegawai** dan **jabatan** dengan struktur sebagai berikut:

Struktur tabel pegawai:

Nama Kolom	Tipe	Null	Key	Ekstra
id_pegawai	Int(5)	NO	PRIMARY	Auto_increment
nama_pegawai	Varchar(30)	NO		
tgl_lahir	Date	YES		
foto	Varchar(50)	YES		
keterangan	Text	YES		
id_jabatan	Int(5)	YES		

Struktur tabel jabatan:

Nama Kolom	Tipe	Null	Key	Ekstra
id_jabatan	Int(5)	NO	PRIMARY	Auto_increment
nama_jabatan	Varchar(20)	NO		

Selanjutnya tambahkan lagi 1 tabel yaitu tabel **user** yang digunakan untuk keperluan login ke aplikasi, dengan skrip sebagai berikut:

```
mysql > CREATE TABLE user(
-> id_user int(5) NOT NULL AUTO_INCREMENT,
-> nama_lengkap varchar(50),
-> username varchar(20),
-> password varchar(50),
-> PRIMARY KEY (id_user)
-> );
Query OK, 0 rows affected (0.44 sec)
```

Tambahkan data pada masing-masing tabel secukupnya untuk memudahkan mengecek keberhasilan penulisan program nantinya, terutama pada tabel user yang digunakan untuk login. Khusus untuk tabel user, pastikan gunakan function MD5() pada data password agar password terenkripsi, sehingga skrip untuk menambahkan data pada tabel user menjadi seperti berikut:

```
mysql > INSERT INTO user VALUES(NULL, 'Daffa Shidqi',
-> 'admin', MD5('admin'));
Query OK, 1 row affected (0.08 sec)
```

Pada skrip di atas, username dan password untuk login sama-sama diset "admin".

5 PHP

5.1 Mengenal PHP

5.1.1 Pengertian dan Aplikasi yang Diperlukan

PHP merupakan kependekan dari *PHP Hypertext Preprocessor* yaitu bahasa pemrograman web yang dapat disisipkan dalam skrip HTML dan bekerja di sisi server. Tujuan dari bahasa ini adalah membantu para pengembangan web untuk membuat web dinamis dengan cepat.

Agar dapat menjalankan PHP harus menyediakan perangkat lunak berikut:

- Web Server (Apache, IIS, Personal Web Server/PWS)
- PHP Server
- Database Server (MySQL, Interbase, MS SQL, dll)

Aplikasi di atas tidak perlu diinstal satu per satu, karena biasanya telah tersedia paket aplikasi yang telah menyediakan ketiganya dalam satu paket aplikasi seperti Xampp atau Appserv.

Jika menggunakan Xampp, maka skrip PHP ditulis di dalam folder **htdocs**. Namun, jika menggunakan Appserv, maka skrip PHP di tulis di dalam folder **www**.

5.1.2 Fungsi PHP

Fungsi PHP dalam pemrograman web diantaranya sebagai berikut:

- Menghasilkan halaman dinamis pada website.
- Membuat, membuka, menulis, membaca, menghapus dan menutup file pada server.
- Memproses data yang dikirim dari form.
- Mengirim dan mengakses cookie.
- Modifikasi data pada database.
- Mengontrol akses user.
- Mengenkripsi data.

5.1.3 Penulisan Sintaks PHP

Skrip PHP dituliskan di antara tanda `<?php` dan `?>` yang memisahkan skrip PHP dengan skrip lainnya. Satu file PHP dapat berisi full skrip PHP atau dapat disisipkan diantara skrip lain seperti HTML, CSS maupun JavaScript. Namun, file yang berisi skrip PHP wajib disimpan dengan ekstensi `*.php` dan disimpan di server (folder `htdocs` atau `www`). Jika disimpan dengan ekstensi HTML atau di simpan di sembarang tempat maka skrip PHP tidak diproses sebagaimana mestinya.

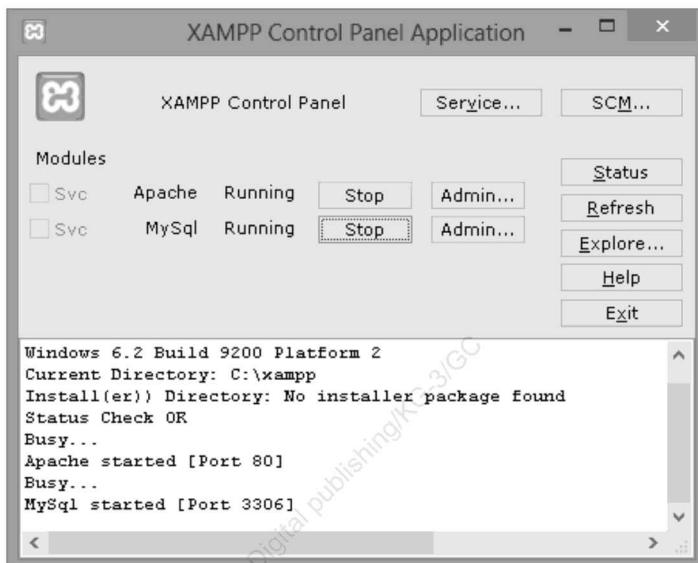
Setiap baris skrip PHP harus diakhiri dengan tanda semicolon (;). Jika tidak, maka akan menampilkan pesan error. Berikut contoh penulisan skrip PHP di dalam skrip HTML.

File 5_1_3_sintax_dasar.php

```
<!DOCTYPE HTML>
<html >
  <head>
    <title>Latihan PHP</title>
  </head>
  <body>
    <?php
      echo "Latihan PHP";
    ?>
  </body>
</html >
```

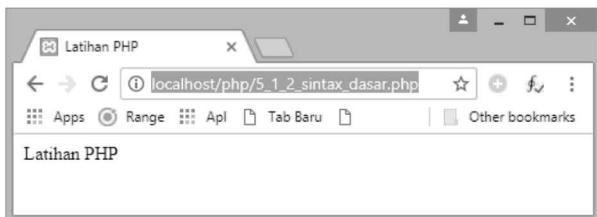
5.1.4 Menjalankan File PHP

Menjalankan file PHP tidak dapat dilakukan seperti saat membuka file HTML yaitu dobel klik pada file. File PHP hanya bisa dibuka dengan memanggil URL melalui web browser dengan mengaktifkan service Apache terlebih dahulu.



Gambar 5.1 Mengaktifkan service Apache dan MySQL

Cara memanggilnya cukup dengan mengetikkan URL dengan format **localhost/nama_folder/nama_file.php** pada address bar browser. Misalnya pada folder htdocs dibuat folder **PHP**, kemudian skrip sebelumnya disimpan di dalamnya, maka cara menjalankan pada browser dengan mengetikkan **localhost/php/5_1_3_sintax_dasar.php** seperti gambar berikut:



Gambar 5.2 Menjalankan file PHP

5.1.5 Skrip HTML dalam PHP

Untuk menampilkan skrip HTML ke browser, pada PHP menggunakan perintah **print** atau **echo**. Keduanya hampir sama, namun echo lebih sering digunakan karena pemrosesannya lebih cepat dibanding print. Penulisan print dan echo dapat menggunakan tanda kurung atau tidak, jadi boleh ditulis **print** atau **echo**, juga dapat ditulis **print()** atau **echo()**.

Tulisan yang ditampilkan dengan echo dapat berupa tulisan biasa atau skrip HTML yang diletakkan di antara tanda petik ganda ("") atau petik tunggal (''). Ada beberapa hal penting yang perlu diperhatikan saat menggunakan echo dan print yaitu sebagai berikut:

- Jangan menuliskan petik ganda ("") dalam text, jika menggunakan echo atau print dengan petik ganda. Begitu juga dengan petik tunggal.
- Jika terpaksa harus menuliskan petik, maka berikan tanda slash (\) sebelum petik.
- Sebagai ganti tanda petik ganda, dapat menggunakan petik tunggal (''), jika menggunakan echo atau print dengan petik ganda. Begitu juga sebaliknya.

Sebagai contoh, perhatikan skrip berikut:

File 5_1_5_echo_print.php

```
<?php
echo "<h2>PHP itu mudah</h2>";
echo("Saya senang belajar \"PHP\" <br>");

print 'Jika terus belajar, ';
print('saya pasti bisa menguasai "PHP"')
?>
```



Gambar 5.3 Perintah echo dan print

5.1.6 Komentar

Menuliskan komentar di dalam skrip PHP dapat dilakukan dengan dua cara. Jika komentar hanya satu baris, maka dapat menggunakan dua garis miring (//) atau tanda pagar (#). Jika komentar lebih dari satu baris, maka komentar diawali dengan tanda /* dan diakhiri dengan tanda */. Contohnya pada skrip berikut:

File 5_1_6_komentar.php

```
<?php
#ini adalah komentar
// ini juga komentar
/* ini komentar
   jika lebih dari satu baris */
?>
```

Jika ditampilkan pada browser, maka tidak akan tampil apa-apa, karena komentar tidak akan ditampilkan pada browser. Fungsi komentar hanya untuk memberi keterangan pada suatu skrip.

5.2 Dasar-Dasar PHP

5.2.1 Variabel

Variabel digunakan untuk menyimpan data berupa teks, angka, dan sebagainya yang nilainya dapat diubah-ubah, sehingga memudahkan

dalam mengakses data tersebut. Nama variabel dapat berupa kata apa saja, tetapi harus mengikuti aturan berikut:

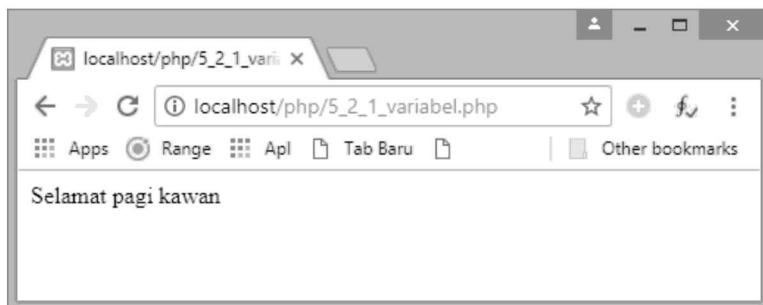
- Nama variabel harus diawali dengan huruf atau underscore (_).
- Nama variabel hanya boleh mengandung huruf, angka dan underscore.
- Nama variabel yang terdiri lebih dari satu kata, dapat dipisahkan dengan underscore.
- Nama variabel tidak boleh menggunakan kata yang merupakan bagian dari skrip PHP.
- Penulisan nama variabel harus diawali dengan tanda dolar (\$).
- Variabel bersifat *case-sensitive*, sehingga huruf besar dan huruf kecil dibedakan.

Pendeklarasian variabel pada PHP tidak perlu menuliskan tipe datanya seperti pada bahasa pemrograman lainnya. PHP akan otomatis mengkonversi ke tipe data yang sesuai. Contoh penggunaan variabel dapat dilihat pada skrip berikut:

File 5_2_1_variabel.php

```
<?php  
    Skat a = "Selamat pagi kawan";  
    echo Skat a;  
?>
```

Jika dijalankan pada browser, hasilnya sebagai berikut:



Gambar 5.4 Contoh penggunaan variabel

5.2.2 Tipe Data

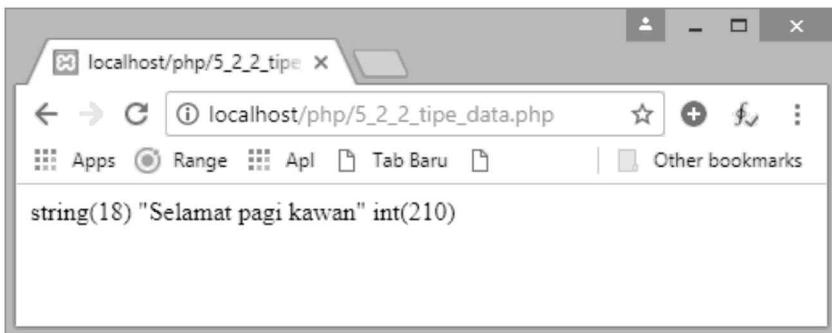
Variabel pada PHP dapat menyimpan berbagai macam tipe data. Beda tipe data dapat melakukan hal yang berbeda pula. Berikut tipe data yang didukung oleh PHP,

- **String**, yaitu tipe data berupa teks atau angka yang ditulis di antara petik ganda, seperti: "Selamat pagi!" atau "12345".
- **Integer**, yaitu tipe data numerik non desimal antara -2,147,483,648 hingga 2,147,483,647.
- **Float**, yaitu tipe data numerik berupa angka desimal. Tanda koma pada angka desimal menggunakan tanda titik, misal: 23.25.
- **Boolean**, yaitu tipe data yang hanya dapat bernilai *true* atau *false*.
- **Array**, yaitu tipe data yang menyimpan banyak nilai dalam satu variabel.
- **Object**, yaitu tipe data yang menyimpan data beserta informasi bagaimana data diproses.
- **Null**, yaitu tipe data yang hanya dapat bernilai NULL.

Untuk menampilkan tipe data beserta nilainya pada PHP dapat menggunakan perintah **var_dump()**. Contohnya seperti file berikut:

File 5_2_2_tipe_data.php

```
<?php
$kat a = "Selamat pagi kawan";
$angka = 210;
var_dump($kat a);
var_dump($angka);
?>
```



Gambar 5.5 Menampilkan tipe data dengan var_dump()

5.2.3 Konstanta

Berbeda dengan variabel, nilai konstanta tidak dapat diubah pada skrip. Konstanta juga digunakan hanya untuk menyimpan data simpel. Penulisan nama konstanta memiliki aturan sama dengan variabel, tetapi tidak menggunakan tanda \$. Dalam praktiknya, konstanta biasanya ditulis dengan huruf kapital semua. Cara mendefinisikan konstanta yaitu dengan perintah **define()** dengan format sebagai berikut:

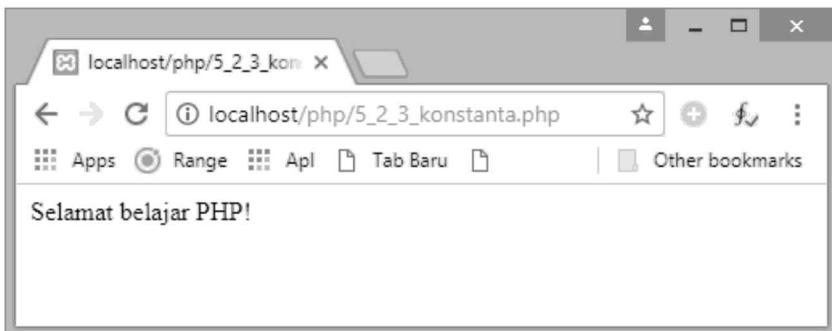
```
define([nama_konstanta], [nilai], [case_sensitive]);
```

Dalam praktiknya, penulisan konstanta biasanya hanya menyertakan parameter `nama_konstanta` dan `nilai`. Secara default `case-sensitive` bernilai *false*.

Contoh penggunaan konstanta dapat dilihat pada skrip berikut:

File 5_2_3_konstanta.php

```
<?php  
define("SALAM", "Selamat belajar PHP!");  
echo SALAM;  
?>
```



Gambar 5.6 Penggunaan konstanta

5.2.4 Operator

Setiap bahasa pemrograman mesti memiliki operator. Operator digunakan untuk melakukan operasi terhadap satu atau lebih data atau variabel sehingga menghasilkan data baru. Ada beberapa jenis operator dalam PHP yaitu:

Operator Aritmetika

Operator aritmatika merupakan operator yang digunakan untuk melakukan perhitungan aritmatika.

Operator	Arti	Contoh
+	Penjumlahan	$2 + 3 = 5$
-	Pengurangan	$6 - 3 = 3$
*	Perkalian	$4 * 3 = 12$
/	Pembagian	$9 / 3 = 3$
%	Modulus / Sisa bagi	$10 \% 4 = 2$

Operator Perbandingan

Operator perbandingan merupakan operator yang digunakan untuk membandingkan 2 nilai atau variabel. Hasil dari operator ini adalah true atau false.

Operator	Arti	Contoh	Hasil
==	Sama dengan	$2 + 3 == 6$	False
===	Identik (sama beserta tipe datanya)	$2 * 3 === "6"$	False
!= atau <>	Tidak sama dengan	$2 + 3 != 6$	True
!==	Tidak identik	$2 * 3 !== "6"$	True
>	Lebih besar dari	$2 + 3 > 4$	True
<	Lebih kecil dari	$2 + 3 < 4$	False
<=	Lebih kecil dari atau sama dengan	$2 + 3 <= 5$	True
>=	Lebih besar dari atau sama dengan	$2 + 3 >= 5$	True

Operator Logika

Operator logika digunakan untuk menggabungkan dua pernyataan atau lebih. Hasil dari operator ini false atau true.

Operator	Arti	Keterangan
and atau &&	Dan	Hasilnya benar jika kedua pernyataan benar
or atau	Atau	Hasilnya benar jika salah satu pernyataan atau keduanya benar
xor	Atau	Hasilnya benar jika salah satu pernyataan benar, tetapi tidak keduanya
!	tidak	Hasilnya benar jika pernyataan salah

Operator Increment dan Decrement

Operator ini digunakan untuk menambahkan atau mengurangi nilai sebuah variabel.

Operator	Nama	Keterangan
++\$x	Pre-increment	Menambahkan variabel \$x dengan 1, lalu mengembalikan nilai \$x
\$x++	Post-increment	Mengembalikan variabel \$x, kemudian menambahkan nilai \$x dengan 1
--\$x	Pre-decrement	Mengurangi variabel \$x dengan 1, lalu mengembalikan nilai \$x
\$x--	Post-decrement	Mengembalikan variabel \$x, kemudian mengurangi dengan 1.

Operator Assignment

Operator ini digunakan untuk menentukan atau mengubah nilai suatu variabel numerik.

Assignment	Sama dengan	Keterangan
$\$x = \y	$\$x = \y	Nilai x sama dengan nilai y
$\$x += \y	$\$x = \$x + \$y$	Nilai x ditambahkan dengan nilai y
$\$x -= \y	$\$x = \$x - \$y$	Nilai x dikurangi dengan nilai y
$\$x *= \y	$\$x = \$x * \$y$	Nilai x dikalikan dengan nilai y
$\$x /= \y	$\$x = \$x / \$y$	Nilai x dibagi dengan nilai y
$\$x %= \y	$\$x = \$x \% \$y$	Nilai x modulus dengan nilai y

Operator String

Operator ini digunakan untuk menggabungkan tipe data string.

Operator	Contoh
.	<code>\$kata = "Saya sedang". " Belajar PHP.</code>
.=	<code>\$kata .= " dengan penuh semangat";</code>

Operator Kondisi

Operator ini digunakan untuk memberikan nilai pada suatu variabel sesuai kondisi yang ditentukan. Jika kondisi **true** maka akan diberi nilai dengan nilai di sebelah kiri tanda titik dua, sedangkan jika kondisi **false** maka akan diberi nilai dengan nilai di sebelah kanan tanda titik dua. Kondisi diletakkan di dalam kurung diikuti dengan tanda ? yang memisahkan dengan nilai.

Operator	Keterangan
? ... : ...	Tanda tanya memisahkan kondisi dengan nilai, sedangkan tanda titik dua memisahkan nilai pertama dan nilai kedua. Nilai pertama dipakai jika kondisi true dan nilai kedua dipakai jika kondisi false.

5.2.5 Array

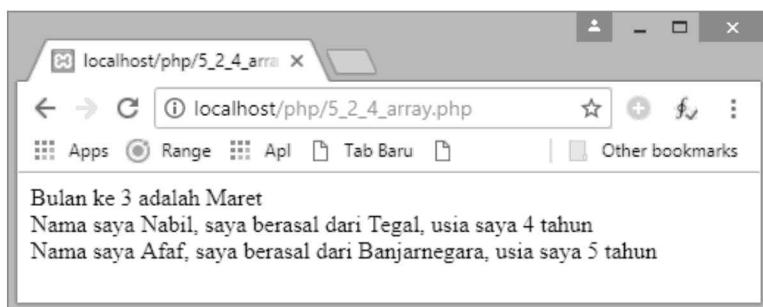
Array merupakan sebuah variabel khusus yang dapat menyimpan nilai lebih dari satu. Array dapat memiliki index (urutan), sehingga dapat

diakses dengan menyebutkan angka index yang dimulai dari nol. Array juga dapat memiliki nama kunci untuk mengidentifikasi masing-masing nilai. Array seperti ini biasa disebut array asosiatif. Contoh penggunaan array dapat dilihat pada skrip berikut:

File 5_2_5_array.php

```
<?php
    // cara 1
    $bulan = array("Januari", "Februari", "Maret", "April",
    "Mei", "Juni", "Jul i", "Agustus", "September", "Okt ober",
    "November", "Desember");
    echo "Bulan ke 3 adalah $bulan[2]";
    $profil = array("nama" => "Nabil", "alamat" => "Tegal",
    "usia" => "6");
    echo "<br>Nama saya $profil[nama], saya berasal dari
$profil[alamat], usia saya $profil[usia] tahun";
    // cara 2
    $profil['nama'] = "Afaf";
    $profil['alamat'] = "Banjarnegara";
    $profil['usia'] = 7;
    echo '<br>Nama saya '. $profil['nama']. ', saya berasal dari
'. $profil['alamat']. ', usia saya '. $profil['usia']. ' tahun';
?>
```

Jika dituliskan di dalam string yang diapit petik ganda, penulisan nama kunci tanpa menggunakan petik. Sedangkan di dalam string yang menggunakan petik tunggal tidak dapat menuliskan array maupun variabel, sehingga ditulis di luar string yang dihubungkan dengan tanda titik. Penulisan nama kunci jika ditulis di luar string menggunakan petik tunggal ('). Hasilnya seperti gambar berikut:



Gambar 5.7 Contoh array

5.2.6 Statement Control

Statement control akan mengatur bagaimana data akan ditampilkan sesuai dengan kondisi tertentu. Statement control menjadi kunci dari logika program. Kesalahan dalam penggunaan statement control, mungkin tidak menampilkan error, tetapi aplikasi tidak berjalan sesuai semesternya.

Ada beberapa statement control yang dapat digunakan pada PHP, yaitu sebagai berikut:

Percabangan dengan if

Statement control ini digunakan untuk menjalankan suatu program tertentu jika sebuah syarat kondisi terpenuhi. Format dari statement control ini adalah:

```
if (kondisi) statement;
```

Format di atas hanya berlaku jika statement hanya satu baris. Jika statement lebih dari satu baris, maka formatnya menjadi:

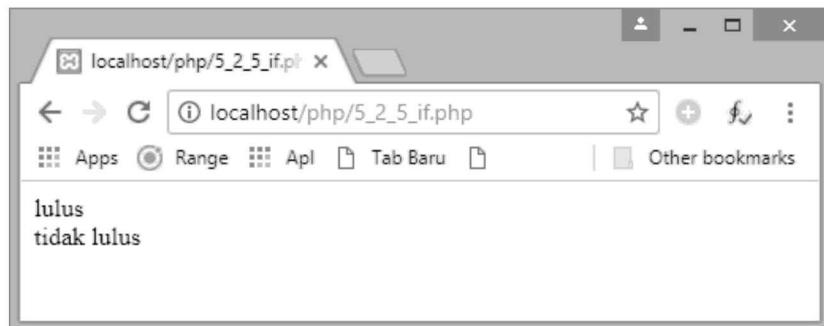
```
if (kondisi)
{
    Statement 1;
    Statement 2;
    ...
}
```

Contoh penggunaanya sebagai berikut:

File 5_2_6_if.php

```
<?php
$nilai = 80;
if ($nilai >= 80) echo "lulus";
$nilai = 70;
if ($nilai < 80)
{
    echo "<br> tidak lulus";
?>
```

Ubah nilai variabel \$nilai antara kurang dari 80 dan lebih dari 80 untuk melihat hasilnya. Hasil dari skrip di atas seperti gambar berikut:



Gambar 5.8 Hasil percabangan dengan if

Percabangan dengan if ... else

Statement control ini digunakan untuk menjalankan program tertentu jika sebuah kondisi terpenuhi sekaligus menyediakan alternatif lain jika kondisi tidak terpenuhi. Format penulisannya sebagai berikut:

```
if (kondisi)
{
    Statement jika benar;
}
else
{
    Statement jika salah;
```

Jika statement hanya satu baris, dapat menggunakan format berikut;

```
if (kondisi) Statement jika benar;
else Statement jika salah;
```

Contoh penggunaanya dapat diperhatikan pada skrip berikut:

File 5_2_6_if_else.php

```
<?php
$nilai = 80;
if ($nilai >= 80) echo "lulus";
else echo "tidak lulus";

$nilai = 70;
if ($nilai >= 80)
{
    echo "<br> lulus";
}
else
{
    echo "<br> tidak lulus";
```

?>

Hasilnya tergantung nilai variabel \$nilai. Skrip di atas akan menghasilkan sama seperti gambar sebelumnya. Untuk mendapatkan hasil yang berbeda, ubah nilai variabel \$nilai.

Percabangan dengan if...elseif... else

Statement kontrol ini digunakan untuk mengecek kondisi berikutnya ketika kondisi pertama tidak terpenuhi. Jika semua kondisi tidak terpenuhi baru dijalankan statement terakhir setelah **else**. Formatnya sebagai berikut:

```
if (kondisi 1)
{
    Statement jika kondisi 1 benar;
}
elseif (kondisi 2)
{
    Statement jika kondisi 2 benar;
}
else
{
    Statement jika semua kondisi salah;
}
```

Contoh penggunaanya dapat diperhatikan pada skrip berikut:

File 5_2_6_if_elseif_else.php

```
<?php
$nilai = 70;
if ($nilai >= 80)
{
    echo "Sangat baik";
}
elseif ($nilai >= 60)
{
    echo "Cukup baik";
}
else
{
    echo "<br> tidak lulus";
?>
```

Pada skrip di atas tentu yang terpenuhi adalah kondisi kedua yaitu nilai lebih atau sama dengan 60, dan menghasilkan keterangan **Cukup baik**.

Percabangan dengan switch

Statement control ini digunakan untuk menjalankan program pada setiap kemungkinan nilai dari sebuah variabel. Formatnya sebagai berikut:

```
switch($variabel)
{
    case "option": statement;
    break;

    case "option": statement;
    break;

    default: statement;
    break;
}
```

Default dijalankan jika tidak ada option yang terpenuhi. Contoh penggunaanya sebagai berikut:

File 5_2_6_switch.php

```
<?php
$skota= "Bandung";
switch($skota)
{
    case "Bandung": echo "Ibukota Jawa Barat"; break;
    case "Surabaya": echo "Ibukota Jawa Timur"; break;
    case "Semarang": echo "Ibukota Jawa Tengah"; break;
    default: echo "Ibukota provinsi lain"; break;
}
?>
```

Hasil skrip di atas tentu saja **Ibukota Java Barat**. Untuk mendapatkan hasil lain, ubah nilai variabel \$kota menjadi Semarang, Surabaya, atau ibukota provinsi lain.

Perulangan dengan while

While digunakan untuk menjalankan program secara berulang-ulang selama kondisi masih terpenuhi. Format penulisannya sebagai berikut:

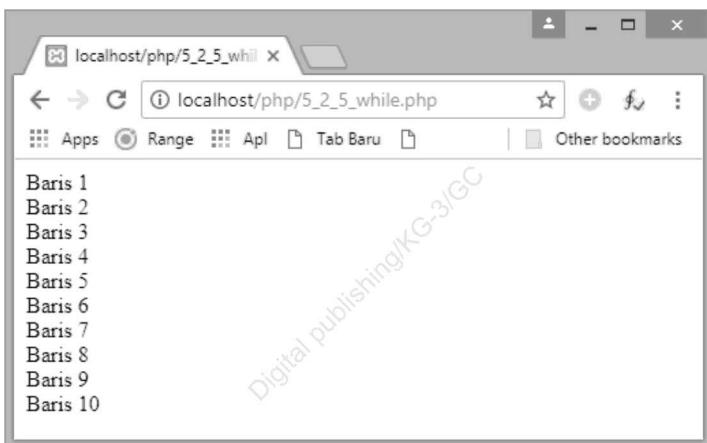
```
while(kondisi)
{
    Statement;
}
```

Berikut contoh penggunaanya:

File 5_2_6_while.php

```
<?php  
    $no = 1;  
    while($no <= 10)  
    {  
        echo " Baris $no <br>";  
        $no++;  
    }  
?>
```

Untuk mendapatkan hasil yang berbeda, ubah angka 10 pada skrip di atas. Hasil dari skrip di atas seperti gambar berikut:



Gambar 5.9 Penggunaan while

Perulangan dengan do ... while

Statement control digunakan untuk menjalankan program secara berulang-ulang selama kondisi yang ditentukan di akhir terpenuhi. Hampir sama dengan while, hanya kondisi ditulis di akhir. Formatnya sebagai berikut:

```
do  
{  
    Statement;  
}  
while(kondisi);
```

Untuk contoh penggunaanya, perhatikan skrip berikut:

File 5_2_6_do_while.php

```
<?php  
$no = 1;  
  
do  
{  
    echo " Baris $no <br>";  
    $no++;  
}  
while($no <= 10)  
?>
```

Hasilnya sama seperti gambar sebelumnya.

For

Pada prinsipnya statement ini sama dengan while, yaitu untuk menjalankan program secara berulang-ulang selama kondisi terpenuhi, tetapi pada statement control ini batas awal dan akhir pengulangan langsung ditentukan. Formatnya sebagai berikut:

```
for (inisialisasi counter; kondisi; increment/decrement counter)  
{  
    Statement;  
}
```

Contoh penggunaan dari statement control ini seperti pada skrip berikut:

File 5_2_6_for.php

```
<?php  
for ($no=1; $no<=10; $no++) {  
    echo " Baris $no <br>";  
}  
?>
```

Hasilnya sama seperti pada gambar yang menggunakan while.

Perulangan Array dengan Foreach

Statement control ini digunakan khusus untuk melakukan perulangan data pada array. Format penulisannya sebagai berikut:

```
foreach($variabel_array as $alias)  
{  
    Statement;  
}
```

Jika berupa array asosiatif, maka dapat menggunakan format seperti berikut:

```
foreach($variabel_array as $key=>$value)
{
    Statement;
}
```

Contoh penggunaanya seperti skrip berikut:

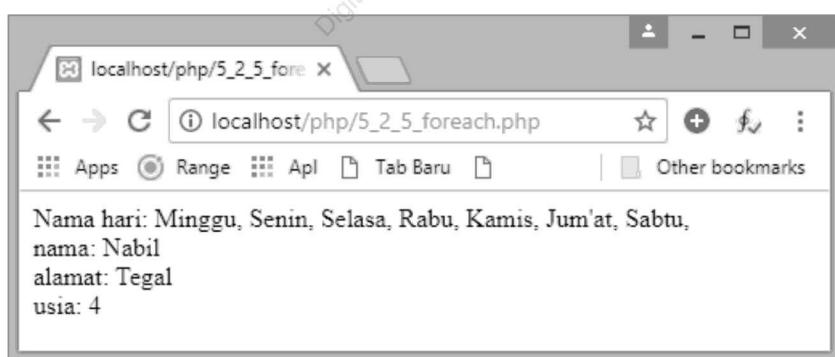
File 5_2_6_foreach.php

```
<?php
    $nama_hari = array("Minggu", "Senin", "Selasa", "Rabu",
"Kamis", "Jum'at", "Sabtu");

echo "Nama hari: ";
foreach($nama_hari as $hari)
{
    echo "$hari, ";
}

$profil = array("nama" => "Nabil", "alamat" => "Tegal",
"usia" => "4");
foreach($profil as $key=>$value)
{
    echo "<br>$key: $value";
}
?>
```

Hasilnya seperti gambar berikut:



Gambar 5.10 Penggunaan foreach

Sama seperti pada JavaScript, perulangan pada PHP juga dapat dikontrol dengan perintah **continue** untuk melewati sebuah

pengulangan dan **break** untuk menghentikan pengulangan sebelum akhir pengulangan tercapai.

5.2.7 Modularitas

Di dalam PHP, sebuah file PHP dapat disisipi file PHP lain, yaitu dengan perintah **include** atau **require**. Keduanya memiliki fungsi yang hampir sama, bedanya ketika terjadi error pada program. Jika menggunakan require, program menghasilkan fatal error dan program berhenti dieksekusi. Sementara jika menggunakan include, program hanya menghasilkan pesan peringatan dan eksekusi program tetap dilanjutkan ke baris berikutnya.

Contoh penggunaanya, perhatikan skrip kedua file berikut:

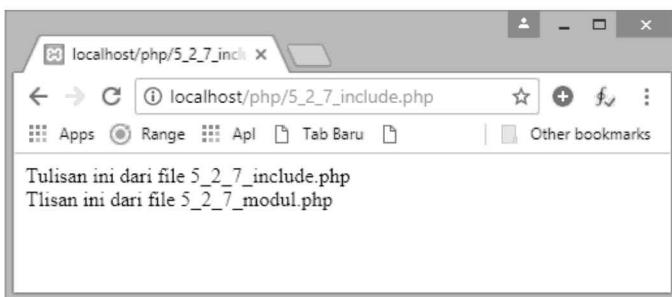
File 5_2_7_include.php

```
<?php  
    echo "Tulisan ini dari file 5_2_7_include.php <br>";  
    include("5_2_7_modul.php");  
?>
```

File 5_2_7_modul.php

```
<?php  
    echo "Tulisan ini dari file 5_2_7_modul.php";  
?>
```

Jika file pertama dijalankan, kedua teks pada kedua file akan tampil sebagai berikut:



Gambar 5.11 *Include file PHP*

5.3 Function

Function digunakan untuk membuat blok program yang sering digunakan di dalam penulisan skrip PHP. Function tidak otomatis dipanggil saat halaman di-load. Function hanya akan dieksekusi ketika sebuah function dipanggil. Pemanggilan function ini dapat dilakukan berkali-kali setiap membutuhkan. Dengan demikian, function akan memudahkan programmer dalam menyingkat skrip yang sering digunakan, terutama yang skripnya terlalu panjang.

PHP telah menyediakan ribuan function yang siap digunakan. Namun, programmer dapat membuat function sendiri sesuai kebutuhan.

5.3.1 Function Dasar

Cara membuat function cukup mudah, yaitu dengan mengikuti aturan format berikut:

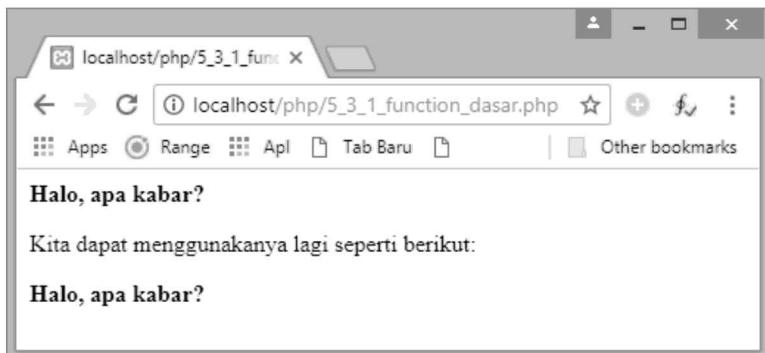
```
function nama_function() {  
    //Kode program di sini  
}
```

Nama function memiliki aturan yang sama dengan variabel, namun tanpa tanda \$. Untuk memanggil function, cukup sebutkan nama function. Contohnya seperti berikut:

File 5_3_1_function_dasar.php

```
<?php  
function ucap_salam() {  
    echo "<b>Hal o, apa kabar?</b>";  
}  
  
ucap_salam();  
  
echo "<p>Ki ta dapat menggunakan lagi seperti berikut:  
</p>";  
ucap_salam();  
?>
```

Hasilnya seperti gambar berikut:



Gambar 5.12 Penggunaan function

5.3.2 Function dengan Argumen

Jika dalam penggunaan function ingin menghasilkan program yang berbeda-beda, tetapi memiliki format yang sama, dapat menggunakan function yang disertai argumen, dimana argumen diletakkan di dalam tanda kurung. Contohnya sebagai berikut:

File 5_3_2_function_argumen.php

```
<?php
function ucapan_salam($nama) {
    echo "Halo, apa kabar <b>$nama</b>" ;
}
ucapan_salam("Nabi l");
echo "<p>Kita dapat menggunakanya lagi seperti berikut:
</p>";
ucapan_salam("Afaf");
?>
```

Pada skrip di atas function memiliki 1 argumen yaitu \$nama. Argumen function dapat lebih dari 1 dengan dipisahkan tanda koma. Dalam pemanggilannya, nilai parameter pengganti argumen harus disertakan.

Sebuah argumen dapat memiliki nilai default, yaitu dengan menyebutkan nilainya sewaktu pembuatan function. Argumen yang memiliki nilai default tidak wajib disebutkan saat pemanggilan function.

Hasil skrip di atas seperti gambar berikut:



Gambar 5.13 Argumen pada function

5.3.3 Function dengan Nilai Kembalian

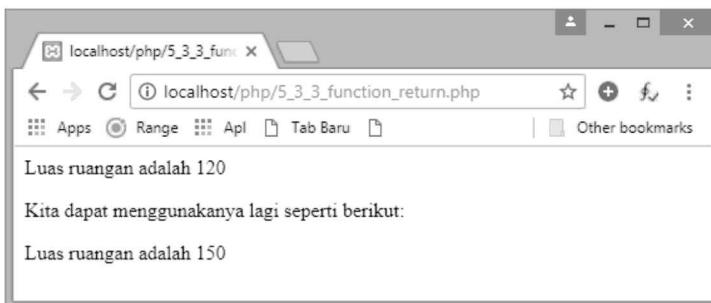
Sebuah function dapat mengembalikan suatu nilai, yaitu dengan perintah **return**. Function yang mengembalikan suatu nilai dapat digunakan sebagai nilai suatu variabel atau dapat ditampilkan langsung menggunakan perintah echo. Contohnya sebagai berikut:

File 5_3_3_function_return.php

```
<?php
function hitung_luas($panjang, $lebar=10) {
    $luas = $panjang * $lebar;
    return $luas;
}

echo "Luas ruangan adalah ". hitung_luas(12);
echo " <p>Kita dapat menggunakanya lagi seperti berikut:</p>";
$panjang = 15;
$lebar = 10;
$luas = hitung_luas($panjang, $lebar);
echo "Luas ruangan adalah $luas";
?>
```

Argumen kedua pada function di atas mempunyai nilai default. Pada pemanggilan pertama argumen tidak disebutkan, maka dianggap 10 sebagai nilai default. Hasilnya sebagai berikut:



Gambar 5.14 Function dengan nilai kembalian

5.3.4 Function untuk Operasi String

Untuk operasi string, PHP telah menyediakan berbagai function yang siap pakai. Berikut daftar function yang biasa digunakan:

Function	Keterangan
addslashes(string)	Memberikan tanda \ pada karakter khusus seperti petik ganda, petik tunggal, dan sebagainya.
chr(ascii)	Menghasilkan karakter dari sebuah kode ASCII.
explode(pemisah, string, batas)	Mengkonversi string menjadi array.
htmlentities(string,flags, character_set, double_encode)	Mengkonversi karakter ke kode HTML.
htmlspecialchars(string,flags, character_set, double_encode)	Mengkonversi karakter tertentu ke kode HTML.
implode(pemisah, array)	Menghasilkan string dari suatu array.
join(pemisah, array)	Menghasilkan string dari suatu array. Alias dari implode().
lcfirst(string)	Mengkonversi karakter pertama string menjadi huruf kecil.
ltrim(string, charlist)	Menghilangkan spasi atau karakter tertentu di sebelah kiri string.
md5(string, raw)	Enkripsi dengan MD5 dari sebuah string. Raw menentukan format output binary atau heksadesimal, diisi true atau false.
nl2br(string, xhtml)	Menyisipkan tag pada setiap ganti baris dari sebuah string. Xhtml diisi true atau false.

<code>number_format(number, desimal, pemisah desimal, pemisah ribuan)</code>	Memformat angka menjadi ribuan.
<code>printf(format, arg1, arg2, arg3, ...)</code>	Menghasilkan string dengan format tertentu.
<code>rtrim(string, charlist)</code>	Menghilangkan spasi atau karakter tertentu pada sebelah kanan string.
<code>str_ireplace(dicari, pengganti, string, jumlah)</code>	Mengganti beberapa karakter pada sebuah string tanpa membedakan besar kecilnya huruf.
<code>str_replace(dicari, pengganti, string, jumlah)</code>	Mengganti beberapa karakter pada sebuah string dengan membedakan besar kecilnya huruf.
<code>str_shuffle(string)</code>	Mengacak karakter pada sebuah string.
<code>str_word_count(string, return, char)</code>	Menghasilkan jumlah karakter dari sebuah string.
<code>stripslashes(string)</code>	Mengembalikan perintah addslashes()
<code>strlen(string)</code>	Menghasilkan panjang sebuah string.
<code>strpos(string, dicari, mulai)</code>	Menghasilkan posisi pertama dari sebuah string pada suatu string. Versi case-insensitive menggunakan strpos().
<code>strripos(string, dicari, mulai)</code>	Menghasilkan posisi terakhir sebuah string pada suatu string. Versi case-insensitive menggunakan strripos().
<code>strtolower(string)</code>	Mengkonversi string ke huruf kecil semua.
<code>strtoupper(string)</code>	Mengkonversi string ke huruf besar semua.
<code>substr(string, mulai, panjang)</code>	Menghasilkan bagian dari sebuah string.
<code>trim(string, charlist)</code>	Menghilangkan spasi atau karakter lain di awal maupun akhir string.
<code>ucfirst(string)</code>	Mengubah huruf pertama string menjadi huruf kapital.
<code>ucwords(string)</code>	Mengubah huruf pertama setiap kata dari sebuah string menjadi huruf kapital.

