

## Daftar Isi

Halaman Sampul .....	i
Halaman Persembahan .....	v
Ucapan Terima Kasih .....	vi
Kata Sambutan .....	vii
Kata Pengantar .....	ix
Halaman Prakata .....	x
Daftar Isi.....	xi
Daftar Gambar.....	xv
Daftar Tabel.....	xvi
<b>BAB 1 Internet dan Web .....</b>	<b>1</b>
1.1. Pengantar Materi.....	1
1.2. Isi Materi.....	2
1.2.1. World Wide Web (WWW) .....	4
1.2.2. Web Statis dan Dinamis .....	5
1.3. Rangkuman .....	7
1.4. Tugas Latihan .....	7
1.5. Pustaka.....	8
<b>BAB 2 Desain Web .....</b>	<b>9</b>
2.1. Pengantar Materi.....	9
2.2. Isi Materi.....	9
2.2.1. Kebutuhan Pengguna .....	9
2.2.2. Desain <i>Wireframe</i> , <i>Mockup</i> , dan <i>Prototype</i> .....	12
2.2.3. Prototype .....	13
2.3. Rangkuman .....	15
2.4. Tugas Latihan .....	15
2.5. Pustaka.....	15
<b>BAB 3 HTML(Hypertext Markup Language) .....</b>	<b>17</b>
3.1. Pengantar Materi.....	17
3.2. Isi Materi.....	18
3.2.1 Struktur HTML .....	18
3.2.1 Elemen HTML .....	20
3.2.2 Atribut .....	25
3.2.3 Penulisan HTML.....	28
3.2.4 Elemen-elemen HTML Penting .....	31
3.2.5 Layout Halaman.....	39

3.2.6 Navigasi .....	43
3.3. Rangkuman.....	44
3.4. Tugas Latihan .....	45
3.5. Pustaka.....	46
BAB 4 HTML Form.....	47
4.1. Pengantar Materi .....	47
4.2. Isi Materi .....	48
4.2.1. Elemen <i>Form</i> .....	48
4.2.2. Bagian <i>Form</i> .....	50
4.2.3. Elemen Input.....	55
4.2.4. Metode Pengiriman Data .....	58
4.3. Rangkuman.....	60
4.4. Tugas Latihan .....	61
4.5. Pustaka.....	62
BAB 5 CSS ( <i>Cascading Style Sheets</i> ).....	63
5.1. Pengantar Materi .....	63
5.2. Isi Materi .....	64
5.2.1. <i>Selector</i> CSS.....	66
5.2.2. Penggunaan CSS.....	69
5.2.3. Properti CSS .....	71
5.3. Rangkuman.....	80
5.4. Tugas Latihan .....	81
5.5. Pustaka.....	83
BAB 6 Javascript .....	84
6.1. Pengantar Materi .....	84
6.2. Isi Materi .....	85
6.2.1. Penggunaan Javascript.....	85
6.2.2. Pemrograman Javascript.....	88
6.2.3. Document Object Model (DOM) .....	90
6.2.4. Metode Menampilkan Data .....	93
6.2.5. Metode Meminta Data .....	96
6.2.6. <i>Event Handler</i> .....	98
6.2.7. Metode Objek <i>String</i> .....	99
6.2.8. Metode Objek Numerik .....	101
6.2.9. Fungsi Buatan Javascript .....	102
6.2.10. Librari jQuery .....	103
6.3. Rangkuman.....	113

6.4. Tugas Latihan .....	113
6.5. Pustaka.....	114
<b>BAB 7 Pemrograman Web .....</b>	<b>116</b>
7.1. Pengantar Materi.....	116
7.2. Isi Materi.....	116
7.2.1. Persiapan Server .....	116
7.2.2. Bahasa <i>script</i> berbasis server .....	117
7.2.3. Pemrograman Dasar.....	119
7.3. Rangkuman.....	123
7.4. Tugas Latihan .....	124
7.5. Pustaka.....	125
<b>BAB 8 PHP (Hypertext Preprocessor) .....</b>	<b>126</b>
8.1. Pengantar Materi.....	126
8.2. Isi Materi.....	126
8.2.1. Penulisan <i>Script</i> PHP .....	127
8.2.2. Tipe Data.....	129
8.2.3. Variabel, Konstanta dan Operator di PHP.....	129
8.2.4. Struktur Kendali.....	135
8.2.5. Struktur Data.....	137
8.2.6. Fungsi PHP Form.....	139
8.2.7. Fungsi Built-In PHP.....	142
8.2.8. Fungsi Buatan PHP .....	146
8.3. Rangkuman.....	146
8.4. Tugas Latihan .....	146
8.5. Pustaka.....	149
<b>BAB 9 PHP Fungsi String.....</b>	<b>150</b>
9.1. Pengantar Materi.....	150
9.2. Isi Materi.....	150
9.2.1. Fungsi PHP Operasi String .....	150
9.2.1. Fungsi Reguler Expression (RegEx).....	152
9.3. Rangkuman.....	157
9.4. Tugas Latihan .....	157
9.5. Pustaka.....	158
<b>BAB 10 Create Read Update Delete (CRUD) .....</b>	<b>159</b>
10.1. Pengantar Materi.....	159
10.2. Isi Materi .....	159

10.2.1. Persiapan <i>Database</i> .....	160
10.2.2. Fungsi PHP MySQLi.....	161
10.2.3. <i>Select</i> Data .....	163
10.2.4. <i>Insert</i> Data .....	166
10.2.5. <i>Edit</i> Data .....	169
10.2.6. <i>Delete</i> Data .....	175
10.3. Rangkuman .....	176
10.4. Tugas Latihan .....	176
10.5. Pustaka.....	177
<b>BAB 11 Session dan Security .....</b>	<b>178</b>
11.1. Pengantar Materi.....	178
11.2. Isi Materi.....	178
11.2.1. Persiapan <i>Database</i> .....	178
11.2.2. Mekanisme <i>Login</i> .....	179
11.2.3. <i>Security Web</i> dengan <i>Session</i> .....	182
11.2.4. Fungsi Keamanan PHP .....	183
11.3. Rangkuman .....	185
11.4. Tugas Latihan .....	185
11.5. Pustaka.....	186
<b>BAB 12 Sertifikasi Junior Web Programmer .....</b>	<b>187</b>
12.1. Pengantar Materi.....	187
12.2. Isi Materi.....	188
12.2.1. Persyaratan Dasar Pemohon Sertifikasi.....	188
12.2.2. Unit Kompetensi.....	189
12.2.3. Bentuk dan Contoh Soal .....	194
12.3. Rangkuman .....	201
12.4. Tugas Latihan .....	202
12.5. Pustaka.....	202

## Daftar Gambar

Gambar 1. 1 Cara Kerja Web Statis.....	6
Gambar 2. 1 Wireframe Sederhana .....	12
Gambar 2. 2 <i>Mockup</i> Halaman Web.....	13
Gambar 2. 3 Tampilan Responsif Bootstrap .....	14
Gambar 3. 1 Struktur Dokumen HTML.....	18
Gambar 3. 2 Dokumen HTML .....	20
Gambar 3. 3 Perbaikan HTML Gambar 3.2 .....	29
Gambar 7. 1 Struktur Kendali Percabangan.....	121
Gambar 7. 2 Struktur Percabangan Majemuk .....	122
Gambar 7. 3 Struktur Kendali Perulangan .....	122
Gambar 8. 1 Kategori Fungsi <i>Built-In</i> PHP .....	143

## Daftar Tabel

Tabel 3. 1 Elemen HTML .....	21
Tabel 3. 2 Tag Usang HTML4 dan Padanannya di HTML5 ....	25s
Tabel 3. 3 Atribut Definisi.....	26
Tabel 3. 4 Atribut Event .....	26
Tabel 3. 5 <i>Style</i> CSS Pengganti Atribut Usang HTML4 .....	27
Tabel 4. 1 Elemen-elemen <i>Form</i> .....	51
Tabel 6. 1 Akses Elemen HTML Dengan Javascript .....	90
Tabel 6. 2 Metode Objek String .....	99
Tabel 6. 3 Metode Numerik .....	102
Tabel 6. 4 Selector jQuery.....	105
Tabel 6. 5 Action jQuery .....	106
Tabel 8. 1 Fungsi <i>Built-In</i> PHP .....	143
Tabel 10. 1 Tipe Data MySQL dan Elemen HTML .....	160
Tabel 10. 2 Fungsi PHP-MySQLi .....	162

# **BAB 1**

## **Internet dan Web**

Internet telah mengubah pola aktivitas manusia baik pertemanan, belanja, pembayaran hingga pembelajaran. Perubahan ini didukung oleh semakin mudahnya akses ke internet. Profil internet Indonesia tahun 2022 yang dikeluarkan Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) menyebutkan 77% penduduk di Indonesia telah menggunakan internet. Namun dari jumlah tersebut, masih sangat sedikit masyarakat yang memiliki *website* sendiri. Sebagai contoh, pelaku UMKM yang seharusnya membutuhkan *website* usaha, hanya 23,75% yang memiliki *website*.