Masing-masing argumen pada function di atas yang dicetak tebal wajib disertakan, sedangkan yang lain bersifat opsional, sehingga tidak harus disertakan. Berikut contoh penggunaan dari function yang disebutkan di atas:

File 5_3_4_function_string.php

```
<?php
$text = "Saya senang belajar PHP";
```

```

echo substr($text, 0, 11) . " (hasil substr())<br>";
echo substr($text, 12) . " (hasil substr())<br>";
echo strToUpper($text) . " (hasil strToUpper())<br>";
echo ucwords($text) . " (hasil ucwords())<br>";
echo str_replace("senang", "suka", $text) . " (hasil
str_replace())<br>";
echo strpos($text, "P") . " (hasil strpos())<br>";
$skata = explode(' ', $text);
echo $skata[1] . " (hasil explode())<br>";
?>

```

Hasilnya seperti gambar berikut:



Gambar 5.15 Hasil operasi string

5.3.5 Function untuk Operasi Tanggal

Function untuk operasi tanggal perlu penjelasan khusus, sehingga masing-masing function dibahas tersendiri.

Function date()

Function ini digunakan untuk mencetak tanggal dengan format tertentu. Format tanggal ditulis di dalam kurung. Beberapa huruf yang dapat digunakan untuk menulis format tanggal yaitu sebagai berikut:

Time:

a : am atau pm

A : AM atau PM

g : jam tanpa 0 di depan, bernilai antara 1 – 12

- G : jam tanpa 0 di depan, bernilai antara 0 – 23
- h : jam dengan 0 di depan, bernilai antara 1 – 12
- H : jam dengan 0 di depan, bernilai antara 00 – 23
- i : menit dengan 0 di depan, bernilai antara 00 – 59
- s : detik dengan 0 di depan, bernilai antara 00 – 59

Day:

- d : tanggal dengan 0 di depan, bernilai antara 01 – 31
- j : tanggal tanpa 0 di depan, bernilai antara 1 – 31
- D : hari dalam mingguan (singkatan), bernilai antara Sun – Sat
- I : hari dalam mingguan, bernilai antara Sunday – Saturday
- w : hari dalam mingguan tanpa 0 di depan, bernilai antara 0 – 6
- z : hari dalam tahunan tanpa 0 di depan, bernilai antara 0 – 365

Month:

- m : bulan dengan 0 di depan, bernilai antara 01 – 12
- n : bulan tanpa 0 di depan, bernilai antara 1 – 12
- M : singkatan dari bulan, bernilai antara Jan – Dec
- F : nama bulan lengkap, bernilai antara January – December
- t : jumlah hari dalam sebulan, bernilai antara 28 – 31

Year:

- L : 1 jika melompati tahun dan 0 jika tidak
- Y : tahun dengan 4 digit
- y : tahun dengan 2 digit (00 – 99)

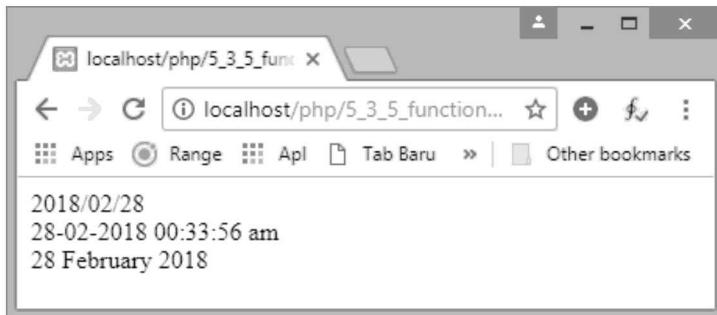
Sebagai contoh, perhatikan skrip berikut:

File 5_3_5_function_date.php

```
<?php
```

```
echo date('Y/m/d');
echo "<br>";
echo date('d-m-Y H:i:s a');
echo "<br>";
echo date('d F Y');
?>
```

Hasilnya seperti gambar berikut:



Gambar 5.16 Format penulisan tanggal

Function mktime()

Function ini digunakan untuk membuat timestamp dengan format sintaks sebagai berikut:

```
mktime(hour, minute, second, month, day, year);
```

Berikut contoh penggunaanya:

File 5_3_5_function_mktime.php

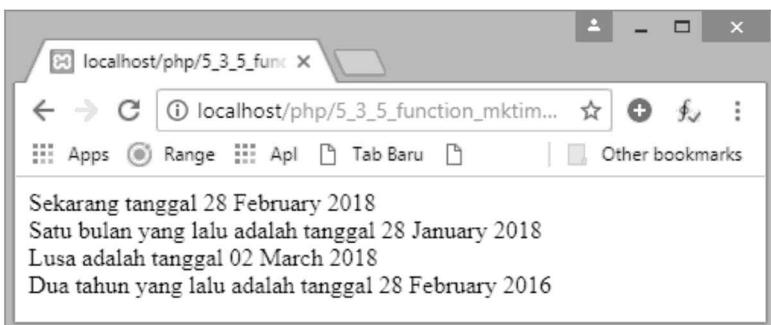
```
<?php
    $sekarang = mktime(12, 13, 40, date('m'), date('d'),
date('Y'));
    echo "Sekarang tanggal ".date('d F Y', $sekarang);

    $bulan_lalu = mktime(0, 0, 0, date('m')-1, date('d'),
date('Y'));
    echo "<br>Satu bulan yang lalu adalah tanggal ".date('d F Y',
$bulan_lalu);

    $lusa = mktime(0, 0, 0, date('m'), date('d')+2, date('Y'));
    echo "<br>Lusa adalah tanggal ".date('d F Y', $lusa);

    $tahun = mktime(0, 0, 0, date('m'), date('d'), date('Y')-2);
    echo "<br>Dua tahun yang lalu adalah tanggal ".date('d F Y',
$tahun);
?>
```

Hasilnya seperti gambar berikut:



Gambar 5.17 Penggunaan function mktime()

Selain 2 function yang paling sering digunakan di atas, ada beberapa function lagi yang berkaitan dengan penggunaan tanggal, yaitu `date_default_timezone_set()` untuk menentukan timezone dan `strtotime()` untuk mengubah string menjadi format tanggal atau waktu.

5.3.6 Function untuk Operasi Matematika

Untuk operasi matematika, banyak sekali function yang dapat digunakan. Berikut ini fungsi dari masing-masing function:

Function	Keterangan
<code>abs(number)</code>	Menghasilkan angka positif dari sebuah angka.
<code>acos(number)</code>	Menghasilkan nilai acos.
<code>asin(number)</code>	Menghasilkan nilai asin.
<code>bindec(binary_string)</code>	Mengkonversi angka biner ke desimal.
<code>atan(number)</code>	Menghasilkan nilai atan.
<code>ceil(number)</code>	Pembulatan angka ke atas.
<code>cos(number)</code>	Menghasilkan nilai cosinus.
<code>decbin(number)</code>	Mengkonversi angka desimal ke biner.
<code>dechex(number)</code>	Mengkonversi angka desimal ke heksadesimal.
<code>decodt(number)</code>	Mengkonversi angka desimal ke oktal.
<code>deg2rad(number)</code>	Mengkonversi derajat ke radian.

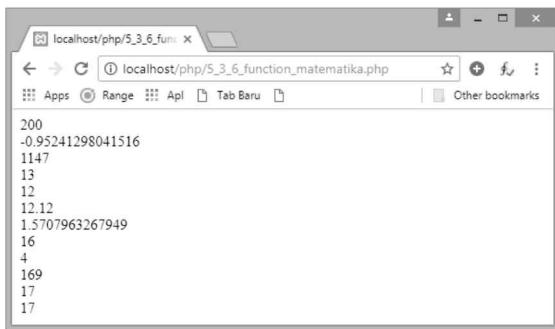
<code>floor(number)</code>	Pembulatan angka ke bawah.
<code>hexdec(hex_string)</code>	Mengkonversi angka heksadesimal ke desimal.
<code>log(number)</code>	Menghasilkan nilai log.
<code>max(array list nilai)</code>	Menghasilkan angka tertinggi dalam sebuah array atau daftar angka yang ditentukan.
<code>min(array list nilai)</code>	Menghasilkan angka terendah dalam sebuah array atau daftar angka yang ditentukan.
<code>octdec(oct_string)</code>	Mengkonversi angka oktal ke desimal.
<code>pi()</code>	Menghasilkan nilai pi.
<code>pow(number1, number2)</code>	Memangkatkan bilangan pertama dengan bilangan kedua.
<code>rad2deg(number)</code>	Mengkonversi radian ke derajat.
<code>rand()</code>	Menghasilkan angka secara acak.
<code>round(number, presisi, mode)</code>	Membulatkan bilangan desimal. Mode dapat diisi PHP_ROUND_HALF_UP, PHP_ROUUND_HALF_DOWN, PHP_ROUND_HALF_EVEN, PHP_ROUND_HALF_ODD.
<code>sin(number)</code>	Menghasilkan nilai sinus.
<code>sqrt(number)</code>	Menghasilkan akar kuadrat.
<code>tan(number)</code>	Menghasilkan nilai tangen.

Argumen dicetak tebal berarti wajib diisi. Contoh penggunaan dari beberapa function di atas dapat diperhatikan pada skrip berikut:

File 5_3_6_function_matematika.php

```
<?php
echo abs(-200) . "<br>";
echo cos(60) . "<br>";
echo bindec("10001111011") . "<br>";
echo ceil(12.123) . "<br>";
echo floor(12.123) . "<br>";
echo round(12.123, 2) . "<br>";
echo deg2rad(90) . "<br>";
echo pow(4, 2) . "<br>";
echo sqrt(16) . "<br>";
echo rand(10, 1000) . "<br>";
echo max(10, 11, 17, 13) . "<br>";
echo max(array(10, 11, 17, 13)) . "<br>";
?>
```

Hasilnya seperti gambar berikut:



Gambar 5.18 Contoh penggunaan function matematika

5.3.7 Function untuk Penanganan File

PHP dapat digunakan untuk memanipulasi direktori maupun file. Function-function yang digunakan untuk keperluan tersebut, diantaranya sebagai berikut:

Function	Keterangan
basename(path, suffix)	Menghasilkan nama file dari sebuah path.
chgrp(file, group)	Mengubah file group.
chmod(file, mode)	Mengubah file mode.
chown(file, owner)	Mengubah file own.
copy(file, file_tujuan)	Meng-copy file.
dirname(path)	Menghasilkan nama direktori dari sebuah path.
disk_free_space(direktori)	Menghasilkan sisa kapasitas penyimpanan sebuah direktori.
disc_total_space(direktori)	Menghasilkan total kapasitas penyimpanan sebuah direktori.
fclose(file)	Menutup file yang sedang dibuka.
feof(file)	Mengecek akhir sebuah file yang dibuka.
fgetc(file)	Menghasilkan karakter dari file yang dibuka.
fgets(file, panjang)	Menghasilkan baris dari sebuah file yang dibuka.
fgetss(file, panjang, tag)	Menghasilkan baris dengan menghapus tag HTML dan PHP dari file yang dibuka.
file(path, include_path, context)	Membaca file ke dalam sebuah array.

<code>file_exists(path)</code>	Mengecek keberadaan file.
<code>file_get_contents(path, include_path, context, awal, panjang_maksimal)</code>	Membaca file ke dalam sebuah string.
<code>file_put_contents(file, data, mode, context)</code>	Menuliskan string ke dalam sebuah file.
<code>fileatime(nama_file)</code>	Menghasilkan waktu terakhir file di akses.
<code>filectime(nama_file)</code>	Menghasilkan waktu terakhir file diubah.
<code>filegroup(nama_file)</code>	Menghasilkan ID grup sebuah file.
<code>filemtime(nama_file)</code>	Menghasilkan waktu terakhir file dimodifikasi.
<code>fileowner(nama_file)</code>	Menghasilkan ID user sebuah file.
<code>fileperms(nama_file)</code>	Menghasilkan permission sebuah file.
<code>filesize(nama_file)</code>	Menghasilkan ukuran sebuah file.
<code>filetype(nama_file)</code>	Menghasilkan tipe sebuah file.
<code>flock(file, lock, block)</code>	Mengunci dan membebaskan file. Lock dapat diisi LOCK_SH, LOCK_EX, LOCK_UN, LOCK_NB.
<code>fopen(nama_file, mode, include_path, context)</code>	Membuka file atau membuat file jika file yang dibuka tidak ada.
<code>fputs(file, string, panjang)</code>	Menuliskan sesuatu pada file yang dibuka.
<code>fread(file, panjang)</code>	Membaca dari file yang dibuka.
<code>fseek(file offset, whence)</code>	Melakukan pencarian pada file yang dibuka. Whence dapat diisi SEEK_SET, SEEK_CUR, SEEK_END.
<code>fstat(file)</code>	Menghasilkan informasi tentang file yang dibuka.
<code>ftell(file)</code>	Menghasilkan posisi saat ini dari file yang dibuka.
<code>fwrite(file, string, length)</code>	Menuliskan sesuatu pada file yang dibuka.
<code>is_dir(file)</code>	Mengecek apakah sebuah file merupakan direktori.
<code>is_executable(file)</code>	Mengecek apakah sebuah file executable.
<code>is_file(file)</code>	Mengecek apakah sebuah file merupakan file biasa.
<code>is_link(file)</code>	Mengecek apakah sebuah file merupakan link.
<code>is_readable(file)</code>	Mengecek apakah sebuah file readable.
<code>is_uploaded_file(file)</code>	Mengecek apakah sebuah file telah ter-upload.
<code>is_writable(file)</code>	Mengecek apakah sebuah file writeable.
<code>mkdir(path, mode, recursive, context)</code>	Membuat direktori.
<code>move_uploaded_file(file, lokasi_baru)</code>	Memindahkan file yang ter-upload ke sebuah direktori.
<code>pathinfo(path, option)</code>	Menghasilkan informasi sebuah path.
<code>realpath(path)</code>	Menghasilkan absolute pathname.
<code>realpath_cache_get()</code>	Menghasilkan entri cache sebuah realpath.

<code>realpath_cache_size()</code>	Menghasilkan ukuran cache sebuah realpath.
<code>rename(nama_lama, nama_baru, context)</code>	Mengubah nama file atau direktori.
<code>rmdir(direktori, context)</code>	Menghapus direktori yang kosong.
<code>tmpnam(direktori, prefix)</code>	Membuat nama unik temporary file.
<code>tmpfile()</code>	Membuat nama unik temporary file.
<code>unmask(mask)</code>	Mengubah permission sebuah file.
<code>unlink(nama_file, context)</code>	Menghapus sfile.

Argumen dicetak tebal menandakan wajib diisi. Contoh penggunaan dari beberapa function di atas dapat diperhatikan pada skrip berikut:

File 5_3_7_function_file.php

```
<?php
$myfile = fopen("hallo.txt", "w");
$txt = "Hello, \n";
fwrite($myfile, $txt);
$txt = "Saya sedang belajar PHP\n";
fwrite($myfile, $txt);
fclose($myfile);
?>
```

Jika file di atas dijalankan, akan menghasilkan file dengan nama hallo.txt dengan isi seperti berikut:



Gambar 5.19 Membuat file dengan PHP

5.4 Variabel Superglobal

Variabel superglobal yaitu variabel khusus bawaan PHP yang dapat diakses dari mana saja baik di luar function maupun di dalam function. Ada beberapa variabel superglobal yang terdapat pada PHP, yaitu sebagai berikut:

5.4.1 Variabel \$GLOBAL

\$GLOBAL digunakan untuk mengakses variabel di mana saja di dalam skrip PHP. PHP akan menyimpan variabel pada array dan dapat dipanggil dengan skrip \$GLOBAL[nama_variabel].

File 5_4_1_superglobal_global.php

```
<?php  
$x = 75;  
$y = 25;  
  
function penambahan() {  
    $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];  
}  
  
penambahan();  
echo $z;  
?>
```

Skrip di atas akan menghasilkan output angka 100.

5.4.2 Variabel \$_SERVER

\$_SERVER menyediakan informasi tentang header, path dan lokasi skrip. Variabel superglobal ini menyediakan banyak elemen, diantaranya sebagai berikut:

Skrip	Keterangan
\$_SERVER['PHP_SELF']	Menghasilkan nama file PHP yang berisi skrip tersebut.
\$_SERVER['GATEWAY_INTERFACE']	Menghasilkan versi Common Gateway Interface (CGI) dari server.
\$_SERVER['SERVER_ADDR']	Menghasilkan IP address dari host server.
\$_SERVER['SERVER_NAME']	Menghasilkan nama dari host server.
\$_SERVER['SERVER_SOFTWARE']	Menghasilkan nama aplikasi server, misalnya: Apache/2.2.4
\$_SERVER['SERVER_PROTOCOL']	Menghasilkan nama dan revisi dari Information Protocol, misalnya: HTTP/1.1.
\$_SERVER['REQUEST_METHOD']	Menghasilkan request method yang digunakan pada halaman apakah GET atau POST.
\$_SERVER['REQUEST_TIME']	Menghasilkan timestamp mulai request.

<code>\$_SERVER['QUERY_STRING']</code>	Menghasilkan query string jika halaman diakses dengan query string.
<code>\$_SERVER['HTTP_ACCEPT']</code>	Menghasilkan Accept header dari request.
<code>\$_SERVER['HTTP_ACCEPT_CHARSET']</code>	Menghasilkan Accept_Charset header dari request.
<code>\$_SERVER['HTTP_HOST']</code>	Menghasilkan host header dari request.
<code>\$_SERVER['HTTP_REFERER']</code>	Menghasilkan URL lengkap dari halaman yang sedang dibuka.
<code>\$_SERVER['HTTPS']</code>	Menghasilkan informasi apakah skrip melalui secure HTTP protocol.
<code>\$_SERVER['REMOTE_ADDR']</code>	Menghasilkan IP address yang digunakan user untuk membuka halaman web.
<code>\$_SERVER['REMOTE_HOST']</code>	Menghasilkan nama host dari user yang membuka halaman web.
<code>\$_SERVER['REMOTE_PORT']</code>	Menghasilkan port yang digunakan oleh mesin user untuk berkomunikasi dengan web server.
<code>\$_SERVER['SCRIPT_FILENAME']</code>	Menghasilkan path absolut dari skrip yang sedang dijalankan.
<code>\$_SERVER['SERVER_PORT']</code>	Menghasilkan port yang digunakan oleh web server.
<code>\$_SERVER['SERVER_SIGNATURE']</code>	Menghasilkan versi server dan virtual host name.
<code>\$_SERVER['PATH_TRANSLATED']</code>	Menghasilkan system base path dari skrip yang sedang berjalan.
<code>\$_SERVER['SCRIPT_NAME']</code>	Menghasilkan path dari skrip yang sedang berjalan.
<code>\$_SERVER['SCRIPT_URI']</code>	Menghasilkan URI dari halaman yang sedang dibuka.

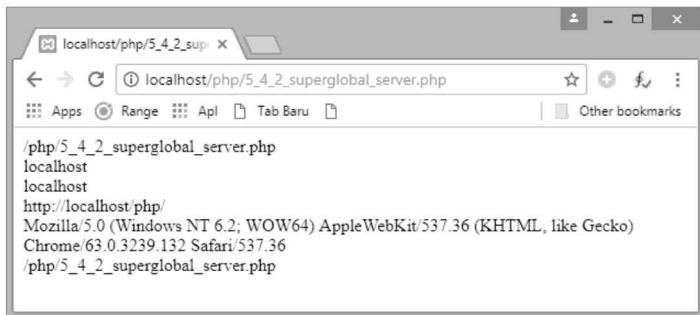
Berikut contoh penggunaanya:

File 5_4_2_superglobal_server.php

```
<?php
echo $_SERVER['PHP_SELF'];
echo "<br>";
echo $_SERVER['SERVER_NAME'];
echo "<br>";
echo $_SERVER['HTTP_HOST'];
echo "<br>";
echo $_SERVER['HTTP_REFERER'];
echo "<br>";
echo $_SERVER['HTTP_USER_AGENT'];
echo "<br>";
echo $_SERVER['SCRIPT_NAME'];
```

?>

Hasilnya sebagai berikut:



Gambar 5.20 Penggunaan variabel `$_SERVER`

Selain dari 2 variabel superglobal di atas, masih ada beberapa lagi, namun akan dibahas secara terpisah. Yang perlu diketahui, setiap variabel superglobal pasti ditulis dengan huruf kapital semua, dan biasanya diawali dengan underscore.

5.5 Session

Sama halnya dengan variabel, session juga digunakan untuk menyimpan suatu nilai. Bedanya jika session telah dibuat dalam suatu file, session dapat diakses oleh file lain selama session belum dihapus atau browser belum ditutup. Session biasanya digunakan untuk pembuatan login user. Dengan session, dapat dicek apakah user sudah login atau belum dengan mengecek session yang dibuat saat login.

Session dibuat menggunakan superglobal variabel `$_SESSION[]`. Sebelum menggunakan session, harus menuliskan function `session_start()`. Sedangkan untuk menghapus session, dapat dilakukan dengan fungsi `session_destroy()`. Jika menutup browser, maka session akan otomatis dihapus.

Berikut contoh skrip untuk membuat session:

File 5_5_session.php

```
<?php  
session_start();
```

```

$_SESSION['username'] = "Administrator";
$_SESSION['password'] = "Rahasia";

echo "Session telah dibuat,
<a href='5_5_session_cek.php'> Cek Session </a>";
?>

```

Jika link Cek Session di atas diklik, akan menuju file lain yang menampilkan data session. Walaupun dalam file yang berbeda, nilai session dapat terbaca, berbeda dengan variabel yang hanya bisa digunakan dalam file yang sama. Berikut skrip file yang menggunakan session:

File 5_5_session_cek.php

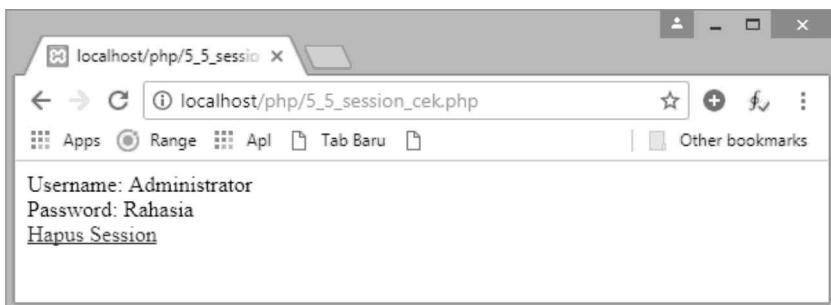
```

<?php
session_start();

if (!empty($_SESSION['username']) and
!empty($_SESSION['password']))
{
    echo "Username: ". $_SESSION['username'];
    echo "<br> Password: ". $_SESSION['password'];
    echo "<br> <a href='5_5_session_hapus.php'> Hapus Session
</a>";
}
else
{
    echo "Session telah dihapus";
    echo "<br> <a href='5_5_session.php'>Buat Session</a>";
}
?>

```

Jika sebelumnya telah membuka file 5_5_session.php, artinya session telah dibuat, maka hasilnya sebagai berikut:



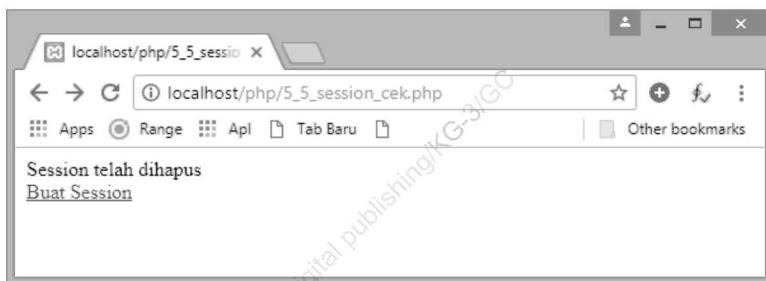
Gambar 5.21 Mengecek dan menampilkan session

Jika link Hapus Session diklik, akan menuju file yang menghapus session. Berikut isi skrip file yang menghapus session:

File 5_5_session_hapus.php

```
<?php  
session_start();  
session_destroy();  
echo "Session telah dihapus,";  
echo "<a href='5_5_session_cek.php'> Cek Session </a>";  
?>
```

Dengan menjalankan skrip di atas, maka session akan dihapus. Coba cek lagi session dengan mengklik link Cek Session, maka tidak lagi menampilkan seperti sebelumnya. Berikut tampilan file 5_5_session_cek.php jika session telah dihapus:



Gambar 5.22 Hasil menghapus session

5.6 Cookie

Hampir mirip dengan session, namun data cookie tidak akan terhapus ketika menutup browser, tetapi akan terhapus setelah batas waktu yang telah ditentukan. Cara membuat cookie yaitu dengan menggunakan function setcookie() dengan format sebagai berikut:

```
setcookie(nama_cookie, nilai, batas_akhir).
```

Batas akhir cookie dihitung dengan satuan detik. Untuk menampilkan cookie menggunakan superglobal variabel `$_COOKIE[nama_cookie]`.

Sebagai contoh, buat 2 buah file yaitu 5_6_cookie.php digunakan untuk membuat cookie dan 5_6_cookie_cek.php digunakan untuk mengecek data cookie.

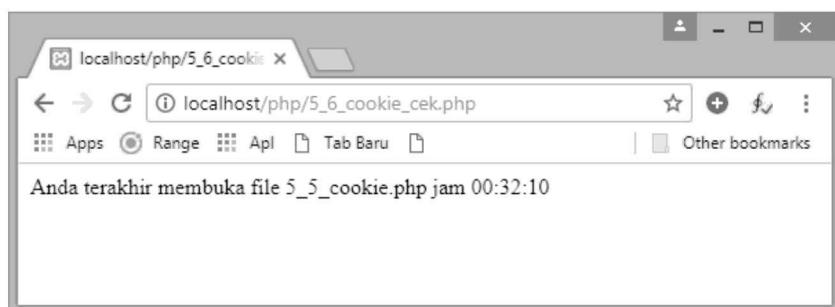
File 5_6_cookie.php

```
<?php  
$batas = time() + 30;  
  
setcookie("kunjunganTerakhir", date('H:i:s'), $batas);  
  
echo "COOKIE telah dibuat, buka file 5_5_cookie_cek.php  
sebelum dan setelah 30 detik";  
?>
```

File 5_6_cookie_cek.php

```
<?php  
if(isset($_COOKIE['kunjunganTerakhir']))  
{  
    echo "Anda terakhir membuka file 5_5_cookie.php  
    jam ". $_COOKIE['kunjunganTerakhir'];  
}  
else  
{  
    echo "Anda terakhir membuka file 5_5_cookie.php  
    lebih dari 30 detik yang lalu";  
}  
?>
```

Buka file 5_6_cookie.php untuk membuat cookie, lalu buka file 5_6_cookie_cek.php sebelum 30 detik. Coba buka lagi setelah 30 detik, perbedaanya sebagai berikut:



Gambar 5.23 Menampilkan cookie sebelum 30 detik



Gambar 5.24 Menampilkan cookie setelah lewat 30 detik

5.7 Penanganan Form

Salah satu fungsi utama PHP adalah memungkinkan adanya interaksi antara aplikasi web dan user. Media untuk melakukan interaksi ini salah satunya melalui form. Untuk memproses data yang dikirim oleh FORM ada beberapa superglobal variabel yang digunakan.

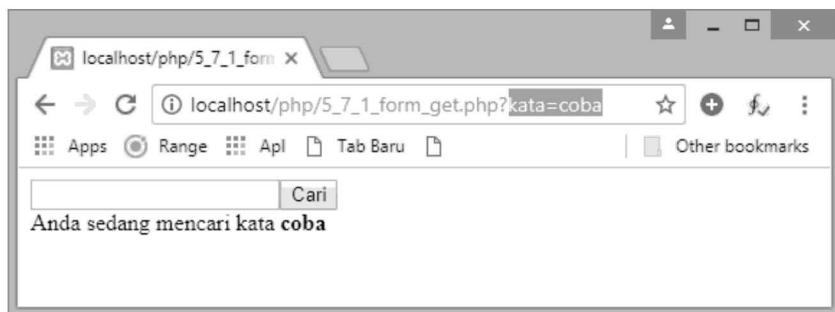
5.7.1 Variabel \$_GET

Variabel `$_GET` menyimpan array data yang dikirim oleh form dengan method GET melalui URL. Data yang dikirim akan tampak pada URL sehingga tidak disarankan untuk mengirim data yang sensitif seperti password. Contoh penggunaanya sebagai berikut:

File 5_7_1_form_get.php

```
<!DOCTYPE HTML>
<html>
  <head> </head>
  <body>
    <form method="get" action="">
      <input type="text" name="kata"><input type="submit" value="Cari">
    </form>
    <?php
      if(isset($_GET['kata'])) {
        echo "Anda sedang mencari kata <b>".$_GET[kata]</b>";
      }
    ?>
  </body>
</html>
```

Masukkan sebuah kata, lalu klik tombol Cari. Perhatikan pada URL, akan tertulis data yang dikirim seperti gambar berikut:



Gambar 5.25 Penggunaan variabel \$_GET

5.7.2 Variabel \$_POST

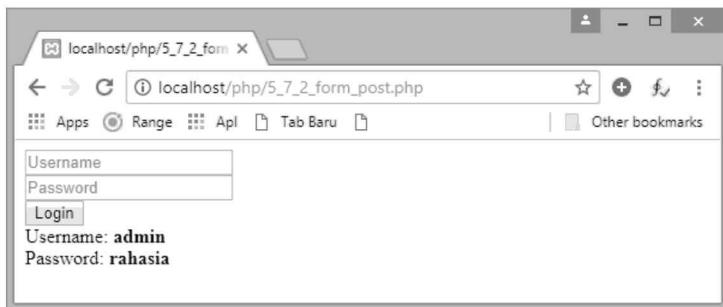
Variabel \$_POST menyimpan array data yang dikirim oleh form dengan method POST melalui HTTP POST. Data yang dikirim tidak akan tampak pada URL sehingga lebih aman digunakan dibanding \$_GET. Contoh penggunaanya sebagai berikut:

File 5_7_2_form_post.php

```
<!DOCTYPE HTML>
<html>
  <head> </head>
  <body>
    <form method="post" action="">
      <input type="text" placeholder="Username"
name="username"><br>
      <input type="password" placeholder="Password"
name="password"><br>
      <input type="submit" value="Login">
    </form>

    <?php
      if($_SERVER[ "REQUEST_METHOD" ] == "POST"){
        echo "Username: <b>$_POST[username]</b><br>";
        echo "Password: <b>$_POST[password]</b>";}
    ?>
  </body>
</html>
```

Masukkan username dan password kemudian klik Login. Perhatikan pada URL, tidak akan tampil data yang dikirim, seperti gambar berikut:



Gambar 5.26 Penggunaan variabel POST

5.7.3 Variabel \$_FILES

Variabel `$_FILES` hanya menyimpan data yang dikirim dari input dengan tipe file. Biasa digunakan untuk proses upload File, agar file dapat ter-upload, pada tag form harus ditambah atribut `enctype="multipart/form-data"`. Contohnya pada skrip berikut:

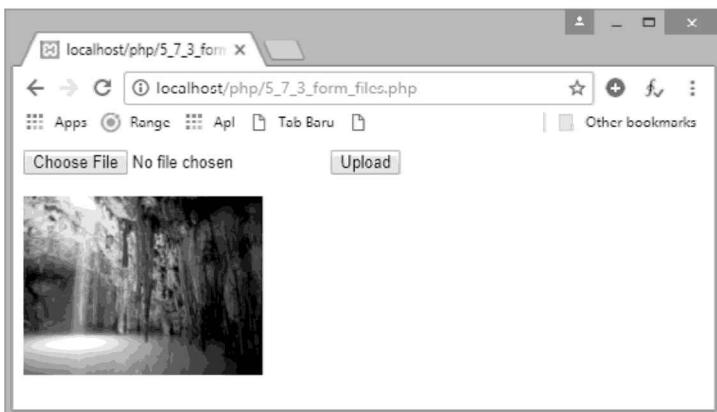
File 5_7_3_form_files.php

```
<!DOCTYPE HTML>
<html>
  <head> </head>
  <body>
    <form method="post" action="" enctype="multipart/form-
data">
      <input type="file" name="foto">
      <input type="submit" value="Upload">
    </form>

    <?php
      if($_SERVER["REQUEST_METHOD"] == "POST"){
        $namafoto = $_FILES['foto']['name'];
        $lokasi_file = $_FILES['foto']['tmp_name'];

        if($namafoto != ""){
          move_uploaded_file($lokasi_file, "file/".$namafoto);
          echo "<br><img src='file/$namafoto' width='200'>";
        }
      }
    ?>
  </body>
</html>
```

Coba upload sebuah foto lalu klik tombol Upload. Hasilnya akan seperti gambar berikut:



Gambar 5.27 Penggunaan variabel `$_FILES`

5.8 Penanganan Database

Penanganan database merupakan tujuan utama dalam penggunaan PHP. Dengan database, data yang diproses oleh PHP dapat disimpan dan dapat digunakan setiap saat. Database yang paling umum digunakan dalam pemrograman PHP yaitu MySQL. Berikut ini beberapa function yang dapat digunakan dalam pemrosesan database.

Function	Keterangan
<code>mysqli_affected_row(koneksi)</code>	Menghasilkan jumlah baris yang terpengaruh oleh operasi MySQL sebelumnya.
<code>mysqli_autocommit(koneksi, mode)</code>	Mengeset modifikasi database langsung diproses atau tidak.
<code>mysqli_change_user(koneksi, username, password, dbname)</code>	Mengubah user pada koneksi database.
<code>mysqli_character_set_name(koneksi)</code>	Menghasilkan karakter default pada koneksi database.
<code>mysqli_close(koneksi)</code>	Menutup koneksi database yang sedang terbuka.
<code>mysqli_commit(koneksi)</code>	Memproses transaksi database saat ini.
<code>mysqli_connect_errno()</code>	Menghasilkan kode error dari koneksi terakhir.
<code>mysqli_connect_error()</code>	Menghasilkan deskripsi error dari koneksi terakhir.

<code>mysqli_connect(host, username, password, dbname, port, socket)</code>	Membuka koneksi baru ke MySQL server.
<code>mysqli_data_seek(result, offset)</code>	Menuju baris tertentu dari data yang dihasilkan.
<code>mysqli_debug(message)</code>	Debug operasi database.
<code>mysqli_dump_debug_info(koneksi)</code>	Menghasilkan info debug ke log.
<code>mysqli_errno(koneksi)</code>	Menghasilkan kode error terakhir dari beberapa operasi database.
<code>mysqli_error_list(koneksi)</code>	Menghasilkan daftar error dari beberapa operasi database.
<code>mysqli_error(koneksi)</code>	Menghasilkan deskripsi error terakhir dari beberapa operasi database.
<code>mysqli_fetch_all(result, result_type)</code>	Menghasilkan array asosiatif, array numerik atau keduanya dari semua hasil operasi database. <code>Result_type</code> dapat diisi <code>MYSQLI_ASSOC</code> , <code>MYSQLI_NUM</code> , <code>MYSQLI_BOTH</code> .
<code>mysqli_fetch_array(result, result_type)</code>	Menghasilkan array asosiatif, array numerik atau keduanya dari hasil operasi database.
<code>mysqli_fetch_assoc(result)</code>	Menghasilkan array asosiatif dari hasil operasi database.
<code>mysqli_fetch_field_direct(result, field_number)</code>	Menghasilkan meta-data sebuah field dari hasil operasi database, sebagai sebuah objek.
<code>mysqli_fetch_field(result)</code>	Menghasilkan field berikutnya dari hasil operasi database, sebagai sebuah objek.
<code>mysqli_fetch_fields(result)</code>	Menghasilkan array objek yang mempresentasikan field dari hasil operasi database.
<code>mysqli_fetch_lengths(result)</code>	Menghasilkan jumlah kolom dari baris saat ini.
<code>mysqli_fetch_object(result, classname, parameter)</code>	Menghasilkan baris dari hasil operasi database sebagai sebuah objek.
<code>mysqli_fetch_row(result)</code>	Menghasilkan array numerik dari satu baris hasil operasi database.
<code>mysqli_field_count(koneksi)</code>	Menghasilkan jumlah kolom dari query sebelumnya.
<code>mysqli_field_seek(result, field_number)</code>	Menuju field tertentu.
<code>mysqli_field_tell(result)</code>	Menghasilkan posisi field saat ini.
<code>mysqli_free_result(result)</code>	Mengosongkan data dari hasil operasi database.
<code>mysqli_get_client_info(koneksi)</code>	Menghasilkan versi client library MySQL.

<code>mysqli_get_connection_stats(koneksi)</code>	Menghasilkan statistik mengenai koneksi client.
<code>mysqli_get_host_info(koneksi)</code>	Menghasilkan nama host server MySQL dan tipe koneksi.
<code>mysqli_get_proto_info(koneksi)</code>	Menghasilkan versi protokol MySQL
<code>mysqli_get_server_info(koneksi)</code>	Menghasilkan versi server MySQL.
<code>mysqli_get_server_version(koneksi)</code>	Menghasilkan versi server MySQL sebagai integer.
<code>mysqli_info(koneksi)</code>	Menghasilkan informasi tentang query yang dieksekusi baru-baru ini.
<code>mysqli_init()</code>	Menginisialisasi MySQLi dan menghasilkan resource untuk digunakan dengan <code>mysqli_real_connection()</code> .
<code>mysqli_insert_id(koneksi)</code>	Menghasilkan id yang dihasilkan otomatis dari query terakhir.
<code>mysqli_more_results(koneksi)</code>	Mengecek apakah ada hasil dari beberapa query.
<code>mysqli_multi_query(koneksi, query)</code>	Memproses beberapa query.
<code>mysqli_next_result(koneksi)</code>	Mempersiapkan hasil berikutnya dari <code>mysqli_multi_query()</code> .
<code>mysqli_num_fields(result)</code>	Menghasilkan jumlah field dari hasil operasi database.
<code>mysqli_num_rows(result)</code>	Menghasilkan jumlah baris dari hasil operasi database.
<code>mysqli_options(koneksi, option, value)</code>	Menambahkan pengaturan pada koneksi database.
<code>mysqli_ping(koneksi)</code>	Mengetes koneksi server atau mencoba koneksi ulang jika koneksi bermasalah.
<code>mysqli_query(koneksi, query, resultmode)</code>	Memproses query ke database. Resultmode dapat diisi <code>MYSQLI_USE_RESULT</code> , <code>MYSQLI_STORE_RESULT</code> .
<code>mysqli_real_connect(koneksi, host, username, password, dbname, port, socket, flag)</code>	Membuka koneksi baru ke MySQL server.
<code>mysqli_real_escape_string(koneksi, escapestring)</code>	Meloloskan karakter khusus pada string untuk digunakan pada perintah SQL.
<code>mysqli_select_db(koneksi, dbname)</code>	Mengubah database yang terkoneksi.

Untuk menggunakan function-function di atas, pastikan telah memiliki database terlebih dahulu yang akan digunakan. Gunakan saja database yang telah dibuat sebelumnya yaitu tabel pegawai. Berikut contoh penggunaanya:

File 5_8_database.php

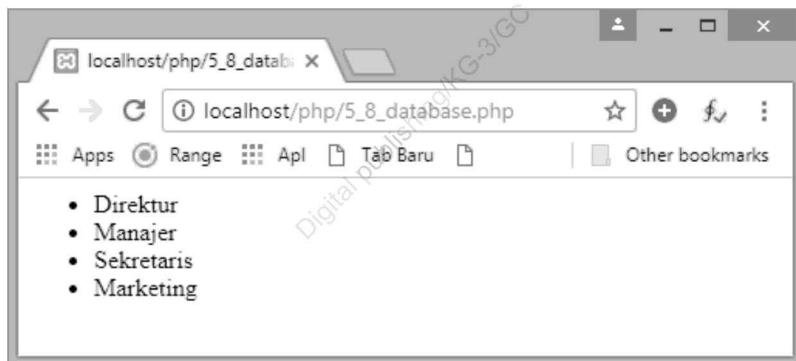
```
<?php
    $con = mysqli_connect("localhost", "root", "", "pegawai");
    if (mysqli_connect_errno()){
        echo "Koneksi gagal: " . mysqli_connect_error();
    }

    $sql = "SELECT * FROM jabatan";
    $result = mysqli_query($con, $sql);

    echo "<ul>";
    while($row = mysqli_fetch_array($result)){
        echo "<li>$row[nama_jabatan]</li>";
    }
    echo "</ul>";

    mysqli_close($con);
?>
```

Hasilnya dapat dilihat pada gambar berikut:



Gambar 5.28 Hasil operasi database

5.9 Mengirim Email

Untuk mencoba mengirim email pada localhost, dapat mengaktifkan Mercury pada Xampp Control Panel jika menggunakan Xampp. Cara membuat email pada mercury penulis sertakan pada file pendukung. Adapun aplikasi yang digunakan untuk menampilkan email dapat menggunakan Mozilla Thunderbird.