Dalam pokok bahasan ini, Anda akan mempelajari pentingnya peran internet dan web, sejarah, dan istilah-istilah yang serupa namun tak sama.

Setelah menyelesaikan Bab ini, Anda diharapkan dapat:

1. Memahami asal muasal internet dan web
2. Memahami istilah-istilah internet dan web
3. Membedakan web statis dan dinamis

### **1.1. Pengantar Materi**

Internet dan web saling mendukung satu sama lain. ARPAnet adalah internet versi pertama yang dibuat pada tahun

1969 yang berhasil mengirimkan paket data pertama kali melalui jaringan. Jaringan ini menghubungkan komputer yang ada di University of California, Los Angeles (UCLA) dan Universitas Stanford. Disusul oleh penemuan protokol komunikasi paket melalui jaringan pada tahun 1974. Protokol ini yang menjadi cikal bakal protokol TCP/IP. Pada masa tersebut, istilah *interconnected network* atau internet mulai digunakan. Pembangunan perangkat untuk *backbone* internet (1986) dan *internet service provider (ISP)* sebagai *internet exchange point* (1989) membentuk infrastruktur internet seperti yang ada sekarang. Seiring dengan perkembangan infrastruktur internet, layanan yang berjalan di atasnya juga mengalami transformasi terutama setelah teknologi *world wide web (www)* sudah dapat digunakan pada tahun 1991. WWW menjadi tonggak awal perkembangan web di dunia dan masih digunakan hingga saat ini.

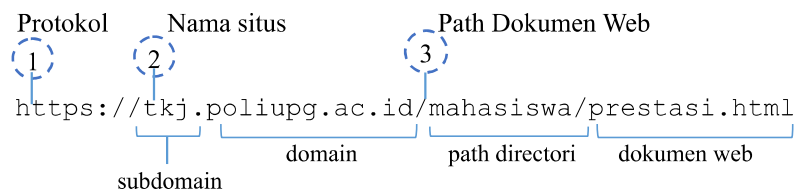
## **1.2. Isi Materi**

Konsep jaringan global yang ditawarkan internet sangat mendukung jangkauan dan kecepatan aktivitas manusia. Internet merupakan setumpuk perangkat keras yang terhubung sebagai jaringan besar dan di atasnya bekerja beberapa layanan. Layanan yang dimaksud antara lain email, web, transfer *file*, dan lain-lain. Sementara untuk melewati layanan ini, digunakan *protocol* sebagai jalur untuk melewati data layanan tersebut, misalnya untuk layanan web digunakan *hypertext transfer*



*protocol* (http) maupun *hypertext transfer protocol secure* (https) dan layanan email melalui protokol pop3.

Layanan web bukan internet. Jika internet merujuk pada keterhubungan jaringan dari seluruh dunia melalui *backbone* telekomunikasi dan *routing* menggunakan alamat IP, maka *world wide web* atau dikenal umum sebagai web merupakan layanan yang memungkinkan keterhubungan dokumen *hypertext* yang memanfaatkan internet sebagai media penghubungnya. Kita dapat melihat isi halaman web yang berisi teks, gambar, video, atau media lainnya serta dapat terhubung satu dengan yang lain melalui *hyperlink*. Alamat web lengkap untuk mengakses lokasi sumber daya pada web disebut *Unified Resources Locator* (URL).



Pola ini menunjukkan protokol komunikasi yang digunakan adalah https dengan nama situs tkj.poliupg.ac.id. Sebuah situs memiliki alamat IP publik, misalnya 159.223.81.66. IP publik merupakan alamat yang dapat dikunjungi dalam jaringan internet. Sementara alamat domain/subdomain adalah alias dari alamat IP yang sebenarnya. Penamaan alias ini diperlukan untuk memudahkan pengguna

mengingat alamat *website*. Perubahan nama alias menjadi alamat IP diatur oleh *Domain Name Server* (DNS). Tanpa DNS, kita harus mengetikkan alamat IP secara langsung untuk mengunjungi sebuah *website*.

### **1.2.1. World Wide Web (WWW)**

Layanan web dimulai dari dibuatnya bahasa markup berisi *hypertext* oleh Tim Berners Lee di awal tahun 90-an. Bahasa ini diberi nama Hypertext Markup Language yang sangat populer dengan nama HTML. WWW merupakan kumpulan halaman-halaman HTML yang saling terkoneksi melalui *hyperlink* dan membentuk samudera informasi. WWW berjalan di atas protokol *hypertext transfer protocol* (*http*). WWW merupakan layanan internet yang paling populer hingga saat ini. Bahkan internet dapat digunakan secara luas setelah adanya layanan *www*.

Mosaic adalah web browser pertama yang digunakan khalayak umum pada tahun 1993. Browser buatan Andressen dan Eric Bina, yang merupakan mahasiswa Universitas Illinois pada saat itu, telah mampu membaca teks dan grafis. Browser dibutuhkan untuk menginterpretasikan dokumen web berisi sintaks HTML menjadi tampilan halaman web.

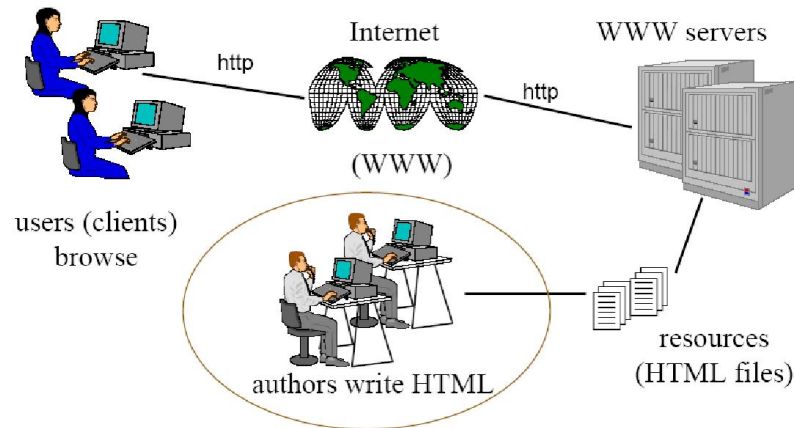
*User*/pengguna bertindak sebagai klien menggunakan browser untuk melakukan permintaan ke server web melalui internet. Di internet, pertama-tama nama *website*/ URL yang diklik akan dikirim ke DNS untuk mendapatkan alamat IP server

*website* yang dituju. Server ini harus selalu hidup untuk merespons permintaan dari pengguna.

### **1.2.2. Web Statis dan Dinamis**

Dokumen web ditulis dengan HTML. Seorang web developer dapat menulis dokumen HTML secara langsung maupun terprogram menggunakan bahasa pemrograman tertentu. Berdasarkan hal tersebut, web dapat terbagi menjadi:

1. Web Statis, memiliki ciri-ciri isi halaman tidak pernah berubah. Dokumen web dapat saja diubah dengan cara mengubah dokumen HTML secara langsung. Biasanya digunakan untuk halaman Kontak Kami atau Tentang Kami. Gambaran cara kerja web statis dapat dilihat pada gambar 1.1. Semua isi dari halaman web terdiri dari dokumen HTML yang ditulis oleh seorang web developer. Ketika diakses, semua *file* yang diminta akan diunduh oleh browser dan dijalankan di sisi *client* (*client side scripting*).