Function yang digunakan untuk mengirim email yaitu **mail()** yang memiliki 5 argumen dengan format sebagai berikut:

```
mail(tujuan, subjek, pesan, header, parameter)
```

Dari 5 argumen, hanya 3 argumen pertama yang wajib disertakan, sedangkan 2 argumen terakhir bersifat opsional. Adapun keterangan dari masing-masing argumen di atas yaitu:

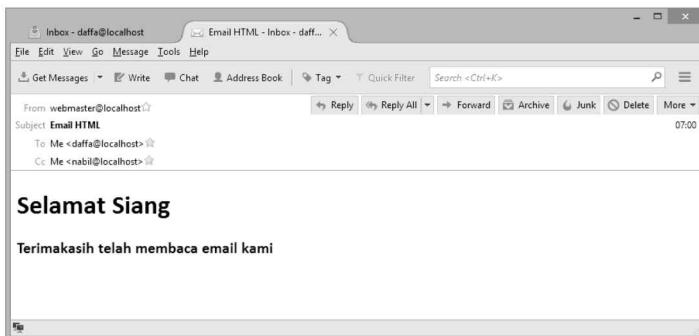
- **Tujuan**, diisi email yang akan dituju.
- **Subjek**, diisi subjek email, biasanya satu baris kalimat yang menjelaskan isi email.
- **Pesan**, diisi pesan yang dikirim baik berupa teks biasa maupun skrip HTML.
- **Header**, diisi tambahan header yang diperlukan seperti From, CC, BCC, dan sebagainya. Jika mengirim email dalam format HTML juga ada tambahan header yang ditulis pada argumen ini.
- **Parameter**, diisi tambahan parameter ke program pengiriman email.

Untuk pengiriman email sederhana, cukup diisi 3 argumen saja. Namun, untuk pengiriman email dengan format HTML harus ada header, seperti pada contoh berikut:

File 5_9_email.php

```
<?php  
$to = "daffa@local host";  
$subject = "Email HTML";  
  
$message = "<h1> Selamat Si ang</1>  
           <h3>Terimakasih telah membaca email kami </h3>";  
  
// Selalu set content-type ketika mengirim email HTML  
$headers = "MIME-Version: 1.0" . "\r\n";  
$headers .= "Content-type: text/html; charset=UTF-8" . "\r\n";  
  
$headers .= 'From: <webmaster@local host>' . "\r\n";  
$headers .= 'Cc: nabil@local host' . "\r\n";  
  
mail($to, $subject, $message, $headers);  
?>
```

Buka file pada browser, lalu lihat pada inbox apakah ada email yang masuk. Berikut tampilan pesan yang terkirim:



Gambar 5.29 Membaca pesan pada Mozilla Thunderbird

5.10 Praktik Membuat Aplikasi

Walaupun telah menguasai seluruh teori tentang PHP, tidak ada jaminan langsung bisa membuat aplikasi. Untuk itu, pada subbab ini akan dibahas praktik membuat aplikasi sederhana yang terdiri dari login, logout dan operasi CRUD. Dengan menguasai praktik ini, pembaca akan lebih mudah memahami proses pembuatan aplikasi dengan PHP.

Praktik ini menggunakan template yang telah dibuat pada pembahasan CSS, sehingga tidak perlu membuat desain dari awal. Copy saja folder template ke folder **xampp/htdocs**, dan ubah namanya menjadi **pegawai** sebagai folder aplikasi.

5.10.1 Membuat Koneksi ke Database

Database yang digunakan pada aplikasi ini adalah database **pegawai** yang telah dibuat sebelumnya, dimana di dalamnya terdapat tabel user, pegawai dan jabatan. Agar terkoneksi dengan database, buat folder **library** dan file **config.php** di dalamnya dengan skrip sebagai berikut:

File library/config.php

```
<?php  
$host = "localhost";  
$user = "root";  
$pass = "";  
$db = "pegawai";  
  
$con = mysqli_connect($host, $user, $pass, $db);
```

```
if (mysql i _connect_errno()) {
    echo "Koneksi gagal : " . mysql i _connect_error();
}
?>
```

Jalankan dengan mengetik **localhost/pegawai/library/config.php**. Jika tidak ada error, berarti koneksi berhasil.

5.10.2 Membuat File Login dan Logout

Untuk membuat halaman login, ubah file **login.html** menjadi file **login.php**, tanpa mengubah skrip di dalamnya. Jika diperhatikan, pada file tersebut terdapat baris skrip sebagai berikut:

```
<form method="post" action="cekl ogi n. php" >
```

Dari skrip di atas, dibutuhkan file **ceklogin.php** yang akan memproses data yang dikirim melalui form login. Buat file tersebut dengan skrip sebagai berikut:

File ceklogi n. php

```
<?php
session_start();
include "library/config.php";
$username = $_POST['username'];
$password = md5($_POST['password']);

$query = mysql i _query($con, "SELECT * FROM user WHERE
$username= '$username' AND password= '$password' ");
$data = mysql i _fetch_array($query);
$jumlah = mysql i _num_rows($query);

if($jumlah > 0){
    $_SESSION['username'] = $data['username'];
    $_SESSION['password'] = $data['password'];

    header ('location: index. php');
} else{
    echo "<p align='center'>Login Gagal </p>";
    echo "<meta http-equiv='refresh' content='2;
url=login. php' >";
}
?>
```

Inti dari skrip di atas yaitu mengecek apakah ada data pada tabel user yang memiliki username dan password sama dengan data yang dikirim. Untuk password menggunakan enkripsi md5. Jika data ditemukan, ditandai dengan jumlah data lebih dari 0, kemudian membuat session username dan password untuk digunakan sebagai penanda pada

halaman lain bahwa user sudah login. Selanjutnya, function **header()** akan mengarahkan ke halaman index.

Selanjutnya buat file logout.php sebagai berikut:

File logout.php

```
<?php
    session_start();
    session_destroy();
    echo "<p align='center'>Anda telah logout!</p>";
    echo "<meta http-equiv='refresh' content='2;
url=login.php'>";
?>
```

Inti dari file ini adalah skrip session_destroy() yang akan menghapus data session username dan password yang dibuat pada file ceklogin.php.

5.10.3 Membuat Halaman Dashboard

Untuk membuat halaman dashboard dibutuhkan 3 file yaitu file **index.php** berisi template yang digunakan oleh semua halaman, file **konten.php** yang berisi skrip untuk mengatur halaman mana yang akan ditampilkan dan file **dashboard.php** yang akan ditampilkan pada bagian konten.

Langsung saja ubah file **dashboard.html** menjadi **index.php**, lalu ubah skripnya menjadi seperti berikut:

File index.php

```
<?php
    session_start();
    ob_start();

    include "library/config.php";

    if(empty($_SESSION['username']) or
empty($_SESSION['password'])){
        echo "<p align='center'> Anda harus login terlebih dahulu!</p>";
        echo "<meta http-equiv='refresh' content='2;
url=login.php'>";
    } else{
        define('INDEX', true);
    }
?>
<!DOCTYPE HTML>
<html>
    <head>
        <title>Dashboard</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width,
initial-scale=1">
```

```

<link rel="stylesheet" href="css/style.css">
</head>
<body>
    <header>
        Apikasi Manajemen Pegawai
    </header>
    <div class="content">
        <aside>
            <ul class="menu">
                <li><a href="?hal=dashboard" class="aktiv">Dashboard</a>
            </li>
                <li><a href="?hal=pegawai">Data Pegawai</a> </li>
                <li><a href="?hal=jabatan">Data Jabatan</a> </li>
                <li><a href="logout.php">Keluuar</a> </li>
            </ul>
        </aside>
        <section class="main">
            <?php include "konten.php"; ?>
        </section>
    </div>
    <footer>
        Copyright © Rohi Abdulloh
    </footer>
</body>
</html>
<?php
}
?>
```

Pada bagian atas terdapat skrip untuk mengecek apakah terdapat session username dan password yang menandakan user sudah login. Jika tidak ada, ditampilkan pesan dan langsung diarahkan ke halaman login. Jika sudah login, baru ditampilkan halaman aplikasi. Sebelum skrip HTML, dibuat dulu konstanta dengan nama INDEX yang akan digunakan untuk memastikan halaman lain hanya dapat dibuka melalui file index.php. Isi tag **<section>** semuanya dihapus dan diganti dengan memanggil file **konten.php** yang akan mengatur file mana yang akan tampil pada bagian tersebut sesuai dengan menu yang diklik. Pada bagian bawah, jangan lupa menutup kurung kurawal sebagai akhir dari if().

Selanjutnya buat file **konten.php** dengan skrip sebagai berikut:

File konten.php

```

<?php
if(!defined('INDEX')) die("");
$halaman = array("dashboard",
    "pegawai", "pegawai_tambah", "pegawai_insert",
    "pegawai_edit", "pegawai_update", "pegawai_hapus",
    "jabatan", "jabatan_tambah", "jabatan_insert",
    "jabatan_edit", "jabatan_update", "jabatan_hapus");
if(isset($_GET['hal'])) $hal = $_GET['hal'];
?>
```

```
else $hal = "dashboard";  
foreach($halaman as $h){  
    if($hal == $h){  
        include "content/$h.php";  
        break;  
    }  
}  
?>
```

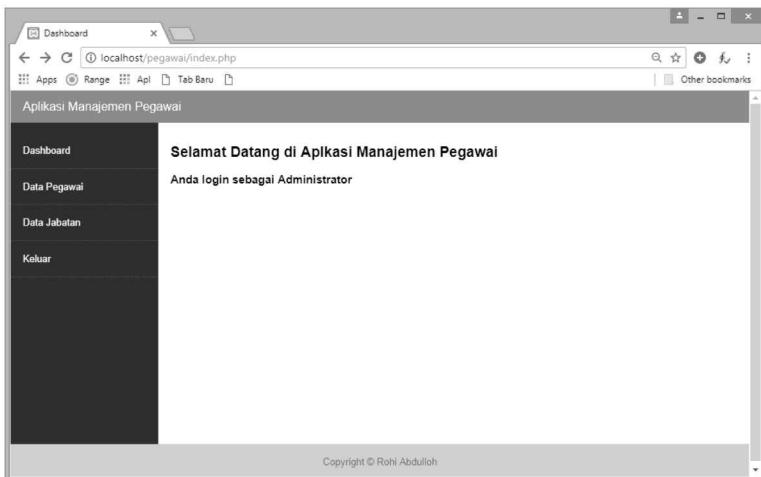
Pada awal skrip di cek apakah telah dibuat konstanta INDEX. Jika tidak mengenali konstanta tersebut, artinya file ini diakses secara langsung, maka ditampilkan saja halaman kosong. Pada baris berikutnya, variabel \$halaman menyimpan daftar nama file yang akan ditampilkan pada bagian konten dalam array. Selanjutnya mengecek data hal pada URL yang dikirim pada setiap link menu. Jika ada, variabel \$hal diisi data tersebut, sedangkan jika kosong variabel \$hal diisi **dashboard** sebagai halaman default. Terakhir, mengecek setiap array halaman, jika sama dengan variabel \$hal, artinya nama halaman tersebut sedang dibuka, maka panggil file tersebut dengan **include** dari folder content.

Selanjutnya buat folder **content** dan file **dashboard.php** di dalamnya dengan skrip sebagai berikut:

File content/dashboard.php

```
<?php  
if(!defined('INDEX')) die("");  
?  
<h1>Selamat Datang di Aplikasi Manajemen Pegawai </h1>  
<h3>Anda login sebagai Administrator</h3>
```

Pada skrip di atas juga dicek konstanta INDEX untuk memastikan file ini dibuka melalui menu, tidak diakses secara langsung. Coba login ke aplikasi, jika semua skrip benar akan tampil sebagai berikut:



Gambar 5.30 Halaman dashboard aplikasi

5.10.4 Membuat Halaman Jabatan

File pada halaman jabatan terdiri dari 6 file yang memiliki fungsi masing-masing sebagai berikut:

- **jabatan.php**, merupakan halaman awal yang akan tampil ketika menu Data Jabatan diklik, akan menampilkan tabel yang berisi daftar jabatan yang telah dimasukkan.
- **jabatan_tambah.php**, menampilkan formulir untuk menambah data jabatan yang akan tampil ketika tombol Tambah diklik.
- **jabatan_insert.php**, berfungsi memproses data yang dikirim dari formulir tambah jabatan untuk dimasukkan ke database.
- **jabatan_edit.php**, menampilkan formulir untuk edit data jabatan yang akan tampil ketika tombol Edit diklik.
- **jabatan_update.php**, berfungsi memproses data yang dikirim dari formulir edit data jabatan untuk dimasukkan ke database.
- **jabatan_hapus.php**, berfungsi untuk menghapus data jabatan dari database, yang akan diakses ketika tombol Hapus diklik.

Semua file di atas diletakkan pada folder content. Adapun isi skrip dari masing-masing file sebagai berikut:

File content/jabatan.php

```
<?php
    if(!defined('INDEX')) die("");
?>

<h2 class="judul">Data Jabatan</h2>
<a class="tombol" href="?hal=jabatan_tambah">Tambah</a>



| No          | Nama Jabatan                  | Aksi                                                                                                                                                                                                 |
|-------------|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <?= \$no ?> | <?= \$data['nama_jabatan'] ?> | <a class="tombol edit" href="?hal=jabatan_edit&amp;id=&lt;?=\$data['id_jabatan'] ?&gt;">Edit</a> <a class="tombol hapus" href="?hal=jabatan_hapus&amp;id=&lt;?=\$data['id_jabatan'] ?&gt;">Hapus</a> |


<?php
}
?>
</tbody>
</table>
```

Skrip di atas diambil dari konten file table.html pada template, ditambahkan skrip SQL untuk mengambil data dari database. Setiap baris data yang dihasilkan dikonversi ke array dengan mysqli_fetch_array() dan dilakukan perulangan hingga akhir baris dengan while. Tanda <?= pada skrip di atas sama fungsinya dengan <?= echo. Pada tombol edit dan hapus pastikan setelah &id= tidak ada spasi.

File content/jabatan_tambah.php

```
<?php
    if(!defined('INDEX')) die("");
?>
```

```

?>

<h2 class="judul" >Tambah Jabatan</h2>
<form method="post" action="?hal=jabatan_insert">

<div class="form-group">
    <label for="nama">Nama</label>
    <div class="input"><input type="text" id="nama" name="nama" /></div>
</div>

<div class="form-group">
    <input type="submit" value="Simpan" class="tombol_simpan" />
    <input type="reset" value="Batal" class="tombol_reset" />
</div>

</form>

```

Skrip di atas diambil dari konten file form.html disesuaikan dengan kebutuhan input data. Atribut action tag <form> diisi link halaman yang memproses data.

File content/jabatan_insert.php

```

<?php
if (!defined('INDEX')) die("");
$query = mysqli_query($con, "INSERT INTO jabatan SET
    nama_jabatan = '$_POST[nama]'");
if ($query) {
    echo "Data berhasil disimpan!";
    echo "<meta http-equiv='refresh' content='1';
url=?hal=jabatan' />";
} else{
    echo "Tidak dapat menyimpan data!<br>";
    echo mysqli_error();
}
?>

```

Skrip utama yaitu yang dicetak tebal, berfungsi untuk memasukkan data ke database. Skrip berikutnya untuk menampilkan pesan ketika skrip berhasil diproses atau ketika gagal diproses.

File content/jabatan_edit.php

```

<?php
if (!defined('INDEX')) die("");
$query = mysqli_query($con, "SELECT * FROM jabatan WHERE
    id_jabatan='$_GET[id]'");
$data = mysqli_fetch_array($query);
?>

<h2 class="judul" >Edit Jabatan</h2>
<form method="post" action="?hal=jabatan_update">

```

```

<input type="hidden" name="id" value=<?= $data['id_jabatan'] ?>>
<div v-cl ass="form-group">
    <label for="nama">Nama</label>
    <div v-cl ass="input">
        <input type="text" id="nama" name="nama" value=<?= $data['nama_jabatan'] ?>>
    </div v>
</div v>
<div v-cl ass="form-group">
    <input type="submit" value="Simpan" cl ass="tombol_simpan">
    <input type="reset" value="Batal" cl ass="tombol_reset">
</div v>
</form>

```

Skrip di atas pada intinya sama dengan file jabatan_tambah.php. Pada bagian atas ditambah skrip untuk mengambil data dari database sesuai id yang dikirim dari tombol Edit, untuk ditampilkan pada form. Nilai atribut action disesuaikan dengan nama file yang memproses. Pada awal form, ditambahkan tag `<input>` dengan type hidden, untuk menampung data id jabatan yang digunakan untuk menentukan data mana yang akan diedit pada file jabatan_update.php. Pada setiap input ditambahkan atribut value untuk menampilkan data yang telah diinput ke database.

File content/jabatan_update.php

```

<?php
if(!defined('INDEX')) die("");
$query = mysqli_query($con, "UPDATE jabatan SET
    nama_jabatan = '$_POST[nama]' 
    WHERE id_jabatan='$_POST[id]' ");
if($query){
    echo "Data berhasil disimpan!";
    echo "<meta http-equiv='refresh' content='1';
url=?hal=jabatan' ";
} else{
    echo "Tidak dapat menyimpan data!<br>";
    echo mysqli_error();
}
?>

```

Skrip di atas hampir sama dengan skrip file jabatan_insert.php. Bedanya pada skrip yang dicetak tebal, yang berfungsi untuk mengubah data yang telah ada pada database.

File content/jabatan_hapus.php

```

<?php
if(!defined('INDEX')) die("");

```

```

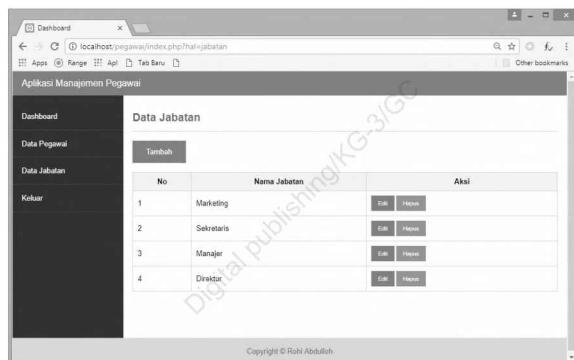
$query = mysql_query($con, "DELETE FROM jabatan WHERE
id_jabatan='$_GET[id]'");

if($query){
    echo "Data berhasil dihapus!";
    echo "<meta http-equiv='refresh' content='1;
url=?hal=jabatan'>";
} else{
    echo "Tidak dapat menyimpan data! <br>";
    echo mysql_error();
}
?>

```

Tak jauh berbeda dengan file `jabatan_update.php` maupun `jabatan_insert.php`. Hanya disesuaikan skrip SQL-nya.

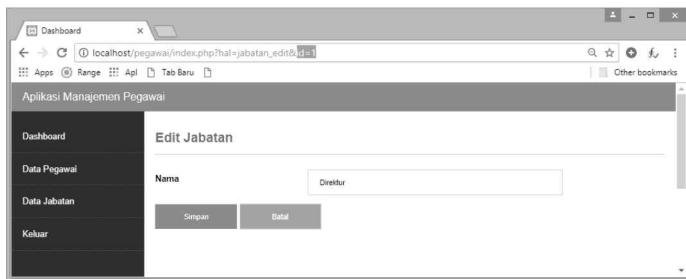
Coba buka menu Data Jabatan untuk melihat hasilnya. Coba juga klik tombol Tambah, Edit dan Hapus untuk melihat fungsi file lain. Jika skrip benar, hasilnya sebagai berikut:



Gambar 5.31 Tampilan halaman data jabatan



Gambar 5.32 Tampilan halaman tambah jabatan



Gambar 5.33 Tampilan halaman edit jabatan

5.10.5 Membuat Halaman Pegawai

Halaman pegawai berisi skrip yang lebih kompleks. Untuk memahami skrip pada halaman ini, pastikan telah memahami konsep CRUD pada halaman jabatan. Sama seperti halaman jabatan, halaman pegawai juga terdiri dari 6 file yang fungsinya juga sama.

File content/pegawai.php

```
<?php
    if (!defined('INDEX')) die("");
?>

<h2 class="judul">Data Pegawai </h2>
<a class="tombol" href="?hal=pegawai_tambah">Tambah</a>

<table class="table">
    <thead>
        <tr>
            <th>No</th>
            <th>Foto</th>
            <th>Nama</th>
            <th>Jenis Kelamin</th>
            <th>Tanggal Lahir</th>
            <th>Jabatan</th>
            <th>Keterangan</th>
            <th>Aksi</th>
        </tr>
    </thead>
    <tbody>
        <?php
            $query = mysqli_query($con, "SELECT * FROM pegawai LEFT JOIN
                jabatan ON pegawai.id_jabatan=jabatan.id_jabatan ORDER BY
                pegawai.id_pegawai DESC");
            $no = 0;
            while ($data = mysqli_fetch_array($query)) {
                $no++;
?>
            <tr>
                <td><?= $no ?></td>
```

```

<td><i mg src="images/<?= $data['foto'] ?>" wi dt h="100"></td>
    <td><?= $data['nama_pegawai'] ?></td>
    <td><?= $data['jenis_kelamin'] ?></td>
    <td><?= $data['tgl_lahir'] ?></td>
    <td><?= $data['namajabatan'] ?></td>
    <td>
        <a class="tombol_edit"
            href="?hal=pegawai_edit&id=<?= $data['id_pegawai'] ?>"> Edi t
        </a>
        <a class="tombol_hapus"
            href="?hal=pegawai_hapus&id=<?= $data['id_pegawai'] ?>&foto=<?=
            $data['foto'] ?>"> Hapus </a>
        </td>
    </tr>
<?php
}
?>
</tbody>
</table>

```

Yang perlu diperhatikan pada skrip di atas yaitu penggunaan LEFT JOIN pada skrip SQL untuk mengambil data dari tabel pegawai dan jabatan sekaligus. Pada tombol Hapus juga mengirimkan data foto melalui URL yang berisi nama file foto yang akan dihapus.

File content/pegawai_tambah.php

```

<?php
if (!defined('INDEX')) die("");
?>

<h2 class="judul" >Tambah Pegawai </h2>
<form method="post" action="?hal=pegawai_insert"
enctype="multipart/form-data">

<div class="form-group">
    <label for="foto">Foto</label>
    <div><input type="file" id="foto" name="foto" /></div>
</div>

<div class="form-group">
    <label for="nama">Nama</label>
    <div><input type="text" id="nama" name="nama" /></div>
</div>

<div class="form-group">
    <label for="jk">Jenis Kelamin</label>
    <div><input type="radio" id="jk" name="jk" value="L" /> Laki-laki
        <input type="radio" id="jk" name="jk" value="P" /> Perempuan
    </div>
</div>

<div class="form-group">
    <label for="tanggal">Tanggal</label>

```

```

<div class="input"><input type="date" id="tanggal" name="tanggal"></div>
</div>

<div class="form-group">
    <label for="jabatan">Jabatan</label>
    <div class="input">
        <select id="jabatan" name="jabatan">
            <option value="">- Pilih Jabatan-</option>
    </select>
</div>

<?php
    $query_jabatan = mysqli_query($con, "SELECT * FROM jabatan");
    while($j = mysqli_fetch_array($query_jabatan)){
        echo "<option value='".$j[id_jabatan]'>" . $j[nama_jabatan] . "</option>";
    }
?>

</div>
</div>
</div>

<div class="form-group">
    <label for="keterangan">Keterangan</label>
    <div class="input"><textarea style="width: 100%" rows="5" id="keterangan" name="keterangan"></textarea></div>
</div>

<div class="form-group">
    <input type="submit" value="Simpan" class="tombol_simpan">
    <input type="reset" value="Batal" class="tombol_reset">
</div>
</form>

```

Perhatikan penambahan atribut **enctype** pada tag **<form>** karena berisi input dengan tipe file untuk upload foto. Perhatikan juga pada input jabatan, data jabatan diambil dari tabel jabatan pada database.

File content/pegawai_insert.php

```

<?php
if(!defined('INDEX')) die("");
$foto = $_FILES['foto']['name'];
$lokasi = $_FILES['foto']['tmp_name'];
$tipefile = $_FILES['foto']['type'];
$ukuranfile = $_FILES['foto']['size'];

$error = "";
if($foto == ""){
    $query = mysqli_query($con, "INSERT INTO pegawai SET
        nama_pegawai = '".$_POST['nama'],
        jenis_kelamin = '".$_POST['jenis_kelamin'],
        tanggal_lahir = '".$_POST['tanggal'],
        id_jabatan = '".$_POST['jabatan'],
        keterangan = '".$_POST['keterangan']
    ");
} else{

```

```

if($tipefile != "image/jpeg" and $tipefile != "image/jpg"
and $tipefile != "image/png"){
    $error = "Tipe file tidak di dukung!";
} elseif($ukuranfile >= 1000000){
    echo $ukuranfile;
    $error = "Ukuran file terlalu besar (lebih dari 1MB)!";
} else{
    move_uploaded_file($lokasi, "images/".$foto);
    $query = mysqli_query($con, "INSERT INTO pegawai SET
        foto = '$foto',
        nama_pegawai = '$POST[nama]',
        jenis_kelamin = '$POST[jk]',
        tanggal_lahir = '$POST[tanggal]',
        id_jabatan = '$POST[jabatan]',
        keterangan = '$POST[keterangan]'");
}
}

if($error != ""){
    echo $error;
    echo "<meta http-equiv='refresh' content='2';
url=?hal=pegawai_tambah>";
} elseif($query){
    echo "Data berhasil disimpan!";
    echo "<meta http-equiv='refresh' content='1';
url=?hal=pegawai'";
} else{
    echo "Tidak dapat menyimpan data!<br>";
    echo mysqli_error();
}
?>

```

Pada skrip di atas terdapat validasi tipe file dan ukuran file. Perhatikan saja skrip yang dicetak tebal.

File content/pegawai_edit.php

```

<?php
if(!defined('INDEX')) die("");
$query = mysqli_query($con, "SELECT * FROM pegawai WHERE
id_pegawai='".$_GET['id']."' ");
$data = mysqli_fetch_array($query);
?>

<h2 class="j udul ">Tambah Pegawai </h2>
<form method="post" action="?hal=pegawai_update"
enctype="multipart/form-data">
    <input type="hidden" name="id" value="?=&$data['id_pegawai']"
?>>

<div class="form-group">
    <label for="foto">Foto</label>
    <div class="input">
        <input type="file" id="foto" name="foto">
        
    </div>
</div>

```

```

<di v cl ass="form-group">
    <l abel for="nama">Nama</l abel>
    <di v cl ass="i nput">
        <i nput type="text" id="nama" name="nama" val ue="?=&
$data['nama_pegawai'] ?>">
    </di v>
</di v>

<di v cl ass="form-group">
    <l abel for="j k">Jeni s Kel ami n</l abel>
    <?php
        if($data['jeni s_kel ami n'] == "L"){
            $l = " checked";
            $p = "";
        }else{
            $l = "";
            $p = "checked";
        }
    ?>
    <i nput type="radi o" id="j k" name="j k" val ue="L" <?= $l ?>>
Laki - laki
    <i nput type="radi o" id="j k" name="j k" val ue="P" <?= $p ?>>
Perempuan
</di v>

<di v cl ass="form-group">
    <l abel for="t anggal">Tanggal </l abel>
    <di v cl ass="i nput">
        <i nput type="date" id="t anggal" name="t anggal"
val ue="?=& $data['tgl _lahi r'] ?>">
    </di v>
</di v>

<di v cl ass="form-group">
    <l abel for="j abatan">Jabat an</l abel>
    <di v cl ass="i nput">
        <sel ect id="j abatan" name="j abatan">
            <option val ue=""> - Pilih Jabat an- </option>
        <?php
            $queryjabatan = mysqli_query($con, "SELECT * FROM
j abatan");
            while($j = mysqli_fetch_array($queryjabatan)){
                echo "<option value='".$j['id_j abatan']."' ";
                if($j['id_j abatan'] == $data['id_j abatan']) echo
" selected";
                echo ">$j[nama_j abatan]</option>";
            }
        ?>
    </sel ect>
    </di v>
</di v>

<di v cl ass="form-group">
    <l abel for="keterangan">Keterangan</l abel>
    <di v cl ass="i nput">
        <t ext area style="width: 100%" rows="5" id="keterangan"
name="keterangan"><?=$data['keterangan'] ?></t ext area>
    </di v>
</di v>

<di v cl ass="form-group">

```

```

<input type="submit" value="Simpan" class="tombol_simpan">
<input type="reset" value="Batal" class="tombol_reset">
</div>
</form>

```

Perhatikan cara menampilkan data pada input jenis kelamin dan jabatan. Pada input jenis kelamin, jika data sesuai dengan database ditampilkan atribut **checked**. Sedangkan pada input jabatan, jika data sesuai dengan database ditambahkan atribut **selected**.

File content/pegawai_update.php

```

<?php
if(!defined('INDEX')) die("");
$foto = $_FILES['foto']['name'];
$lokasi = $_FILES['foto']['tmp_name'];
$tipefile = $_FILES['foto']['type'];
$ukuranfile = $_FILES['foto']['size'];
$error = "";
if($foto == ""){
    $query = mysqli_query($con, "UPDATE pegawai SET
        nama_pegawai = '$POST[nama]',
        jenis_kelamin = '$POST[jk]',
        tanggal_lahir = '$POST[tanggal]',
        id_jabatan = '$POST[jabatan]',
        keterangan = '$POST[keterangan]'
        WHERE id_pegawai = '$POST[id]'");
} else{
    if($tipefile != "image/jpeg" and $tipefile != "image/png"){
        $error = "Tipe file tidak di dukung!";
    } elseif($ukuranfile >= 1000000){
        echo $ukuranfile;
        $error = "Ukuran file terlalu besar (lebih dari 1MB)!";
    } else{
        $query = mysqli_query($con, "SELECT * FROM pegawai
        WHERE id_pegawai = '$POST[id]'");
        $data = mysqli_fetch_array($query);
        if(file_exists("images/$data[foto]")){
            unlink("images/$data[foto]");
            move_uploaded_file($lokasi, "images/".$foto);
        }
        $query = mysqli_query($con, "UPDATE pegawai SET
            foto = '$foto',
            nama_pegawai = '$POST[nama]',
            jenis_kelamin = '$POST[jk]',
            tanggal_lahir = '$POST[tanggal]',
            id_jabatan = '$POST[jabatan]',
            keterangan = '$POST[keterangan]'
            WHERE id_pegawai = '$POST[id]'");
    }
}
if($error != ""){
    echo $error;
}

```

```

        echo "<meta http-equiv='refresh' content='2';
url=?hal=pegawai_edit&id=$_POST[id]>";
    } else if ($query) {
        echo "Data berhasil disimpan!";
        echo "<meta http-equiv='refresh' content='1';
url=?hal=pegawai'>";
    } else{
        echo "Tidak dapat menyimpan data!<br>";
        echo mysql_error();
    }
?>

```

Perhatikan pada skrip yang dicetak tebal, jika user mengisi data foto, tidak hanya meng-upload foto baru tetapi juga menghapus foto yang lama.

File content/pegawai_hapus.php

```

<?php
if(!defined('INDEX')) die("");
if(file_exists("images/".$_GET[foto]))
unlink("images/".$_GET[foto]);
$query = mysql_query($con, "DELETE FROM pegawai WHERE
id_pegawai='".$GET[id']."'");

if($query) {
    echo "Data berhasil dihapus!";
    echo "<meta http-equiv='refresh' content='1;
url=?hal=pegawai'>";
} else{
    echo "Tidak dapat menyimpan data!<br>";
    echo mysql_error();
}
?>

```

Sama seperti file pegawai_update.php, skrip di atas juga terdapat skrip untuk menghapus file foto. Hanya saja nama foto diambil dari data foto pada URL.

Jika semua skrip benar, hasilnya seperti berikut:

No	Foto	Nama	Jenis Kelamin	Tanggal Lahir	Jabatan	Keterangan	Aksi
1		Kaesang Ayu	P	2002-08-16	Sekretaris		<button>Edit</button> <button>Hapus</button>
2		Afaf Firdaus	L	1994-04-13	Manajer		<button>Edit</button> <button>Hapus</button>
3		Daffa Shidqi	L	1998-07-09	Direktur	Pimpinan perusahaan	<button>Edit</button> <button>Hapus</button>

Gambar 5.34 Tampilan halaman data pegawai

Tambah Pegawai

Foto

Nama

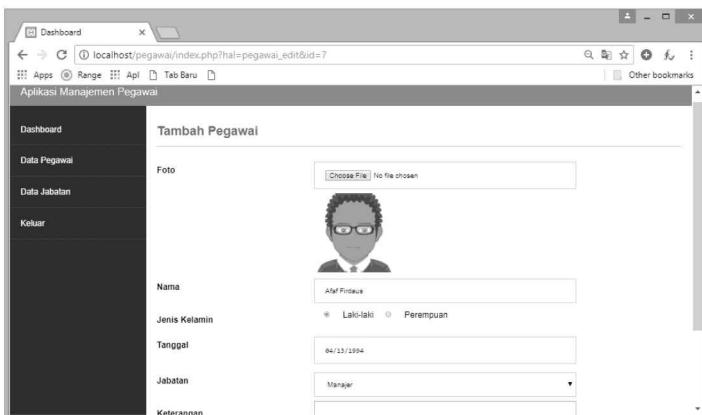
Jenis Kelamin Laki-laki Perempuan

Tanggal

Jabatan

Keterangan

Gambar 5.35 Tampilan halaman tambah pegawai



Gambar 5.36 Tampilan halaman edit pegawai

5.11 Bonus File Contoh

Untuk mendukung latihan agar lebih menguasai PHP, penulis sertakan bonus file contoh PHP yang terletak pada folder PHP/BONUS. Folder tersebut berisi berbagai contoh function buatan yang sering digunakan dalam pembuatan aplikasi PHP, seperti function tanggal indonesia, format rupiah, angka terbilang, dan sebagainya.

6 JAVASCRIPT

6.1 Mengenal JavaScript

6.1.1 Pengertian

JavaScript merupakan bahasa pemrograman web yang pemrosesanya dilakukan di sisi client. Karena berjalan di sisi client, JavaScript dapat dijalankan hanya dengan menggunakan browser. Berbeda dengan PHP yang bekerja di sisi server, untuk menjalankan skrip JavaScript tidak memerlukan refresh pada browser.

JavaScript biasanya dijalankan ketika ada event tertentu yang terjadi pada halaman web. Baik event yang dilakukan oleh user, maupun event yang terjadi karena adanya perubahan pada halaman website.

6.1.2 Meletakkan Skrip JavaScript

Ada berbagai cara meletakkan kode JavaScript pada dokumen HTML. Pembaca bebas menggunakan cara yang paling sesuai dengan selera, namun biasanya disesuaikan dengan kebutuhan atau keadaan. Berikut berbagai cara meletakkan kode Javasacript.

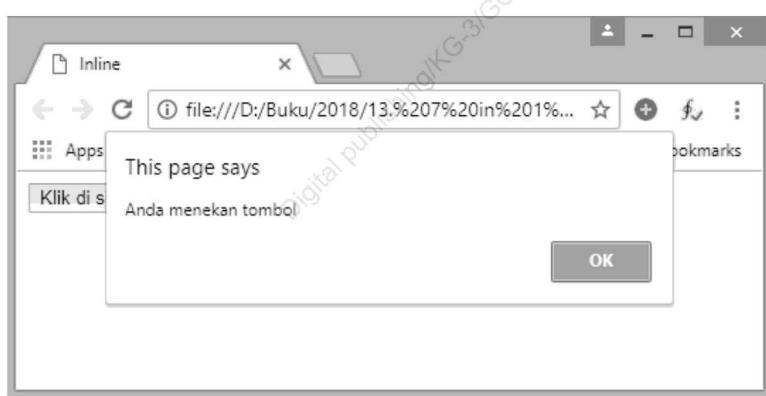
Menggunakan Event Handler (Inline)

Cara ini biasanya digunakan jika kode JavaScript hanya sedikit atau sekedar memanggil function. Cara meletakkan kode JavaScript melalui cara ini yaitu dengan meletakkan kode setelah attribut event pada tag HTML. Pada contoh skrip berikut, akan menampilkan kotak pesan ketika dokumen baru dibuka dan ketika klik tombol.

File 6_1_2_inline.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Inline</title>
  </head>
  <body onload="alert('Selamat Datang!')">
    <button onclick="alert('Anda menekan tombol')">Klik di
  si ni </button>
  </body>
</html>
```

Hasilnya seperti gambar berikut:



Gambar 6.1 Hasil kode Inline JavaScript

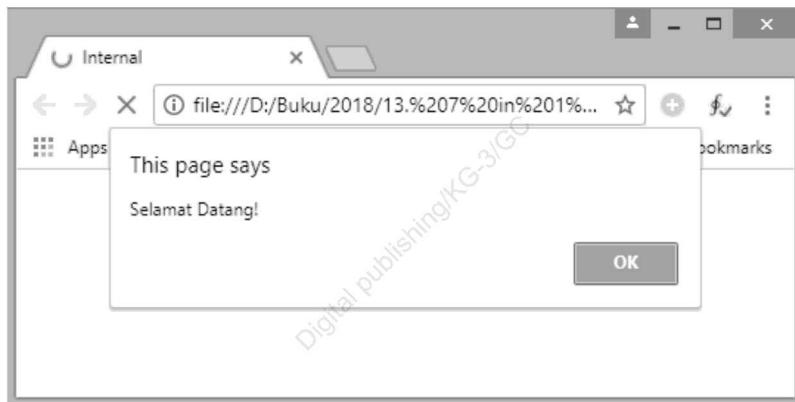
Menggunakan Tag <script> (Internal)

Caranya yaitu dengan meletakkan kode JavaScript di dalam tag `<script>...</script>`. Tag ini dapat diletakkan di dalam tag `<head>` maupun `<body>`. Cara ini biasanya digunakan jika kode JavaScript terdiri dari beberapa baris tetapi tidak terlalu panjang. Biasanya menggunakan attribut `type="text/javascript"` untuk membedakan dengan skrip dari bahasa lain tetapi tidak wajib.

File 6_1_2_internal.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Internal</title>
    <script type="text/javascript">
      alert("Selamat Datang!");
    </script>
  </head>
  <body>
    <script>
      document.write("Latihan JavaScript");
    </script>
  </body>
</html>
```

Hasilnya seperti gambar berikut:



Gambar 6.2 Hasil kode Internal JavaScript

Menggunakan tag <script src=""> (Eksternal)

Caranya yaitu dengan memisahkan kode JavaScript dengan kode HTML. Kode JavaScript disimpan dengan file tersendiri dengan format .js. File tersebut kemudian dipanggil dengan tag <script src=""></script> pada bagian <head>. Cara ini biasanya digunakan jika kode JavaScript sangat panjang. Pada skrip berikut kode JavaScript disimpan dengan nama file **6_1_2_skrip.js** dan dipanggil melalui file **6_1_2_eksternal.html**.