Gambar 1. 1 Cara Kerja Web Statis

2. Web Dinamis, membutuhkan pemrograman web, sehingga isi halaman dapat diprogram sesuai kebutuhan. Isi web akan berubah tanpa melakukan perubahan pada dokumen HTML. Beberapa bahasa pemrograman web yang populer adalah PHP dan Javascript. Pemrograman yang memungkinkan terjadi perubahan dari sebuah halaman web secara dinamis. Misalnya tanggal yang menyesuaikan dengan waktu server, berita yang selalu berubah setiap waktu, hingga tampilan yang berbeda-beda di tiap dashboard pengguna. Semuanya diproses di server terlebih dahulu, kemudian server menjawab request dengan mengirimkan *resource* ke *client*. Web menjadi dinamis berkat *script* yang berjalan di sisi server (*server side scripting*)

### 1.3. Rangkuman

1. Internet adalah jaringan global berupa infrastruktur yang berupa *hardware*, sedangkan web adalah layanan yang memungkinkan untuk dilewatkan melalui internet.
2. Internet berkembang pesat dengan kemunculan *www* yang memungkinkan interkoneksi data secara luas
3. Pembuatan web membutuhkan kemampuan desain dan pemrograman, yaitu membuat tampilan serta menggunakan program untuk menciptakan dinamika sebuah web.

### 1.4. Tugas Latihan

1. Jelaskan pengertian dari:
  - a. *Social media*
  - b. *E-commerce*
  - c. Internet
  - d. Web
  - e. Domain Name Server (DNS)
2. Jelaskan perbedaan
  - a. Unified Resource Locator (URL) dan Unified Resource Identifier (URI)
  - b. Web dan *website*
  - c. Web Statis dan dinamis
  - d. *Social media* dan *Instant Messenger*
3. Jelaskan cara kerja web statis!

4. Apa fungsi pemrograman dalam membangun sebuah web dinamis?
5. Sebutkan bahasa pemrograman yang dapat digunakan untuk membangun web dinamis

### **1.5. Pustaka**

1. Brpadbamdsearch.com, [Online] tersedia di <https://www.broadbandsearch.net/blog/who-invented-the-internet-full-history>, diakses tanggal 16 Juni 2022
2. Duckett, J., HTML & CSS Design and Build *Websites*, John Wiley & Sons, 2011
3. Gil, Paul, [Online] tersedia di <https://www.lifewire.com/top-internet-terms-for-beginners-2483381>, diakses tanggal 16 Juni 2022

## **BAB 2**

### **Desain Web**

Dalam pokok bahasan ini, Anda akan mempelajari pentingnya peran desain web, memahami kebutuhan pengguna, dan tahapan-tahapan desain.

Setelah menyelesaikan Bab ini, Anda diharapkan dapat:

1. Memahami peran desain web
2. Memahami kebutuhan desain bagi pengguna
3. Menjelaskan tahapan desain

#### **2.1. Pengantar Materi**

Membangun sebuah *website* bukan hanya membutuhkan keterampilan bahasa pemrograman. Namun, pengetahuan tentang target pengguna *website* adalah kunci kesuksesan sebuah *website*. Setelah memahami target pengguna barulah tampilan web dirancang. Untuk menjawab kebutuhan pengguna (*user*) terhadap *website* dibutuhkan seorang *User Interface* dan *User Experience (UI/ UX) designer*.

#### **2.2. Isi Materi**

##### **2.2.1. Kebutuhan Pengguna**

Pengunjung *website* adalah pengguna utama *website* kita, sehingga kita perlu melihat dari perspektif pengunjung, antara lain:

#### A. Pengunjung tidak membaca keseluruhan halaman

Meskipun *website* penuh dengan informasi, pengunjung hanya akan menghabiskan beberapa detik untuk melihat-lihat. Penting untuk meletakkan inti dari halaman di kalimat awal, buat mereka melihat paragraf pendek dan petunjuk-petunjuk pada halaman. Gunakan gambar atau objek dinamis untuk menonjolkan petunjuk-petunjuk tersebut

#### B. Lebih Sedikit Lebih Baik

Buat paragraf, halaman, dan bagian-bagian lainnya dalam halaman sependek mungkin, agar halaman tidak dipenuhi dengan teks yang dapat membuat pengunjung jenuh. Jika memang banyak hal yang ingin disampaikan, tampilkan judul dan atau potongan bagian saja, sementara sisanya diletakkan di halaman yang berbeda.

#### C. Navigasi

Buat navigasi yang konsisten pada semua halaman di dalam situs. Hindari buat *hyperlink* di dalam paragraf yang mengarahkan pengunjung ke tiap halaman dalam situs. Ini akan mengacaukan konsistensi dalam struktur navigasi situs. Jika ingin menggunakan *hyperlink*, tambahkan di bagian akhir paragraf atau di bagian menu.

#### D. Kecepatan akses

Menurut penelitian yang dilakukan google, 53 persen pengunjung akan meninggalkan *website* Anda jika loadingnya lebih dari tiga detik. Dampaknya 79%



pengunjung tidak ingin kembali lagi, 49% di antara mereka kehilangan kepercayaan pada bisnis Anda, 44% di antaranya memberitahukan kekecewaan mereka ke orang lain. Jika proses *download* lambat, sebaiknya tampilkan grafis dengan ukuran *file* kecil.

E. Biarkan pengunjung berbicara

Pengunjung situs adalah “pelanggan” kita. Mereka bisa memberikan informasi-informasi untuk perbaikan kita. Mudahkan pengunjung menghubungi anda agar banyak masukan yang kita dapatkan dari berbagai aspek.

F. Perhatikan monitor pengguna

Tidak semua orang punya monitor yang sama dengan komputer Anda. Pastikan melakukan tes di ukuran monitor target pengunjung anda. Jika perlu, gunakan teknologi yang responsif dengan ukuran monitor.

G. Perhatikan browser yang digunakan

Untuk menguji coba situs, gunakan beberapa browser yang berbeda. Hal ini penting agar situs anda ditampilkan sebagaimana mestinya. Perhatikan perkembangan browser yang banyak digunakan pengunjung

H. *Plug-in* apa yang dimiliki pengguna

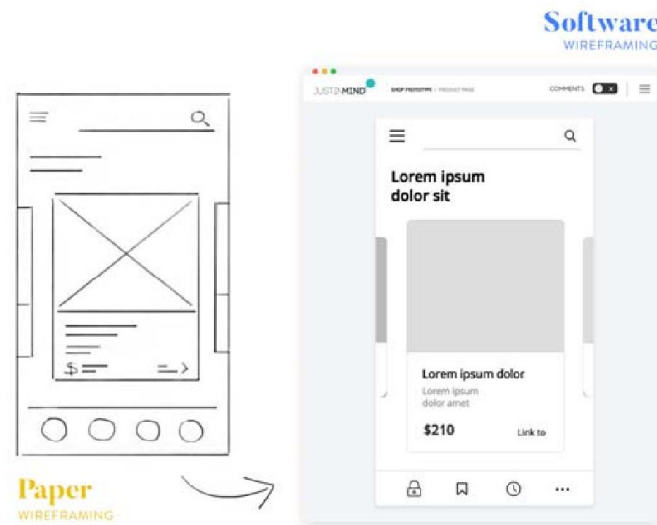
Format audio dan video biasanya memerlukan program terpisah (*plug-in*) untuk bisa diputar di halaman *website*. Pastikan pengunjung memiliki akses ke *plug-in* untuk menampilkan konten dalam situs kita.

I. Manfaatkan *search engine*

Beberapa *search engine* populer dapat digunakan sebagai media untuk memperkenalkan situs.

### 2.2.2. Desain *Wireframe*, *Mockup*, dan *Prototype*

*Wireframe* adalah kerangka desain paling dasar sebuah halaman web. Struktur layout berupa kerangka berbentuk tabel.



(a) Menggunakan Kertas

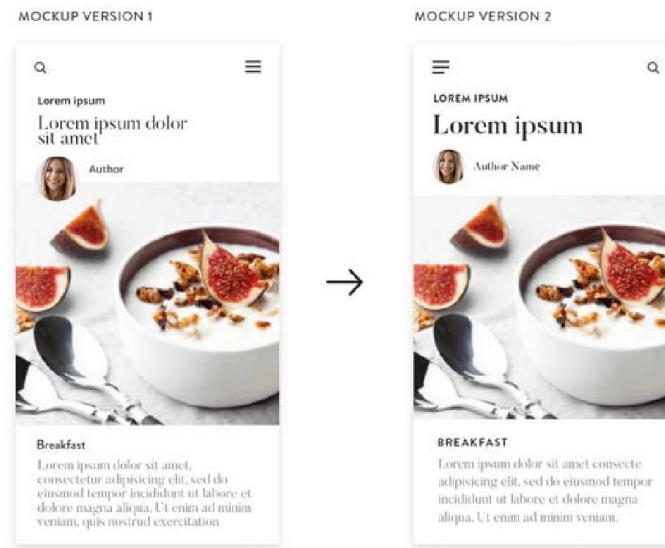
(b) Menggunakan Aplikasi

Gambar 2. 1 Wireframe Sederhana  
(Adiseshiah, 2016)

Fungsinya untuk mendapatkan umpan balik dari *user* sebelum fokus ke detail visualisasi. *Wireframe* belum menunjukkan halaman web yang sebenarnya.

Sementara *mockup* sudah memperlihatkan wujud grafis halaman *website*, hanya saja belum dapat berinteraksi langsung

dengan pengguna. Sementara *prototype* sudah dapat dikatakan halaman *website* yang sudah siap diakses hanya saja belum selengkap versi final. Perbedaan *wireframe* dan *mockup* dapat dilihat pada gambar 2.1 dan 2.2. Banyak aplikasi *online* (daring) yang tersedia untuk membuat *wireframe*, *mockup* bahkan *prototype*, misalnya Figma (figma.com)



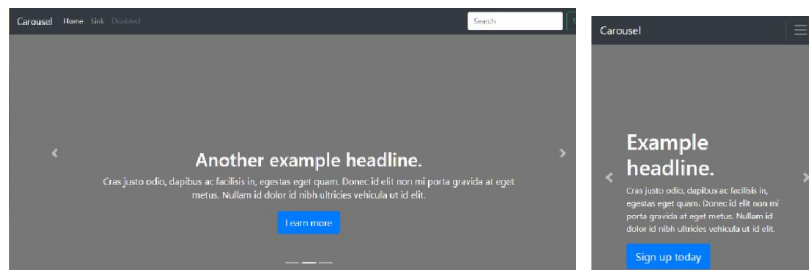
Gambar 2. 2 *Mockup* Halaman Web  
(Adiseshiah, 2016)

### 2.2.3. Prototype

Untuk membuat tampilan halaman yang interaktif membutuhkan tiga layer utama, yaitu *content layer* (file .html), *presentation layer* (file .css), dan *behaviour layer* (file .js). HTML membentuk konten halaman, membangun struktur halaman dan menambahkan makna semantik. Halaman HTML

berfungsi sebagai penghubung *file* CSS and Javascript. CSS memberikan aturan pada halaman HTML yang menentukan bagaimana konten HTML ditampilkan antara lain *background, border, box dimension, color, font, etc.* Javascript mengatur cara halaman bertingkah laku, meningkatkan interaksi dengan pengguna.

Terdapat *framework* desain yang dibuat untuk menggabungkan tiga layer ini. Yang paling populer adalah bootstrap. Framework bootstrap sangat mudah digunakan, sehingga banyak template yang dibuat berbasis bootstrap. Selain itu bootstrap juga bersifat responsif, artinya tampilan dapat menyesuaikan dengan ukuran perangkat yang digunakan pengguna, sehingga dipastikan bisa bekerja secara optimal di semua ukuran layar. Bootstrap juga dapat digunakan semua browser. Kekurangan bootstrap adalah *resource* yang digunakan cukup besar baik *resource* css, js, glyph-icon, serta jQuery. Namun, tidak semua resource tersebut yang terpakai. Contoh tampilan web menggunakan bootstrap seperti pada gambar 2.3.



(a) Tampilan Desktop

(b) Tampilan mobile

Gambar 2. 3 Tampilan Responsif Bootstrap

### 2.3. Rangkuman

1. Desain web harus dibuat berdasarkan kebutuhan pengguna
2. Gambaran desain dapat dimulai dari *wireframe*, *mockup* hingga membuat *prototype*
3. Desain *prototype* menggunakan tiga layer yaitu HTML, CSS, dan Javascript.

### 2.4. Tugas Latihan

1. Buatlah desain *wireframe website* sekolah secara langsung berhadapan dengan pihak sekolah anda. Mintalah umpan balik untuk perbaikan desain anda.
2. Buatlah desain *mockup* halaman depan *website* tersebut menggunakan *tools* yang tersedia
3. Pastikan projek anda tersimpan untuk digunakan pada pembuatan *prototype* di materi selanjutnya.
4. Bagaimana tiga layer desain terhubung satu sama lain?
5. Apa saja kelebihan dan kekurangan menggunakan *framework* desain?

### 2.5. Pustaka

1. Boulton, M., Designing for The Web, [Online] tersedia di <https://designingfortheweb.co.uk/> [Diakses tanggal 16 Juni 2022]
2. Adiseshiah, E. G., 2016. Wireframes vs. Mockups: what's

- the best option?. [Online] tersedia di <https://www.justinmind.com/blog/wireframes-and-mockups-whats-the-best-option/> [Diakses tanggal 16 Juni 2022].
3. Nayoan, A., 2021. 8+ Cara Mempercepat Loading *Website* dan Blog Anda [Online], tersedia di <https://www.niagahoster.co.id/blog/cara-mempercepat-loading-blog> [Diakses tanggal 16 Juni 2022]
  4. Bootstrap, Example – Bootstrap [Online] tersedia di <https://getbootstrap.com/docs/4.0/examples/> [Diakses tanggal 16 Juni 2022]

## **BAB 3**

### **HTML(Hypertext Markup Language)**

Dalam pokok bahasan ini, Anda akan mempelajari bentuk dasar dari HTML dan penggunaan dari beberapa tag dasar yang sering digunakan dalam pengembangan aplikasi web.

Setelah menyelesaikan bab ini, Anda diharapkan dapat:

1. Memahami bentuk dasar dari halaman web HTML
2. Menggunakan tag HTML dalam membuat halaman web
3. Membuat halaman HTML dengan komponen yang sesuai dengan kebutuhan

#### **3.1. Pengantar Materi**

*Hypertext Markup Language* atau HTML adalah bahasa markup yang digunakan untuk membuat sebuah halaman web. HTML5 adalah standar HTML terbaru yang digunakan untuk menulis sebuah dokumen web. Sebelumnya, standar yang digunakan adalah HTML 4 yang memiliki standar penulisan yang identik dengan XHTML yang lebih rapi dan ketat. XHTML dibuat untuk memudahkan *web developer* beralih dari HTML ke XML.

Namun dalam perjalanannya, HTML5 lebih mendominasi dengan mendapatkan dukungan semua browser, sehingga HTML5 lebih banyak digunakan hingga saat ini. Kelebihan lain HTML5 adalah:

- Memiliki elemen-elemen modern antara lain elemen video, audio dan *canvas* yang membuat halaman web semakin terlihat modern dengan mendukung multimedia
- Tag penutup dapat dihilangkan untuk *tag* kosong
- Adanya tag semantik, seperti *header*, *footer*, *section*, *article*, dan *nav* yang dapat mengurangi penggunaan *tag*
- Tidak *case-sensitive*
- Dukungan API yang memungkinkan pengguna berbagi lokasi

## **3.2. Isi Materi**

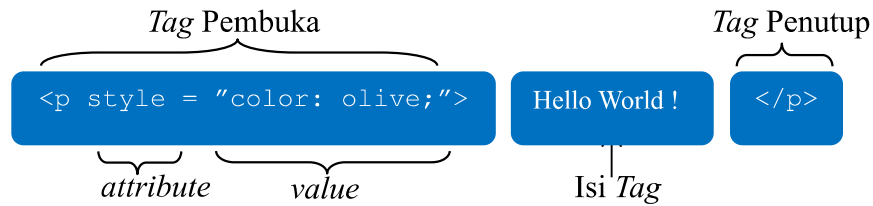
### **3.2.1 Struktur HTML**

Sebuah dokumen web diinterpretasi oleh browser berdasarkan informasi dari struktur HTML. Struktur HTML pada gambar 3.1 menunjukkan elemen pada HTML bersifat bersarang.

Gambar 3. 1 Struktur Dokumen HTML



Sebuah elemen HTML ditulis dengan simbol *tag*. *Tag* yang kosong atau *empty tag* adalah *tag* yang tidak memiliki isi tag. Misalnya *tag breaking space* `<br>`.