File 6_1_2_skrip.js

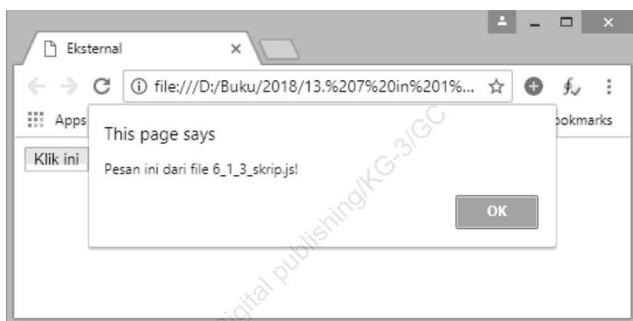
```
function hello() {
```

```
    alert("Pesan ini dari file 6_1_2_skrip.js!");
}
```

File 6_1_2_eksternal.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Eksternal </title>
    <script src="6_1_2_skrip.js"></script>
  </head>
  <body>
    <button onclick="hallo()">Klik ini </button>
  </body>
</html>
```

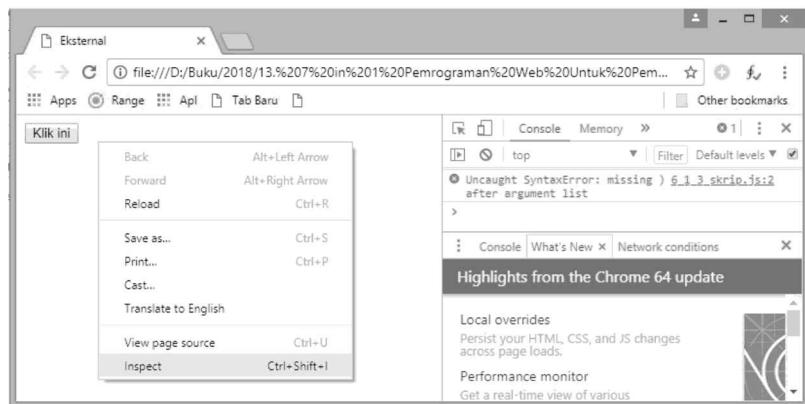
Hasilnya seperti gambar berikut:



Gambar 6.3 Hasil kode External JavaScript

6.1.3 Mengatasi Error pada JavaScript

Dalam menuliskan program pasti sering mengalami error pada program. Error pada JavaScript tidak akan menampilkan pesan apapun pada browser seperti PHP, jadi akan sedikit menyulitkan bagi programmer. Tapi, tidak perlu khawatir, karena beberapa browser menyediakan tool untuk mengatasi error pada JavaScript. Seperti pada browser Google Chrome disediakan menu Developer Tool. Biasanya pesan error pada program JavaScript akan tampil pada tab Console. Fitur ini dapat diakses dengan cara klik kanan lalu pilih Inspect (inspeksi) atau tekan Ctrl+Shift+I.



Gambar 6.4 Mengakses Console Log

Pada gambar di atas, tampak pesan error berwarna merah pada bagian kanan browser.

6.1.4 Menampilkan Hasil Program

Ada berbagai cara menampilkan hasil program yang dibuat dengan JavaScript baik untuk tujuan utama dibuatnya program maupun untuk menampilkan pesan kesalahan ketika terjadi kesalahan pada program.

Menampilkan Hasil Program dengan alert()

Cara ini digunakan yaitu dengan menampilkan hasil program melalui kotak pesan. Caranya dengan meletakkan teks atau variabel di dalam tanda kurung fungsi **alert()**. Pada beberapa contoh sebelumnya juga sudah ditunjukkan beberapa contoh menampilkan hasil program melalui **alert()**.

Menampilkan hasil program dengan write()

Cara ini dilakukan dengan menuliskan teks atau variabel di dalam tanda kurung skrip **document.write()**. Hasil program yang ditampilkan dengan cara ini akan langsung tampil pada halaman web.

Menampilkan hasil program di dalam tag HTML

Cara ini dilakukan dengan membuat tag container terlebih dahulu yang diberi atribut id tertentu. Kemudian mendeteksi tag tersebut dengan method **getElementById()** dimana id dituliskan di dalam kurung skrip tersebut. Untuk menampilkan hasil program, tuliskan teks atau variabel setelah properti **innerHTML** yang dipisahkan dengan tanda sama dengan. Penggunaan cara ini harus disertai dengan penggunaan event **onload**.

Menampilkan hasil program dengan console log

Cara ini akan menampilkan hasil program pada tag console pada menu Development Tool. Caranya yaitu dengan menggunakan skrip **console.log()**. Data atau variabel yang akan ditampilkan ditulis di dalam kurung. Cara ini biasanya hanya untuk menampilkan informasi selama proses pembuatan program.

Berikut ini contoh berbagai cara menampilkan data seperti yang telah disebutkan di atas:

```
File 6_1_4_menampilkan_data.html
<!DOCTYPE HTML>
<html>
  <head>
    <script>
      window.onload = function(){
        alert('Selamat Datang');
        document.getElementById('judul').innerHTML =
        "Mudahnya Belajar JavaScript";
        console.log('Skrip sukses di eksekusi !');
      }
    </script>
  </head>
  <body>
    <h2 id="judul"></h2>
    <script>
      document.write('Dengan semangat belajar, aku pasti
      bisa.');
    </script>
  </body>
</html>
```

Hasil dari skrip di atas seperti gambar berikut:



Gambar 6.5 Berbagai cara menampilkan data

6.1.5 Komentar

Komentar pada pemrograman merupakan teks yang tidak akan dieksekusi sebagai program. Komentar biasanya untuk memberi keterangan pada skrip program. Cara menuliskan komentar pada JavaScript yaitu dengan meletakkan tanda // sebelum komentar jika komentar hanya satu baris. Jika komentar lebih dari satu baris, maka komentar ditulis di antara tanda /* dan */. Berikut ini contoh penulisan komentar:

File 6_1_5_komentar.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <script>
      //Ini komentar satu baris

      /*
        Ini komentar lebih dari satu baris
        komentar ini tidak akan di eksekusi sebagai program
      */
    </script>
  </head>
  <body>
  </body>
</html>
```

6.2 Dasar-Dasar JavaScript

6.2.1 Variabel

Variabel digunakan untuk menyimpan data berupa teks, angka, dan sebagainya yang nilainya dapat diubah-ubah, sehingga memudahkan dalam mengakses data tersebut. Dalam penulisan variabel, ada aturan yang harus dipatuhi agar tidak menghasilkan error pada program. Dalam JavaScript, aturan penulisan variabel sebagai berikut:

- Jangan menggunakan kata-kata yang menjadi kode JavaScript atau dikenal reserved keyword. Contohnya: break, boolean, else, and, var, true, enum, switch, this, throw, long, new, dan sebagainya.
- Variabel tidak boleh diawali dengan angka atau simbol. Variabel harus diawali dengan huruf atau underscore. Contoh: panjang, _tinggi, dan sebagainya. Sedangkan yang tidak dibolehkan seperti: 7ingi, 4las, @las, dan sebagainya.
- Variabel dalam JavaScript bersifat **case-sensitive** artinya huruf kecil dan huruf besar dibedakan. Jadi, antara tinggi dan Tinggi adalah dua variabel yang berbeda.

Variabel pada JavaScript dideklarasikan dengan **var**. Contohnya sebagai berikut:

```
<script>
  var panjang;
  var tinggi ;
</script>
```

Selain dengan cara di atas, dapat juga mendeklarasikan beberapa variabel sekaligus dengan satu **var**. Misalnya sebagai berikut:

```
<script>
  var panjang, tinggi ;
</script>
```

Variabel juga dapat dideklarasikan dengan langsung mendefinisikan nilai awal. Misalnya sebagai berikut:

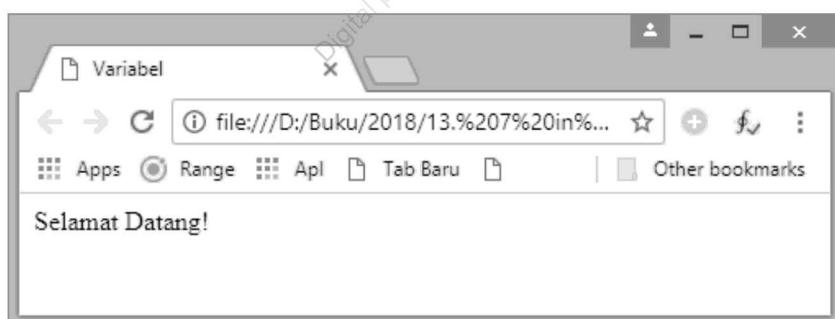
```
<script>
  var panjang = 200;
  var tinggi = 150;
  var kata = "JavaScript menyenangkan";
</script>
```

Pada JavaScript terdapat batasan dalam mengakses variabel atau yang disebut *scope variable*. JavaScript mengenal variabel **global** dan variabel **local**. Variabel **global** merupakan variabel yang dapat dipakai dimana saja baik di luar function maupun di dalam function. Sedangkan variabel **local** yaitu variabel yang hanya dapat digunakan di dalam function. Variabel yang dideklarasikan di luar function akan menjadi variabel **global** dan variabel yang dideklarasikan di dalam function akan menjadi variabel **local**. Contohnya sebagai berikut:

File 6_2_1_variabel.html

```
<!DOCTYPE HTML>
<html>
  <head></head>
  <body>
    <script>
      var a = "Selamat Datang!";
      function selamat () {
        var b = "Selamat pagi!";
        document.write(a);
      }
      selamat();
      document.write(b);
    </script>
  </body>
</html>
```

Hasilnya seperti gambar berikut:



Gambar 6.6 Penggunaan variabel

Pada gambar di atas, variabel **a** berhasil ditampilkan walaupun dideklarasikan di luar function dan digunakan di dalam function. Ini namanya variabel **global**. Sedangkan variabel **b** tidak tampil karena dideklarasikan di dalam function dan digunakan di luar function. Ini namanya variabel **local**. Variabel local hanya dapat digunakan di dalam

function dimana variabel tersebut dideklarasikan dan tidak dapat digunakan di luar function atau di function lain.

6.2.2 Tipe Data

Setiap bahasa pemrograman pasti mengenal yang namanya tipe data. Tipe data pada JavaScript sedikit berbeda dengan bahasa pemrograman lain. Tipe data pada JavaScript bersifat seolah-olah sebagai objek. Sehingga setiap variabel pada JavaScript akan langsung memiliki method.

JavaScript mengenal 3 tipe data dasar yaitu sebagai berikut:

- **Numbers** : contoh 125, 20.50, dsb.
- **Strings** : contoh "Hallo!", "200", dsb.
- **Boolean** : hanya ada dua nilai yaitu True dan False.

Selain tiga tipe di atas, JavaScript juga mengenal tipe data **null** dan **undefined**. **Null** merupakan nilai dari sebuah variabel yang tidak diberi nilai saat didefinisikan. Sedangkan **undefined** merupakan hasil yang didapat dari proses berikut:

- Nilai dari pemanggilan variabel yang belum dideklarasikan.
- Nilai dari pemanggilan elemen array yang tidak ada.
- Nilai dari pemanggilan property objek yang tidak ada.
- Nilai dari pemanggilan function yang tidak mengembalikan nilai.
- Nilai dari parameter function yang tidak memiliki argumen.

Dapat disimpulkan bahwa perbedaan antara null dan undefined adalah bahwa null biasanya diperoleh dari kondisi normal yang sudah direncanakan, sedangkan undefined biasanya diperoleh dari kesalahan program yang tidak direncanakan.

6.2.3 Operator

Operator merupakan sebuah simbol atau kata yang digunakan untuk melakukan operasi terhadap satu atau lebih data atau variabel sehingga menghasilkan data baru. Variabel atau data yang dioperasikan disebut

dengan **operand**. Javascript mengenal beberapa jenis operator yaitu sebagai berikut:

Operator Aritmetika

Operator aritmetika merupakan operator yang digunakan untuk melakukan perhitungan data numerik. Daftar operator aritmatika dan cara penggunaanya dapat dilihat pada tabel berikut:

Operator	Keterangan
+	Untuk penjumlahan, misal A = 10 dan B = 4, maka: A + B menghasilkan 14
-	Untuk pengurangan, misal A = 10 dan B = 4, maka: A - B menghasilkan 6
*	Untuk perkalian, misal A = 10 dan B = 4, maka: A * B menghasilkan 40
/	Untuk pembagian, misal A = 10 dan B = 4, maka: A / B menghasilkan 2.5
%	Untuk mendapatkan sisa pembagian (modulus), misal A = 10 dan B = 4, maka : A % B menghasilkan 2
++	Berarti ditambah satu, misal A = 10, maka: A++ menghasilkan 11
--	Berarti dikurangi satu, misal A = 10, maka: A-- menghasilkan 9

Operator Perbandingan

Operator perbandingan digunakan untuk membandingkan dua variabel atau data. Hasil dari perbandingan tersebut akan menghasilkan nilai **true** atau **false**. Daftar operator perbandingan dan penjelasannya dapat dilihat pada table berikut:

Operator	Keterangan
==	Berarti sama dengan, misal A = 10 dan B = 4, maka: (A == B) menghasilkan false
!=	Berarti tidak sama dengan, misal A = 10 dan B = 4, maka:

	(A != B) menghasilkan true.
>	Berarti operand pertama lebih besar dari operand ke dua, misal A = 10 dan B = 4, maka: (A > B) menghasilkan true
<	Berarti operand pertama lebih kecil dari operand kedua, misal A = 10 dan B = 4, maka (A < B) menghasilkan false
>=	Berarti lebih besar atau sama dengan, misal A = 10 dan B = 4, maka: (A >= B) menghasilkan true dan (10 >= A) juga menghasilkan true
<=	Berarti lebih kecil atau sama dengan, misal A = 10 dan B = 4, maka: (A <= B) menghasilkan false, sedangkan (10 <= A) menghasilkan true

Operator Logika

Operator logika digunakan untuk mengoperasikan data boolean. Hasil dari perbandingan tersebut akan menghasilkan nilai **true** atau **false**. Daftar operator logika dan penjelasanya dapat dilihat pada table berikut:

Operator	Keterangan
&&	Menghasilkan nilai true jika kedua operand bernilai true (logika And). Misal A = true dan B = false, maka: (A && B) menghasilkan false
	Menghasilkan nilai true jika salah satu atau kedua operand bernilai true (logika Or). Misal A = true dan B = false, maka: (A B) menghasilkan true
!	Membalikkan nilai sebuah variabel (logika Not). Misal A = true, maka: (!A) menghasilkan false

Operator Bitwise

Operator bitwise akan membandingkan nilai biner dua bilangan. Misalnya 2 akan dikonversi ke biner menjadi 10 dan 3 akan dikonversi ke biner menjadi 11. Jika 2 dan 3 dibandingkan dengan operator bitwise, maka yang dibandingkan adalah nilai biner-nya. Hasil dari operasi berupa biner yang akan dikonversi lagi dan ditampilkan dalam bentuk desimal. Daftar operator bitwise dan penjelasanya dapat dilihat pada table berikut:

Operator	Keterangan
&	Menghasilkan angka 1 jika kedua angka yang dibandingkan 1 dan menghasilkan 0 jika salah satu atau kedua angka 0 (bitwise And). Misal, A = 2 (binernya 10) dan B = 3 (binernya 11), maka: (A & B) menghasilkan 2 Maksudnya: digit pertama 1 & 1 menghasilkan 1, digit kedua 0 & 1 menghasilkan 0. Jika digabungkan hasilnya 10 jika didesimalkan menjadi 2.
	Menghasilkan angka 1 jika salah satu atau kedua angka yang dibandingkan 1 dan menghasilkan 0 jika kedua angka 0 (bitwise Or). Misal, A = 2 (binernya 10) dan B = 3 (binernya 11), maka: (A B) menghasilkan 3
^	Menghasilkan angka 1 jika salah satu angka yang dibandingkan 1 dan menghasilkan 0 jika kedua angka 0 atau kedua angka 1 (bitwise Xor). Misal, A = 2 (binernya 10) dan B = 3 (binernya 11), maka: (A ^ B) menghasilkan 1
~	Membalikkan suatu angka yang 1 menjadi 0 dan yang 0 menjadi 1 (bitwise Not). Biasanya hasilnya lebih besar satu angka tapi bernilai negatif. Misal, A = 3 (binernya 11), maka: (~A) menghasilkan -4
<<	Menggeser ke kiri bit operand pertama sejumlah angka operand kedua dan diisi dengan 0 (Left Shift). Misal A = 4 (binernya 100), maka: (A<<2) menghasilkan 16 (binernya 10000)
>>	Menggeser ke kanan bit operand pertama sejumlah angka operand kedua (Right Shift). Misal A = 9 (binernya 1001), maka: (A>>1) menghasilkan 4 (binernya 100)
>>>	Hampir sama dengan >> hanya di sebelah kiri selalu 0.

Operator Assignment

Operator ini digunakan untuk memberikan nilai kepada suatu variabel menggunakan tanda sama dengan. Daftar operator assignment dan penjelasannya dapat dilihat pada tabel berikut:

Operator	Keterangan
=	Memberi nilai pada suatu variabel dengan nilai, variabel lain atau hasil perhitungan di sebelah kanan. Misal A = 3 dan B = 4, maka: $C = A + B$ menghasilkan nilai C = 3 + 4 yaitu 7
+=	Menambah nilai suatu variabel dengan nilai atau variabel lain. Misal A = 3 dan B = 4, maka: $A += B$ artinya nilai A ditambah nilai B, sehingga nilai A menjadi 7

<code>-=</code>	Mengurangi nilai suatu variabel dengan nilai atau variabel lain. Misal A = 3 dan B = 4, maka: B -= A artinya nilai B dikurangi nilai A, sehingga nilai B menjadi 1
<code>*=</code>	Mengalikan nilai suatu variabel dengan nilai atau variabel lain. Misal A = 3 dan B = 4, maka: B *= A artinya nilai B dikalikan nilai A, sehingga nilai B menjadi 12
<code>/=</code>	Membagikan nilai suatu variabel dengan nilai atau variabel lain. Misal A = 12 dan B = 4, maka: A /= B artinya nilai A dibagi nilai B, sehingga nilai A menjadi 3
<code>%=</code>	Mengambil sisa bagi suatu variabel dengan nilai atau variabel lain. Misal A = 12 dan B = 4, maka: A %= B artinya nilai sisa bagi A dengan B, sehingga nilai A menjadi 0

Operator Kondisi

Operator ini digunakan untuk memberikan nilai pada suatu variabel sesuai kondisi yang ditentukan. Jika kondisi **true** maka akan diberi nilai dengan nilai di sebelah kiri tanda titik dua, sedangkan jika kondisi **false** maka akan diberi nilai dengan nilai di sebelah kanan tanda titik dua. Kondisi diletakkan di dalam kurung diikuti dengan tanda ? yang memisahkan dengan nilai.

Operator	Keterangan
<code>? ... : ...</code>	Tanda tanya memisahkan kondisi dengan nilai, sedangkan tanda titik dua memisahkan nilai pertama dan nilai kedua. Nilai pertama dipakai jika kondisi true dan nilai kedua dipakai jika kondisi false.

Berikut contoh beberapa penggunaan operator:

File 6_2_3_operator.html

```
<!DOCTYPE HTML>
<html>
<head></head>
<body>
<script>
var A = 10, B = 4;
var C = A % B;
document.write("10 % 4 = " + C + '<br>');

A++;
document.write('Nilai A sekarang: ' + A + '<br>');

var D = A > B;
document.write('10 > 4 hasilnya: ' + D + '<br>');

```

```

var E = false, F = true;
var G = E && F;
document.write('false && true hasilnya: ' + G +
'<br>');

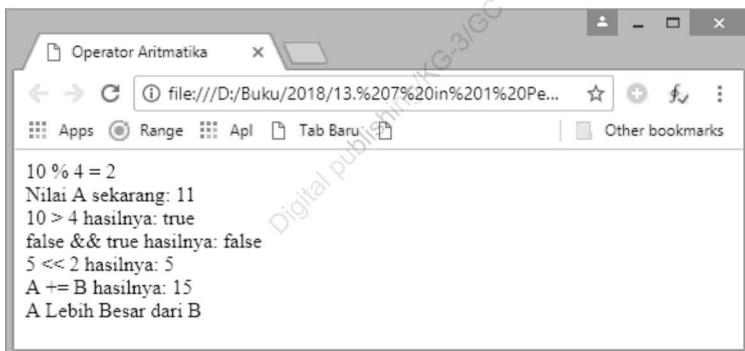
var H = 5, I = 2;
var J = 5 << 2;
document.write('5 << 2 hasilnya: ' + H + '<br>');

A += B;
document.write('A += B hasilnya: ' + A + '<br>');

var hasil = (A>B) ? "A Lebih Besar dari B" : "A Lebih
Kecil dari B";
document.write(hasil);
</script>
</body>
</html >

```

Pada skrip di atas, masing-masing jenis operator diwakili oleh satu operator. Untuk mencoba operator lain pada masing-masing jenis operator silakan bereksperimen sendiri. Hasil skrip di atas seperti gambar berikut:



Gambar 6.7 Hasil penggunaan operator

6.2.4 Statement Control

Statement control pada bahasa pemrograman digunakan untuk menjalankan suatu program sesuai dengan kondisi tertentu. Ada berbagai jenis statement control pada JavaScript yang dapat digunakan sesuai kebutuhan.

Percabangan dengan if

Percabangan ini digunakan untuk menjalankan suatu program jika kondisi terpenuhi (**true**) dan jika kondisi tidak terpenuhi tidak ada alternatif lain untuk dijalankan. Penulisan kondisi biasanya menggunakan variabel dengan tipe **boolean** atau menggunakan operator **logika** yang akan menghasilkan tipe data boolean.

Penggunaan **if** dapat ditulis dengan format sebagai berikut:

```
if (kondisi) pernyataan;
```

Jika pernyataan lebih dari satu baris, cara di atas tidak dapat digunakan, tapi menggunakan format berikut:

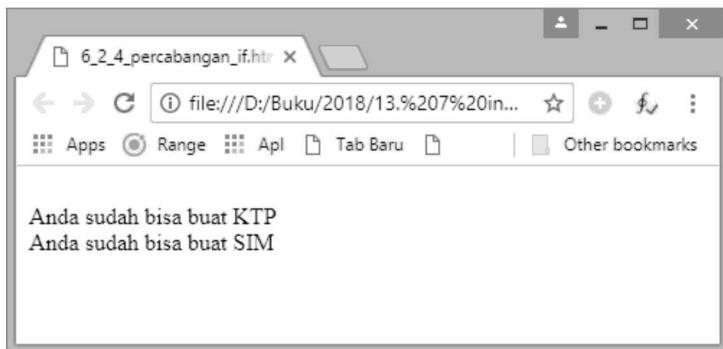
```
if (kondisi) {  
    pernyataan baris pertama;  
    pernyataan baris kedua;  
}
```

Berikut ini contoh percabangan menggunakan **if** pada JavaScript:

File 6_2_4_percabangan_if.html

```
<!DOCTYPE HTML>  
<html>  
  <head>  
    <script>  
      var umur = 20;  
      var dewasa;  
  
      if(umur > 17) dewasa = true;  
      if(umur > 20) document.write("Anda sudah boleh  
menikah");  
  
      if(dewasa){  
        document.write("<br>Anda sudah bisa buat KTP");  
        document.write("<br>Anda sudah bisa buat SIM");  
      }  
    </script>  
  </head>  
  <body>  
  </body>  
</html>
```

Hasil skrip di atas seperti gambar berikut:



Gambar 6.8 Hasil percabangan dengan if

Percabangan dengan if ... else ...

Percabangan ini digunakan untuk menjalankan program jika kondisi terpenuhi, dan jika kondisi tidak terpenuhi ada alternatif untuk dijalankan. Cara menggunakannya yaitu dengan format sebagai berikut:

```
if(kondisi) pernyataan jika terpenuhi;  
else pernyataan jika tidak terpenuhi;
```

Jika pernyataan lebih dari satu baris, maka formatnya menjadi seperti berikut:

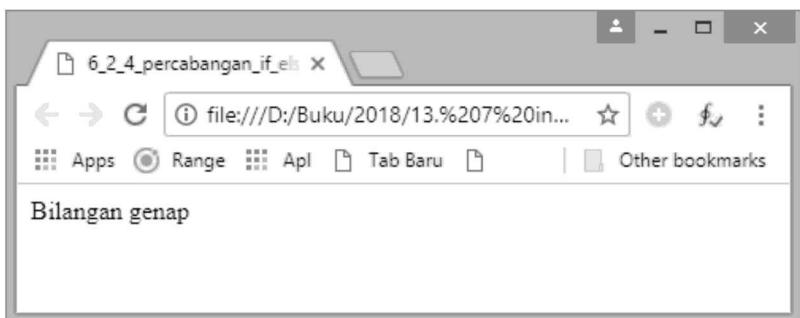
```
if(kondisi){  
    pernyataan jika terpenuhi;  
} else{  
    pernyataan jika tidak terpenuhi;  
}
```

Berikut ini contoh penggunaan percabangan dengan if Else pada JavaScript:

File 6_2_4_percabangan_if_else.html

```
<!DOCTYPE HTML>  
<html>  
  <head>  
    <script>  
      var angka = 20;  
      if(angka%2 == 0) document.write("Bilangan genap");  
      else document.write("Bilangan ganjil");  
    </script>  
  </head>  
  <body>  
  </body>  
</html>
```

Hasil dari perintah di atas seperti gambar berikut:



Gambar 6.9 Hasil percabangan dengan if...else...

Percabangan dengan if ... else if ... else ...

Percabangan ini digunakan untuk menjalankan program jika kondisi terpenuhi, dan jika kondisi tidak terpenuhi ada pengecekan kondisi berikutnya hingga tidak ada lagi kondisi untuk dicek baru dijalankan alternatif terakhir. Cara menggunakannya yaitu dengan format sebagai berikut:

```
if(kondisi){  
    pernyataan jika kondisi terpenuhi ;  
} else if(kondisi kedua){  
    pernyataan jika kondisi kedua terpenuhi ;  
} else if(kondisi ketiga){  
    pernyataan jika kondisi ketiga terpenuhi ;  
} else{  
    pernyataan jika tidak ada kondisi terpenuhi ;  
}
```

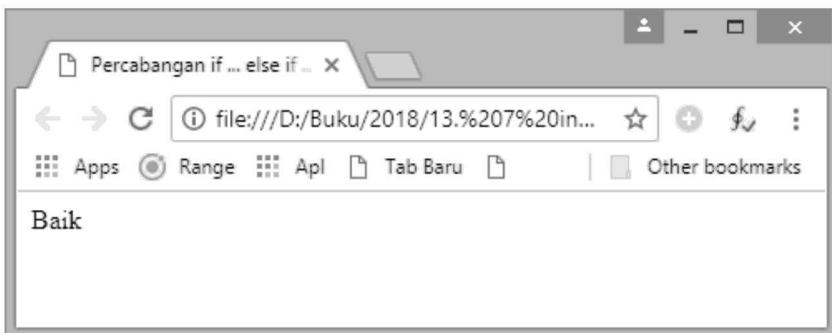
Berikut ini contoh penggunaanya:

File 6_2_4_percabangan_if_else_if_else.html

```
<!DOCTYPE HTML>  
<html>  
  <head>  
    <title>Percabangan if ... else if ... else</title>  
    <script>  
      var nilai = 80;  
      if(nilai > 90) document.write("Sangat Baik");  
      else if(nilai > 75) document.write("Baik");  
      else if(nilai > 60) document.write("Cukup");  
      else document.write("Kurang");  
    </script>  
  </head>  
<body>
```

```
</body>  
</html>
```

Hasilnya seperti gambar berikut:



Gambar 6.10 Hasil percabangan dengan if...else if...else...

Percabangan dengan switch

Percabangan ini merupakan alternatif untuk menggantikan percabangan dengan if ... else if ... else. Namun, jika variabel yang dicek sama, cara ini lebih disarankan daripada menggunakan if ... else if ... else. Cara penggunaanya yaitu sebagai berikut:

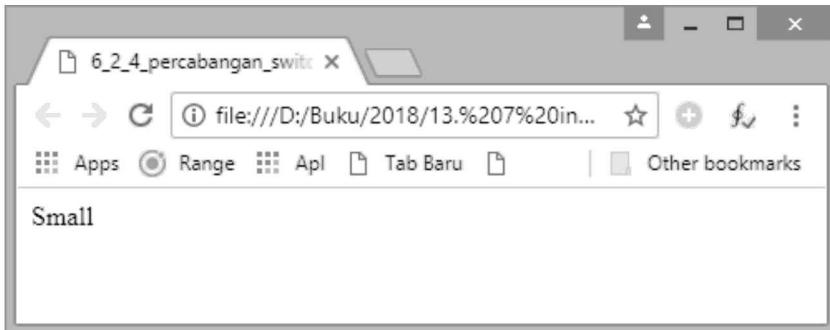
```
switch (variabel){  
    case kondisi 1: pernyataan;  
        break;  
    case kondisi 2: pernyataan;  
        break;  
    case kondisi 3: pernyataan;  
        break;  
    default: pernyataan default;  
}
```

Contoh penggunaanya, perhatikan pada skrip berikut:

```
<!DOCTYPE HTML>  
<html>  
    <head>  
        <script>  
            var ukuran = "S";  
            switch(ukuran){  
                case "L": document.write("Large");  
                    break;  
                case "M": document.write("Medium");  
                    break;  
                default: document.write("Small");  
            }  
        </script>  
    </head>  
    <body>  
        <h1>Hello World</h1>  
    </body>  
</html>
```

```
        }
    </script>
</head>
<body>
</body>
</html >
```

Hasilnya seperti gambar berikut:



Gambar 6.11 Hasil percabangan dengan switch

Perulangan dengan while

Perulangan ini digunakan untuk menjalankan program berulang-ulang selama kondisi yang ditentukan terpenuhi. Cara penggunaanya yaitu sebagai berikut:

```
while(kondisi) {
    pernyataan;
}
```

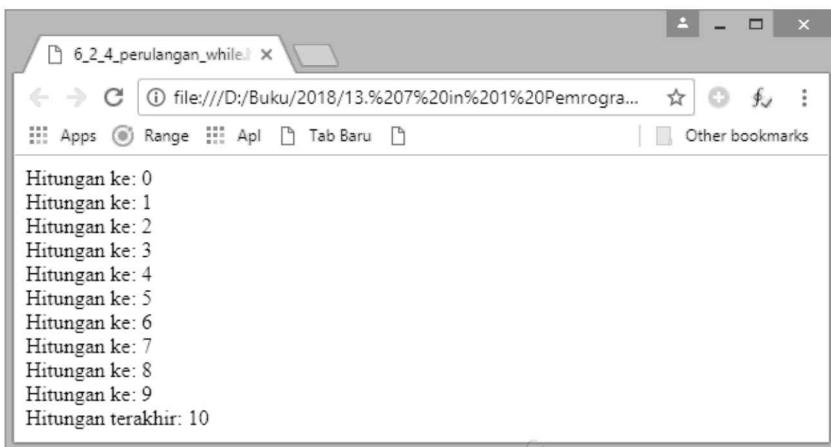
Berikut contoh penggunaan perulangan dengan while:

6_2_4_perulangan_while.html

```
<!DOCTYPE HTML>
<html >
  <head>
    <script>
      var hitung = 0;
      while(hitung < 10){
        document.write("Hitungan ke: " + hitung + "<br>");
        hitung++;
      }
      document.write("Hitungan terakhir: " + hitung +
    " <br> ");
    </script>
  </head>
  <body>
```

```
</body>  
</html>
```

Skrip di atas akan menghasilkan seperti gambar berikut:



Gambar 6.12 Hasil perulangan dengan while

Perulangan dengan do ... while

Statement control ini digunakan untuk menjalankan program secara berulang-ulang selama kondisi yang ditentukan di akhir terpenuhi. Hampir sama dengan while, hanya kondisi ditulis di akhir. Cara penggunaanya yaitu sebagai berikut:

```
do{  
    Pernyataan;  
} while(kondisi);
```

Contoh penggunaanya dapat dilihat pada skrip berikut:

File 6_2_4_perulangan_do_while.html

```
<!DOCTYPE HTML>  
<html>  
  <head>  
    <script>  
      var hitung = 0;  
      do{  
        document.write("Hitungan ke: " + hitung + "<br>");  
        hitung++;  
      } while (hitung<10);  
      document.write("Hitungan mencapai " + hitung + "<br>");  
    </script>  
  </head>  
<body>  
  </body>  
</html>
```

```
</head>
<body>
</body>
</html >
```

Hasilnya sama dengan perulangan menggunakan **while**.

Perulangan dengan for

Perulangan ini digunakan untuk menjalankan program berulang-ulang dengan menentukan nilai awal, nilai akhir, dan penambahan/pengurangan nilai. Cara penggunaanya sebagai berikut:

```
for (ini si al i sasi counter; kondisi; increment/decrement counter) {
    pernyataan;
}
```

Contoh penggunaanya seperti skrip berikut:

File 6_2_4_perulangan_for.html

```
<!DOCTYPE HTML>
<html >
  <head>
    <script>
      var hitung;
      for (hitung=0; hitung<10; hitung++) {
        document.write("Hitungan ke: " + hitung + "<br>");
      }
      document.write("Hitungan mencapai " + hitung + "<br>");
    </script>
  </head>
  <body>
  </body>
</html >
```

Hasilnya sama dengan perulangan menggunakan **while**.

6.2.5 Continue dan Break

Continue digunakan untuk melewati suatu perulangan karena kondisi tertentu. Sedangkan **break** digunakan untuk menghentikan paksa suatu perulangan sebelum kondisi perulangan berakhir karena kondisi tertentu.

Contoh penggunaanya dapat diperhatikan pada skrip berikut:

File 6_2_5_continue_break.html

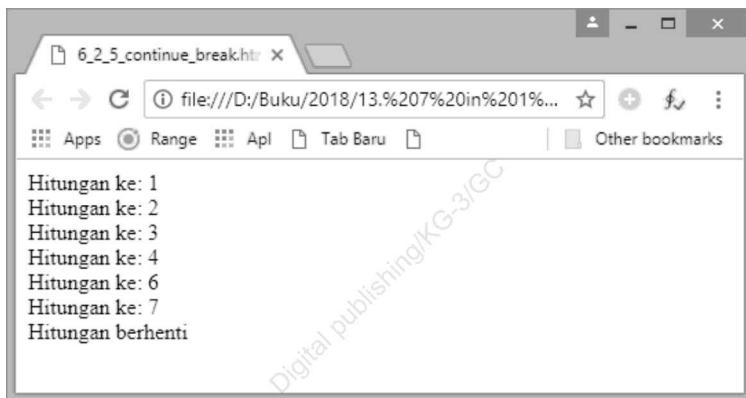
```
<!DOCTYPE HTML>
<html >
```

```

<head>
<script>
    var hitung = 0;
    while(hitung<10) {
        hitung++;
        if(hitung == 5) continue; // dilanjutkan
        if(hitung == 8) break; // berhenti
        document.write("Hitungan ke: " + hitung + "<br>");
    }
    document.write("Hitungan berhenti");
</script>
</head>
<body>
</body>
</html >

```

Hasil dari skrip di atas seperti gambar berikut:



Gambar 6.13 Contoh penggunaan continue dan break

Pada gambar di atas tampak tidak ada hitungan ke-5 dan pada hitungan ke-8 perulangan dihentikan.

6.3 Function

Function merupakan blok program yang dapat digunakan secara berulang-ulang dengan memanggil nama function. Function dapat mengembalikan sebuah nilai dengan perintah **return** atau tanpa mengembalikan nilai. Pemanggilan function yang mengembalikan nilai dapat digunakan sebagai nilai suatu variabel. Cara menulis function yaitu sebagai berikut:

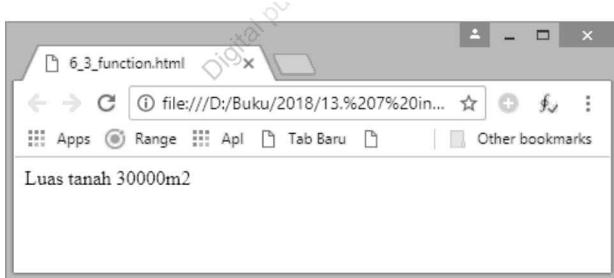
```
function nama_fungsi (parameter 1, parameter 2) {  
    pernyataan;  
    return variabel;  
}
```

Berikut ini contoh dari penggunaan function.

File 6_3_function.html

```
<!DOCTYPE HTML>  
<html>  
    <head>  
        <script>  
            function hitung_luas(p, l){  
                var L = p * l;  
                return L;  
            }  
            function tampilkan_hasil(){  
                var teks = "Luas tanah " + hitung_luas(200, 150) +  
                "m2";  
                document.write(teks);  
            }  
            tampilkan_hasil();  
        </script>  
    </head>  
    <body>  
    </body>  
</html>
```

Hasilnya seperti gambar berikut:



Gambar 6.14 Contoh penggunaan function

6.4 Event

Event merupakan suatu perlakuan yang dilakukan terhadap suatu elemen HTML. Ada berbagai macam event yang dapat digunakan, yaitu sebagai berikut:

- onblur, ketika suatu elemen tidak focus
- onchange, ketika nilai suatu elemen diubah
- onclick, ketika suatu elemen diklik
- ondblclick, ketika suatu elemen didobelklik
- onerror, ketika error
- onfocus, ketika suatu elemen sedang fokus
- onkeydown, ketika tombol huruf pada keyboard ditekan
- onkeypress, ketika tombol keyboard ditekan
- onkeyup, ketika tombol keyboard di lepas
- onload, ketika dokumen di-load
- onmousedown, ketika tombol mouse ditekan
- onmousemove, ketika mouse digerakkan
- onmouseout, ketika mouse keluar dari area suatu elemen
- onmouseover, ketika mouse melewati area suatu elemen
- onmouseup, ketika tombol mouse dilepas setelah diklik
- onreset, ketika form direset
- onresize, ketika elemen diubah ukuranya
- onselect, ketika suatu elemen dipilih
- onsubmit, ketika data form dikirim
- onunload, ketika keluar dari suatu dokumen

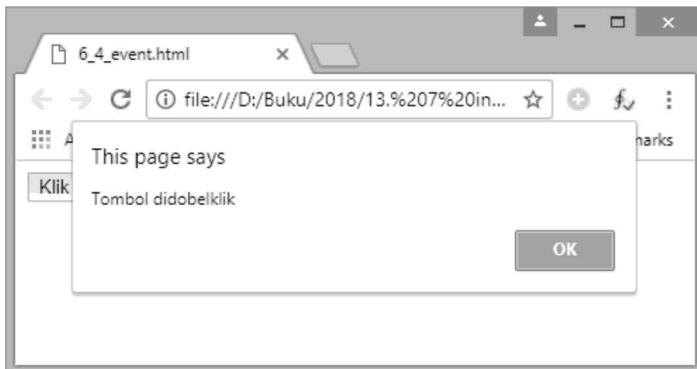
Contoh penggunaanya dapat diperhatikan pada skrip berikut:

File 6_4_event.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <script>
      function dobel_klik() {
        alert("Tombol di dobel klick");
      }
    </script>
  </head>
  <body>
    <button ondblclick="dobel_klik()">
      Klik dan Dobel klick</button>
  </body>
</html>
```

```
</body>  
</html>
```

Coba jalankan, lalu dobel klik pada tombol, maka akan tampil alert seperti gambar berikut:



Gambar 6.15 Contoh penggunaan event

6.5 Kotak Dialog

Ada 3 macam kotak dialog pada JavaScript yaitu dengan function **alert()**, **confirm()**, dan **prompt()**. Alert() digunakan untuk menampilkan pesan tertentu kepada pengguna. Confirm() hampir sama dengan alert, tetapi terdapat 2 tombol yaitu Oke dan Batal, fungsinya untuk meminta konfirmasi pada pengguna. Sedangkan prompt() digunakan untuk meminta masukan data dari pengguna.

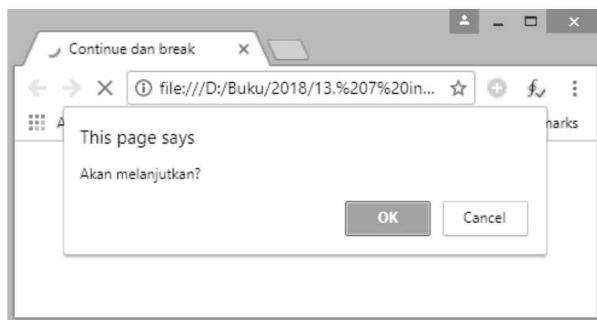
Contoh penggunaanya sebagai berikut:

File 6_5_kotak_dialog.html

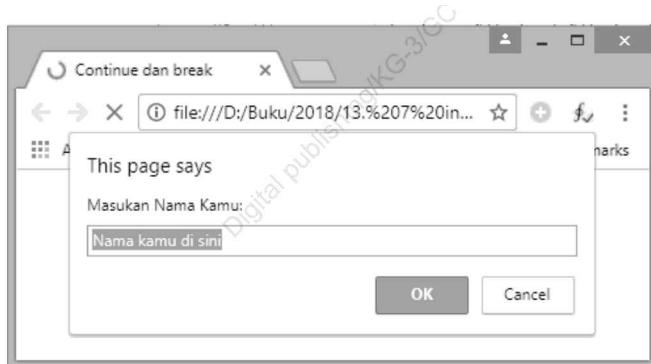
```
<!DOCTYPE HTML>  
<html>  
  <head>  
    <title>Continue dan break</title>  
    <script>  
      alert("Selamat Datang");  
      confirm("Akan melanjutkan?");  
      prompt("Masukan Nama Kamu: ", "Nama kamu disini");  
    </script>  
  </head>  
<body>  
</body>
```

```
</html >
```

Hasil untuk kotak dialog dengan confirm() dan prompt() seperti pada gambar berikut:



Gambar 6.16 Hasil penggunaan confirm()



Gambar 6.17 Hasil penggunaan prompt()

6.6 Objek String

Seperti telah dijelaskan sebelumnya bahwa setiap variabel akan menjadi objek sehingga akan otomatis memiliki properti dan method. Untuk variabel dengan tipe data string memiliki properti dan method sebagai berikut:

Properti / Method	Keterangan
length	Menghasilkan jumlah karakter dari suatu teks.
anchor(nama)	Menghasilkan link sama seperti menggunakan tag .
big()	Menjadikan teks menjadi lebih besar karena diapit tag <big>.
blink()	Menjadikan teks berkedip karena diapit tag <blink>.
bold()	Menjadikan teks bercetak tebal karena diapit tag .
fontcolor(warna)	Menjadikan teks berwarna karena diapit tag .
fontsize(ukuran)	Mengubah ukuran teks karena diapit tag .
italics()	Menjadikan teks miring karena diapit tag <i>.
link(url)	Merubah teks menjadi link karena diapit tag .
small()	Menjadikan teks kecil karena diapit tag <small>.
strike()	Menjadikan teks tercoret karena diapit tag <strike>.
sub()	Menjadikan teks subscript karena diapit tag <sub>.
sup()	Menjadikan teks superscript karena diapit tag <sup>.
toLowerCase()	Mengubah teks menjadi huruf kecil.
toUpperCase()	Mengubah teks menjadi huruf besar.
charAt(x)	Menghasilkan karakter pada posisi x pada string.
charCodeAt(x)	Menghasilkan nilai Unicode dari karakter pada posisi x pada suatu string
concat(teks1, teks2, ...)	Menggabungkan dua atau lebih string menjadi satu.
fromCharCode(c1, c2,)	Menghasilkan string yang dibuat menggunakan urutan dari Unicode.
indexOf(substr, mulai)	Mencari dan menghasilkan nomor index.
charCodeAt(x)	Menghasilkan nilai unicode value dari karakter pada posisi x pada suatu string.
concat(string1, string2, ...)	Menggabungkan beberapa string yang ada di dalam kurung.
fromCharCode(c1, c2, ...)	Menghasilkan string yang dibuat menggunakan urutan dari nilai Unicode.
indexOf(substr, start)	Menghasilkan nomor index dari karakter atau substring yang ada di dalam kurung.
match(regexp)	Mencocokkan suatu string dengan pola regular expression.
replace(regexp, replace_text)	Menukar suatu string berdasarkan pola regular expression.

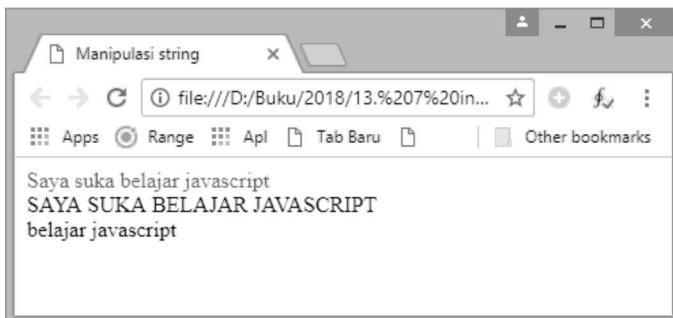
search(regexp)	Mengetes apakah string cocok dengan pola regular expression, jika cocok akan menghasilkan index dari yang cocok.
slice(start, end)	Menghasilkan substring dari suatu string dimulai dari parameter pertama dan diakhiri parameter kedua.
split(delimiter, limit)	Memecah string menjadi array berdasarkan tanda pemisah yang ditentukan.
substr(start, length)	Menghasilkan potongan dari suatu string yang dimulai dari parameter pertama sebanyak parameter kedua.

Berikut contoh penggunaan dari beberapa method dan properti di atas. Untuk contoh method dan properti yang lain silakan bereksperimen sendiri.

File 6_6_string.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Manipulasi string</title>
    <script>
      var teks = "Saya suka belajar JavaScript";
      document.write(teks. fontcolor('red') + '<br>');
      document.write(teks. toUpperCase() + '<br>');
      document.write(teks. substr(10, 18));
    </script>
  </head>
  <body>
  </body>
</html>
```

Hasilnya seperti gambar berikut:



Gambar 6.18 Contoh manipulasi objek string

6.7 Objek Window dan Dokumen

Tidak hanya variabel yang menjadi sebuah objek, window browser dan dokumen juga dianggap sebagai objek, sehingga dapat diterapkan method. Ada beberapa method yang digunakan untuk manipulasi window atau dokumen, yaitu sebagai berikut.

Properti / Method	Keterangan
reload()	Digunakan untuk mereload window, contoh: window.location.reload()
close()	Digunakan untuk menutup window, contoh: window.close()
open()	Digunakan untuk membuka window baru dengan aturan yang ditentukan.
print()	Digunakan untuk membuka jendela print, contoh: window.print()
location	Digunakan untuk pindah ke halaman lain. window.location = "url"

6.8 Method Tanggal dan Waktu

Tanggal dan waktu juga merupakan objek yang memiliki properti dan method. Ada berbagai method untuk memanipulasi tanggal atau waktu. Untuk menggunakan method tersebut, harus memanggil objek Date() terlebih dahulu seperti berikut:

```
var tanggal = new Date();
```

Selanjutnya method tersebut digunakan pada variabel tanggal. Adapun daftar method yang bisa dipakai sebagai berikut:

Properti / Method	Keterangan
getDate()	Menghasilkan tanggal dalam satu bulan.
getDay()	Menghasilkan nama hari.
getHours()	Menghasilkan jam dimulai dari 0 hingga 23
getMinutes()	Menghasilkan menit.
getSeconds()	Menghasilkan detik.

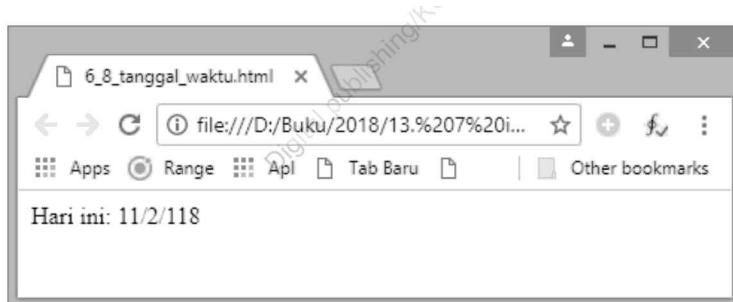
getMonth()	Menghasilkan bulan dimulai dari 0 hingga 11
getYear()	Menghasilkan tahun
getTime()	Menghasilkan waktu yang lengkap.

Berikut ini contoh penggunaan beberapa method di atas untuk memanipulasi data tanggal dan waktu:

File 6_8_tanggal_waktu.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <script>
      var waktu = new Date();
      document.write('Hari ini: ' + waktu.getDate() + '/'
      + waktu.getMonth() + '/' + waktu.getFullYear());
    </script>
  </head>
  <body>
  </body>
</html>
```

Hasilnya seperti gambar berikut:



Gambar 6.19 Contoh manipulasi objek tanggal

6.9 Memahami Array

Array merupakan sebuah variabel yang dapat menampung beberapa data sekaligus yang terindeks secara teratur. Untuk mendeklarasikan array ada beberapa cara yaitu sebagai berikut:

Cara 1

```
var buah = new Array();
buah[0] = "Mangga";
buah[1] = "Apel";
buah[2] = "Jeruk";
```

Cara 2

```
Var buah = new Array("Mangga", "Apel", "Jeruk");
```

Cara 3

```
Var buah = ["Mangga", "Apel", "Jeruk"];
```

Cara menggunakan array yaitu dengan menyebutkan nama variabel dan indeks yang diletakkan di dalam kurung siku. Misalnya:

```
document.write(buah[1]);
```

Skrip di atas akan menghasilkan tulisan "**Mangga**", karena indeks dimulai dari 0.

Beberapa properti dan method yang bisa digunakan untuk memanipulasi array yaitu sebagai berikut:

Properti / Method	Keterangan
length	Menghasilkan jumlah elemen dalam satu array
join(separator)	Digunakan untuk menggabungkan nilai array menjadi string yang dipisahkan dengan tanda yang ditentukan.
concat(array2, array3, ...)	Menggabungkan dua array atau lebih menjadi array baru.
pop()	Menghapus elemen terakhir pada array.
push(item1, item2, ...)	Menambahkan satu atau lebih elemen baru ke bagian akhir array.
reverse()	Membalik urutan elemen array.
shift()	Menghapus elemen pertama dari suatu array
unshift(item1, item2, ...)	Menambahkan satu atau lebih elemen pada awal array.

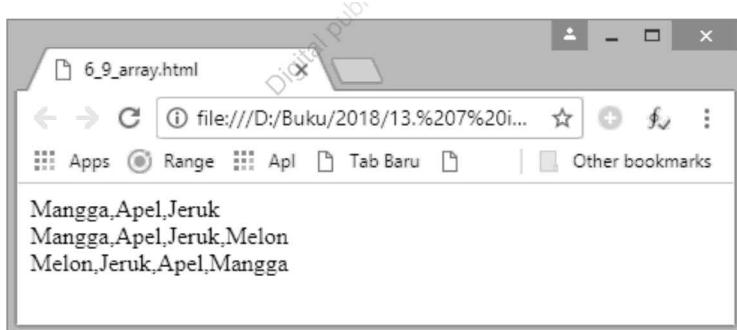
sort()	Mengurutkan elemen array.
splice(index, length,	Menghapus atau menambahkan item pada array.
toString()	Mengkoversi array ke string.
isArray(objek)	Mengecek apakah sebuah objek merupakan array.

Berikut contoh manipulasi array dengan beberapa method di atas:

File 6_9_array.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <script>
      var buah = ["Mangga", "Apel", "Jeruk"];
      document.write(buah + '<br>');
      buah.push("Melon");
      document.write(buah + '<br>');
      buah.reverse();
      document.write(buah + '<br>');
    </script>
  </head>
  <body>
    </body>
</html>
```

Hasilnya seperti gambar berikut:



Gambar 6.20 Hasil manipulasi array

6.10 Objek JavaScript

JavaScript menggunakan *Prototype-based OOP (PBO)*, artinya objek pada JavaScript hanya sebuah kumpulan nilai dan tidak dibuat

menggunakan class seperti pada bahasa pemrograman yang lain. Properti dari objek JavaScript dapat berisi tipe data dasar, juga dapat berisi array, function atau objek. Jadi, memungkinkan adanya objek di dalam objek.

6.10.1 Membuat dan Mengakses Objek

Untuk membuat objek pada JavaScript sama dengan membuat variabel tetapi nilainya diawali tanda { dan diakhiri tanda }. Masing-masing properti dipisahkan dengan koma (,), Sedangkan antara properti dan nilainya dipisahkan titik dua (:).

Untuk mengakses objek properti dari sebuah objek dapat dengan format **objek['properti']** atau dengan format **objek.properti**. Namun, lebih sering digunakan format yang kedua.

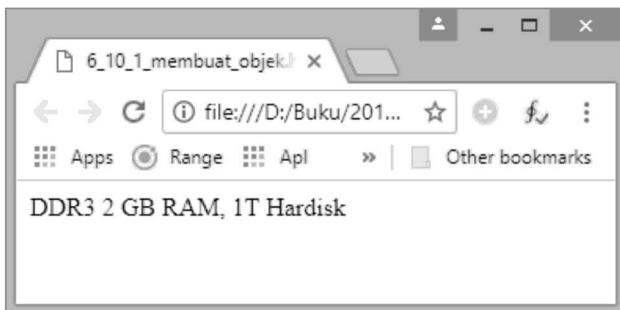
Berikut contoh skrip membuat dan mengakses objek.

File 6_10_1_membuat_objek.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <script>
      komputer = {
        so : 'Linux',
        ram : 'DDR3 2 GB',
        hardisk : '1T'
      }

      document.write(komputer['ram'] + ' RAM, ');
      document.write(komputer.hardisk + ' Hardisk');
    </script>
  </head>
  <body></body>
</html>
```

Skrip di atas membuat objek dengan nama **komputer**, di dalamnya terdapat 3 properti yaitu **so**, **ram** dan **hardisk**. Di bawahnya mencoba mengakses dengan 2 cara, hasilnya seperti gambar berikut:



Gambar 6.21 Membuat dan mengakses objek

6.10.2 Mengubah dan Menghapus Properti

Properti dari objek dapat diubah kapan pun dengan cara memberi nilai pada properti dipisahkan dengan sama dengan (=). Bahkan jika properti yang diubah ternyata tidak ada, properti akan ditambahkan. Sedangkan untuk menghapus properti dapat menggunakan perintah **delete**. Contohnya seperti berikut:

File 6_10_2_mengubah_properti.html

```
<!DOCTYPE HTML>
<html>
<head>
<script>
komputer = {
    so : 'Li nux',
    ram : 'DDR3 2 GB',
    hardisk : '1T'
}

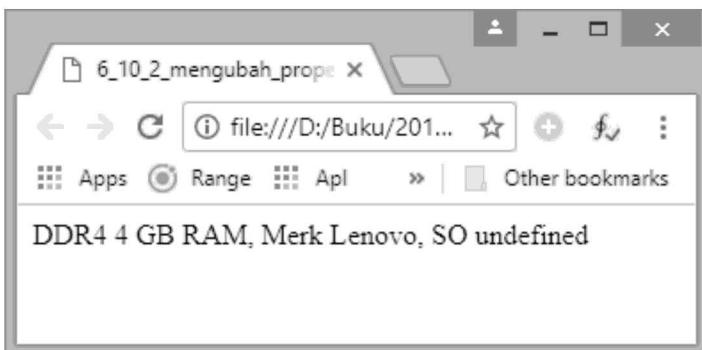
komputer.ram = 'DDR4 4 GB';
document.write(komputer.ram + ' RAM, ');

komputer.merk = 'Lenovo';
document.write('Merk ' + komputer.merk);

delete komputer.so;
document.write(' SO ' + komputer.so)
</script>
</head>
<body></body>
</html >
```

Pada skrip di atas mencoba mengubah properti **ram** menjadi "DDR4 4 GB". Kemudian menambahkan properti **merk** dengan nilai "Lenovo".

Terakhir menghapus properti `so`, sehingga ketika diakses akan menghasilkan `Undefined`. Hasilnya seperti gambar berikut:



Gambar 6.22 Mengubah dan menghapus properti

6.10.3 Berbagai Nilai Properti

Seperti telah dijelaskan sebelumnya, nilai dari properti selain menggunakan tipe data dasar juga dapat diberi nilai array, function maupun objek. Untuk membuktikannya, buat saja skrip berikut:

File 6_10_3_nilai_properti.html

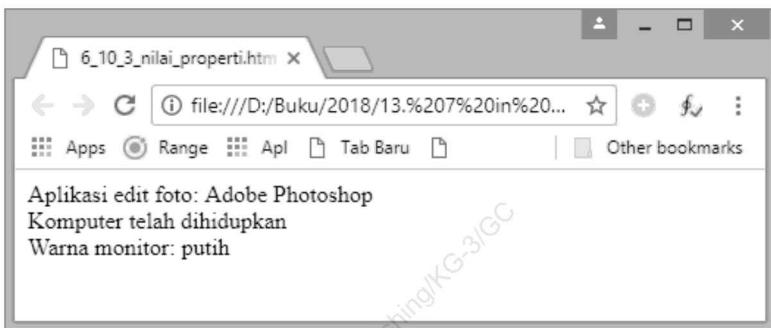
```
<!DOCTYPE HTML>
<html>
<head>
<script>
komputer = {
    so : 'Linux',
    ram : 'DDR3 2 GB',
    aplikasi : ['Ms. Word', 'Ms. Excel', 'Adobe
Photoshop'],
    hidup : function(){
        var text = 'Komputer telah dihidupkan';
        text.toUpperCase();
        return text;
    },
    monitor: {
        merk : 'Samsung',
        ukuran : '21 in',
        warna : 'putih'
    }
}
document.write('Aplikasi edit foto: ' +
komputer.aplikasi[2]);
```

```

document.write('<br>' + komputer.hi_dup());
document.write('<br>Warna monitor: ' +
komputer.monitor.warna);
</script>
</head>
<body></body>
</html>

```

Pada skrip di atas ditambah properti **aplikasi** dengan nilai berupa array, properti **hidup** dengan nilai berupa function dan properti **monitor** dengan nilai berupa objek. Properti dengan nilai function disebut juga **method**. Pada bagian bawah ditunjukkan cara mengakses properti dari masing-masing nilai. Hasilnya seperti gambar berikut:



Gambar 6.23 Contoh berbagai nilai properti

6.11 Perintah setTimeout() dan setInterval()

Perintah **setTimeout()** digunakan untuk menjalankan suatu program setelah batas waktu yang ditentukan dan hanya dijalankan 1 kali. Perintah ini biasanya disimpan dalam suatu variabel sebagai identitas yang dapat digunakan untuk membatalkan program dengan perintah **clearTimeout()**. Ketika perintah **clearTimeout()** dijalankan, maka program yang seharusnya dijalankan pada batas waktu tertentu dengan **setTimeout()** tidak jadi dijalankan. Format penggunaan **setTimeout()** seperti berikut:

```
variabelId = setTimeout(expression, timeout);
```

VariableId adalah variabel yang menjadi identitas. *Expression* adalah function yang akan menjalankan program. Sedangkan *timeout* adalah

batas dijalankanya program dalam milisecond, artinya 1000 berarti 1 detik.

Perintah **setInterval()** digunakan untuk menjalankan program berkali-kali dengan jarak waktu tertentu. Perintah ini dapat digunakan untuk membuat animasi pada JavaScript. Perintah ini juga biasa disimpan dalam suatu variabel yang dapat digunakan untuk menghentikan program dengan perintah **clearInterval()**. Format penggunaan **setInterval()** sebagai berikut:

```
variableId = setInterval(expression, timeout);
```

Sebagai contoh, perhatikan skrip berikut:

File 6_11_setInterval_setTimeOut.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <script>
      var angka = 0;
      var timer;
      var timeout;

      function jalankanTimer() {
        timeout = setTimeout('setTimer()', 3000);
      }

      function setTimer() {
        timer = setInterval('tampilkanAngka()', 100);
      }

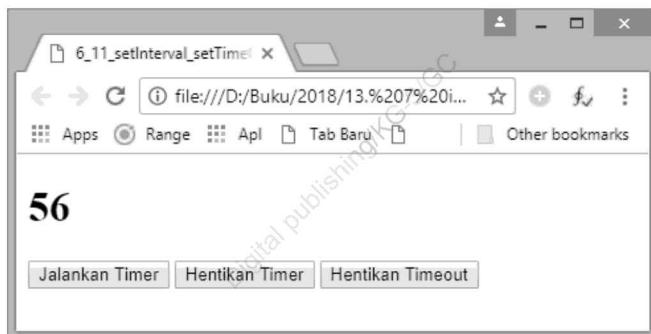
      function tampilkanAngka() {
        document.getElementById('timer').innerHTML = angka;
        angka++;
      }

      function hentikanTimer() {
        clearInterval(timer);
      }

      function hentikanTimeout() {
        clearTimeout(timeout);
      }

    </script>
  </head>
  <body>
    <h1 id="timer">0</h1>
    <button onclick="jalankanTimer()">Jalankan Timer</button>
    <button onclick="hentiakanTimer()">Hentiakan Timer</button>
    <button onclick="hentikanTimeout()">Hentikan
      Timeout</button>
  </body>
</html>
```

Skrip di atas terdiri dari 5 function. Function pertama yaitu **jalankanTimer()** dijalankan ketika tombol Jalankan Timer diklik. Karena pada function tersebut menggunakan perintah **setTimeout()** dengan durasi 3000 milisecond, maka program dijalankan setelah 3 detik. Program yang dijalankan yaitu function **setTimer()** yang menggunakan perintah **setInterval()** dengan durasi 200 milisecond untuk menjalankan perintah function **tampilkanAngka()**. Function ini akan menampilkan angka kemudian menaikkan nilai angka. Dengan demikian ketika tombol Jalankan Timer diklik, 3 detik kemudian angka akan ditampilkan dan meningkat terus menerus dengan interval 200 milisecond. Tombol Hentikan Timer akan menjalankan function **hentikanTimer()** yang digunakan untuk menghentikan timer. Sedangkan tombol Hentikan Timeout akan menjalankan function **hentikanTimeout()** yaitu membatalkan perintah **setTimeout** pada function **jalankanTimer()**. Hasilnya seperti gambar berikut:



Gambar 6.24 Contoh perintah **setInterval()** dan **setTimeout()**

7

JQUERY

7.1 Mengenal jQuery

7.1.1 Pengertian jQuery

JQuery merupakan salah satu dari sekian banyak *JavaScript library*, yaitu kumpulan fungsi JavaScript yang siap pakai, sehingga mempermudah dan mempercepat dalam membuat kode JavaScript. Dengan menggunakan jQuery, skrip JavaScript yang panjang dapat disingkat menjadi beberapa baris kode saja.

Untuk dapat menggunakan jQuery dengan maksimal, Akan lebih baik jika pembaca mempunyai pengetahuan tentang HTML, JavaScript dan CSS.

7.1.2 Kelebihan jQuery

Dibandingkan dengan pesaingnya seperti Prototype, Mootools, YUI, dan library JavaScript lainnya, jQuery memiliki kelebihan tersendiri. Berikut beberapa alasan mengapa harus memilih jQuery dibandingkan yang lainnya:

- JQuery telah banyak dipakai oleh website-website terkemuka di dunia.

- Kompatibel dengan semua browser yang sering digunakan seperti Mozilla Firefox, Internet Explorer, Safari, Google Chrome, dan Opera.
- Kompatibel dengan semua versi CSS.
- Dokumentasi, tutorial dan contoh-contohnya lengkap.
- Didukung oleh komunitas yang besar dan aktif.
- Ketersediaan plugin yang sangat banyak jumlahnya.
- File-nya hanya satu dan ukurannya kecil, sehingga cepat aksesnya.
- Open source (gratis) dengan lisensi dari GNU General Public License dan MIT License.

7.1.3 Kemampuan jQuery

Dalam membuat aplikasi website, jQuery Akan sangat membantu dalam menciptakan desain website yang handal. Berikut beberapa kemampuan yang dimiliki jQuery dalam pembuatan aplikasi berbasis web:

- Memanipulasi Elemen HTML
- Memanipulasi CSS
- Penanganan event terhadap elemen HTML
- Membuat efek dan animasi
- Memudahkan penggunaan AJAX
- Menyederhanakan kode JavaScript lainnya

7.1.4 Mendapatkan Library jQuery

Library jQuery dapat di-download dari website resminya yaitu jquery.com/download. Terdapat skrip untuk development yang lebih dikhususkan untuk para pengembang website dan skrip yang telah dikompres sehingga ukurannya lebih kecil yang lebih dikhususkan untuk para pengguna. Skrip yang telah dikompres biasanya ditambah kata ".min" di belakang namanya. Saat buku ini ditulis, jQuery telah mencapai versi 3.3.1.

7.1.5 Cara Menggunakan jQuery

Cara penggunaan jQuery cukup mudah. Berikut ini langkah-langkah menggunakan jQuery dalam website:

- Simpan jQuery dalam sebuah folder yang Anda inginkan. Biasanya pada website diletakkan pada folder script atau js.
- Buat file HTML dalam folder yang Sama atau dalam folder lain, kemudian panggil jQuery di antara tag <head> dan </head>. Contoh pemanggilan jQuery dapat dilihat pada skrip berikut (perhatikan tulisan yang dicetak tebal).

```
<html>
  <head>
    <script src="js/jquery-3.3.1.min.js"> </script>
  </head>
  <body>
    </body>
<html>
```

- Buatlah kode HTML di antara tag <body> dan </body>, lalu beri atribut id atau class pada tag yang ingin dimanipulasi dengan jQuery. Misalnya akan menampilkan foto, maka skripnya menjadi seperti berikut:

```
<html>
  <head>
    <script src="js/jquery-3.3.1.min.js"> </script>
  </head>
  <body>
    
  </body>
<html>
```

- Buatlah skrip jQuery dengan menyesuaikan id atau class yang telah diberikan pada tag . Misalnya ingin membuat foto menghilang ketika diklik, maka skripnya menjadi seperti berikut:

```
<html>
  <head>
    <script src="js/jquery-3.3.1.min.js"> </script>
    <script>
      $('document').ready(function(){
        $('#foto').click(function(){
          $(this).fadeOut();
        });
      });
    </script>
  </head>
  <body>
    
  </body>
<html>
```

```

        });
    });
</script>

</head>
<body>
    
</body>
</html>

```

Penulisan skrip jQuery harus diawali dengan skrip `$(document).ready (function () {dan diakhiri dengan});`; Tanda dolar pada skrip tersebut merupakan simbol resmi dari jQuery yang harus dituliskan setiap Akan menuliskan selector. Selain dengan skrip di atas, skrip jQuery juga dapat diawali dengan skrip yang lebih singkat yaitu `$(function () {dan diakhiri dengan tanda});`

7.2 Memahami Selector

Selector adalah skrip yang digunakan untuk memilih suatu elemen HTML yang Akan dimanipulasi dengan jQuery. Selector pada jQuery ditulis di dalam kurung setelah tanda dolar. Format penulisan selector adalah sebagai berikut:

```
$('sel ectors')
```

Selector pada jQuery dibagi menjadi beberapa kelompok, yaitu:

7.2.1 Basic Selector

Selector kelompok ini merupakan selector dasar sesuai yang dipakai pada CSS. Berikut daftar dan penjelasannya:

Selector	Penjelasan
Selector All	Memilih semua elemen HTML, ditandai dengan tanda *. \$(*)
Selector tag	Memilih tag HTML dengan cara menulik nama tagnya langsung. Misalnya memilih tag <p> maka ditulisnya menjadi: \$('p')
Selector class	Memilih tag HTML yang memiliki atribut class tertentu. Cara penulisannya dengan menaruh tanda titik(.) sebelum nama

	<p>class. Misalnya memilih tag <p class="merah">, maka penulisan selector menjadi:</p> <pre><code>\$('.merah') atau \$('.p.merah')</code></pre>
Selector id	<p>Memilih tag HTML yang memiliki atribut id tertentu. Cara penulisannya dengan menaruh tanda pagar (#) sebelum nama id. Misalnya memilih tag <p id="hijau">, maka penulisan selector menjadi:</p> <pre><code>('#hijau')</code></pre>
Multiple Selector	<p>Memilih elemen HTML dengan beberapa selector sekaligus, dimana masing-masing selector dipisahkan dengan tanda koma (,). Misalnya:</p> <pre><code>('p, h2, .merah, #hijau')</code></pre>

7.2.2 Attribute Selector

Selector ini ditulis dengan menyebutkan Nama tag diikuti Nama atribut yang diletakkan di dalam tanda kurung siku ([...]). Pada umumnya antara atribut dan nilainya dipisahkan Sama dengan (=). Namun, ada beberapa variasi lain dalam penulisan selector ini yaitu sebagai berikut:

Selector	Penjelasan
[name]="value"]	<p>Memilih tag dengan nilai atribut sama dengan atau diawali dengan kata yang disebutkan. Misalnya memilih atribut hreflang="en", hreflang="en-UK" dan hreflang="english" dapat dituliskan:</p> <pre><code>('a[hreflang="en"]')</code></pre>
[name*="value"]	<p>Memilih tag dengan nilai atribut yang mengandung kata yang disebutkan. Misalnya memilih tag input dengan atribut name="mandor", name ="spiderman", dan name="semangka" dapat dituliskan:</p> <pre><code>('input[name*="man"]')</code></pre>
[name~="value"]	<p>Memilih tag dengan nilai atribut yang mengandung kata yang disebutkan tetapi dipisahkan oleh spasi. Misalnya memilih tag input dengan atribut name="milk man" dan name ="man news" dapat dituliskan:</p> <pre><code>('input[name~="man"]')</code></pre>

[name\$="value"]	Memilih tag dengan nilai atribut yang diakhiri kata yang disebutkan. Misalnya memilih tag input dengan atribut name="spiderman" dan name ="batman" dapat dituliskan: <code>\$('input [name*=" man"]')</code>
[name="value"]	Memilih tag dengan nilai atribut sama persis dengan kata yang disebutkan.
[name!="value"]	Memilih tag dengan nilai atribut tidak sama dengan kata yang disebutkan.
[name^="value"]	Memilih tag dengan nilai atribut yang diawali kata yang disebutkan. Misalnya memilih tag img dengan atribut title="apel" dan title ="anggur" dapat dituliskan: <code>\$('img[title^="a"]')</code>
[name]	Memilih tag yang memiliki atribut yang disebutkan tanpa mempedulikan nilainya. Misalnya memilih tag div dengan atribut id="apel", id="nanas" dan id="jeruk" dapat dituliskan: <code>\$('div[id]')</code>
[name="value"] [name="value"]	Memilih tag yang memiliki semua atribut yang disebutkan dengan nilai yang sama pula. Penyebutan nama atribut bisa sampai 2, 3 atau bahkan lebih.

7.2.3 Filter Selector

Selector ini digunakan untuk memilih elemen dengan keadaan tertentu menggunakan menggunakan basic selector diikuti Nama filter. Cara penulisannya dengan menyebutkan basic selector diikuti Nama filter yang diawali dengan tanda titik dua (:). Daftar selector dan penjelasanya dapat dilihat pada tabel berikut:

Selector	Penjelasan
:animated	Memilih elemen yang sedang dalam progress animasi.
:eq(index)	Memilih elemen dengan urutan seperti yang disebutkan di dalam kurung. Nomor urut dimulai dari nol. Misalnya memilih tag p ke-3 maka dapat dituliskan: <code>\$('p: eq(2)')</code>
:even	Memilih elemen dengan urutan genap.

:first	Memilih elemen dengan urutan pertama.
:focus	Memilih elemen yang sedang fokus.
:gt()	Memilih elemen dengan urutan lebih besar dari angka yang disebutkan.
:header	Memilih elemen yang termasuk header seperti H1, H2, H3, dan sebagainya.
:lang()	Memilih elemen dengan bahasa yang ditentukan.
:last	Memilih elemen dengan urutan terakhir.
:lt()	Memilih elemen dengan urutan lebih kecil dari angka yang disebutkan.
:not()	Memilih elemen yang tidak termasuk selector yang disebutkan di dalam kurung.
:odd	Memilih elemen dengan urutan ganjil.
:root	Memilih elemen yang menjadi root dokumen.
:target	Memilih elemen yang menjadi target URL.
:first-child	Memilih elemen di dalamnya dengan urutan pertama. Misalnya memilih tag li pertama di dalam tag ul, skripnya menjadi: \$('ul:first-child')
:first-of-type	Memilih elemen pertama dengan tipe yang sama pada suatu container.
:last-child	Memilih elemen di dalamnya dengan urutan terakhir.
:last-of-type	Memilih elemen terakhir dengan tipe yang sama pada satu container.
:nth-child()	Memilih elemen di dalamnya dengan urutan seperti yang disebutkan di dalam kurung.
:nth-last-child()	Memilih elemen di dalamnya dengan urutan seperti yang disebutkan di dalam kurung, tetapi urutan dimulai dari yang terakhir.
:nth-last-of-type()	Memilih elemen di dalamnya dengan urutan seperti yang disebutkan dengan tipe yang sama dimulai dari yang terakhir.
:nth-of-type()	Memilih elemen di dalamnya dengan urutan seperti yang disebutkan dengan tipe yang sama.
:only-child	Memilih elemen yang hanya terdapat satu tag di dalam container (parent).
:only-of-type	Memilih elemen yang hanya terdapat satu tag dalam container dengan tipe yang sama.
:contains()	Memilih elemen yang mengandung teks yang ditentukan.
:empty	Memilih elemen yang tidak memiliki konten.
:has()	Memilih elemen yang mengandung tag yang ditentukan di dalam kurung.

:parent	Memilih container (parent) dari selector yang disebutkan sebelumnya.
:hidden	Memilih elemen yang disembunyikan.
:visible	Memilih elemen yang tidak disembunyikan.

7.2.4 Form Selector

Selector kelompok ini khusus memilih elemen di dalam form. Penulisannya sama dengan filter selector. Daftar selector dan penjelasanya sebagai berikut:

Selector	Penjelasan
:button	Memilih elemen dengan tipe button.
:checkbox	Memilih tag input dengan tipe checkbox.
:checked	Memilih tag input type radio, checkbox, atau elemen select yang terpilih.
:disabled	Memilih elemen dengan atribut disabled.
:enabled	Memilih elemen yang tidak memiliki atribut disabled.
:file	Memilih tag input dengan tipe file.
:focus	Memilih elemen yang sedang fokus.
:image	Memilih tag input dengan tipe image.
:input	Memilih semua tipe input termasuk textarea, select dan button.
:password	Memilih tag input dengan tipe password,
:radio	Memilih tag input dengan tipe radio.
:reset	Memilih tag input dengan tipe reset.
:selected	Memilih elemen yang terpilih.
:submit	Memilih tag input dengan tipe submit.
:text	Memilih tag input dengan tipe text.

7.2.5 Hierarchy Selector

Selector kelompok ini digunakan untuk memilih suatu elemen pada posisi tertentu terhadap elemen lain. Yang termasuk dalam kelompok ini yaitu:

Selector	Penjelasan
parent > child	Memilih suatu elemen yang di dalam suatu container. Misalnya memilih tag p yang berada di dalam tag div, maka skripnya: <code>\$('div > p')</code>
Prev + next	Memilih suatu elemen yang berada setelah elemen lain. Misalnya memilih tag p yang posisinya tepat setelah tag div, maka skripnya: <code>\$('div + p')</code>
Prev ~ next	Memilih semua elemen yang posisinya setelah elemen lain. Misalnya memilih semua tag p yang posisinya setelah tag div, maka skripnya: <code>\$('div ~ p')</code>

7.3 Memahami Effect

Effect merupakan salah satu jenis method pada jQuery yang digunakan untuk memanipulasi elemen HTML yang telah dipilih oleh selector seperti menyembunyikan, menampilkan, menganimasikan, dan sebagainya. Effect ditulis setelah selector, yang dipisahkan dengan tanda titik (.) dan diikuti tanda kurung. Tanda kurung bisa diisi dengan pengaturan effect tersebut, atau bisa dikosongkan. Format penulisan effect adalah sebagai berikut:

```
$('selector').nama_effect()
```

Beberapa effect jQuery yang sering dipakai beserta penjelasannya dapat dilihat pada tabel berikut:

Effect	Penjelasan
<code>animate({parameter}, speed, callback)</code>	Digunakan untuk membuat animasi dengan cara merubah style CSS yang ditulis di dalam kurung kurawal. Parameter diisi kode css. Speed diisi <i>slow</i> , <i>fast</i> atau angka dalam milliseconds. Callback diisi function yang dijalankan jika animasi selesai. Contoh: <code>\$('img').animate({top: '20px'}, 300)</code>
<code>hide(speed)</code>	Digunakan untuk menyembunyikan elemen yang dipilih oleh selector. Contoh:

	<code>\$('img').hide('slow')</code>
<code>show(speed)</code>	Digunakan untuk menampilkan kembali elemen yang disembunyikan dengan <code>hide()</code> . Contoh: <code>\$('img').show(200)</code>
<code>toggle(speed)</code>	Digunakan untuk menampilkan dan menyembunyikan elemen secara bergantian, jika sebelumnya tampil maka disembunyikan dan jika sebelumnya disembunyikan maka akan ditampilkan. Contoh: <code>\$('img').toggle()</code>
<code>fadeIn(speed, callback)</code>	Digunakan untuk menampilkan elemen yang dipilih secara perlahan-lahan dengan lama waktu yang ditentukan. Contoh: <code>\$('img').fadeIn('slow')</code>
<code>fadeOut(speed, callback)</code>	Digunakan untuk menyembunyikan elemen yang dipilih secara perlahan-lahan dengan lama waktu yang ditentukan.
<code>fadeTo(speed, opacity, callback)</code>	Digunakan untuk membuat animasi menuju tingkat transparansi tertentu sesuai angka yang dituliskan di dalam kurung. Contoh: <code>\$('img').fadeTo('slow', 0.5)</code>
<code>fadeToggle(speed, callback)</code>	Digunakan untuk menampilkan dan menyembunyikan elemen yang dipilih seperti perintah <code>toggle()</code> , tetapi dengan animasi fade.
<code>slideDown(speed, callback)</code>	Digunakan untuk menampilkan elemen yang dipilih dengan animasi turun.
<code>slideUp(speed, callback)</code>	Digunakan untuk menyembunyikan elemen yang dipilih dengan animasi naik.
<code>slideToggle(speed, callback)</code>	Digunakan untuk menampilkan dan menyembunyikan elemen yang dipilih secara bergantian, jika sebelumnya disembunyikan maka akan ditampilkan dengan animasi turun dan jika sebelumnya ditampilkan maka akan disembunyikan dengan animasi naik.
<code>clearQueue()</code>	Digunakan untuk menghapus antrian animasi yang dibuat dengan perintah <code>animate()</code> .
<code>delay(interval)</code>	Digunakan untuk mengatur jeda antara dua animasi. Interval diisi angka dalam millisecond.
<code>stop()</code>	Digunakan untuk menghentikan animasi yang sedang berjalan.
<code>finish()</code>	Menghentikan animasi berjalan, menghapus antrian dan menyelesaikan semua animasi pada elemen yang sama.

Berikut contoh penggunaan beberapa effect pada HTML:

File 7_3_effect.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <script src="jquery-3.3.1.min.js"></script>
    <style>
      .animasi {
        position: absolute; top: 0; left: 0;
      }
    </style>
    <script>
      $(function() {
        $('.animasi').animate({left: 500}, 'slow');
        $('.fade, .slide').hide();
        $('.fade').fadeIn('slow');
        $('.slide').slideDown('slow');
        $('.slide').click(function(){
          $('.toggle').fadeToggle();
        });
      });
    </script>
  </head>
  <body>
    
    
    
    
  </body>
</html>
```

Jalankan skrip di atas, lalu klik gambar urutan ketiga (kedua setelah gambar pertama bergeser ke kanan).

7.4 Event

7.4.1 Memahami Event

Event merupakan skrip yang akan mendeteksi sebuah aksi dari user misalnya ketika user menekan mouse, memindahkan mouse, dan sebagainya. Penulisan event sama dengan penulisan effect, namun dalam tanda kurung yang ditulis setelah event biasanya berisi function yang akan dijalankan. Format penulisan event adalah sebagai berikut:

```
$( 'selector' ).event( function() {
```

Agar lebih memahami perbedaan selector, effect dan event, silakan perhatikan skrip berikut:

```

<html>
<head>

<script src="jquery-3.1.1.min.js"> </script>
<script type="text/javascript">
    $('document').ready(function(){
        $('#foto').click(function(){
            $(this).fadeOut();
        });
    });
</script>

</head>
<body>
    
</body>
<html>

```

Pada skrip di atas, event-nya adalah **click()**, yang maksudnya ketika elemen HTML yang disebutkan oleh selector, yaitu tag HTML yang mempunyai id "foto" diklik, maka sebuah function dijalankan. Di dalam function ada effect yang bekerja yaitu effect **fadeOut ()** yang fungsinya untuk menyembunyikan foto. Hasilnya, ketika foto diklik, foto akan menghilang secara perlahan-lahan.

7.4.2 Berbagai Jenis Event

Event dibagi menjadi beberapa kelompok sebagai berikut:

Browser event

Event dapat dilakukan pada sebuah elemen browser. Berikut daftar dan penjelasannya:

Event	Penjelasan
<i>error()</i>	Mendeteksi ketika terjadi error pada JavaScript.
<i>resize()</i>	Mendeteksi ketika jendela browser diubah ukurannya.
<i>scroll()</i>	Mendeteksi ketika melakukan scroll pada konten browser.

Berikut contoh penggunaan event pada browser:

File 7_4_2_event_browser.html

```

<!DOCTYPE HTML>
<html>

```

```

<head>
    <script src="js/jquery-3.3.1.min.js"></script>
    <style>
        img{
            position: absolute; top: 0; left: 0;
        }
    </style>

<script>
$(function(){
    $(window).resize(function(){
        $('h2:first').text('Lebar browser '+$(this).width());
        $('h2:last').text('Tinggi browser '+$(this).height());
    });
});
</script>

</head>
<body>
    <h2> </h2>
    <h2> </h2>
</body>
</html>

```

Buka browser dan coba ubah ukuran browser, hasilnya seperti berikut:



Gambar 7.1 Hasil event browser

Keyboard Event

Event ini terjadi ketika terdapat melakukan sesuatu terhadap keyboard seperti menekan tombol maupun ketika melepasnya. Beberapa event yang termasuk kelompok ini yaitu:

Event	Penjelasan
<i>keypress()</i>	Mendeteksi ketika tombol pada keyboard ditekan.
<i>keydown()</i>	Mendeteksi ketika tombol pada keyboard ditekan sebelum dilepas.
<i>keyup()</i>	Mendeteksi ketika tombol pada keyboard dilepas setelah ditekan.

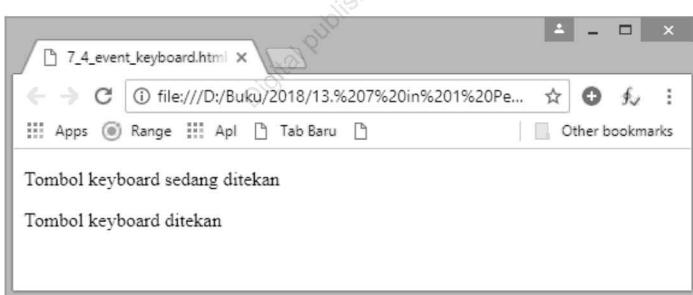
Berikut contoh dari penggunaan event di atas:

File 7_4_2_event_keyboard.html

```
<!DOCTYPE HTML>

<html>
  <head>
    <script src="js/jquery-3.3.1.min.js"></script>
    <script>
      $(function() {
        $(window).keydown(function() {
          $('p').text('Tombol keyboard sedang ditekan');
        }).keyup(function() {
          $('p').text('Tombol keyboard dilepas');
        }).keypress(function() {
          $('div').text('Tombol keyboard ditekan');
        });
      });
    </script>
  </head>
  <body>
    <p></p>
    <div></div>
  </body>
</html>
```

Buka pada browser, lalu tekan tombol keyboard untuk melihat perubahannya.



Gambar 7.2 Hasil event keyboard

Mouse Event

Event mouse terjadi ketika ada perlakuan pada mouse baik ketika menekan tombol mouse maupun ketika menggeser mouse. Yang termasuk dalam event ini yaitu:

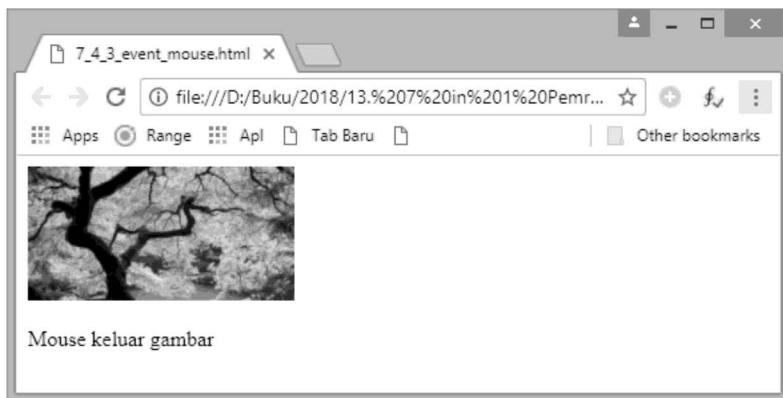
Event	Penjelasan
<i>click()</i>	Mendeteksi ketika tombol kiri mouse diklik.
<i>contextmenu()</i>	Mendeteksi ketika tombol kanan mouse diklik.
<i>dblclick()</i>	Mendeteksi ketika tombol kiri mouse diklik 2 kali.
<i>hover()</i>	Mendeteksi ketika pointer mouse memasuki elemen dan ketika keluar elemen. Diikuti 2 function sekaligus, dimana function pertama dijalankan ketika pointer mouse masuk dan function kedua dijalankan ketika pointer keluar.
<i>mousedown()</i>	Mendeteksi ketika tombol mouse ditekan sebelum dilepaskan.
<i>mouseup()</i>	Mendeteksi ketika tombol mouse dilepas setelah ditekan.
<i>mouseenter()</i>	Mendeteksi ketika pointer mouse memasuki elemen.
<i>mouseleave()</i>	Mendeteksi ketika pointer mouse meninggalkan elemen.
<i>mousemove()</i>	Mendeteksi ketika mouse digerakkan.
<i>mouseover()</i>	Mendeteksi ketika mouse digerakkan sementara pointer berada di dalam elemen.
<i>mouseout()</i>	Mendeteksi ketika mouse digerakkan keluar elemen.
<i>toggle()</i>	Mendeteksi ketika mouse diklik dan diikuti 2 function, dimana function pertama dijalankan ketika klik pertama dan klik ganjil berikutnya, sedangkan function kedua dijalankan ketika klik kedua dan klik genap berikutnya.

Berikut ini contoh penggunaan event di atas:

File 7_4_3_event_mouse.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <script src="js/jquery-3.3.1.min.js"></script>
    <script>
      $(function() {
        $('img').mouseenter(function() {
          $('p').text('Mouse masuk gambar');
        }).mouseleave(function() {
          $('p').text('Mouse keluar gambar');
        }).mousedown(function() {
          $('p').text('Mouse ditekan');
        }).mouseup(function() {
          $('p').text('Mouse dilepas');
        });
      });
    </script>
  </head>
  <body>
    
    <p></p>
  </body>
</html>
```

Event yang lain dapat ditambahkan sendiri. Jalankan skrip di atas, lalu gerakkan pointer mouse masuk dan keluar gambar. Coba juga tekan mouse dan lepaskan tombol mouse di dalam gambar. Hasilnya seperti berikut:



Gambar 7.3 Hasil event mouse

Form Event

Form event yaitu event yang berkaitan dengan perlakuan terhadap form. Berbeda dengan event lain, event pada form dapat dibuat dengan program menggunakan method yang Sama dengan event tetapi tanpa argumen. Berikut yang termasuk kelompok event jenis ini:

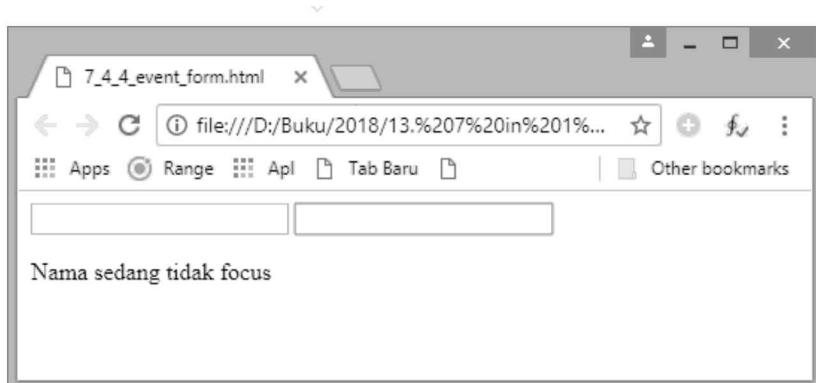
Event	Penjelasan
<code>blur()</code>	Mendeteksi ketika elemen form tidak dalam keadaan fokus.
<code>change()</code>	Mendeteksi ketika nilai suatu elemen form diubah.
<code>focus()</code>	Mendeteksi ketika elemen form sedang fokus.
<code>focusin()</code>	Mendeteksi ketika elemen form berubah dari tidak fokus menjadi fokus.
<code>focusout()</code>	Mendeteksi ketika elemen form berubah dari fokus menjadi tidak fokus.
<code>select()</code>	Mendeteksi ketika salah satu pilihan pada elemen form dipilih.
<code>submit</code>	Mendeteksi ketika tombol submit pada form diklik.