Penjelasan tag di atas adalah sebagai berikut:

1. **Tag** merupakan simbol/tanda pembuka dan atau penutup dari sebuah elemen HTML.
2. **Atribut** merupakan ciri-ciri *Tag* yang dituliskan pada *Tag* pembuka.
3. **Isi Tag** merupakan isi dari *Tag* HTML biasanya berupa teks
4. **Elemen** merupakan komponen utama HTML yang mencakup *tag*, atribut, serta isi *tag*.

Berdasarkan pemahaman bentuk perintah HTML diatas, Anda akan belajar komponen-komponen HTML dasar selanjutnya. Penulisan *tag* di dalam dokumen HTML dapat dilihat pada gambar 3.2.

```

        <td>Alfreds Futterkiste</td>
        <td>Maria Anders</td>
        <td>Germany</td>
    </tr>
    <tr>
        <td>Centro comercial Moctezuma</td>
        <td>Francisco Chang</td>
        <td>Mexico</td>
    </tr>
</table>

</body>
</html>

```

**Tampilan:**

**A basic HTML table**

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico

**3.2.5 Layout Halaman**

Elemen div sangat penting perannya dalam layout halaman. Meski di HTML5 telah ada elemen semantik yang mengatur layout, tetapi div masih sangat dibutuhkan untuk mengelola pembagian *section* sebuah halaman. Contohnya:

```

<html>
  <body>
    <div style ="border:1px solid blue;
    width:800px; height:100px; text-
    align:center"> HEADER </div>

    <div style="float:left; width:150px;
    border:2px solid red;
    height:100px;">Sidemenu </div>

    <div style="float:left; width:646px;
    border:1px solid green; height:100px;">
    CONTENT </div>

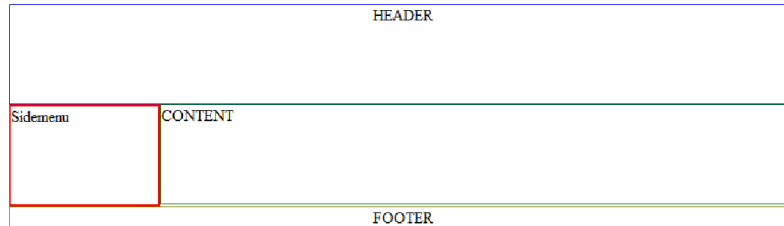
```

```

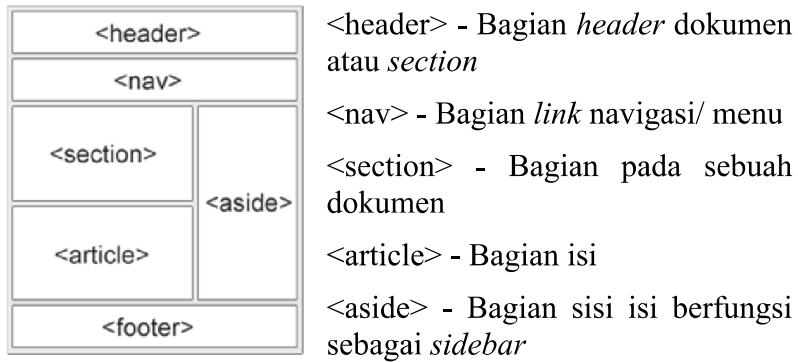
<div style ="clear:both; border:1px solid
olive; width:800px; text-align:center;">
FOOTER </div>
</body>
</html>

```

**Tampilan:**



Selain menggunakan elemen div, kita juga dapat menggunakan elemen semantik pada HTML5, yaitu:



`<aside>` - Bagian sisi isi berfungsi sebagai *sidebar*

`<footer>` - Bagian *footer* dokumen atau *section*

Contoh penggunaan elemen layout semantik:

```

<html>
  <body>
    <header>
      <h2>Cities</h2>
    </header>

    <section>
      <nav>
        <ul>

```

```

        <li><a href="#">London</a></li>
        <li><a href="#">Paris</a></li>
        <li><a href="#">Tokyo</a></li>
    </ul>
</nav>

<article>
    <h1>London</h1>
    <p>London is the capital city of
    England. It is the most populous city
    in the United Kingdom, with a
    metropolitan area of over 13 million
    inhabitants.</p>
</article>
</section>

<footer>
    <p>Footer</p>
</footer>

</body>
</html>

```

Elemen semantik tidak dibuat untuk menggantikan elemen div, melainkan menguatkan penamaan sesuai dengan maknanya, tetapi tidak secara otomatis memiliki *style*, sehingga elemen-elemen ini tetap perlu diberi *style* seperti pada div, seperti berikut ini:

```

<style>
    * {
        box-sizing: border-box;
    }

    /* Style header */
    header {
        background-color: #666;
        padding: 30px;
        text-align: center;
        font-size: 35px;
        color: white;
    }

    /* Kolom navigasi bagian kiri */

```

```

nav {
    float: left;
    width: 30%;
    background: #ccc;
    padding: 20px;
}

/* Style list dalam menu */
nav ul {
    list-style-type: none;
    padding: 0;
}

article {
    float: left;
    padding: 20px;
    width: 70%;
    background-color: #f1f1f1;
}

/* Clear floats after the columns */
section::after {
    content: "";
    display: table;
    clear: both;
}

/* Style the footer */
footer {
    background-color: #777;
    padding: 10px;
    text-align: center;
    color: white;
}

/* Style responsif untuk nav dan article */
@media (max-width: 600px) {
    nav, article {
        width: 100%;
        height: auto;
    }
}
</style>

```

\*) Penulisan *style* model ini akan dijelaskan pada bab 4

## Tampilan:



### 3.2.6 Navigasi

Navigasi dalam halaman web dibangun menggunakan HTML dan CSS. Sebagai contoh:

```
<style>
  .clear{clear:both;}
  ul{list-style-type:none; margin:0;padding:0;}
  ul#main li{background:white;float:left;}
  ul#main li a{display:block;padding:5px 10px;
  border-bottom:1px solid #ccc;}
  ul#main li a:hover{background:#eee;}
  ul#main li ul li{float:none;}
  ul#main li ul {position:absolute;display:none;}
  ul#main li ul li
  ul{position:absolute;display:none;}
  ul#main li:hover ul{display:block;}
</style>
<div>
  <ul id="main" style="background:red;">
    <li><a href="#">Home</a></li>
    <li><a href="#">Tutorial</a>
      <ul>
        <li><a href="#">Java</a></li>
        <li><a href="#">PHP</a></li>
        <li><a href="#">Delphi</a></li>
      </ul>
    </li>
    <li><a href="#">Contact Us</a>
  </ul>
```

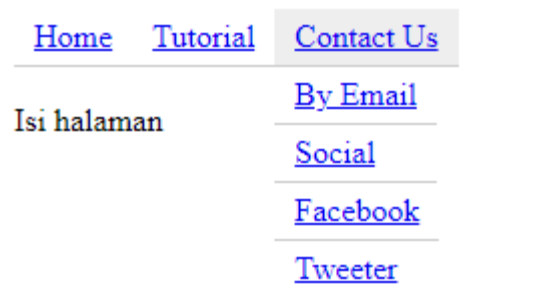
```

        <li><a href="#">By Email</a></li>
        <li><a href="#">Social </a>
            <ul>
                <li><a href="#">Facebook</a></li>
                <li><a href="#">Tweeter</a></li>
            </ul>
        </li>
    </ul>
</li>
</ul>
</div>

<div class="clear">
    <p><br/>Isi halaman </p>
</div>

```

**Tampilan:**



\*) *Style* menggunakan CSS akan dijelaskan secara rinci pada bab berikutnya.