Berikut contoh penggunaan event di atas:

File 7_4_2_event_form.html

```
<!DOCTYPE HTML>
<html>
<head>
<script src="js/jquery-3.3.1.min.js"></script>
<script>
$(function(){
    $('input[name="nama"]').focus(function(){
        $('p').text('Nama sedang fokus');
    }).focusout(function(){
        $('input[name="alamat"]').focus();
        $('p').text('Nama sedang tidak focus');
    });
});
</script>
</head>
<body>
<form>
    <input type="text" name="nama">
    <input type="text" name="alamat">
</form>
<p></p>
</body>
</html>
```

Perhatikan skrip yang dicetak tebal, selain untuk mendeteksi fokus method focus () juga dapat digunakan untuk membuat elemen menjadi fokus. Event yang lain bisa dicoba sendiri. Coba jalankan lalu fokus pada input pertama. Selanjutnya, klik di luar input, maka otomatis alamat menjadi fokus.



Gambar 7.4 Penggunaan event form

7.5 JQuery dan CSS

JQuery dan CSS tidak dapat dipisahkan satu sama lain. Keduanya merupakan pasangan yang serasi ketika digunakan bersamaan untuk memanipulasi desain web menjadi lebih mantap.

Dalam bekerjasama dengan CSS untuk menciptakan desain web yang powerful, jQuery memiliki kemampuan untuk mengambil dan memanipulasi nilai properti pada CSS. Cara mengambil nilai properti pada CSS adalah dengan menggunakan perintah `css()`. Contoh penggunaanya adalah sebagai berikut:

```
$(body).css('background')
```

Skrip di atas digunakan untuk mengambil nilai `background` dari tag `<body>` yang ditentukan pada skrip CSS. Untuk mengubah nilai CSS, maka nama properti dan nilainya dipisahkan dengan tanda koma seperti berikut:

```
$(body).css('background', 'red')
```

Skrip di atas akan mengubah `background` dari tag `<body>` menjadi merah. Jika properti yang diubah lebih dari satu properti, maka properti yang satu dan yang lainnya dipisahkan dengan tanda koma, sedangkan setiap properti dan nilainya dipisahkan dengan titik dua (:). Perhatikan skrip berikut:

```
$(body).css({'background': 'red', 'color': 'blue'})
```

Skrip di atas akan mengubah `background` tag `<body>` menjadi merah dan warna text yang ditulis di dalamnya menjadi biru.

Selain method `css()`, ada beberapa method lain yang khusus untuk mengambil nilai properti tertentu seperti yang terlihat pada tabel berikut:

Method	Penjelasan
<code>height()</code>	Mendapatkan nilai tinggi suatu elemen atau mengatur tinggi elemen jika diisi angka di dalam kurung.
<code>innerHeight()</code>	Mendapatkan atau mengatur tinggi elemen termasuk padding tetapi tidak termasuk border.
<code>width()</code>	Mendapatkan atau mengatur lebar elemen.
<code>innerWidth()</code>	Mendapatkan atau mengatur lebar elemen termasuk padding tetapi tidak termasuk border.

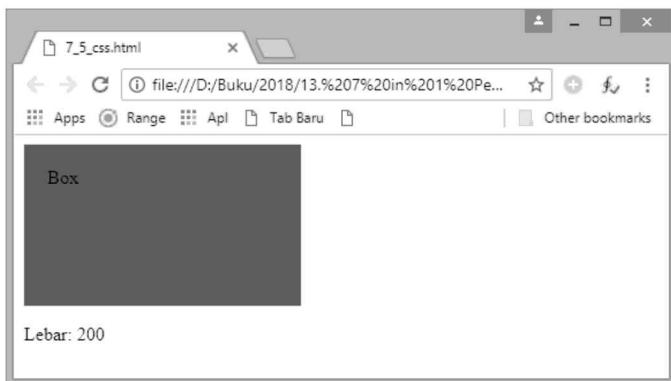
<code>offset()</code>	Mendapatkan nilai koordinat dari elemen relatif terhadap dokumen.
<code>outerHeight()</code>	Mendapatkan atau mengatur tinggi elemen termasuk margin.
<code>outerWidth()</code>	Mendapatkan atau mengatur lebar elemen termasuk margin.
<code>position()</code>	Mendapatkan nilai koordinat dari elemen relatif terhadap parent.
<code>scrollLeft()</code>	Mengatur atau mendapatkan posisi scroll bar secara horizontal.
<code>scrollTop()</code>	Mengatur atau mendapatkan posisi scroll bar secara vertikal.

Berikut ini contoh manipulasi CSS dengan jQuery dan penggunaan beberapa method di atas.

File 7_5_css.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <script src="js/jquery-3.3.1.min.js"></script>
    <script>
      $(function() {
        $('div').css({
          background: 'red',
          width: 200,
          height: 100,
          padding: 20px
        });
        $('p').text('Lebar: '+$('div').width());
      });
    </script>
  </head>
  <body>
    <div>Box</div>
    <p></p>
  </body>
</html>
```

Hasilnya seperti gambar berikut:



Gambar 7.5 Manipulasi CSS dengan jQuery

7.6 Manipulasi HTML dengan jQuery

jQuery memiliki kemampuan untuk memanipulasi elemen HTML seperti menghapus, menduplikasi dan menyisipkan elemen baru. Dalam memanipulasi elemen HTML jQuery memiliki berbagai macam cara. Berikut beberapa perintah jQuery yang digunakan untuk memanipulasi elemen HTML:

Method	Penjelasan
<code>clone()</code>	Menduplikasi suatu elemen yang dipilih oleh selector.
<code>wrap()</code>	Membungkus masing-masing elemen yang dipilih oleh selector dengan elemen baru yang disebutkan di dalam kurung.
<code>unwrap()</code>	Menghapus elemen HTML yang membungkus elemen yang dipilih oleh selector (parent).
<code>wrapAll()</code>	Membungkus semua elemen yang dipilih oleh selector dengan elemen baru.
<code>wrapInner()</code>	Membungkus konten setiap elemen yang dipilih dengan elemen baru.
<code>prepend()</code>	Menyisipkan elemen baru di dalam elemen yang dipilih oleh selector pada bagian awal.
<code>append()</code>	Menyisipkan elemen baru di dalam tag yang dipilih oleh selector pada bagian akhir.
<code>prependTo()</code>	Menyisipkan elemen yang disebutkan oleh selector ke dalam suatu elemen target pada bagian awal.

<code>appendTo()</code>	Menyisipkan elemen yang disebutkan oleh selector ke dalam suatu elemen pada bagian akhir.
<code>text()</code>	Mengambil atau mengubah teks di dalam suatu elemen. Jika argumen tidak disebutkan, akan menghasilkan (mengambil) teks dari elemen yang dimaksud, sedangkan jika argumen disebutkan berupa teks akan mengubah teks sebelumnya dengan teks yang disebutkan.
<code>html()</code>	Mengambil atau mengubah skrip HTML di dalam suatu elemen. Cara kerjanya sama dengan <code>text()</code> .
<code>val()</code>	Mengambil atau mengubah nilai suatu input form.
<code>before()</code>	Menambahkan elemen baru sebelum elemen yang dipilih oleh selector.
<code>after()</code>	Menambahkan elemen baru setelah elemen yang disebutkan oleh selector.
<code>replaceWith()</code>	Mengganti elemen yang dipilih oleh selector dengan elemen baru.
<code>next()</code>	Memilih elemen yang terletak setelah elemen yang dipilih oleh selector.
<code>prev()</code>	Memilih elemen yang terletak sebelum elemen yang dipilih oleh selector.
<code>parent()</code>	Memilih elemen yang membungkus elemen yang dipilih oleh selector (parent).

Selain memanipulasi elemen HTML, jQuery juga memiliki kemampuan untuk memanipulasi atribut suatu elemen. Perintah yang digunakan untuk memanipulasi atribut HTML yaitu sebagai berikut:

Method	Penjelasan
<code>attr()</code>	Mengambil atau mengubah nilai atribut dari elemen yang dipilih oleh selector.
<code>removeAttr()</code>	Menghapus atribut dari elemen yang disebutkan oleh selector.
<code>addClass()</code>	Menambahkan atribut class ke dalam elemen yang dipilih oleh selector.
<code>removeClass()</code>	Menghapus atribut class yang disebutkan di dalam kurung pada elemen yang dipilih oleh selector.
<code>toggleClass()</code>	Menambahkan class baru pada elemen yang dipilih oleh selector jika sebelumnya belum ada dan menghapusnya jika sebelumnya sudah ada.

File 7_6_eleman.html

```
<!DOCTYPE HTML>
```

```

<html>
  <head>
    <script src="js/jquery-3.3.1.min.js"></script>
  </script>

$(function() {
  $('.box').wrap('<div class="container"></div>');
  $('.inner').wrapAll('<div class="innerBox"></div>');
  $('.box').after('<div class="end">Complete</div>');
  $('.box').prepend('<div class="start">Begin</div>');
  $('.inner').addClass('content');
});

</script>
</head>
<body>
  <div class="box">
    <div class="inner">One</div>
    <div class="inner">Two</div>
    <div class="inner">Three</div>
  </div>
</body>
</html>

```

Skrip di atas, akan menghasilkan skrip HTML seperti berikut:

```

...
<div class="container">
  <div class="box">
    <div class="start">Begin</div>
    <div class="innerBox">
      <div class="inner content">One</div>
      <div class="inner content">Two</div>
      <div class="inner content">Three</div>
    </div>
  </div>
  <div class="end">Complete</div>
</div>
...

```

Skrip di atas diambil dari tab Element pada browser Google Chrome. Tab tersebut dapat diakses dengan klik kanan lalu pilih Inspeksi atau tekan Ctrl+Shift+I. Skrip yang dicetak tebal merupakan hasil dari manipulasi oleh jQuery, sedangkan yang tidak tebal adalah aslinya.

7.7 Menggunakan Plugin jQuery

Plugin jQuery merupakan skrip jQuery yang dibuat oleh pihak ketiga yang dapat digunakan untuk tujuan tertentu pada aplikasi website secara instan sehingga penggunaan jQuery menjadi sangat mudah.

Sebenarnya, seorang web programmer tidak harus menguasai skrip jQuery untuk menerapkan jQuery pada website. Asal tahu cara menggunakan plugin jQuery, sudah dapat menggunakan jQuery pada berbagai komponen website.

Salah satu keunggulan jQuery, terdapat ribuan plugin yang dapat digunakan secara gratis di internet. Cara menggunakan plugin cukup mudah, karena sama seperti menggunakan effect jQuery. Hanya dengan menuliskan skrip beberapa baris sudah dapat menerapkan plugin jQuery. Secara umum, plugin jQuery dapat digunakan dengan langkah-langkah sebagai berikut:

- Panggil file plugin seperti saat memanggil file jQuery. Jika plugin mempunyai file CSS, maka panggil juga file CSS-nya di antara tag <head> dan </head>. Cara pemanggilan file CSS dan plugin jQuery sebagai berikut:

```
<link rel="stylesheet" type="text/css"
      href="plugins/folder_plugin/nama_plugin.css">
<script type="text/javascript"
      src="plugins/folder_plugin/nama_plugin.js"></script>
```

- Tambahkan id atau class pada elemen HTML yang akan diterapkan plugin.

```
<div id="nama_id" > </div>
```

- Gunakan plugin jQuery dengan aturan skrip sebagai berikut:

```
<script type="text/javascript">
$(function(){
    $('#nama_id').nama_plugin();
})</script>
```

Jika ada pengaturan pada plugin, biasanya formatnya menjadi seperti berikut:

```
<script type="text/javascript">
$(function(){
    $('#nama_id').nama_plugin({
        parameter1 : nilai_parameter,
        parameter2 : nilai_parameter,
        ...
});</script>
```

```
});  
</script>
```

Berikut contoh penggunaan plugin jQuery untuk membuat zoom image pada galeri.

File 7_7_plugin.html

```
<!DOCTYPE HTML>  
<html>  
  <head>  
    <link rel="stylesheet" href="fancybox/  
jquery.fancybox.min.css">  
    <script src="js/jquery-3.3.1.min.js"></script>  
    <script src="fancybox/jquery.fancybox.min.js"></script>  
    <script>  
      $(function() {  
        $('#galeri a').fancybox();  
      });  
    </script>  
  </head>  
  <body>  
    Klik gambar untuk memperbesar!  
    <div id="galeri">  
      <a href="images/1.jpg">  
          
      </a>  
      <a href="images/2.jpg">  
          
      </a>  
      <a href="images/3.jpg">  
          
      </a>  
      <a href="images/4.jpg">  
          
      </a>  
    </div>  
  </body>  
</html>
```

Pada skrip dia atas menggunakan plugin fancybox yang terdiri dari file **jquery.fancybox.min.css** dan **jquery.fancybox.min.js**. Kedua file tersebut diletakkan pada folder **fancybox**. Hasilnya seperti gambar berikut:



Gambar 7.6 Contoh penggunaan plugin

7.8 Berbagai Contoh Plugin jQuery

Di antara ribuan plugin yang ada di internet, berikut beberapa contoh plugin untuk berbagai keperluan dalam pemrograman web yang dikelompokkan berdasarkan fungsinya:

Kategori Plugin	Contoh Plugin
Form Validation	jQuery Form Validator, Bootstrap Validator, FormValidation, Parsley, Smoke, Validate, jQuery.validity, h5Validate.
Calendar & Date Picker	FullCalendar, CLNDR, Simple Event Calendar, Zabuto Calendar, gDatepicker, DateTimePicker, PickMeUp.
Image Slider	Nivo Slider, bxSlider, SliceBox, MasterSlider, Owl Carousel, FlexSlider, Jssor jQuery Slider, Elastic Slider, Pgw Slider, jcSlider, Zoom Slider, WOW Slider,
Image Gallery	Fancybox, Lightbox, Flip Gallery, Metro, jGallery, Bootstrap Image Gallery, Balanced Gallery.
Table	Param Query, Data Table, WATables, tableSorter, Tabulator, jsGrid.
WYSIWYG Text Editor	Froala, ContentTools, TinyMCE, Bootstrap3-wysiwyg, Summernote, CKEditor, Redactor.
Chart	HightchartTable, jChartFX, Chart.js, jqPlot, Chartist.js, Dynatable.js.
Navigation	jQuery Stellarnav, Megamenu.js, Responsive-collapse.js, Responsive-menu.js, eosMenu, okayNav.

Color Picker	Tinycolorpicker, ColorPicker, Spectrum, Pick-a-Color, Farbastic, ColPick, ColorJoe.
--------------	---

Plugin-plugin di atas dapat di-download dengan mudah, tinggal ketikan saja namanya pada kotak pencarian Google. Selain plugin di atas, masih banyak plugin lain bahkan hingga ribuan yang dapat di-download dengan gratis baik untuk keperluan yang sama seperti di atas, maupun untuk keperluan lainnya. Bahkan ada plugin yang dapat digunakan untuk berbagai macam keperluan seperti plugin **Jquery-UI**. Plugin ini dapat digunakan untuk membuat tab, accordion, datepicker, Dialog, Drag & Drop, Tooltip, Slider, dan sebagainya.

7.9 Membuat Plugin Sendiri

Bagi yang ingin membuat plugin sendiri, caranya juga cukup mudah. Berikut format dasar pembuatan plugin:

```
(function ($) {
  $ .fn. namaPlugin = function () {
    //skrip plugin
  }
})(jQuery);
```

Kata yang dicetak tebal di atas diganti nama plugin yang ingin dibuat dan juga sebagai method yang digunakan ketika plugin digunakan. Jika plugin yang dibuat memiliki option, maka formatnya menjadi seperti berikut:

```
(function ($) {
  $ .fn. namaPlugin = function( options ) {
    var setting = $ .extend({{
      option1 : "nilai default",
      option2 : "nilai default",
      ...
    }, options);
    //skrip plugin
  }
})(jQuery);
```

Sebagai contoh, buat file **pluginku.js** yang diisi dengan skrip plugin sebagai berikut:

File pluginku.js

```
(function ($) {
```

```

$.fn.pluginku = function( options ){
    var setting = $.extend({
        warna : '#fff',
        background : '#f00',
    }, options);

    return this.css({
        color: setting.warna,
        backgroundColor: setting.background,
        width: 200,
        height: 100,
        padding: 10
    })
};

}(jQuery));

```

Pada skrip plugin di atas, nama plugin yaitu **pluginku**, memiliki 2 option yaitu **warna** dan **background**. Selanjutnya buat file yang menerapkan plugin di atas sebagai berikut:

File 7_8_plugin_custom.html

```

<!DOCTYPE HTML>
<html>
    <head>
        <script src="js/jquery-3.3.1.min.js"></script>
        <script src="js/pluginku.js"></script>
        <script>
            $(function(){
                $('.normal').pluginku();
                $('.biru').pluginku({
                    warna: 'yellow',
                    background: 'blue'
                });
            });
        </script>
    </head>
    <body>
        <div class="normal">Box normal</div>
        <div class="biru">Box biru</div>
    </body>
</html>

```

Pada skrip di atas terdapat 2 box dengan class normal dan biru. Pada box normal diterapkan plugin tanpa option sehingga menggunakan nilai yang dimasukkan pada skrip plugin. Sedangkan pada box biru diterapkan plugin dengan option. Hasilnya, box biru akan memiliki background biru dan teks berwarna kuning seperti yang ditentukan pada option di atas.

7.10 Bonus Skrip Contoh

Untuk menunjang proses pembelajaran jQuery, bersama dengan file pendukung penulis sertakan berbagai contoh skrip jQuery tanpa plugin yang dapat ditemukan pada folder **BONUS**. Pada folder tersebut 27 contoh penggunaan jQuery tanpa plugin seperti membuat animasi teks, slideshow gambar, animasi pointer mouse, animasi menu, animasi tombol, animasi tab, tooltip, kotak dialog, dan sebagainya.

8

BOOTSTRAP

8.1 Mengenal Bootstrap

8.1.1 Pengertian Bootstrap

Bootstrap merupakan salah satu framework CSS paling popular dari sekian banyak framework CSS yang ada. Bootstrap memungkinkan desain sebuah web menjadi responsif sehingga dapat dilihat dari berbagai macam ukuran device dengan tampilan tetap menarik. Bootstrap juga membuat proses pengaturan desain menjadi lebih cepat karena tidak perlu lagi banyak menulis CSS, bahkan hampir tidak perlu kecuali jika memerlukan pengaturan desain yang berbeda dengan style Bootstrap. Bootstrap telah didukung oleh hampir semua browser baik pada desktop maupun mobile.

8.1.2 Mendapatkan Bootstrap

Bootstrap dapat didapatkan melalui website resminya yaitu getbootstrap.com. Dari hasil download Bootstrap terdapat banyak sekali file dan folder di dalamnya, namun yang paling utama yaitu folder **dist**. Folder ini yang perlu di salin ke direktori website yang akan menggunakan Bootstrap.

8.1.3 Struktur File Bootstrap

Di dalam folder **dist** terdiri dari 2 folder yaitu **css** dan **js** dengan struktur sebagai berikut:

```
dist/
  -- css/
    -- bootstrap.css
    -- bootstrap.css.map
    -- bootstrap.min.css
    -- bootstrap.min.css.map
    -- bootstrap-grid.css
    -- bootstrap-grid.css.map
    -- bootstrap-grid.min.css
    -- bootstrap-grid.min.css.map
    -- bootstrap-reboot.css
    -- bootstrap-reboot.css.map
    -- bootstrap-reboot.min.css
    -- bootstrap-reboot.min.css.map
  -- js/
    -- bootstrap.bundle.js
    -- bootstrap.bundle.min.js
    -- bootstrap.js
    -- bootstrap.min.js
```

Penjelasan dari masing-masing file di atas yaitu sebagai berikut:

- **bootstrap.css**, berisi seluruh skrip CSS Bootstrap termasuk layout, content, component dan utilities.
- **bootstrap-grid.css**, hanya berisi skrip CSS untuk keperluan layout, tidak mendukung komponen dan style HTML standar Bootstrap.
- **bootstrap-reboot.css**, hanya berisi skrip CSS untuk mengubah desain standar HTML, tidak mendukung layout dan komponen Bootstrap.
- **bootstrap.bundle.js**, berisi skrip JavaScript Bootstrap termasuk popper.js di dalamnya.
- **bootstrap.js**, berisi skrip JavaScript Bootstrap tidak termasuk popper.js.

Semua file yang diakhiri **.min** berarti bentuk file yang dikompres dari file yang namanya sama, sehingga ukurannya lebih kecil dan lebih disarankan untuk digunakan.

8.1.4 Cara Menggunakan Bootstrap

Cara menggunakan Bootstrap cukup mudah. Langkahnya seperti berikut:

1. Salin folder dist ke folder tempat penyimpanan file website. Sebaiknya ubah namanya menjadi **bootstrap** agar mudah dikenali.
2. Tambahkan tag <meta name="viewport"> pada elemen <head> pada file template yang menggunakan bootstrap untuk membuat web responsive.
3. Panggil file **bootstrap.min.css** pada elemen <head>.
4. Panggil library **jQuery** dan file **bootstrap.min.js** atau **bootstrap.bundle.min.js** di bagian akhir elemen <body>.

Dengan demikian, maka struktur skrip yang menggunakan Bootstrap menjadi seperti berikut:

File 8_1_4_struktur_dasar.html

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link rel="stylesheet" href="bootstrap/css/bootstrap.min.css">
  </head>
  <body>
    <h1>Hello, world!</h1>
    <script src="js/jquery-3.3.1.min.js"></script>
    <script src="bootstrap/js/bootstrap.bundle.min.js">
  </script>
  </body>
</html>
```

8.2 Layout

Layout pada Bootstrap menggunakan grid system yang terdiri dari baris dan kolom dan dibungkus dengan container. Container merupakan tag <div> yang diberi class **container** untuk lebar tidak memenuhi layar atau **container-fluid** untuk lebar memenuhi layar. Baris merupakan tag <div> yang diberi class **row**. Adapun kolom merupakan tag <div> yang diberi class diawali dengan **col-**. Nama class pada kolom disesuaikan

dengan spesifikasi breakpoint dan lebar kolom dengan format sebagai berikut:

```
<div class="col-{breakpoint}-{lebar kolom}">
```

Breakpoint diisi sesuai dengan target breakpoint diantaranya: **sm** ($\geq 576px$), **md** ($\geq 768px$), **lg** ($\geq 992px$), **xl** ($\geq 1200px$). Jika target layar kurang dari 576px, breakpoint tidak perlu diisi, jadi langsung diikuti lebar kolom. Lebar kolom merupakan angka dari 1 sampai 12, dimana 12 untuk lebar memenuhi container.

Adapun struktur penulisan dari container, baris dan kolom yaitu sebagai berikut:

```
<div class="container">
  <div class="row">
    <div class="col-md-3">
    <div class="col-md-7">
    </div>
    <div class="row">
      <div class="col-md-7">
      <div class="col-md-3">
    </div>
  </div>
</div>
```

Ada beberapa ketentuan mengenai layout pada Bootstrap yaitu:

- Menggunakan class **col** tanpa diikuti spesifikasi breakpoint dan lebar kolom akan membuat kolom memiliki lebar sama dan memenuhi lebar container.
- Jika ada 3 atau lebih kolom dan salah satu kolom ditentukan lebarnya, maka kolom lain menyesuaikan sisanya dengan masing-masing kolom lebarnya sama.
- Jika salah satu kolom diberi class **col-{breakpoint}-auto**, maka lebar kolom mengikuti lebar konten (isi dari kolom tersebut) dan kolom lain menyesuaikan.
- Untuk membagi kolom menjadi 2 baris atau lebih dan masing-masing kolom lebarnya sama, maka untuk memisahkan masing-masing baris dapat menggunakan tag **<div>** dengan class **w-100**.
- Semua kolom berapapun lebarnya akan disusun bertumpuk pada layar yang kurang dari nilai breakpoint yang ditentukan. Jika ingin tetap disusun horizontal, gunakan class **col** saja atau **col-{lebar kolom}**.

- Untuk mendapatkan susunan dan lebar kolom yang berbeda pada setiap ukuran layar yang berbeda dapat menggunakan kombinasi class untuk kolom, misalnya **col-md-4** dan **col-sm-8** jika digunakan bersamaan akan menghasilkan lebar kolom 4 pada layar sedang dan lebar kolom 8 pada layar kecil.
- Gunakan class **align-items-start**, **align-items-center** atau **align-items-end** bersama class **row** untuk mengatur perataan baris secara vertikal.
- Gunakan class **align-self-start**, **align-self-center** atau **align-self-end** bersama class **col** untuk mengatur perataan kolom secara vertikal.
- Gunakan class **justify-content-start**, **justify-content-center**, **justify-content-end**, **justify-content-around** atau **justify-content-between** bersama class **row** untuk mengatur perataan kolom secara horizontal.
- Gunakan class **no-gutters** untuk menghilangkan jarak antar kolom.
- Gunakan class **order-first**, **order-last** atau **order-{posisi}** untuk mengatur urutan kolom dimana posisi dapat diisi dengan angka 1 sampai 12.
- Gunakan class **offset-{breakpoint}-{offset}** untuk mengatur offset kolom. Offset dapat diisi angka 1 sampai 12.
- Gunakan class **mr-auto** atau **ml-auto** untuk mengatur margin kanan atau margin kiri auto pada kolom.
- Sebuah kolom dapat berisi baris lagi di dalamnya sehingga dapat membagi kolom menjadi beberapa baris dan kolom yang lebih kecil di dalamnya.

Berikut contoh penggunaan grid untuk membuat layout:

File 8_2_layout.html

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
    scale=1, shrink-to-fit=no">
    <link rel="stylesheet"
      href="bootstrap/css/bootstrap.min.css">
    <style>
      .main{ background: #ccc; }
      .slide{ background: #aaa; height: 400px; }
      .image{ background: #888; }
```

```

        .article{ background: #999; }
        .image, .article{ height: 100px; margin-top: 10px; }
    </style>
</head>
<body>

<div class="container">
    <div class="row">
        <div class="col-sm-8 main">
            <div class="row justify-content-center">
                <div class="col-6 image"></div>
            </div>
            <div class="row">
                <div class="col-3 image"></div>
                <div class="col article"></div>
            </div>
            <div class="row">
                <div class="col-3 image"></div>
                <div class="col article"></div>
            </div>
        </div>
        <div class="col side"></div>
    </div>
</div>

<script src="js/jquery-3.3.1.min.js"></script>
<script src="bootstrap/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

Pada skrip di atas ada sedikit tambahan skrip CSS untuk memberi background pada masing-masing kolom dan mengatur tingginya. Class yang tidak dicetak tebal bukan termasuk class dari Bootstrap. Hasilnya seperti berikut:



Gambar 8.1 Penggunaan grid pada layout

8.3 CSS Utilities

Bootstrap telah mengatur ulang sebagian besar style HTML. Sehingga, tanpa menambahkan skrip CSS pun pada HTML tampilanya akan terasa berbeda dibanding ketika tidak menggunakan Bootstrap. Tentu saja tampilanya lebih rapi dan enak dipandang. Selain itu, Bootstrap menyediakan berbagai **class instant** untuk menghasilkan style tertentu pada elemen HTML. Cara menggunakanya hanya dengan menerapkan class yang disediakan dengan atribut class pada elemen HTML. Dengan demikian, dapat mempersingkat proses pembuatan desain web. Berikut ini daftar class yang disediakan Bootstrap untuk mengatur style elemen HTML.

Kategori	Class	Penjelasan
Heading	<i>display-1, display-2, display-3, display-4</i>	Membuat heading lebih besar, dimana semakin kecil angkanya hasilnya semakin besar. Digunakan pada tag <h1>.
Paragraf	<i>lead</i>	Membuat paragraf dengan teks lebih besar. Digunakan pada tag <p>.
Blockquote	<i>blockquote</i>	Membuat teks cuplikan. Digunakan bersama tag <blockquote>.
	<i>blockquote-footer</i>	Menuliskan nama orang pada cuplikan. Digunakan bersama tag <footer> di dalam tag <blockquote>.
List	<i>list-unstyled</i>	Membuat list tanpa simbol. Digunakan pada tag .
	<i>list-inline</i>	Membuat list tersusun secara horizontal. Digunakan pada tag bersama tag yang diberi class <i>list-inline-item</i> .
Image	<i>img-fluid</i>	Membuat gambar memenuhi container. Digunakan pada tag .
	<i>img-thumbnail</i>	Membuat gambar thumbnail yang memiliki border dan padding.
Table	<i>table</i>	Membuat desain tabel khas Bootstrap. Digunakan pada tag <table>.
	<i>table-dark</i>	Membuat tabel dengan background hitam.
	<i>thead-light, thead-dark</i>	Membuat background berwarna pada header tabel. Digunakan pada tag <thead>.

	<i>table-striped</i>	Membuat tabel belang-belang.
	<i>table-bordered</i>	Membuat tabel memiliki border.
	<i>table-hover</i>	Membuat baris tabel berubah warna background ketika ditunjuk mouse.
	<i>table-sm</i>	Membuat tabel dengan padding lebih sempit.
	<i>table-{color}</i>	Membuat background baris tabel berwarna. Umumnya digunakan pada tag <tr>. Color dapat diisi dengan <i>active</i> , <i>primary</i> , <i>secondary</i> , <i>success</i> , <i>danger</i> , <i>warning</i> , <i>info</i> , <i>light</i> dan <i>dark</i> .
	<i>table-responsive</i> , <i>table-responsive-{breakpoint}</i>	Membuat tabel responsive, yaitu muncul scroll bar horizontal pada lebar layar tertentu. Digunakan pada tag <div> yang membungkus tabel.
Border	<i>border</i> , <i>border-top</i> , <i>border-bottom</i> , <i>border-right</i> , <i>border-left</i>	Memberi border pada suatu elemen.
	<i>border-0</i> , <i>border-top-0</i> , <i>border-bottom-0</i> , <i>border-right-0</i> , <i>border-left-0</i>	Menghilangkan border pada elemen yang memiliki border.
	<i>border-{color}</i>	Membuat border berwarna pada elemen. Color dapat diisi dengan <i>primary</i> , <i>secondary</i> , <i>success</i> , <i>danger</i> , <i>warning</i> , <i>info</i> , <i>light</i> , <i>dark</i> dan <i>white</i> .
	<i>rounded</i> , <i>rounded-top</i> , <i>rounded-right</i> , <i>rounded-bottom</i> , <i>rounded-left</i>	Membuat sudut elemen menjadi melengkung (tidak lancip).
Floate	<i>float-left</i> , <i>float-right</i>	Mengatur perataan elemen.
	<i>clearfix</i>	Menormalkan pengaruh dari float-left dan float-right.
Close Icon	<i>close</i>	Membuat ikon silang.
Text	<i>text-{color}</i>	Mengatur warna teks. Color dapat diisi dengan <i>primary</i> , <i>secondary</i> , <i>success</i> , <i>danger</i> , <i>warning</i> , <i>info</i> , <i>light</i> , <i>dark</i> , <i>muted</i> dan <i>white</i> .
	<i>text-hide</i>	Menyembunyikan teks.
	<i>text-left</i> , <i>text-center</i> , <i>text-right</i> , <i>text-justify</i>	Mengatur perataan teks.
	<i>text-{breakpoint}-{left / right / center}</i>	Mengatur perataan teks pada ukuran layar tertentu. Contoh: <i>text-md-left</i> , <i>text-sm-center</i> .
	<i>text nowrap</i>	Membuat teks tidak ganti baris saat memenuhi container.
	<i>text-truncate</i>	Membuat teks terpotong ketika

		memenuhi container.
	<i>text-capitalize, text-uppercase, text-lowercase</i>	Mengatur besar kecilnya huruf.
	<i>font-weight-bold, font-weight-normal, font-weight-light, font-italic</i>	Untuk mengatur ketabalan dan style font.
Color	<i>bg-{color}</i>	Mengatur warna background pada elemen. Color dapat diisi dengan <i>primary, secondary, success, danger, warning, info, light, dark</i> dan <i>white</i> .
Display	<i>d-{value}, d-{breakpoint}-{value}</i>	Mengatur display elemen dimana value dapat diisi dengan <i>none, inline, inline-block, block, table, table-cell, table-row, flex</i> dan <i>inline-flex</i> . Selain dapat diisi dengan <i>sm, md, lg</i> dan <i>lg</i> , breakpoint juga dapat diisi dengan <i>print</i> untuk mengatur display elemen pada saat di-print. Contoh: <i>d-md-block, d-lg-inline</i> .
Embed	<i>embed-responsive</i>	Mengatur tag <code><iframe></code> agar responsive. Digunakan pada tag <code><div></code> yang didalamnya terdapat tag <code><iframe></code> dengan class <i>embed-responsive-item</i> .
	<i>embed-responsive-{aspect ratio}</i>	Mengatur aspect ratio pada tag <code><iframe></code> dimana aspect ratio dapat diisi <i>21by9</i> (21:9), <i>16by9</i> (16:9), <i>4by3</i> (4:3), dan <i>1by1</i> (1:1). Digunakan pada tag <code><div></code> bersama class <i>embed-responsive</i> .
Position	<i>position-{value}</i>	Mengatur position elemen dimana value dapat diisi <i>static, relative, absolute, fixed</i> dan <i>sticky</i> .
	<i>fixed-top, fixed-bottom</i>	Mengatur position elemen menjadi fixed top atau fixed bottom.
	<i>sticky-top, sticky-bottom</i>	Mengatur position elemen menjadi sticky top atau sticky bottom.
Screen Reader	<i>sr-only, sr-only-focusable</i>	Mengatur agar elemen hanya dapat diakses melalui <i>screen reader</i> .
Sizing	<i>w-25, w-50, w-75, w-100</i>	Mengatur lebar elemen dalam satuan persen.
	<i>h-25, h-50, h-75, h-100</i>	Mengatur tinggi elemen dalam satuan persen.
Padding & Margin	<i>{property}{side}-{size}</i>	Mengatur margin atau padding elemen. Properti dapat diisi <i>m</i> atau <i>/</i> yang merupakan singkatan margin dan padding. Side dapat diisi <i>t, b, l, r, x</i> dan <i>y</i> yang merupakan singkatan

		dari top, bottom, left dan right. Adapun x menunjukkan kanan dan kiri, sedangkan y menunjukkan atas dan bawah. Size dapat diisi 0 sampai 5 atau auto. Contoh: p-5, m-2, ml-3, pt-auto, mx-0.
Vertical Alignment	<i>align-{value}</i>	Mengatur perataan secara vertikal dimana value dapat diisi <i>baseline</i> , <i>top</i> , <i>middle</i> , <i>bottom</i> , <i>text-top</i> , dan <i>text-bottom</i> .
Visibility	<i>visible</i> , <i>invisible</i>	Mengatur visibility elemen dimana visible berarti terlihat sedangkan invisible berarti tidak terlihat.

Berikut ini contoh penggunaan dari beberapa class di atas, terutama pada tabel yang sering digunakan.

File 8_3_utilities.html

```
...
<div cl ass="cont ai ner">
<h1 cl ass="di spl ay-4"> Jadwal Uji an </h1>

<div cl ass="table-responsi ve">
<table cl ass="table table-bordered table-striped table-hover">
<thead>
<tr cl ass="table-primary">
<th>Hari </th>
<th>Jam ke 1</th>
<th>Jam ke 2</th>
</tr>
</thead>
<tbody>
<tr>
<td cl ass="text-uppercase">Seni n</td>
<td cl ass="text-center">Bahasa I ndonesi a</td>
<td cl ass="text-center bg-danger text-white">Matemati ka</td>
</tr>
<tr>
<td cl ass="text-uppercase">sel asa</td>
<td cl ass="text-center">Bahasa I nggris</td>
<td cl ass="text-center">Kej uruan</td>
</tr>
</tbody>
</table>
</div>
</div>
```

Pada skrip di atas dan pada skrip selanjutnya sengaja tidak dituliskan secara lengkap karena pada bagian atas maupun bawah pada setiap file

pasti sama, sehingga tinggal ditambahkan saja mengikuti file yang lain. Jadi, tanda titik tiga pada skrip di atas menandakan masih ada skrip pada bagian tersebut. Hasil dari skrip di atas seperti gambar berikut:

The screenshot shows a web browser window with the title '8_3_utilities.html'. The page displays a table titled 'Jadwal Ujian' (Schedule). The table has three columns: 'Hari' (Day), 'Jam ke 1' (Time 1), and 'Jam ke 2' (Time 2). The data is as follows:

Hari	Jam ke 1	Jam ke 2
SENIN	Bahasa Indonesia	Matematika
SELASA	Bahasa Inggris	Kejuruan

Gambar 8.2 Hasil penggunaan css utilities

8.4 Komponen Bootstrap

8.4.1 Button

Button pada Bootstrap dapat dibuat dengan tag `<a>`, `<button>`, `<label>` dan `<input>` yang diberi class `btn`. Desain button pada Bootstrap memiliki banyak variasi. Berikut ini berbagai class yang dapat digunakan bersama class `btn` untuk menghasilkan desain button yang berbeda.

Class	Penjelasan
<code>btn-{color}</code>	Membuat tombol dengan warna tertentu. Color dapat diisi <code>primary</code> , <code>secondary</code> , <code>success</code> , <code>danger</code> , <code>warning</code> , <code>info</code> , <code>light</code> dan <code>dark</code> .
<code>btn-link</code>	Membuat tombol dengan desain seperti link.
<code>btn-outline-{color}</code>	Membuat tombol dengan outline berwarna. Warna yang dapat digunakan sama seperti sebelumnya.
<code>btn-lg</code> , <code>btn-sm</code>	Mengubah ukuran tombol. Kata <code>lg</code> berarti besar (<code>large</code>) dan <code>sm</code> berarti kecil (<code>small</code>).
<code>active</code>	Membuat tombol aktif.
<code>disabled</code>	Membuat tombol tidak dapat diklik. Jika pada tag <code><a></code> disertai

	dengan atribut aria-disabled="true".
<i>btn-block</i>	Membuat tombol memenuhi elemen induk.
<i>btn-group</i>	Membuat group tombol. Diterapkan pada tag <div> yang membungkus lebih dari 1 tombol (button).
<i>btn-group-toggle</i>	Membuat tombol yang juga berfungsi sebagai checkbox atau radio. Diterapkan pada tag <div> yang membungkus tag <label> atau <a> dengan class btn. Digunakan bersama atribut data-toggle="buttons". Jika sebagai radio, digunakan bersama class btn-group.
<i>btn-toolbar</i>	Membuat toolbar dengan grup tombol. Digunakan pada tag <div> yang membungkus 1 atau lebih grup tombol.
<i>btn-group-lg</i> , <i>btn-group-sm</i>	Mengatur ukuran grup tombol. Digunakan bersama dengan class btn-group.

Berikut contoh penggunaan tombol dan beberapa class di atas:

File 8_4_1_button.html

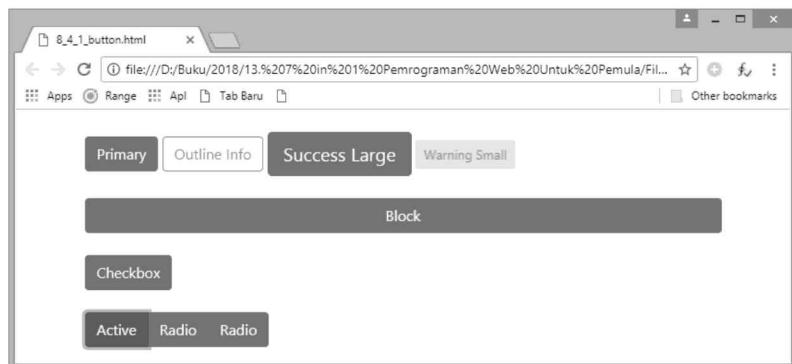
```
...
<a href="#" class="btn btn-primary">Primary</a>
<button class="btn btn-outline-info">Outline Info</button>
<a href="#" class="btn btn-success btn-lg active">Success Large</a>
<a href="#" class="btn btn-warning btn-sm disabled">Success Small</a><br><br>

<input type="button" class="btn btn-block btn-danger" value="Block"><br>

<div class="btn-group-toggle" data-toggle="buttons">
  <label class="btn btn-secondary active">
    <input type="checkbox" checked autocomplete="off" > Checkbox
  </label>
</div><br>

<div class="btn-group btn-group-toggle" data-toggle="buttons">
  <label class="btn btn-primary active">
    <input type="radio" name="options" id="option1" checked>
  Active
  </label>
  <label class="btn btn-primary">
    <input type="radio" name="options" id="option2" > Radio
  </label>
  <label class="btn btn-primary">
    <input type="radio" name="options" id="option3" > Radio
  </label>
</div>
<br>
```

Hasilnya seperti berikut:



Gambar 8.3 Contoh penggunaan button

8.4.2 Alert

Alert digunakan untuk menampilkan pesan kepada pengunjung website. Alert dibuat menggunakan tag `<div>` dengan tambahan class **alert**. Untuk membuat berbagai variasi desain alert dapat menggunakan class-class berikut:

Class	Penjelasan
<code>alert-{color}</code>	Membuat alert dengan background warna. Color dapat diisi <i>primary</i> , <i>secondary</i> , <i>success</i> , <i>danger</i> , <i>warning</i> , <i>info</i> , <i>light</i> dan <i>dark</i> .
<code>alert-link</code>	Menandai link di dalam alert sehingga warnanya menyesuaikan warna yang digunakan pada alert. Digunakan pada tag <code><a></code> .
<code>alert-heading</code>	Membuat heading di dalam alert. Digunakan pada tag <code><h1></code> sampai <code><h6></code> .
<code>alert-dismissible</code>	Membuat alert dapat disembunyikan dengan tombol tertentu.

Contoh penggunaan alert bersama class-class di atas dapat dilihat pada skrip berikut:

File 8_4_2_alert.html

```
<div class="alert alert-primary" role="alert">
    Ini alert primary dengan <a href="#" class="alert-link">
    sebuah link </a>. Silakan klik link.
</div>
```

```

<div class="alert alert-success" role="alert">
  <h4 class="alert-heading">Judul alert!</h4>
  <p>Ini contoh alert dengan judul.</p>
</div>

<div class="alert alert-warning alert-dismissible fade show"
role="alert">
  <p>Klik silang untuk menyembunyikan alert</p>
  <button type="button" class="close" data-dismiss="alert"
aria-label="Close">
    <span aria-hidden="true">&times;</span>
  </button>
</div>
...

```



Gambar 8.4 Contoh penggunaan alert

8.4.3 Badge

Badge digunakan untuk memberi background pada teks yang biasanya berupa angka dengan border yang lebih membulat. Badge dapat diletakkan pada berbagai komponen lain seperti button, nav, navbar dan sebagainya. Badge biasanya dibuat dengan tag `` yang diberi class `badge`. Untuk mengatur warna badge dapat menggunakan class `badge-{color}` dimana color dapat diisi `primary`, `secondary`, `success`, `warning`, `danger`, `info`, `light` dan `dark`. Untuk membuat badge berbentuk kapsul dapat menggunakan class `badge-pill`. Berikut contoh penggunaanya:

File 8_4_3_badge.html

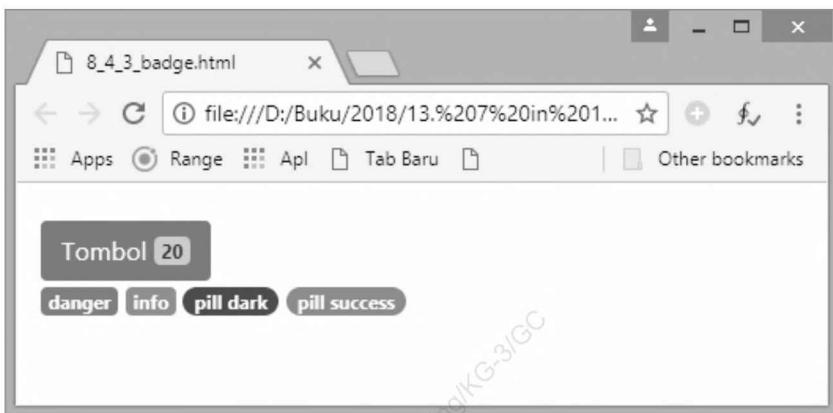
```
<button type="button" class="btn btn-primary"> Tombol
```

```
<span class="badge badge-warning">20</span>
</button><br>

<span class="badge badge-danger">danger</span>
<span class="badge badge-info">info</span>
<span class="badge badge-dark badge-pill">pill dark</span>
<span class="badge badge-success badge-pill">pill success</span>
...

```

Hasilnya seperti gambar berikut:



Gambar 8.5 Contoh penggunaan badge

8.4.4 Breadcrumb

Breadcrumb digunakan untuk menunjukkan posisi halaman yang sedang dibuka dari halaman utama website. Breadcrumb dibuat menggunakan tag `` atau `` yang diberi class `breadcrumb`. Masing-masing item breadcrumb diberi class `breadcrumb-item`. Pemisah antar link akan otomatis ditambahkan pada CSS sehingga tidak perlu menambahkannya secara manual. Untuk alasan accessibility, akan lebih baik jika membungkus breadcrumb dengan tag `<nav>` dengan atribut `aria-label="breadcrumb"` dan menambahkan atribut `aria-current="page"` dan class `active` pada item terakhir.

Skrip 8_4_4Breadcrumb.html

```
...
<nav aria-label="breadcrumb">
  <ol class="breadcrumb">
    <li class="breadcrumb-item"><a href="#">Home</a></li>

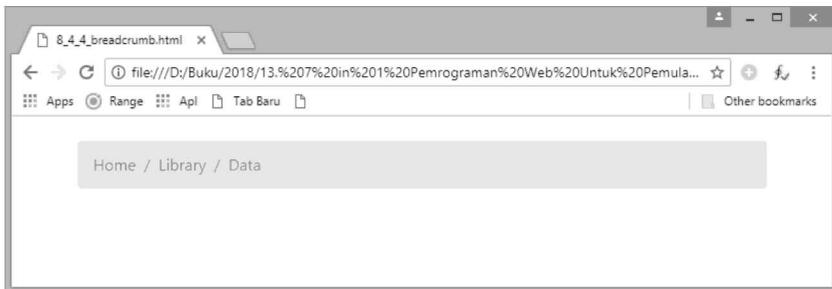
```

```

<li class="breadcrumb-item"><a href="#">Library</a></li>
<li class="breadcrumb-item active" aria-current="page">Data</li>
</ol>
</nav>
...

```

Skrip di atas akan menghasilkan breadcrumb seperti gambar berikut:



Gambar 8.6 Contoh penggunaan breadcrumb

8.4.5 Card

Card merupakan sebuah box dengan border tipis yang dapat diisi dengan berbagai macam konten untuk berbagai macam keperluan. Card dibuat dengan tag <div> yang diberi class **card**. Dalam card tersebut dapat terdiri dari berbagai macam konten dan masing-masing konten memiliki class khusus.

Berikut berbagai class yang digunakan pada card:

Class	Penjelasan
<i>card-header</i>	Membuat header card.
<i>card-footer</i>	Membuat footer card.
<i>card-body</i>	Membuat bagian utama card yang dapat diisi berbagai macam konten.
<i>card-title</i>	Membuat judul di dalam card bersama tag <h1> sampai <h6>.
<i>card-subtitle</i>	Membuat judul yang lebih kecil di dalam card.
<i>card-link</i>	Membuat link bersama tag <a>.
<i>card-img-top</i>	Menampilkan gambar pada bagian atas konten card bersama tag .

<i>card-img-bottom</i>	Menampilkan gambar pada bagian bawah konten card bersama tag .
<i>card-text</i>	Menampilkan teks paragraf bersama tag <p> di dalam card.
<i>card-header-tabs</i>	Membuat header card yang juga difungsikan sebagai tab.
<i>card-header-pills</i>	Membuat header card yang juga difungsikan sebagai nav pills.
<i>card-img-overlay</i>	Membuat gambar sebagai background pada card.
<i>card-group</i>	Membuat grup card yang berisi sejumlah card.
<i>card-deck</i>	Membuat grup card yang berisi sejumlah card, dan masing-masing card memiliki jarak dan sama tingginya.
<i>card-columns</i>	Membuat grup card dan masing-masing card di dalamnya disusun secara vertikal.

Berikut contoh skrip untuk membuat card:

File 8_4_5_card.html

```
<div class="card w-75">
  
  <div class="card-body">
    <p class="card-text">Lorem ipsum dolor sit ...</p>
  </div>
  <div class="card-footer">Footer</div>
</div>
...
```

Skrip di atas merupakan contoh sederhana penggunaan card. Pada pembahasan komponen Bootstrap yang lain akan dibahas lagi penggunaan card bersama komponen yang lebih kompleks. Skrip di atas menghasilkan card seperti gambar berikut:



Gambar 8.7 Contoh penggunaan card

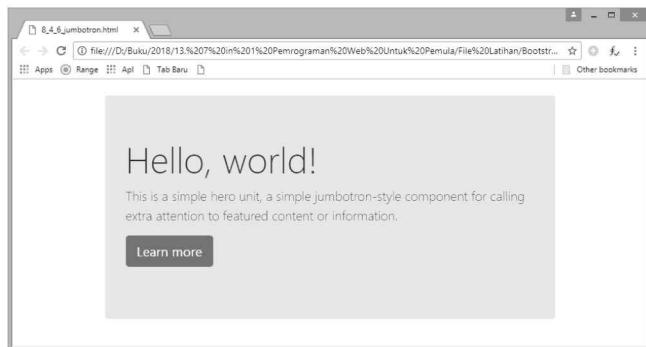
8.4.6 Jumbotron

Jumbotron digunakan untuk memberi blok pada elemen website yang ingin ditonjolkan atau kata-kata kunci pada website dan biasanya diletakkan pada bagian atas website di bawah menu utama. Untuk membuat jumbotron yaitu dengan menambah class **jumbotron** pada tag <div>.

Berikut contoh penggunaanya:

Skrip 8_4_6_jumbotron.html

```
<div cl ass="j umbotron">
  <h1 cl ass="di spl ay- 4">Hello, world!</h1>
  <p cl ass="l ead">This is a simple hero unit...</p>
  <p cl ass="l ead">
    <a cl ass="bt n bt n- primary bt n- lg" href="#" rol e="button">Learn more</a>
  </p>
</di v>
...
```



Gambar 8.8 Contoh penggunaan jumbotron

8.4.7 List Group

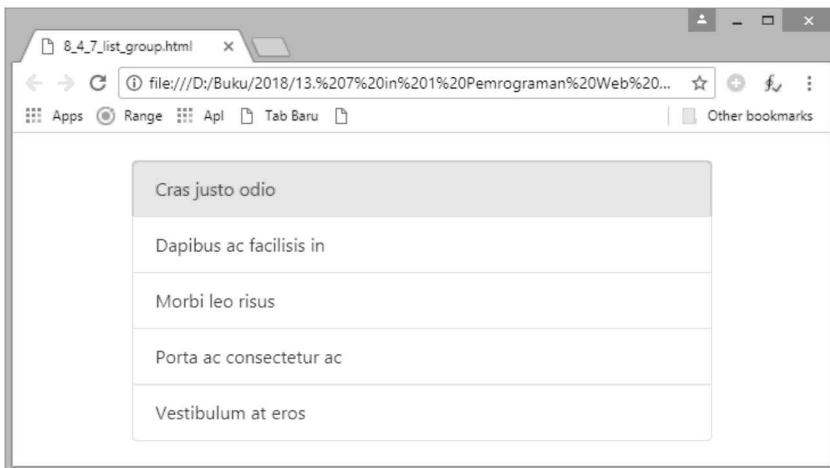
List group akan menampilkan list dengan tampilan yang lebih elegan. Selain itu, dapat juga menampilkan berbagai macam komponen di dalam list group. List group dibuat dengan tag `` yang diberi class **list-group**. Adapun masing-masing list item dibuat dengan tag `` yang diberi class **list-group-item**. List group juga dapat dibuat dengan tag `<div>` dan masing-masing item list dengan tag `<a>` jika item juga berfungsi sebagai link. Pada tag `<a>` tambahkan class **list-group-item-action**.

Untuk menghilangkan border di sekeliling list group dapat ditambahkan class **list-group-flush** pada tag ``. List item pada list group juga dapat diberi warna dengan class **list-group-item-{color}**. Berikut contoh skrip untuk membuat list group:

File 8_4_7_list_group.html

```
<ul class="list-group">
  <li class="list-group-item list-group-item-primary">Cras
justo odi o</li>
  <li class="list-group-item">Dapi bus ac facilisis in</li>
  <li class="list-group-item">Morbi leo risus</li>
  <li class="list-group-item">Porta ac consectetur ac</li>
  <li class="list-group-item">Vestibulum at eros</li>fca
</ul>
...
```

Hasil dari skrip di atas seperti gambar berikut:



Gambar 8.9 Contoh penggunaan list group

8.4.8 Pagination

Pagination merupakan kumpulan link yang digunakan untuk berpindah ke nomor halaman yang lain. Pagination dibuat menggunakan tag yang diberi class **pagination** dan masing-masing item diberi class **page-item**. Adapun link halaman diberi class **page-link**. Untuk menghasilkan ukuran yang berbeda dapat menambahkan class **pagination-lg** atau **pagination-sm**.

Berikut contoh penggunaan pagination:

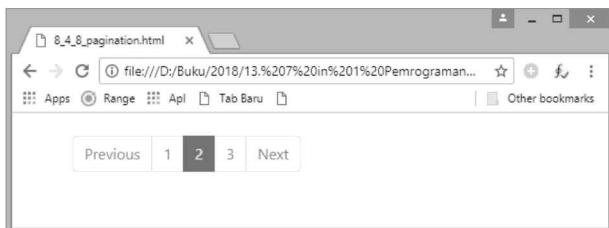
Skrip 8_4_8_pagination.html

```
<ul class="pagination">
    <li class="page-item disabled">
        <a class="page-link" href="#">Previous</a>
    </li>
    <li class="page-item">
        <a class="page-link" href="#">1</a>
    </li>
    <li class="page-item active">
        <a class="page-link" href="#">2 <span class="sr-only" style="display:none">(current)</span></a>
    </li>
    <li class="page-item">
        <a class="page-link" href="#">3</a>
    </li>
    <li class="page-item">
```

```
</li>
</ul>
...

```

Skrip di atas menghasilkan pagination seperti berikut:



Gambar 8.10 Contoh penggunaan pagination

8.4.9 Media Object

Media object merupakan elemen khusus untuk mengatur layout berbagai konten website seperti komentar blog, tweet, dan sebagainya, dengan susunan avatar di sebelah kiri dan teks paragraf di sebelah kanan.

Media objek dibuat menggunakan tag `<div>` dengan tambahan class **media**. Di dalamnya berisi image dan tag `<div>` yang diberi class **media-body** sebagai konten utama. Berikut contoh penggunaan media object.

File 8_4_9_media_object.html

```
...
<div class="media">
  
  <div class="media-body">
    <h5 class="mt-0">Media heading</h5>
    Cras sit amet nibh libero, ...
  <div class="media mt-3">
    <a class="pr-3" href="#">
      
    </a>
    <div class="media-body">
      <h5 class="mt-0">Media heading</h5>
      Cras sit amet nibh libero, ...
    </div>
  </div>
</div>

```

```

</div>
<div v-cl ass="media">
  
  <div v-cl ass="media-body">
    <h5 cl ass="mt-0">Media heading</h5>
    Cras sit amet nibh libero, ...
  </div>
</div>
...

```

Pada skrip di atas terdapat media lagi di dalam media body, sehingga hasilnya seperti gambar berikut:



Gambar 8.11 Contoh penggunaan media object

8.4.10 Form

Bootstrap menyediakan desain yang menarik pada berbagai elemen pada form. Setiap elemen memiliki class khusus yang harus ditambahkan ketika membuat form agar form terlihat menarik. Daftar class yang dapat digunakan pada elemen form dapat dilihat pada tabel berikut:

Class	Penjelasan
<i>form-group</i>	Diberikan pada tag <div> yang membungkus satu set elemen input yang terdiri dari label, alat input dan teks bantuan atau pesan validasi.
<i>form-control</i>	Diberikan pada tag <input>, <select> atau <textarea> yang berfungsi sebagai alat inputan.

<i>form-text</i>	Diberikan pada tag <small> setelah alat input untuk membuat teks bantuan.
<i>form-control-file</i>	Diberikan khusus pada tag <input> dengan tipe file.
<i>form-control-lg, form-control-sm</i>	Diberikan pada alat input bersama class form-control untuk mengatur ukuran alat input.
<i>form-control-plaintext</i>	Diberikan pada alat input menggantikan class form-control untuk menghilangkan desain default dan akan ditampilkan seperti teks biasa.
<i>form-check</i>	Diberikan pada tag <div> yang di dalamnya berisi checkbox atau radio.
<i>form-check-box</i>	Diberikan pada tag <input> dengan tipe checkbox atau radio menggantikan class form-control.
<i>form-check-label</i>	Diberikan pada tag <label> yang menjadi label checkbox.
<i>form-check-inline</i>	Digunakan bersama class form-check untuk membuat tampilan checkbox atau radio tersusun horizontal.
<i>form-row</i>	Diberikan pada tag <div> menggantikan class row untuk penggunaan grid pada form.
<i>col-form-label</i>	Diberikan pada tag <label> pada form yang tersusun horizontal.
<i>col-form-label-sm, col-form-label-lg</i>	Diberikan pada tag <label> untuk mengatur ukuran label form.
<i>form-inline</i>	Diberikan pada tag <form> untuk membuat form yang disusun dalam satu baris.
<i>is-valid, is-invalid</i>	Diberikan pada alat input untuk mengatur tipe validasi.
<i>valid-feedback, invalid-feedback</i>	Diberikan pada tag <div> setelah alat input untuk menandai teks validasi.
<i>custom-select</i>	Diberikan pada tag <select> untuk membuat desain select box yang menarik.
<i>custom-select-sm, custom-select-lg</i>	Mengatur ukuran select box.
<i>custom-control</i>	Diberikan pada tag <div> yang akan berisi checkbox atau radio dengan desain khusus.
<i>custom-checkbox</i>	Digunakan bersama class custom-control untuk membuat desain checkbox lebih menarik.
<i>custom-radio</i>	Digunakan bersama class custom-control untuk membuat desain radio lebih menarik.
<i>custom-control-input</i>	Digunakan pada tag <input> dengan tipe checkbox atau radio yang menggunakan desain khusus.
<i>custom-control-label</i>	Diberikan pada tag <label> yang menjadi label dari custom-checkbox atau custom-radio.
<i>custom-control-line</i>	Digunakan bersama class custom-control untuk membuat tampilan checkbox atau radio menjadi mendatar.
<i>valid-tooltip, invalid-</i>	Menggantikan class valid-feedback atau invalid-feedback

<i>tooltip</i>	untuk membuat tampilan pesan validasi seperti tooltip.
<i>custom-file</i>	Diberikan pada tag <div> yang di dalamnya berisi tag <input> dengan tipe file untuk membuat desain input file lebih menarik.
<i>custom-file-input</i>	Diberikan pada tag <input> dengan tipe file untuk membuat desain input file lebih menarik.
<i>custom-file-label</i>	Diberikan pada label untuk tag <input> yang diberi class custom-file-input.

Berikut ini contoh penggunaan berbagai class dalam menampilkan form:

File 8_4_10_form.html

```
<form>
  <div class="form-row">
    <div class="form-group col-md-6">
      <label for="inputEmail4">Email</label>
      <input type="email" class="form-control is-valid" id="inputEmail4" placeholder="Email" />
      <div class="valid-feedback">Email is valid</div>
    </div>
    <div class="form-group col-md-6">
      <label for="inputPassword4">Password</label>
      <input type="password" class="form-control is-invalid" id="inputPassword4" placeholder="Password" />
      <div class="invalid-feedback">Password not valid</div>
    </div>
  </div>

  <div class="form-group">
    <label for="inputAddress">Address</label>
    <input type="text" class="form-control form-control-lg" id="inputAddress" placeholder="1234 Main St" />
  </div>

  <div class="form-group row">
    <label for="inputCity" class="col-sm-2 col-form-label">State</label>
    <div class="col-sm-4">
      <select id="inputCity" class="custom-select">
        <option selected>Choose...</option>
        <option>...</option>
      </select>
    </div>
  </div>

  <div class="form-group">
    <div class="custom-control custom-radio custom-control-line">
      <input type="radio" id="customRadio1" name="customRadio" class="custom-control-input" />
      <label for="customRadio1" class="custom-control-label">Toggle this custom radio</label>
    </div>
  </div>
</form>
```

```

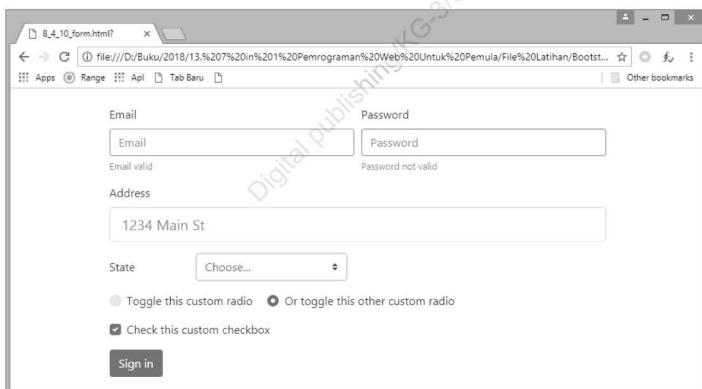
<div class="custom-control custom-radio custom-control-inline">
    <input type="radio" id="customRadio2" name="customRadio" class="custom-control-input">
    <label class="custom-control-label" for="customRadio2">Or toggle this other custom radio</label>
</div>
</div>

<div class="form-group">
    <div class="custom-control custom-checkbox">
        <input type="checkbox" id="customCheck1" class="custom-control-input">
        <label class="custom-control-label" for="customCheck1">Check this custom checkbox</label>
    </div>
</div>

<button type="submit" class="btn btn-primary">Sign in</button>
</form>
...

```

Skrip di atas sudah termasuk contoh penggunaan validasi, form horisontal, penggunaan grid, ukuran form dan custom control. Hasilnya seperti gambar berikut:



Gambar 8.12 Contoh penggunaan form

8.4.11 Input Group

Input group yaitu input dengan tambahan tombol atau teks yang menempel di sebelah kanan atau di sebelah kirinya. Input group dibuat dengan tag `<div>` yang diberi class **input-group**. Class lain yang digunakan untuk menyusun input group yaitu sebagai berikut:

Class	Penjelasan
<i>input-group-prepend</i>	Diberikan pada tag <div> sebelum alat input untuk menambahkan tombol atau teks di sebelah kiri alat input.
<i>input-group-append</i>	Diberikan pada tag <div> setelah alat input untuk menambahkan tombol atau teks di sebelah kanan alat input.
<i>input-group-sm</i> , <i>input-group-lg</i>	Digunakan bersama class input-group untuk mengatur ukuran input group.

Berikut contoh penggunaan input group dan variasinya:

File 8_4_11_input_group.html

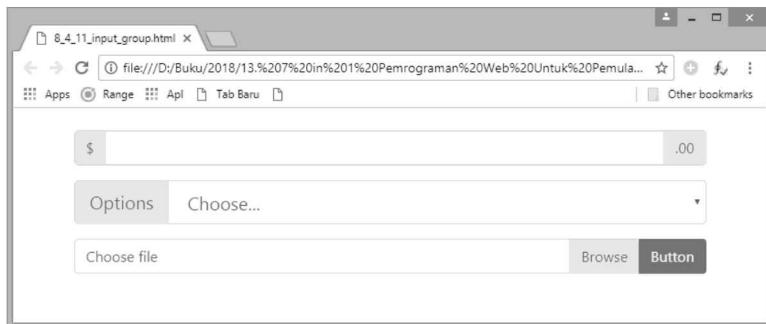
```
<div v class="input-group mb-3">
  <div v class="input-group-prepend">
    <span class="input-group-text">$</span>
  </div v>
  <input type="text" class="form-control" <div v class="input-group-append">
    <span class="input-group-text">.00</span>
  </div v>
</div v>

<div v class="input-group input-group-lg mb-3">
  <div v class="input-group-prepend">
    <label class="input-group-text">
      for="inputGroupSelect01">Options</label>
  </div v>
  <select class="form-control" id="inputGroupSelect01">
    <option selected>Choose...</option>
    <option value="1">One</option>
    <option value="2">Two</option>
    <option value="3">Three</option>
  </select>
</div v>

<div v class="input-group">
  <div v class="custom-file">
    <input type="file" class="custom-file-input" id="inputGroupFile04">
    <label class="custom-file-label">
      for="inputGroupFile04">Choose file</label>
  </div v>
  <div v class="input-group-append">
    <button class="btn btn-primary" type="button">Button</button>
  </div v>
</div v>
...
```

Pada contoh di atas terdiri dari penggunaan input group dengan input text, select dan custom file. Termasuk penggunaan input group dengan

variasi ukuran, input group dengan teks dan input group dengan tombol. Hasilnya seperti gambar berikut:



Gambar 8.13 Contoh penggunaan input group

8.4.12 Progress Bar

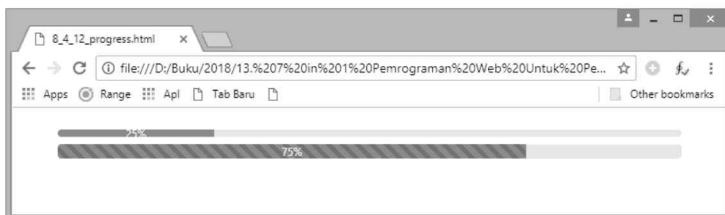
Progress bar merupakan elemen yang memberikan informasi persentase sebuah proses yang telah dilalui dalam bentuk bar. Progress bar dibuat dengan tag `<div>` yang diberi class **progress** di dalamnya berisi tag `<div>` dengan class **progress-bar**. Variasi yang dapat digunakan yaitu class **progress-bar-striped** untuk membuat progress bar dengan tekstur garis-garis dan class **progress-bar-animated** untuk membuat progress bar yang mengandung animasi.

Contoh penggunaanya dapat dilihat pada skrip berikut:

File 8_4_12_progress.html

```
<div class="progress" style="height: 4px">
  <div class="progress-bar bg-success" role="progressbar" aria-valuenow="75" aria-valuemin="0" aria-valuemax="100" style="width: 25%">25%</div>
</div>
<div class="progress">
  <div class="progress-bar progress-bar-striped progress-bar-animated" role="progressbar" aria-valuenow="75" aria-valuemin="0" aria-valuemax="100" style="width: 75%">75%</div>
</div>
...
```

Pada skrip di atas, untuk mengatur tinggi bar menggunakan inline CSS dengan properti height. Untuk variasi warna dapat menggunakan class `bg-{color}`. Hasilnya seperti gambar berikut:



Gambar 8.14 Contoh penggunaan progress bar

8.4.13 Tooltip

Tooltip yaitu teks yang muncul ketika pointer mouse menunjuk suatu elemen website. Untuk membuat tooltip, tambahkan atribut `data-toggle="tooltip"` dan atribut `title="..."`. Teks yang menjadi tooltip, yaitu teks yang ditulis pada atribut title. Tooltip juga dapat berupa kode HTML yang ditulis pada atribut title. Untuk mengatur posisi tooltip dapat menggunakan atribut `data-placement="top|bottom|left|right"`. Untuk menampilkan tooltipnya, gunakan skrip jQuery seperti berikut:

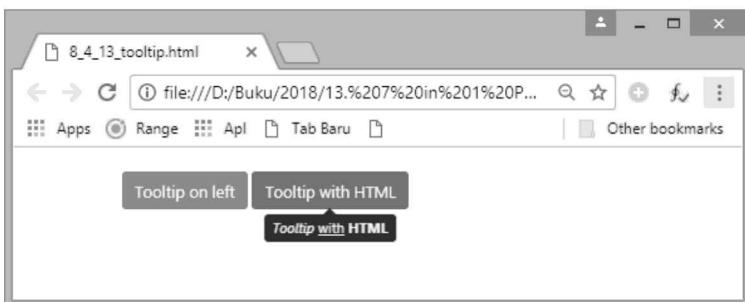
```
$(function () {
  $('[data-toggle="tooltip"]').tooltip()
})
```

Berikut contoh skrip tooltip dalam berbagai posisi:

File 8_4_13_tooltip.html

```
<button type="button" class="btn btn-secondary"
data-toggle="tooltip" data-placement="left" title="Tooltip on
left">
  Tooltip on left
</button>
<button type="button" class="btn btn-secondary"
data-toggle="tooltip" data-html="true" title=<em>Tooltip</em>
<u>with</u> <b>HTML</b>>
  Tooltip with HTML
</button>
...
```

Skrip di atas terdiri dari 2 tooltip dimana salah satunya menggunakan teks biasa dan satunya lagi dengan kode HTML. Hasilnya seperti gambar berikut:



Gambar 8.15 Contoh penggunaan tooltip

8.4.14 Popover

Popover hampir sama dengan tooltip, hanya saja pada popover dapat berisi judul dan konten. Selain itu, popover baru muncul ketika suatu elemen diklik dan akan menghilang ketika klik yang kedua kalinya. Popover dibuat dengan menambahkan atribut **data-toggle="popover"**. Sama dengan tooltip, untuk mengatur penempatan dapat menggunakan atribut **data-placement=""**.

Berikut ini contoh penggunaan popover pada HTML:

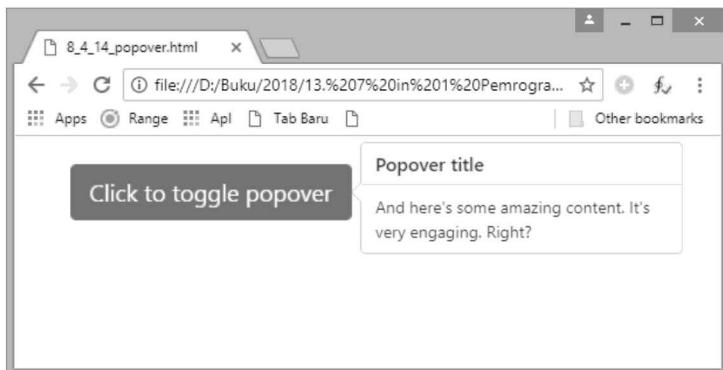
File 8_4_14_popover.html

```
<button type="button" class="btn btn-danger"
data-toggle="popover" title="Popover title" data-content="And here's ... "
```

Pada skrip di atas, isi atribut **title** akan dijadikan sebagai judul popover, sedangkan atribut **data-content** dijadikan sebagai konten popover. Popover juga menggunakan plugin jQuery, sehingga dalam penggunaanya perlu menambahkan skrip sebagai berikut:

```
$(function () {
  $('[data-toggle="popover"]').popover()
})
```

Hasilnya seperti gambar berikut:



Gambar 8.16 Contoh penggunaan popover

8.4.15 Dropdown

Dropdown yaitu suatu elemen, biasanya berupa list yang tampil ketika sebuah tombol diklik. Dropdown dibuat dengan tag <div> yang diberi class **dropdown-menu** dan masing-masing item dibuat dengan tag <a> yang diberi class **dropdown-item**. Adapun tombol yang mengakses dropdown diberi atribut **data-toggle="dropdown"**. Ada beberapa class yang digunakan bersama dropdown yaitu sebagai berikut:

Class	Penjelasan
<i>dropdown-toggle</i>	Diberikan pada tombol (button) yang mengakses dropdown.
<i>dropdown-toggle-split</i>	Digunakan pada tombol yang mengakses dropdown agar segitiga penunjuk tidak menggabung dengan tombol.
<i>dropup, dropright, dropleft</i>	Digunakan bersama class btn-group yang membungkus tombol agar arah dropdown mengarah ke sisi lain (tidak ke bawah).
<i>dropdown-menu-right</i>	Digunakan bersama class dropdown-menu agar menu dropdown rata kanan dengan tombol.
<i>dropdown-header</i>	Membuat item dropdown yang berfungsi sebagai header.

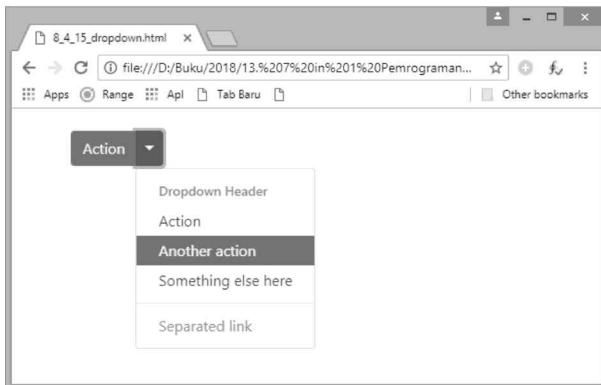
<i>dropdown-divider</i>	Membuat garis pemisah antara 2 item dropdown.
<i>active</i>	Menunjukkan item dropdown yang aktif.
<i>disabled</i>	Menunjukkan item dropdown yang tidak dapat diklik.

Berikut contoh penggunaan dropdown beserta variasinya:

File 8_4_15_dropdown.html

```
<div class="btn-group">
  <button type="button" class="btn btn-danger">Action</button>
  <button type="button" class="btn btn-danger dropdown-toggle dropdown-toggle-split" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
    <span class="sr-only">Toggle Dropdown</span>
  </button>
  <div class="dropdown-menu">
    <h6 class="dropdown-header">Dropdown Header</h6>
    <a class="dropdown-item" href="#">Action</a>
    <a class="dropdown-item active" href="#">Another action</a>
    <a class="dropdown-item" href="#">Something else here</a>
    <div class="dropdown-divider"></div>
    <a class="dropdown-item disabled" href="#">Separated link</a>
  </div>
</div>
<...>
```

Pada skrip di atas menggunakan dropdown header dan dropdown divider. Terdapat juga contoh penggunaan class **active** dan **disabled**. Hasilnya seperti gambar berikut:



Gambar 8.17 Contoh penggunaan dropdown

8.4.16 Collapse

Collapse yaitu elemen yang dapat ditampilkan dan disembunyikan dengan cara mengarahkan target ke elemen tersebut. Berbeda dengan dropdown yang bersifat absolute, collapse tidak bersifat absolute sehingga akan berpengaruh terhadap elemen di sekitarnya. Collapse dibuat dengan tag <div> yang diberi class **collapse**. Tombol yang membuka dan menutup collapse diberi atribut **data-toggle="collapse"**. Untuk mengarahkan target ke collapse, sebuah tombol dapat menggunakan atribut **href=""** atau **data-target=""**.

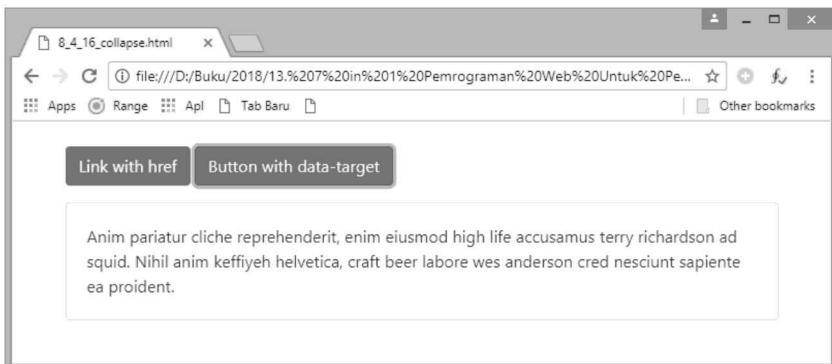
Contoh penggunaan collapse dapat dilihat pada skrip berikut:

File 8_4_16_collapse.html

```
...
<p>
  <a class="btn btn-primary" data-toggle="collapse"
    href="#collapseExample" role="button" aria-expanded="false"
    aria-controls="collapseExample">
    Link with href
  </a>
  <button class="btn btn-primary" type="button" data-
    toggle="collapse" data-target="#collapseExample" aria-
    expanded="false" aria-controls="collapseExample">
    Button with data-target
  </button>
</p>
<div class="collapse" id="collapseExample">
  <div class="card card-body">
    Anim pariatur cliche reprehenderit, ...
  </div>
</div>
...

```

Skrip di atas terdiri dari 2 tombol yang mengakses 1 target collapse, salah satunya menggunakan atribut href sedangkan yang satunya lagi menggunakan atribut data-target. Hasilnya seperti gambar berikut:



Gambar 8.18 Contoh penggunaan collapse

8.4.17 Modal

Modal yaitu kotak dialog yang dapat ditampilkan dengan suatu tombol. Modal terdiri dari header, konten dan footer. Modal dapat diisi dengan berbagai macam konten seperti tabel, form bahkan layout dengan grid. Modal dibuat dengan tag <div> yang diberi class **modal**. Tombol yang mengakses modal diberi atribut **data-toggle="modal"**. Sedangkan untuk mengarahkan ke target modal, pada tombol diberi atribut **data-target=""**. Berbagai class yang digunakan pada modal diantaranya sebagai berikut:

Class	Penjelasan
<i>modal-dialog</i>	Diberikan pada tag <div> yang berada di dalam tag <div> dengan class modal.
<i>modal-content</i>	Diberikan pada tag <div> yang berada di dalam tag <div> dengan class modal-dialog.
<i>modal-header</i>	Diberikan pada tag <div> untuk menunjukkan header modal.
<i>modal-body</i>	Diberikan pada tag <div> untuk menunjukkan konten utama modal.
<i>modal-footer</i>	Diberikan pada tag <div> untuk menunjukkan footer modal.
<i>modal-title</i>	Digunakan pada tag heading untuk membuat judul modal di dalam modal header.
<i>modal-dialog-centered</i>	Digunakan bersama class modal-dialog untuk membuat posisi modal dialog tepat di tengah layar secara vertikal.

<i>fade</i>	Digunakan bersama class modal untuk mengatur animasi menjadi fade saat modal tampil dan menghilang.
<i>modal-lg, modal-sm</i>	Digunakan bersama class modal-dialog untuk mengatur ukuran modal.

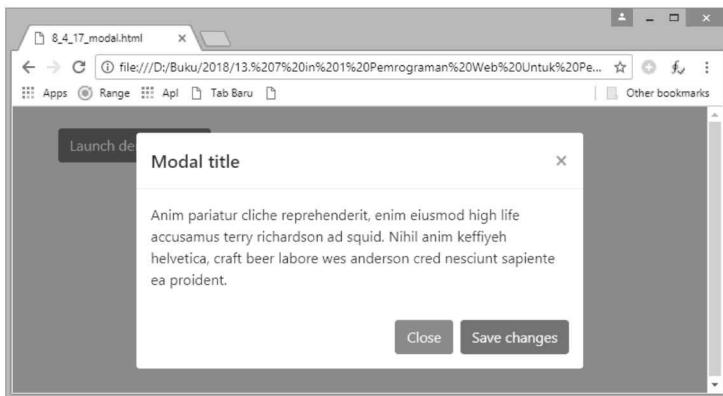
Contoh penggunaan modal dapat dilihat pada skrip berikut:

File 8_4_17_modal.html

```
<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModalCenter">
    Launch demo modal
</button>

<div class="modal fade" id="exampleModalCenter" tabindex="-1" role="dialog" aria-labelledby="exampleModalCenterTitle" aria-hidden="true">
    <div class="modal-dialog modal-lg modal-dialog-centered" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="exampleModalLongTitle">
                    Modal title</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                Anim pariatur cliche reprehenderit, ...
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
                <button type="button" class="btn btn-primary">Save changes</button>
            </div>
        </div>
    </div>
</div>
...
```

Contoh skrip modal di atas menggunakan variasi class **modal-lg** dan **modal-dialog-centered**. Hasilnya seperti gambar berikut:



Gambar 8.19 Contoh penggunaan modal

8.4.18 Carousel

Carousel yaitu elemen yang menampilkan slideshow gambar. Carousel dibuat dengan tag <div> yang diberi class **carousel**. Berbagai class yang digunakan pada carousel diantaranya sebagai berikut:

Class	Penjelasan
<i>carousel-inner</i>	Diberikan pada tag <div> yang berada di dalam tag <div> dengan class carousel untuk membuat konten carousel.
<i>carousel-item</i>	Diberikan pada tag <div> untuk membuat item carousel.
<i>active</i>	Digunakan bersama class carousel-item untuk menunjukkan item yang sedang aktif.
<i>carousel-control-prev</i>	Digunakan pada tag <a> yang berfungsi sebagai pengontrol carousel ke kiri.
<i>carousel-control-next</i>	Diberikan pada tag <a> yang berfungsi sebagai pengontrol carousel ke kanan.
<i>carousel-control-prev-icon</i>	Membuat ikon panah ke kiri.
<i>carousel-control-next-icon</i>	Membuat ikon panah ke kanan.
<i>carousel-caption</i>	Digunakan pada tag <div> yang berisi deskripsi gambar pada item carousel.
<i>carousel-indikator</i>	Diberikan pada tag yang menjadi indikator carousel.