**3.3. Rangkuman**

*Hypertext Markup Language* atau HTML adalah bahasa markup yang digunakan untuk membuat sebuah halaman web. Isinya terdiri dari berbagai elemen yang membentuk struktur sebuah halaman web. Beberapa tag yang sering digunakan ialah:

1. <html> </html> sebagai pembuka halaman HTML
2. <h1> - <h6> sebagai *heading* halaman

3. `<p></p>` sebagai paragraf teks halaman
4. `<ul>` / `<ol>` sebagai list item
5. `<a>` sebagai *link*
6. `<img>` sebagai pengisi gambar
7. `<table>` sebagai pembuat *table*
8. `<div>` sebagai *layout* halaman
9. `<ul>`, `<a>` sebagai navigasi

### 3.4. Tugas Latihan

1. Buatlah *prototype* untuk *homepage* halaman web sekolah yang berisi profil sekolah menggunakan *tag-tag* HTML paragraf, list item, *link*, gambar dan *table*.
2. Buatlah sebuah dokumen HTML sebagai berikut

```
<body alink=' style.css'>
  <p><font size="10">Paragraf dengan ukuran
  font 10</font></p>
  <ul type="circle">
    <li>List pertama</li>
    <li>List kedua </li>
  </ul>
  <table cellpadding="10">
    <tr>
      <td width="100">Kolom pertama</td>
      <td width="200">Kolom kedua</td>
    </tr>
  </table>
</body>
```

3. Ubahlah HTML nomor 2 agar sesuai dengan standar HTML5
4. Buatlah dokumen HTML untuk tabel berikut:



		Siswa	
		Laki-laki	Perempuan
Kelas	1	501	480
	2	500	470

5. Buatlah navigasi menu berisi 3 menu yaitu Beranda, Produk, dan Tentang Kami. Pada menu produk terdapat kategori Buku, Novel, dan Komik.

### 3.5. Pustaka

1. Duckett, J., *HTML & CSS Design and Build Websites*, John Wiley & Sons, 2011
2. Hogan, B.P., *HTML5 & CSS3 Develop with Tomorrows Standards Today*, USA, Pragmatic Programmers, 2010
3. Sari, A.O., *dkk, Web Programming*, Graha Ilmu, Jakarta, 2019
4. <https://www.w3schools.com/html>

## BAB 4

### HTML Form

HTML *Form* digunakan untuk mengambil input dari pengguna. HTML *form* sangat penting dalam pengembangan web karena merupakan jalur utama data dari *user* ke aplikasi.

Setelah menyelesaikan bab ini, Anda diharapkan dapat:

1. Memahami cara membuat *form*
2. Menuliskan kode untuk pengiriman data pada *form* HTML
3. Memahami cara menangkap data dari *form* HTML

#### 4.1. Pengantar Materi

Formulir paling banyak digunakan untuk membuat buku tamu atau umpan balik. Sebuah *form* dalam HTML di definisikan menggunakan *tag* HTML `<form>` dan di tutup dengan `</form>`

```
<form>
  .
  form elements
  .
</form>
```

Dalam sebuah *form* kemudian beberapa elemen pendukung untuk input di berikan seperti `<input>`, `<label>`, `<textarea>`, dan lainnya.

## 4.2. Isi Materi

### 4.2.1. Elemen *Form*

*Form* memiliki beberapa Atribut wajib yang dapat dituliskan sebagai berikut:

```
<FORM METHOD="post" ACTION="proses.php">...</FORM>
```

└──────────────────┬──────────────────┘

Memberitahu cara      Memberitahu server tujuan  
informasi dikirim      informasi akan dikirim

*Form* HTML harus di dukung oleh beberapa atribut wajib, yaitu:

#### 1. Atribut `action=""`

Atribut `action` mendefinisikan tujuan pengiriman data dalam sebuah form. Umumnya, `action` mengarahkan pengiriman data pada sebuah link / *file* khusus pada server yang berfungsi sebagai penangkap data dan pengolah data.

```
<form action="aksi_insert.php">
    .
    form elements
    .
</form>
```

#### 2. Atribut `method=""`

Atribut `method` digunakan untuk mendefinisikan cara sebuah form dalam mengirim data. Terdapat dua cara yang bias digunakan yaitu `method POST` dan `method GET`. Metode `GET` merupakan metode *default* pengiriman data sebuah *form*. Data dikirimkan ke server dengan meletakkan data pada bagian akhir

URL. Sebagai contoh, URL *action* menunjuk ke `proses.php` dan terdapat elemen dengan nama “kota” dengan nilai “Jakarta” dan “telepon” dengan nilai “2503645”, maka URL akhir yang dikirim ke server adalah `proses.php?kota=Jakarta&telepon=2503645`.

Tidak seperti metode GET, metode POST mengirimkan data secara terpisah dari URL dalam format standar input. *Script* mengambil data form dari input dan dapat menyimpan data dalam jumlah banyak menggunakan struktur data *array*. Data yang dikirim lebih aman karena tidak tampak pada URL. Metode POST cocok digunakan untuk pengiriman data besar dan *file* sedangkan GET digunakan untuk pengiriman data form yang ringkas melalui URL.

```
<form action="aksi_insert.php" method="POST">
    .
    form elements
    .
</form>
```

### 3. Atribut `enctype=""`

`Enctype` merupakan atribut optional pada *form* yang tidak selalu di butuhkan. *Enctype* baru akan dibutuhkan jika, form di rancang dengan pilihan upload *file*.

```
<form action="aksi_insert.php" method="POST"
enctype="multipart/form-data">
    .
    form elements
    .
</form>
```

Terdapat juga Atribut tambahan seperti:

- Id, nama identitas yang dapat digunakan pada saat menggunakan Javascript() dengan fungsi getElementById().
- Name, nama *form* yang digunakan pada saat melakukan pemrosesan data pada form.

#### 4.2.2. Bagian *Form*

Pada *form*, terdapat elemen-elemen khusus yang dapat digunakan untuk membuat berbagai macam bentuk input dari *user*, seperti pada tabel 3.1. Beberapa elemen yang umum digunakan yaitu:

- `<input>`

Salah satu elemen *form* yang paling sering digunakan adalah elemen `<input>`. Elemen `<input>` dapat ditampilkan dalam beberapa cara, tergantung pada atribut *type*.

```
<html>
  <body>

    <h2>The input Element</h2>

    <form action="/action_page.php">
      <label for="fname">First name:</label><br>
      <input type="text" id="fname"
        name="fname"><br><br>
      <input type="submit" value="Submit">
    </form>

  </body>
</html>
```


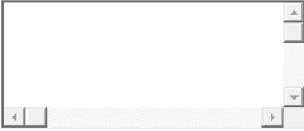

## Tampilan :

### The input Element

First name:

Submit

Tabel 4. 1 Elemen-elemen *Form*

Elemen Form	Keterangan	Tampilan Halaman
<u>TEXTBOX</u> What's your name?   <INPUT TYPE="text" NAME="email" />	Tipe input textbox dengan nama "email"	What's your email ? 
<u>TEXT AREA</u> Have you got anything to tell me?  <TEXTAREA NAME="comment" ROWS="13" COLS="50"> </TEXTAREA>	Nama textarea "comment" dengan jumlah baris 13 dan jumlah kolom 50 karakter	Have you got anything to tell me? 
<u>DROP DOWN BOX</u> What's your favorite color? <select name="favcolor"> <option selected>red <option>orange <option>yellow	Memulai drop down box dengan nama "favcolor", dengan opsi default "red"	What's your favorite color? 

<pre>&lt;option&gt;green &lt;option&gt;gray&lt;/opti on&gt; &lt;/select&gt;&lt;br&gt;</pre>		
<p><b><u>CHECK BOXES</u></b> I enjoy (check all that apply):&lt;br/&gt; &lt;INPUT TYPE="checkbox" NAME="fav"&gt; sports &lt;INPUT TYPE="checkbox" NAME="movies"&gt; horror movies</p>	<p>Tipe input adalah checkbox dengan nama "fav"</p>	<p>I enjoy (check all that apply): <input checked="" type="checkbox"/> sports <input type="checkbox"/> horror movies</p>
<p><b><u>RADIO BUTTONS</u></b> What is your Gender? &lt;br&gt; &lt;INPUT TYPE="radio" NAME="gender" VALUE="male" CHECKED&gt; MALE &lt;INPUT TYPE="radio" NAME="gender" VALUE="female"&gt; FEMALE</p>	<p>Input bertipe radio dengan nama "gender" dengan nilai "male" dan "female" dengan nilai default yang ter-checked adalah male</p>	<p>What is your Gender? <input checked="" type="radio"/> MALE <input type="radio"/> FEMALE</p>
<p><b><u>SUBMIT BUTTON</u></b> &lt;INPUT TYPE="submit" VALUE="Mail My</p>	<p>Tipe input submit dengan nilai "Mail My Comments"</p>	<p><input type="submit" value="Mail My comments"/> <input type="submit" value="Clear Form"/></p>

```
function nama(parameter1, parameter2, ...) {  
    // kode diletakkan di sini  
}
```

Semua variabel yang diletakkan di dalam fungsi bersifat lokal yang hanya dikenali di dalam fungsi itu saja. Untuk mengembalikan hasil fungsi gunakan fungsi *return*.