Contoh penggunaan crousel dapat dilihat pada skrip berikut:

File 8_4_18_carousel.html

```
<div id="carousel" class="carousel slide" data-slide="carousel">

<ol class="carousel-indicators">
    <li data-target="#carousel" data-slide-to="0" class="active"></li>
    <li data-target="#carousel" data-slide-to="1"></li>
    <li data-target="#carousel" data-slide-to="2"></li>
</ol>

<div class="carousel-inner">
    <div class="carousel-item active">
        
        <div class="carousel-caption d-none d-md-block">
            <h5>Caption Title</h5>
            <p>Caption content here</p>
        </div>
    </div>
    <div class="carousel-item">
        
    </div>
    <div class="carousel-item">
        
    </div>
    </div>

    <a class="carousel-control-prev" href="#carousel" role="button" data-slide="prev">
        <span class="carousel-control-prev-icon" aria-hidden="true"></span>
        <span class="sr-only">Previous</span>
    </a>

    <a class="carousel-control-next" href="#carousel" role="button" data-slide="next">
        <span class="carousel-control-next-icon" aria-hidden="true"></span>
        <span class="sr-only">Next</span>
    </a>
</div>
...
```

Skrip di atas merupakan contoh skrip crousel yang lengkap yaitu terdapat indikator, tombol pengontrol dan caption. Hasilnya seperti gambar berikut:



Gambar 8.20 Contoh penggunaan carousel

8.4.19 Nav

Nav yaitu elemen yang digunakan sebagai navigasi, baik navigasi antar halaman yang berbeda maupun navigasi antarsemesta. Nav dibuat dengan tag `` yang diberi class **nav**. Adapun masing-masing item nav yang dibuat dengan tag `` diberi class **nav-item**. Nav juga dapat dibuat dengan tag `<nav>` dan `<a>` saja, sehingga skrip lebih simpel. Nav memiliki beberapa variasi, sehingga ada beberapa macam class yang dapat digunakan. Berikut daftar class yang berperan dalam menyusun desain nav:

Class	Penjelasan
<code>nav-link</code>	Diberikan pada tag <code><a></code> yang berfungsi sebagai link pada nav.
<code>active, disabled</code>	Diberikan pada tag <code><a></code> untuk mengatur status aktif atau disable.
<code>nav-tabs</code>	Digunakan bersama class nav untuk membuat nav dengan gaya tab.
<code>nav-pills</code>	Digunakan bersama class nav untuk membuat nav dengan gaya pill.
<code>nav-fill</code>	Digunakan bersama class nav untuk membuat nav memenuhi lebar yang disediakan.
<code>dropdown</code>	Digunakan bersama class nav-item jika item nav mengandung dropdown.
<code>tab-content</code>	Diberikan pada tag <code><div></code> yang digunakan sebagai konten tab,

	jika nav digunakan untuk navigasi tab.
<i>tab-pane</i>	Diberikan pada tag <div> yang menjadi masing-masing segment pada konten tab.
<i>show, active</i>	Digunakan bersama class tab-pane untuk menunjukkan tab yang sedang tampil (aktif).
<i>fade</i>	Digunakan bersama class tab-pane untuk membuat animasi fade saat tab diakses.

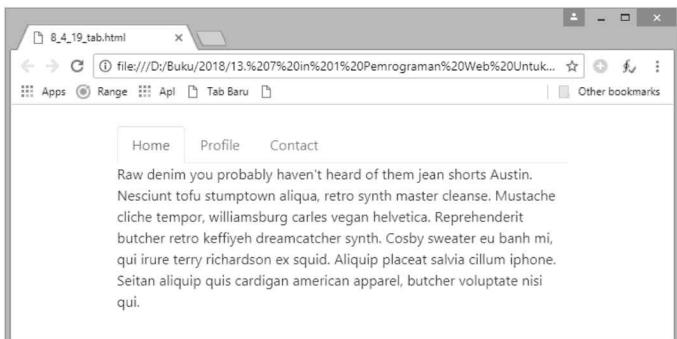
Penggunaan nav salah satunya adalah untuk membuat tab, seperti yang terlihat pada skrip berikut:

File 8_4_19_nav.html

```
<ul class="nav nav-tabs" id="myTab" role="tablist">
  <li class="nav-item">
    <a class="nav-link active" id="home-tab" data-toggle="tab"
      href="#" role="tab" aria-controls="home" aria-
      selected="true">Home</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" id="profile-tab" data-toggle="tab"
      href="#" role="tab" aria-controls="profile" aria-
      selected="false">Profile</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" id="contact-tab" data-toggle="tab"
      href="#" role="tab" aria-controls="contact" aria-
      selected="false">Contact</a>
  </li>
</ul>

<div class="tab-content" id="myTabContent">
  <div class="tab-pane fade show active" id="home"
    role="tabpanel" aria-labelledby="home-tab">
    Raw denim you probably haven't heard ...
  </div>
  <div class="tab-pane fade" id="profile" role="tabpanel" aria-
    labeledby="profile-tab">
    Food truck fixie locavore, accusamus ...
  </div>
  <div class="tab-pane fade" id="contact" role="tabpanel" aria-
    labeledby="contact-tab">
    Etsy mixtape wayfarers, ethical wes ...
  </div>
</div>
```

Jika digunakan pada tab, maka link nav diberi atribut **data-toggle="tab"** seperti pada skrip di atas. Hasilnya seperti gambar berikut:



Gambar 8.21 Contoh penggunaan nav pada tab

8.4.20 Navbar

Navbar digunakan sebagai navigasi utama yang memiliki sifat responsive, dimana ketika pada layar kecil navbar hanya menampilkan tombol yang ketika diklik akan membuka item navbar secara keseluruhan. Navbar digunakan pada tag <nav> yang diberi class **navbar**. Di dalam navbar terdiri dari beberapa elemen yang masing-masing memiliki class sendiri seperti pada tabel berikut:

Class	Penjelasan
<i>navbar-brand</i>	Digunakan pada tag <a> yang menunjukkan nama web atau nama perusahaan.
<i>navbar-nav</i>	Diberikan pada tag yang berfungsi sebagai navigasi pada navbar.
<i>navbar-toggler</i>	Digunakan pada tag <button> untuk membuat tombol yang membuka dan menutup navigasi pada layar kecil.
<i>navbar-toggler-icon</i>	Digunakan pada tag di dalam tag <button> untuk menampilkan ikon 3 garis pada tombol.
<i>navbar-text</i>	Menambahkan teks biasa pada navbar yang memiliki rata tengah vertikal.
<i>navbar-collapse</i>	Digunakan bersama class collapse pada tag <div> yang membungkus tag agar menu navigasi tersembunyi pada layar kecil dan akan tampil ketika tombol diklik dalam bentuk dropdown.
<i>navbar-{color}</i>	Digunakan bersama class bg-{color} untuk membuat navbar dengan berbagai variasi warna background.
<i>navbar-expand-</i>	Digunakan bersama class navbar untuk mengatur pada

{breakpoint}	layar ukuran berapa menu akan ditampilkan tidak dalam bentuk dropdown.
--------------	--

Berikut ini contoh penggunaan navbar, yang di dalamnya juga terdapat form.

File 8_4_20_navbar.html

```
...
<nav class="navbar navbar-expand-lg navbar-dark bg-primary" >
  <a class="navbar-brand" href="#">Navbar</a>

  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarContent" aria-controls="navbarContent" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>

  <div class="collapse navbar-collapse" id="navbarContent" >

    <ul class="navbar-nav mr-auto" >
      <li class="nav-item active" >
        <a class="nav-link" href="#">Home <span class="sr-only" >(current)</span></a>
      </li >
      <li class="nav-item" >
        <a class="nav-link" href="#">Link</a>
      </li >
      <li class="nav-item dropdown" >
        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false" >
          Dropdown
        </a>

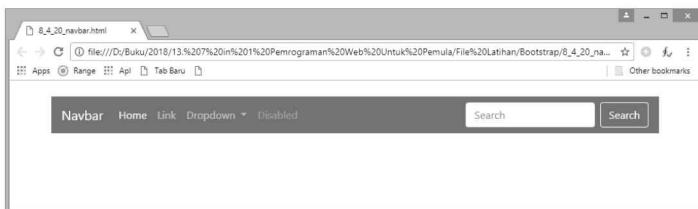
        <div class="dropdown-menu" aria-labelledby="navbarDropdown" >
          <a class="dropdown-item" href="#">...</a>
        </div >
      </li >
    </ul >

    <form class="form-inline my-2 my-lg-0" >
      <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="Search" >
      <button class="btn btn-outline-light my-2 my-sm-0" type="submit" >Search</button>
    </form>

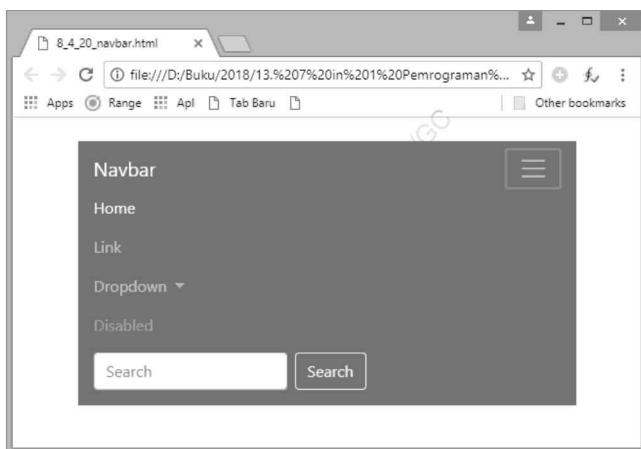
  </div>
</nav>
...

```

Pada contoh skrip di atas sudah termasuk contoh penggunaan dropdown dan form pencarian di dalam navbar. Hasilnya seperti gambar berikut:



Gambar 8.22 Tampilan navbar pada layar yang lebar



Gambar 8.23 Tampilan navbar pada layar kecil

8.4.21 Scrollspy

Scrollspy merupakan navigasi yang otomatis update sesuai dengan posisi scroll pada konten untuk menunjukkan link mana yang sedang aktif pada layar. Cara menggunakannya yaitu dengan menambah atribut **data-spy="scroll"** dan **data-target=""** pada tag `<body>`. Atribut data-target diisi id atau class navigasi yang digunakan sebagai scrollspy. Selain itu juga dapat menggunakan jQuery seperti skrip berikut:

```
$('body').scrollspy({ target: '#navbar-example' })
```

Yang perlu diperhatikan juga, nilai atribut **href** pada navigasi harus sesuai dengan id pada target. Misal pada navigasi memiliki atribut **href="#one"**, maka pada target harus memiliki atribut **id="one"**.

Contoh penggunaanya dapat dilihat pada skrip berikut:

File 8_4_21_scrollspy.html

```
<body data-spy="scroll" data-target=".navbar" data-offset="50">

<nav class="navbar fixed-top navbar-light bg-light">
  <ul class="nav nav-pills">
    <li class="nav-item">
      <a class="nav-link" href="#one">One</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#two">Two</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#three">Three</a>
    </li>
  </ul>
</nav>

<div class="container"><br>
<div id="one" style="padding-top: 50px">
  <h4>One</h4>
  <p>Veniam marfa mustache skateboard, ...</p>
</div>

<div id="two" style="padding-top: 50px">
  <h4>Two</h4>
  <p>Occaecat commodo aliqua del ectus. ...</p>
</div>

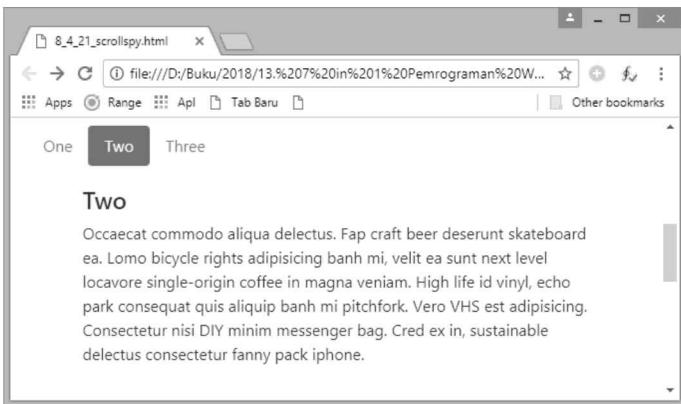
<div id="three" style="padding-top: 50px">
  <h4>Three</h4>
  <p>In incidunt echo park, officia ...</p>
</div>

</div>

<script src="js/jquery-3.3.1.min.js"></script>
<script src="bootstrap/js/bootstrap.bundle.min.js"></script>
</body>
...<
```

Pada skrip di atas, berisi 3 paragraf dimana masing-masing paragraf diletakkan pada tag div dengan id **one**, **two** dan **three**. Target scrollspy diarahkan pada navbar. Maka, pada navbar masing-masing link memiliki atribut href dengan nilai **#one**, **#two** dan **#three**. Untuk mencobanya,

buka pada browser lalu scroll pada konten. Hasilnya seperti gambar berikut:



Gambar 8.24 Contoh penggunaan scrollspy

8.5 Penerapan JQuery dan Bootstrap pada Aplikasi

Pada bab sebelumnya, belum dibahas bagaimana menerapkan jQuery pada aplikasi. Penerapannya akan dibahas sekaligus pada bab ini bersama dengan penerapan Bootstrap.

Pada praktiknya, dalam penggunaan Bootstrap biasanya tetap dibutuhkan penambahan skrip CSS untuk melengkapi kebutuhan design yang belum mampu dilakukan dengan Bootstrap. Namun, pada pembahasan ini penulis sebisa mungkin memaksimalkan penggunaan Bootstrap tanpa tambahan CSS.

8.5.1 Mengatur Halaman Login

Perubahan desain menggunakan Bootstrap pasti akan berdampak pada skrip HTML yang lebih panjang. Hal, ini karena penggunaan class yang banyak pada beberapa tag HTML. Termasuk pada halaman login. Untuk menggunakan style Bootstrap, halaman login dapat diubah seperti berikut:

File login.php

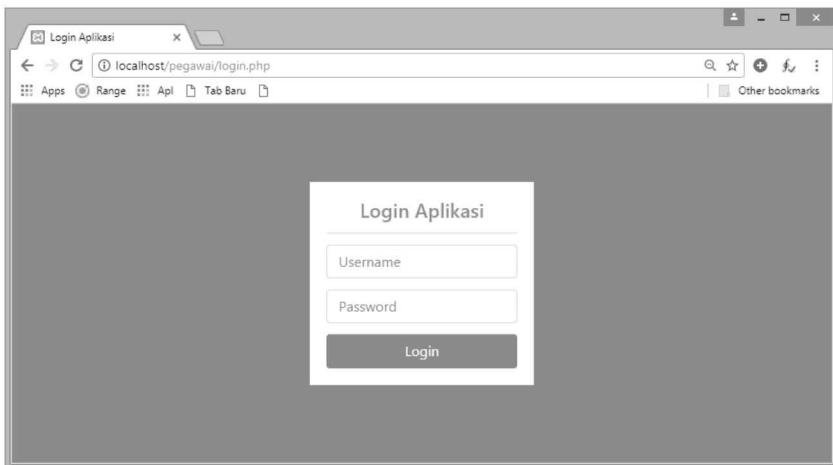
```
<!DOCTYPE html>
<html lang="en" class="h-100">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
    scale=1, shrink-to-fit=no">
    <title>Login Aplikasi </title>

    <link href="bootstrap/css/bootstrap.min.css"
    rel="stylesheet">
  </head>
  <body class="h-100 bg-info d-flex align-items-center">
    <div class="container">
      <div class="row">
        <div class="col-sm-6 col-md-4 mx-auto bg-light p-4">

<h3 class="text-center text-info pb-3 mb-3 border-bottom">Login
Aplikasi </h3>

<form method="post" action="ceklogin.php">
  <input class="form-control form-control-lg mb-3" type="text"
placeholder="Username" name="username">
  <input class="form-control form-control-lg mb-3"
type="password" placeholder="Password" name="password">
  <input class="btn btn-info btn-lg btn-block" type="submit"
value="Login">
</form>
      </div>
    </div>
  </body>
</html>
```

Perhatikan pada class-class yang dicetak tebal. Class-class tersebut yang menentukan posisi form login. Class **h-100** pada tag `<html>` akan membuat tinggi dokumen 100%. Dengan demikian ketika diterapkan class **h-100**, **d-flex** dan **align-items-center** pada tag `<body>` akan membuat container rata tengah secara vertikal. Untuk membuat form rata tengah horizontal, sebenarnya ada banyak cara. Pada skrip di atas penulis menggunakan class **mx-auto**. Class-class yang lain dapat diubah-ubah sesuai desain yang diinginkan oleh pembaca, tanpa harus mengikuti skrip di atas. Hasilnya seperti gambar berikut:



Gambar 8.25 Halaman login dengan Bootstrap

8.5.2 Mengatur Layout

Layout aplikasi akan dibuat responsive, dimana ketika tampil pada perangkat mobile, menu disembunyikan. Pembaca pasti sudah tahu caranya. Tentu saja menggunakan komponen navbar. Tapi, bagaimana jika pada layar yang lebar menu ingin ditampilkan di sebelah kanan. Jawabannya, perhatikan skrip berikut:

File index.php

```
<?php
session_start();
ob_start();

include "library/config.php";

if(empty($_SESSION['username']) or
empty($_SESSION['password'])) {
    echo "<p align='center'> Anda harus login terlebih dahulu! </p>";
    echo "<meta http-equiv='refresh' content='0;
url=login.php'>";
} else{
    define('INDEX', true);
?>

<!doctype html>
<html lang="en" class="h-100">
<head>
<meta charset="utf-8">
```



```

        </a>
    </li>
    <li class="list-group-item bg-dark">
        <a class="nav-link text-white" href="?hal=jabatan">
            <i class="oi oi-sort-descending"></i> Data Jabatan
        </a>
    </li>
    <li class="list-group-item bg-dark">
        <a class="nav-link text-white" href="logout.php">
            <i class="oi oi-account-logout"></i> Keluar
        </a>
    </li>
</ul>

</nav>

<div class="col-md-10 col-sm-9 offset-md-2 offset-sm-3 mb-3">
    <section>
        <?php include "konten.php"; ?>
    </section>
    </div>
    </div>
</div>

<div class="bg-light fixed-bottom">
    <p class="m-2 text-center text-muted">Copyright ©
Rohi Abdulloh</p>
</div>

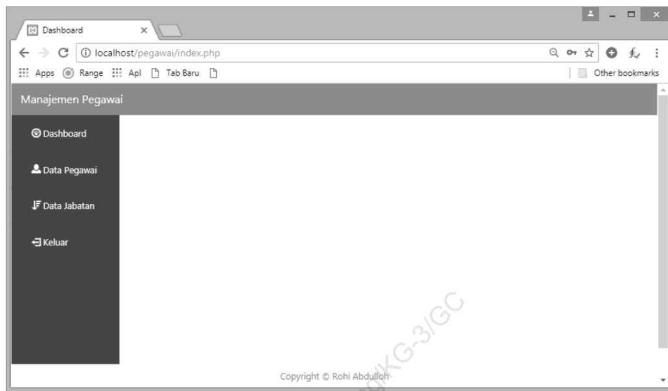
<script src="js/jquery-3.3.1.min.js"></script>
<script src="bootstrap/js/bootstrap.bundle.min.js">
</script>
</body>
</html>
<?php
}
?>
```

Pada skrip di atas, navigasi dibuat 2, pertama pada navbar dan yang kedua pada kolom grid dibuat menggunakan list group. Pada tag `` di dalam navbar diberi class **d-sm-none** sehingga pada layar yang lebar menu pada navbar tidak tampil. Sebagai gantinya, pada navigasi kedua yaitu pada tag `<nav>` ditambah class **d-none** dan **d-sm-block** sehingga hanya tampil pada layar yang lebar dan pada layar kecil disembunyikan. Selain itu, navigasi kedua juga ditambah class **position-fixed** sehingga tidak terpengaruh oleh scroll pada konten utama. Namun, dampak dari class ini akan membuat kolom grid sebelah kanan sebagai konten utama menumpuk pada menu. Untuk itu, pada kolom konten ditambah class **offset-sm-3** dan **offset-md-2**. Agar list group pada navigasi memenuhi lebar kolom, ditambahkan class **list-group-flush** pada tag ``.

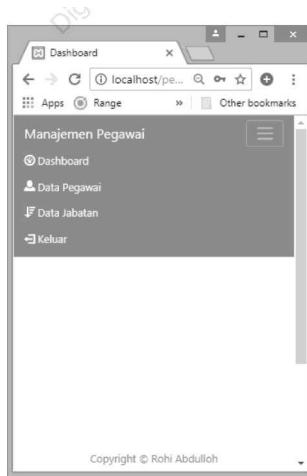
Jika diperhatikan pada bagian atas, terdapat skrip memanggil file **open-iconic-bootstrap.min.css**. File ini merupakan style dari open-iconic

yang digunakan untuk menampilkan ikon pada menu, karena pada Bootstrap 4 tidak menyediakan ikon. File tersebut dapat di-download pada <http://useiconic.com/open>. Pada aplikasi ini, file tersebut disimpan pada folder **open-iconic**. Cara menggunakannya dapat dilihat pada skrip di atas, yaitu dengan menambah class **oi** dan **oi-{nama ikon}** pada tag **<i>** atau tag ****.

Hasil dari skrip di atas seperti gambar berikut:



Gambar 8.26 Layout aplikasi pada layar lebar



Gambar 8.27 Layout aplikasi pada layar kecil

8.5.3 Mengatur Halaman Dashboard

Pada skrip sebelumnya halaman dashboard disembunyikan sementara dengan menambah tanda // pada include "konten.php". Untuk menampilkannya kembali hapus tanda tersebut.

Untuk mengatur halaman dashboard akan digunakan jumbotron dan ditambah list-group untuk menampilkan jumlah data yang telah diinput. Sehingga skripnya berubah seperti berikut:

File konten/dashboard.php

```
<?php
    if(!defined('INDEX')) die("");
?>

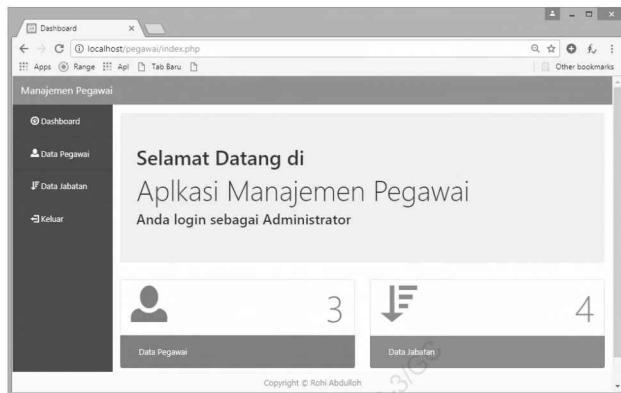
<div class="jumbotron mt-3">
    <h1>Selamat Datang di </h1>
    <h1 class="display-4">Aplikasi Manajemen Pegawai </h1>
    <h3>Anda login sebagai Administrador</h3>
</div>

<?php
    $jumlah_pegawai = mysqli_num_rows(mysqli_query($con, "SELECT * 
FROM pegawai"));
    $jumlah_jabatan = mysqli_num_rows(mysqli_query($con, "SELECT * 
FROM jabatan"));
?>

<div class="row mb-3 pb-3">
    <div class="col-sm-6 mb-3">
        <ul class="list-group">
            <li class="list-group-item text-danger">
                <i class="oi oi-person display-3"></i>
                <span class="display-3 float-right">
                    <?= $jumlah_pegawai ?>
                </span>
            </li>
            <li class="list-group-item bg-danger">
                <a href="?hal=jabatan" class="nav-link text-white">Data Pegawai </a>
            </li>
        </ul >
    </div >
    <div class="col-sm-6 mb-3">
        <ul class="list-group">
            <li class="list-group-item text-success">
                <i class="oi oi-sort-descending display-3"></i>
                <span class="display-3 float-right">
                    <?= $jumlah_jabatan ?>
                </span>
            </li>
            <li class="list-group-item bg-success">
                <a href="?hal=jabatan" class="nav-link text-white">Data Jabatan</a>
            </li>
        </ul >
    </div >
</div>
```

```
</di v>  
</di v>
```

Pada skrip di atas ditambah skrip untuk mengambil jumlah data pada tabel pegawai dan jabatan. Jumlah tersebut kemudian ditampilkan pada list group dengan ditambah class **display-3** agar tampil besar dan **float-right** agar di sebelah kanan. Hasilnya seperti berikut:



Gambar 8.28 Halaman dashboard dengan Bootstrap

8.5.4 Mengatur Tampilan Tabel

Pada halaman jabatan maupun pegawai, tampilan tabel akan diubah menggunakan Bootstrap dan plugin jQuery DataTable. Plugin ini akan membuat tampilan tabel lebih menarik disertai dengan fasilitas pencarian, pagination dan sorting. Pada file jabatan.php, skripnya diubah seperti berikut:

File konten/jabatan.php

```
<?php  
    if (!defined('INDEX')) die("");  
?>  
<h4 class="mt-2">Data Jabatan</h4>  
<hr>  
<a class="btn btn-success" href="?hal=jabatan_tambah"><i  
class="oi oi-plus"></i> Tambah</a>  
  
<div class="table-responsive mt-3">  
<table class="table table-striped table-hover table-bordered">  
    <thead>
```

```

<tr>
    <th>No</th>
    <th>>Nama Jabatan</th>
    <th>Aksi </th>
</tr>
</thead>
<tbody>
<?php
    $query = mysqli_query($con, "SELECT * FROM jabatan ORDER BY
id_jabatan DESC");
    $no = 0;
    while($data = mysqli_fetch_array($query)) {
        $no++;
?>
    <tr>
        <td><?= $no ?></td>
        <td><?= $data['nama_jabatan'] ?></td>
        <td>
            <a class="btn btn-sm btn-info"
            href="?hal=jabatan_edit&id=<?= $data['id_jabatan'] ?>"> <i
            class="oi oi-pencil"></i> Edit </a>
            <a class="btn btn-sm btn-danger"
            href="?hal=jabatan_hapus&id=<?= $data['id_jabatan'] ?>"> <i
            class="oi oi-trash"></i> Hapus </a>
        </td>
    </tr>
<?php
?>
    </tbody>
</table>
</div>

```

Perubahan paling mencolok yaitu pada tag `<table>` yang diberi class untuk mengatur tampilan tabel dan pada link tambah, edit dan hapus yang dibuat menjadi tombol dengan class **btn** dan dibuat warna yang berbeda-beda. Pada tombol tersebut juga ditambah ikon yang membuat tampilan lebih menarik.

Untuk menerapkan plugin DataTable, download dulu plugin dari <http://www.datatables.net>. Selanjutnya, buat folder **plugin**, lalu pindahkan folder **dist** dari plugin DataTable ke folder plugin tersebut. Kebanyakan plugin biasanya menempatkan file plugin pada folder dist, untuk itu penulis ubah foldernya menjadi **DataTable**. Kerena desain menggunakan Bootstrap 4, maka style yang digunakan juga yang diperuntukkan untuk Bootstrap 4. Panggil file style pada bagian head file index.php sebagai berikut:

File index.php

```

<link href="plugins/datatables/css/datatables.bootstrap4.min.css"
rel="stylesheet">
...

```

Selanjutnya, panggil file plugin jQuery di bagian bawah setelah pemanggilan skrip jQuery, serta buat skrip untuk menerapkan plugin DataTable seperti berikut:

File index.php

```
...
<script>
src="pl ugi n/Dat aTabl es/j s/j query. dat aTabl es. mi n.j s"></script>
<script>
src="pl ugi n/Dat aTabl es/j s/dat aTabl es. boot str ap4. mi n.j s"></script>
>

<script>
$(function(){
    $('.table').DataTable();
});
</script>
...

```

File yang dipanggil ada 2, yaitu file core dari plugin DataTable dan file skrip khusus untuk integrasi dengan Bootstrap. Cara menerapkannya ke tabel dapat dilihat pada skrip yang dicetak tebal. Hasilnya akan seperti gambar berikut:

No	Nama Jabatan	Aksi
1	Marketing	Edit Hapus
2	Sekretaris	Edit Hapus
3	Manajer	Edit Hapus
4	Direktur	Edit Hapus

Gambar 8.29 Tampilan tabel dengan Bootstrap dan plugin DataTable

8.5.5 Mengatur Tampilan Form

Pada form tambah dan edit jabatan hanya menerapkan style Bootstrap, penulis yakin pembaca tidak kesulitan mengubahnya. Pada form tambah dan edit pegawai, akan menggunakan style Bootstrap dengan berbagai jenis custom control, serta menambahkan plugin jQuery **bootstrap-datepicker** yang dapat di-download di <https://github.com/uxsolutions/bootstrap-datepicker> dan plugin **summernote** untuk mengubah textarea dengan penambahan berbagai tool pengeditan teks yang dapat didownload di <https://summernote.org>.

Pada file pegawai_tambah.php, skripnya berubah menjadi seperti berikut:

File konten/pegawai_tambah.php

```
<?php
    if (!defined('INDEX')) die("");
?>

<h4 class="mt-2">Tambah Pegawai </h4>
<hr>
<form class="mb-5" method="post" action="?hal=pegawai_insert"
      enctype="multipart/form-data">

    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Foto</label>
        <div class="col-sm-4">
            <div class="custom-file">
                <label for="foto" class="custom-file-label"></label>
                <input class="custom-file-input" type="file" id="foto"
name="foto">
            </div>
        </div>
    </div>

    <div class="form-group row">
        <label for="nama" class="col-sm-2 col-form-label">Nama</label>
        <div class="col-sm-4">
            <input class="form-control" type="text" id="nama"
name="nama">
        </div>
    </div>

    <div class="form-group row">
        <label class="col-sm-2 col-form-label">Jenis
Kelamin</label>
        <div class="col-sm-4">
            <div class="custom-control custom-radio custom-control-
inline">
                <input class="custom-control-input" type="radio"
id="jkl" name="jkl" value="L">
                <label class="custom-control-label" for="jkl">Laki -
Laki </label>
            </div>
        </div>
    </div>
</form>
```

```

        </div>
    <div class="custom-control custom-radio custom-control-inline">
        <input class="custom-control-input" type="radio" id="jkp" name="jk" value="P">
        <label class="custom-control-label">Perempuan</label>
    </div>
    <div class="form-group row">
        <label for="tanggal" class="col-sm-2 col-form-label">Tanggal</label>
        <div class="col-sm-2">
            <input class="form-control" type="text" id="tanggal" name="tanggal">
        </div>
    </div>

    <div class="form-group row">
        <label for="jabatan" class="col-sm-2 col-form-label">Jabatan</label>
        <div class="col-sm-4">
            <select class="custom-select" id="jabatan" name="jabatan">
                <option value=""> - Pilih Jabatan - <option>
                <?php
                    $query_jabatan = mysqli_query($con, "SELECT * FROM
                jabatan");
                    while($j = mysqli_fetch_array($query_jabatan)) {
                        echo "<option value=' $j[id_jabatan]'>$j[nama_jabatan]</option>";
                    }
                ?>
            </select>
        </div>
    </div>

    <div class="form-group row">
        <label for="keterangan" class="col-sm-2 col-form-label">Keterangan</label>
        <div class="col-sm-8">
            <textarea class="form-control" id="keterangan" name="keterangan"></textarea>
        </div>
    </div>

    <button type="submit" class="btn btn-info"><i class="oi oi-task"></i> Simpan</button>
    <button type="reset" class="btn btn-warning"><i class="oi oi-circle-x"></i> Batal </button>
</form>

```

Pada skrip di atas, disertai beberapa contoh penggunaan custom control yaitu pada input file, input radio dan select. Perhatikan saja pada class-class yang dicetak tebal.

Untuk menerapkan plugin bootstrap-datepicker dan summernote seperti biasa panggil file style pada bagian head dan panggil plugin pada

bagian bawah beserta penerapannya pada file index.php. Perhatikan skrip berikut:

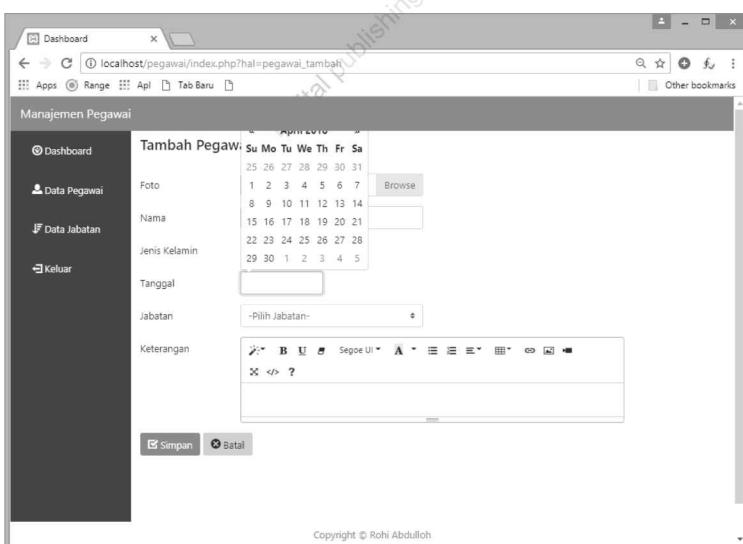
File index.php

```
<link href="plugin/bootstrap-datepicker/css/bootstrap-datepicker3.min.css" rel="stylesheet">
<link href="plugin/summernote/summernote-bs4.css" rel="stylesheet">

...
<script src="plugin/bootstrap-datepicker/js/bootstrap-datepicker.js"></script>
<script src="plugin/summernote/summernote-bs4.min.js"></script>

<script>
$(function(){
    $('#table').DataTable();
    $('#tanggal').datepicker();
    $('#keterangan').summernote();
});
</script>
```

Pemanggilan file style dan skrip plugin setelah pemanggilan style dan skrip plugin sebelumnya. Hasilnya seperti berikut:



Gambar 8.30 Tampilan form dengan Bootstrap dan plugin jQuery

Perubahan-perubahan pada file lain silakan pembaca dapat mengubahnya sendiri menyesuaikan contoh di atas, atau jika mengalami kendala dapat melihat pada file final project yang penulis sertakan.

8.6 Bonus File Contoh

Seperti materi-materi sebelumnya, penulis juga menyertakan beberapa contoh skrip Bootstrap yang diambil dari hasil download Bootstrap dengan sedikit modifikasi. File-file tersebut dapat ditemukan pada folder Bootstrap/BONUS.

9

PENUTUP

Tujuh materi utama yang dibahas pada buku sudah sangat cukup digunakan untuk membuat berbagai aplikasi berbasis web. Apalagi didukung dengan Bootstrap yang dapat membuat desain aplikasi web lebih cepat dan tampil responsive dalam berbagai ukuran layar. Ditambah lagi kemampuan jQuery untuk menciptakan berbagai efek visual yang menarik pada desain web. Yang lebih penting lagi, pada buku ini telah dibahas contoh pembuatan aplikasi lengkap dengan fasilitas login dan operasi CRUD serta upload gambar ke aplikasi. Tinggal bagaimana kreatifitas pembaca dalam memanfaatkan semua pengetahuan yang dimiliki untuk menciptakan aplikasi yang diinginkan. Namun, perlu diperhatikan, ilmu pemrograman web tidak sesederhana itu. Masih banyak sekali hal yang belum bisa dibahas pada buku ini. Untuk itu, penulis akan melanjutkan materi yang dibahas pada buku ini ke level yang lebih tinggi.

Materi yang akan dibahas selanjutnya yaitu sebagai berikut:

- **PDO**, yaitu sebuah metode agar aplikasi berbasis PHP dapat menggunakan berbagai jenis database dengan lebih mudah.
- **OOP**, yaitu metode pemrograman dengan menggunakan konsep objek untuk memudahkan programmer dengan menerapkan konsep objek yang ada pada kehidupan sehari-hari ke dalam pemrograman web.
- **MVC**, yaitu metode dalam pemrograman dengan memisahkan komponen utama yang membangun aplikasi yaitu manipulasi data, user interface dan bagian yang mengontrol aplikasi. Ketiga

komponen utama aplikasi tersebut biasa disebut *Model*, *View* dan *Controller* yang merupakan kepanjangan dari MVC.

- **AJAX**, yaitu solusi untuk permasalahan pada web tradisional, dimana ketika klien mengirim permintaan ke server harus menunggu respon dari server, baru bisa melanjutkan aktivitas berikutnya. Pada AJAX, selama proses menunggu respon dari server, user tetap dapat melanjutkan proses selanjutnya dan proses komunikasi data antara klien dan server tanpa ada refresh pada browser.
- **Laravel**, merupakan framework PHP yang paling popular saat ini. Tujuan framework sendiri untuk memudahkan programmer sehingga dalam pembuatan aplikasi web tidak perlu diawali dari nol. Proses pembuatan aplikasi yang dibuat oleh tim dengan lebih dari satu programmer biasanya dibuat menggunakan framework.
- **TypeScript**, merupakan pengembangan dari JavaScript yang telah mendukung konsep OOP berbasis class.
- **Angular**, merupakan framework JavaScript untuk membuat *single page application* layaknya aplikasi Google. Walaupun URL pada address bar berubah layaknya aplikasi biasa, tetapi tidak ada refresh pada halaman secara keseluruhan. Hanya pada sebagian elemen web yang terjadi perubahan.

Semua materi di atas akan sangat menarik untuk dibahas, mengingat sudah banyak buku yang juga telah membahas masing-masing materi tersebut secara terpisah dan penulis berencana akan menjadikannya dalam 1 buku. Untuk mewujudkannya tentu penulis membutuhkan dukungan dari pembaca. Untuk itu, silakan sampaikan kritik dan saran mengenai isi buku ini melalui instagram atau email penulis. Khusus untuk konsultasi, penulis hanya melayani melalui email.

KATA PENGANTAR



Rohi Abdulloh merupakan seorang Freelance Web Programer yang juga bekerja sebagai pengajar Multimedia, Jaringan Komputer dan Pemrograman. Memiliki kompetensi di bidang Desain Grafis, Pemrograman Web, Multimedia, dan Animasi. Sangat senang dengan dunia komputer dan selalu aktif meningkatkan kemampuannya terutama yang berhubungan dengan pemrograman web dan multimedia. Penulis sudah aktif menulis sejak 2013

dan beberapa buku yang telah diterbitkan diantaranya mengenai CSS3, HTML5 Canvas, jQuery, PHP, MySQLi, AJAX, Bootstrap, Zurb Foundation, OOP, MVC, Wordpress Developer, Laravel dan Android. Buku-buku sebelumnya mendapatkan banyak pujian dari para pembaca, disamping buku-bukunya memiliki ciri khas dengan studi kasus yang menarik serta pembahasan yang kata pembaca mudah dipahami, juga kesediaan penulis untuk selalu membantu dan menyemangati pembacanya setiap ada kesulitan. Untuk konsultasi mengenai isi buku dapat melalui email rohi.abdulloh@gmail.com. Informasi mengenai buku-buku penulis dapat dilihat di Instagram [@rohi_abdulloh](https://www.instagram.com/rohi_abdulloh).

7 IN 1

PEMROGRAMAN WEB

UNTUK PEMULA



Teknologi pemrograman web berkembang begitu cepat. Bagi pemula, tentu akan sangat tertinggal jika tidak cepat mengejarnya. Buku ini membahas 7 materi utama dalam mempelajari pemrograman web. Ketujuh bahasan ini akan sangat membantu pemula yang ingin menjadi web programmer dalam waktu yang singkat.

Pembahasan dimulai dari pengetahuan dasar tentang pemrograman web, dilanjutkan dengan pembahasan 7 materi pemrograman web satu per satu disertai dengan contoh skrip beserta hasilnya. Disertai juga pembuatan aplikasi sederhana yang akan membantu pembaca menguasai pembuatan modul aplikasi.

Untuk menunjang latihan pembaca, penulis juga menyertakan puluhan bonus skrip aplikatif. Dengan menguasai buku ini, Anda benar-benar dapat menjadi web programmer yang siap dan mampu membuat aplikasi web secara mandiri.

Pembahasan dalam buku mencakup:

- Dasar-dasar web programming
- HTML
- CSS
- MySQL
- PHP
- Javascript
- jQuery
- Bootstrap

PT ELEX MEDIA KOMPUTINDO
Kompas Gramedia Building
Jl. Palmerah Barat 29-37, Jakarta 10270
Telp. (021) 53650110-53650111, Ext 3214
Webpage: <http://elexmedia.id>

Kelompok
Pemrograman
Keterampilan
<input checked="" type="checkbox"/> Tingkat Pemula
<input checked="" type="checkbox"/> Tingkat Menengah
<input type="checkbox"/> Tingkat Mahir
Jenis Buku
<input checked="" type="checkbox"/> Referensi
<input checked="" type="checkbox"/> Tutorial
<input type="checkbox"/> Latihan

KOMPUTER

718051350

Harga P. Jawa **Rp74.800,-**



SU

9 786020 1479439