Untuk memanggil fungsi, tuliskan nama fungsi serta parameter yang dibutuhkan fungsi tersebut. Sebagai contoh:

```
<script>  
    function ganjilgenap(nilai_local) {  
        if(nilai_local % 2 == 0)  
            return "Genap";  
        else  
            return "Ganjil";  
    }  
    var nilai = 5;  
    alert (nilai + " termasuk bilangan " +  
ganjilgenap(nilai));  
</script>
```

#### 6.2.10. Librari jQuery

jQuery menawarkan cara yang lebih sederhana dalam melakukan tugas Javascript, bebas dari pengkodean Javascript yang rumit, sehingga pengaplikasiannya cepat dan konsisten serta dapat berjalan pada browser apapun. Penyederhanaan lainnya antara lain:

- Elemen HTML maupun selektor CSS merupakan *selector* di jQuery.
- Akses *selector* lebih menyeluruh dan fleksibel jika dibandingkan dengan menggunakan kueri DOM.
- Performansi yang lebih baik karena metode-metode jQuery



mbolehkan update hirarki DOM dan dapat menjelajahi hirarki elemen dengan satu baris perintah, sehingga tugas sulit seperti animasi elemen sangat mudah dilakukan.

- Penanganan *event* dimungkinkan melalui metode-metode yang memasukkan *event listeners* pada elemen yang diinginkan tanpa melakukan pengkodean berarti.

Untuk menggunakan *library* ini, sisipkan kode berikut pada dokumen HTML:

1. Sumber diunduh dari <http://jquery.com> dan disimpan di folder root web

```
<head>
  <script src="jquery-3.6.0.min.js"></script>
</head>
```

2. Sumber disimpan di CDN (Content Delivery Network).

Misalnya menggunakan CDN dari Google.

```
<head>
  <script
    src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
</head>
```

Syntax dasar jQuery:

```
$(selector).action()
```

Tanda \$ mendefinisikan akses jQuery. Sebuah *selector* melakukan "query (atau menemukan)" elemen HTML. Beberapa *selector* pada jQuery pada tabel 6.4:

Tabel 6. 4 Selector jQuery

Selector	Merujuk
<code>\$(this)</code>	Elemen sendiri
<code>\$("p")</code>	Elemen <code>&lt;p&gt;</code>
<code>\$(".test")</code>	Elemen dengan nama class "test"
<code>\$("#p.test")</code>	Elemen <code>&lt;p&gt;</code> yang memiliki class "test"
<code>\$(p:first)</code>	Elemen <code>&lt;p&gt;</code> pertama dalam dokumen
<code>\$("#test")</code>	Elemen dengan id="test"
<code>\$(tr:even)</code>	Elemen <code>&lt;tr&gt;</code> bernomor genap
<code>\$( "[href]" )</code>	Elemen yang memiliki Atribut "href"
<code>\$( "p:nth-child(2)" )</code>	Elemen <code>&lt;p&gt;</code> pada urutan/ anak kedua dalam elemen parent-nya
<code>\$( "div &gt; p" )</code>	Elemen <code>&lt;p&gt;</code> yang merupakan anak langsung dari elemen <code>&lt;div&gt;</code>
<code>\$( "div p" )</code>	Elemen <code>&lt;p&gt;</code> yang merupakan keturunan dari elemen <code>&lt;div&gt;</code> (tidak harus langsung)
<code>\$( "div + p" )</code>	Elemen <code>&lt;p&gt;</code> yang bertetangga langsung dengan elemen <code>&lt;div&gt;</code>
<code>\$( "div ~ p" )</code>	Elemen <code>&lt;p&gt;</code> yang muncul setelah elemen <code>&lt;div&gt;</code>
<code>\$( "input:not(:empty)" )</code>	Elemen <code>&lt;input&gt;</code> yang tidak kosong
<code>\$( ":input" ) :</code> <code>\$( ":text" )</code> <code>\$( ":checkbox" )</code> <code>\$( ":radio" )</code> <code>\$( ":file" )</code> <code>\$( ":submit" )</code> <code>\$( ":selected" )</code> <code>\$( ":checked" )</code> <code>\$( ":enabled" )</code> <code>\$( ":disabled" )</code>	Elemen <code>&lt;input&gt;</code> : Tipe text Tipe checkbox Tipe radio Tipe <i>file</i> Tipe submit Semua item dropdown terpilih Semua radio/checkbox terpilih Semua elemen form aktif Semua elemen form tidak aktif
<code>\$( "a[target!='_blank']" )</code>	Elemen yang memiliki Atribut target bukan blank

<code>\$("[title^='Helo']")</code>	Elemen dengan Atribut title diawali kata "Helo"
<code>\$("[href\$='.jpg']")</code>	Elemen dengan Atribut href diakhiri kata ".jpg"

Sebuah *selector* memiliki *action()* yang disebut juga *method* yang akan dikerjakan pada *selector* tersebut. Beberapa manipulasi yang dapat dilakukan:

- Manipulasi elemen HTML/DOM
- Manipulasi *style* CSS
- Manipulasi event
- Manipulasi efek dan animasi
- Asynchronous Javascript and XML
- Lainnya

Masing-masing contoh *method* dapat dilihat pada tabel 6.5.

Tabel 6. 5 Action jQuery

Method	Deskripsi
<b>HTML/CSS</b>	
<code>addClass() / removeClass() / toggleClass()</code>	Menambahkan/ menghapus/ toggle kelas pada selector
<code>hasClass()</code>	Mengecek apakah selector memiliki class tertentu
<code>after() / before()</code>	Menambahkan konten setelah/ sebelum selector dan selector diletakkan di awal
<code>insertAfter() / insertBefore()</code>	Sama dengan after/ before, hanya parameter selector diletakkan di belakang
<code>empty()</code>	Menghapus semua konten (elemen dan teks) dari selector

<code>text()</code>	Mengambil atau mengambil semua teks di dalam selector
<code>html()</code>	Mengambil atau menambahkan semua konten di dalam selector
<code>val()</code>	Mengambil atau menambahkan nilai dari Atribut value pada elemen-elemen form
<code>attr("Atribut")</code>	Mengambil atau menambahkan nilai pada Atribut html
<code>css()</code>	Menambahkan <i>style</i> css pada selector
<code>prop()</code>	Mengambil atau menambahkan nilai pada properti <i>style</i>
<code>prepend()/append()</code>	Menambahkan konten yang diletakkan di awal/ di akhir
<code>prependTo()/appendTo()</code>	Sama dengan <code>prepend/append</code> , hanya parameter selector diletakkan di belakang
<code>scrollTop()/scrollLeft()</code>	Mengambil atau menambahkan posisi scrollbar vertikal/ horizontal dari selector
<code>height()/width()</code>	Mengambil atau menambahkan tinggi/ lebar dari selector
<code>innerHeight()/innerWidth()</code>	Mengambil atau menambahkan tinggi/ lebar dari selector, termasuk padding namun border tidak
<code>outerHeight()/outerWidth()</code>	Mengambil atau menambahkan tinggi/ lebar dari selector, termasuk padding dan border
<b>Jelajah elemen</b>	
<code>first()/last()</code>	Memilih elemen pertama / terakhir yang sesuai dengan selector
<code>parent()/</code>	Memilih elemen parent / child dari

children()	selector
next()/ prev()	Memilih elemen setelah / sebelum selector
closest()	Memilih elemen parent pertama dari selector
not()	Memilih elemen yang tidak sesuai
find()	Memilih elemen turunan yang sesuai dari selector
eq(n)	Memilih selector yang sesuai dengan nomor indeks n
slice(n)	Mengeluarkan selector yang sesuai sebanyak n
each(function())	Mengeksekusi fungsi untuk tiap elemen
<b>Event: Mouse Event</b>	
click()	Event klik
dblclick()	Event double klik
mouseover()/ mouseout()	Event mouseover/ mouseout
hover()	Event mouseover + mouseout
scroll()	Event mouse discroll
submit()	Event form disubmit
event.pageX()	Membaca posisi mouse dari kiri dokumen
event.pageY()	Membaca posisi mouse dari atas dokumen
<b>Event: Keyboard Event</b>	
change()	Event perubahan pada elemen input
blur()	Event blur pada elemen input
focus()	Event fokus pada elemen input

<code>keypress()</code>	Event keyboard ditekan pada elemen input
<b>Event: Lainnya</b>	
<code>event.preventDefault()</code>	Mencegah aksi default dari event
<code>event.which()</code>	Membaca key keyboard atau mouse yang ditekan dalam kode numerik
<code>select()</code>	Memilih teks dalam elemen input
<code>trigger(event)</code>	Memicu event tertentu
<code>ready()</code>	Mendefinisikan fungsi yang dieksekusi ketika DOM selesai dipasang
<b>Efek</b>	
<code>animate(style)</code>	Menjalankan animasi dengan <i>style</i> yang diinginkan
<code>show()/hide()/toggle()</code>	Memunculkan/ menyembunyikan/ toggle elemen selector dengan beberapa level durasi
<code>fadeIn()/fadeOut()/fadeToggle()</code>	Memunculkan/ menyembunyikan/ toggle selector secara fading dengan beberapa level durasi
<code>slideUp()/slideDown()/slideToggle()</code>	Memunculkan/ menyembunyikan/ toggle selector secara sliding dengan beberapa level durasi
<b>Asynchronous Javascript and XML</b>	
<code>\$.ajax()</code>	Memanggil data dengan metode asynchronous
<code>\$.get("target", {object}, function(output))</code>	Memanggil data dari alamat target server dengan metode HTTP GET, dapat menyertakan data tambahan berupa objek dan kembalian berupa output
<code>\$.getJSON()</code>	Memanggil data dalam format JSON dengan metode HTTP GET

<code>\$.getScript()</code>	Mengambil (dan mengeksekusi) Javascript dengan metode HTTP GET
<code>\$.post()</code>	Mengambil data dari server dengan metode HTTP POST
<code>load()</code>	Mengambil data dari server dan meletakkannya hasilnya pada selector
<code>serialize()</code>	Menyatukan elemen-elemen form dalam format nama=nilai menjadi sepenggal string sebelum dikirim
<code>serializeAll()</code>	Menyatukan elemen-elemen form menjadi array nama dan nilainya

Berikut contoh pemanfaatan jQuery:

- Dokumen HTML Form

```

<!DOCTYPE html>
<style>
  .row{
    display: flex; justify-content: space-between;
    margin:10px;
  }
</style>
<form method="post" id="form_cari">
  <div class="row">
    <label for="prodi">Pekerjaan</label>
    <select id="pekerjaan" name= "jenis_pekerjaan">
      <option value="">Pilih jenis
      pekerjaan...</option>
      <option value="pegawai">Pegawai</option>
      <option value="buruh">Buruh</option>
    </select>
  </div>
  <div class="row">
    <label for="kelas">Status Pekerjaan</label>
    <select id="status" name="status_pekerjaan">
      <option value="">Status belum
      tersedia...</option>
    </select>
  </div>
  <div class="row">

```

```

<button id="bfilter" type="submit"> Cari
</button>
</div>
<div id="form_status"></div>
</form>
<script src="jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function() {
    $("#form_cari ").submit(kirim_data);
    $("#pekerjaan").change(function(){
        pekerjaan = $(this).val();
        $.post("coba.php",{pekerjaan: pekerjaan},
        function(output) {
            $("#status").html(output);
            console.log(output);
        });
    });
});
function kirim_data(e){
    if($("#pekerjaan").val() != '' &&
    $("#status").val() != ''){
        $("#form_status").html("Data berhasil
        terkirim");
        var data = $(this).serialize();
        console.log(data);
    }else
        $("#form_status").html("Periksa kembali
        isian pekerjaan dan status");
    e.preventDefault();
}
</script>

```

- Script coba.php sebagai script yang menerima data serta memberi respon balik yang sesuai

```

<?php
$pekerjaan = $_POST['pekerjaan'];
echo "<option value=''>Pilih status
$pekerjaan</option>";
if($pekerjaan == 'pegawai'){
    echo "<option value='1'>Kontrak</option>";
    echo "<option value='2'>Tetap</option>";
}else if($pekerjaan == 'buruh'){
    echo "<option value='1'>Harian</option>";
    echo "<option value='2'>Proyek</option>";
}
?>

```



## Tampilan:

- HTML Form



The screenshot shows a form with two labels: "Pekerjaan" and "Status Pekerjaan". Below "Pekerjaan" is a dropdown menu with the text "Pilih jenis pekerjaan..." and a list of options: "Pilih jenis pekerjaan...", "Pegawai", and "Buruh". Below "Status Pekerjaan" is a "Cari" button.

- Script coba.php merespon perubahan dropdown pekerjaan

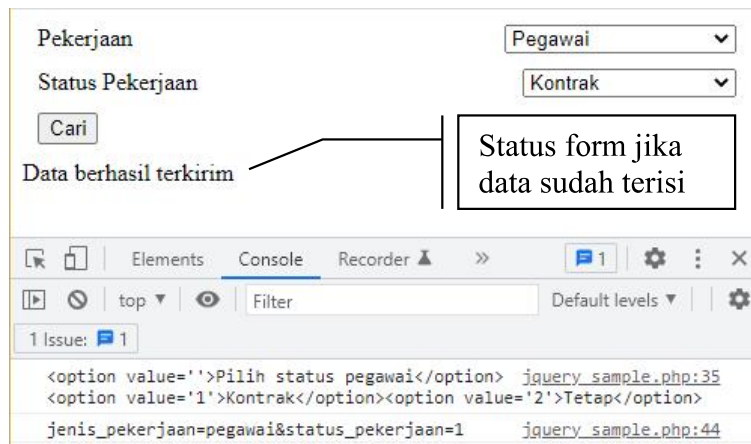


The screenshot shows a browser window with a form. The "Pekerjaan" dropdown is set to "Pegawai". The "Status Pekerjaan" dropdown is open, showing options: "Pilih status pegawai", "Kontrak", and "Tetap". Below the form is a "Cari" button. The browser's developer console is open, showing the following HTML code snippet:

```
<option value=''>Pilih status pegawai</option>
<option value='1'>Kontrak</option>
<option value='2'>Tetap</option>
```

Respon dari  
coba.php

- Script coba.php merespon submit form yaitu melalui fungsi kirim\_data()



The screenshot shows a browser window with a form. The "Pekerjaan" dropdown is set to "Pegawai". The "Status Pekerjaan" dropdown is set to "Kontrak". Below the form is a "Cari" button. The text "Data berhasil terkirim" is displayed below the "Cari" button. The browser's developer console is open, showing the following HTML code snippet:

```
<option value=''>Pilih status pegawai</option>
<option value='1'>Kontrak</option>
<option value='2'>Tetap</option>
jenis_pekerjaan=pegawai&status_pekerjaan=1
```

Status form jika  
data sudah terisi

Format data form  
hasil fungsi serialize  
yang siap kirim

### 6.3. Rangkuman

Javascript merupakan bahasa pemrograman yang berfungsi untuk membuat halaman web menjadi interaktif. Javascript dapat terhubung dengan halaman web HTML dengan memanfaatkan Object Model. Javascript dapat memilih objek pada halaman web berdasarkan *tag*, nama id atau nama *class*. Input dan output Javascript menggunakan beberapa metode, demikian juga untuk metode penanganan *string* dan numerik.

### 6.4. Tugas Latihan

1. Buatlah sebuah form yang menerima input nama dan email, kemudian tampilkan nama dan email tersebut dalam bentuk alert “Terima kasih *Nama!* Bukti akan dikirimkan ke *email* Saudara”.
2. Buatlah fungsi checkEmail agar string yang diinputkan wajib memiliki karakter @ dan . memanfaatkan fungsi match.
3. Lakukan pengecekan format email menggunakan metode checkEmail kemudian buatlah sebuah paragraph di samping isian email yang menampilkan pesan Email Valid atau Invalid menggunakan metode getElement
4. Tentukan hasil dari kode berikut:

```
<script>
var bil = 1;
do {
  document.write(bil+ "<br>");
  bil++;
}
```