



# BUKU AJAR PEMROGRAMAN BERBASIS WEB

Ika Ratna Indra Astutik  
Mochamad Alfian Rosid



# **BUKU AJAR PEMROGRAMAN BERBASIS WEB**

Oleh

**Ika Ratna Indra Astutik, S.Kom., M.T.  
Mochamad Alfian Rosid, S.Kom, M.Kom.**



Diterbitkan oleh

**UMSIDA PRESS**

**BUKU AJAR**  
**PEMROGRAMAN BERBASIS WEB**

**Penulis :**

Ika Ratna Indra Astutik, S.Kom., M.T.

Mochamad Alfian Rosid, S.Kom, M.Kom.

**ISBN : 978-623-7578-22-2**

**Editor :**

Septi Budi Sartika, M.Pd

M. Tanzil Multazam , S.H., M.Kn.

**Copy Editor :**

Fika Megawati, S.Pd., M.Pd.

**Design Sampul dan Tata Letak :**

Mochamad Nashrullah, S.Pd

**Penerbit :**

UMSIDA Press

**Redaksi :**

Universitas Muhammadiyah Sidoarjo

Jl. Mojopahit No 666B

Sidoarjo, Jawa Timur

**Cetakan pertama, Agustus 2019**

© Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dengan suatu apapun  
tanpa ijin tertulis dari penerbit.

## TINJAUAN MATA KULIAH

### 1. Deskripsi Singkat Mata Kuliah

Mata kuliah Pemrograman Berbasis Web membahas berbagai hal terkait dengan teknologi web, Bahasa pemrograman berbasis web serta pembuatan aplikasi berbasis web, yaitu sejarah perkembangan teknologi web, konsep dasar web, Hypertext Markup language (HTML), Cascading Style Sheet (CSS), Javascript, PHP dan Mysql.

### 2. Kegunaan Mata Kuliah

Dengan mengikut mata kuliah pemrograman berbasis web mahasiswa akan lebih memahami dan mengetahui tentang website dan aplikasi berbasis website serta mahasiswa juga mampu merancang dan membuat sebuah aplikasi berbasis web dalam berbagai bidang yang bisa di akses secara *offline* atau *online*.

### 3. Standart Kompetensi

- a. Mahasiswa dapat memahami dan mengetahui teknologi yang mendukung pemrograman berbasis web.
- b. Mahasiswa dapat menerapkan HTML sebagai dasar bahasa pemrograman web.
- c. Mahasiswa dapat memahami dan menerapkan CSS dalam pembuatan aplikasi web terutama tampilan web.
- d. Mahasiswa dapat menerapkan PHP dan MYSQL dalam pembuatan aplikasi web dinamis.

#### 3.1. Kompetensi Dasar

- a. Mampu mengetahui teknologi yang mendukung pemrograman berbasis web.
- b. Mampu menerapkan HTML sebagai dasar bahasa pemrograman web.
- c. Mampu menerapkan CSS dalam pembuatan aplikasi berbasis web terutama dalam desain tampilan web.
- d. Mampu menerapkan bahasa pemrograman PHP dan basis data MySQL dalam pembuatan aplikasi web dinamis.

### 3.2. Indikator

- a. Mahasiswa dapat membuat program sederhana menggunakan HTML.
- b. Mahasiswa dapat membuat tampilan web menggunakan CSS.
- c. Mahasiswa mampu membuat aplikasi berbasis web dinamis menggunakan bahasa pemrograman PHP dan MySQL.

## KATA PENGANTAR

Assalamualaikum wa Rahmatullah wa Barakatuh

Puji syukur kami panjatkan ke hadirat Allah SWT karena atas rahmat dan karunia-Nya kami bisa menyelesaikan pembuatan buku ajar "**Pemrograman Berbasis web**" bagi mahasiswa Informatika.

Buku ini, merupakan buku pegangan untuk mendukung proses belajar mahasiswa matakuliah "pemrograman berbasis web" Program Studi Informatika, yang secara spesifik berisi tentang perkembangan teknologi web, pengenalan dan sintaks pemrograman berbasis web seperti *HyperText Markup Language* (HTML), Javascript, *Cascading Style Sheets* (CSS), pemrograman server-side dengan *Hypertext Preprocessor* (PHP) dan MySQL serta penerapan dalam studi kasus. Kami menyadari, bahwa dalam buku ajar yang kami susun ini masih banyak kekurangan. Kami berharap pembaca maupun pengguna buku ajar ini dapat memberikan masukan atau kritik yang bersifat membangun agar kami bisa melakukan perbaikan di masa mendatang.

Ucapan terima kasih kepada pihak-pihak yang membantu terselesainya buku ajar ini. semoga buku ini dapat bermanfaat terutama untuk mahasiswa dalam proses pembelajaran pemrograman berbasis web.

Wassalamualaikum wa Rahmatullah wa Barakatuh

Sidoarjo, Juli 2019

Penyusun

## DAFTAR ISI

|   |          |
|---|----------|
| Cover.....                                    | i        |
| Tinjauan Mata Kuliah.....                     | v        |
| Kata Pengantar .....                          | vii      |
| Daftar Isi .....                              | viii     |
| <b>BAB 1 PENGANTAR PEMROGRAMAN WEB .....</b>  | <b>1</b> |
| 1.1. Pengertian Website 1                     |          |
| 1.2. Mengetahui Pemrograman Berbasis Web..... | 2        |
| 1.3. Kelebihan Pemrograman Berbasis Web.....  | 2        |
| 1.4. Bahasa Pemrograman Web.....              | 3        |
| 1.5. Aplikasi Pengembang Web.....             | 3        |
| 1.6. Istilah Dalam Pemrograman Web .....      | 4        |
| <b>BAB 2 HTML.....</b>                        | <b>6</b> |
| 2.1. HTML .....                               | 6        |
| 2.2. Struktur Dasar HTML .....                | 6        |
| 2.3. Unsur HTML.....                          | 7        |
| 2.3.1.Tag HTML.....                           | 7        |
| 2.3.2. Elemen di HTML .....                   | 9        |
| 2.3.3. Atribut di HTML .....                  | 10       |
| 2.4. Komentar di HTML.....                    | 12       |
| 2.5. Format Teks di HTML.....                 | 13       |
| 2.5.1. Heading Style HTML .....               | 13       |
| 2.5.2. Physical Style di HTML.....            | 14       |

|   |           |
|---|-----------|
| 2.5.3 Font Style di HTML .....                  | 16        |
| 2.5.4. Karakter dan Simbol Khusus di HTML ..... | 20        |
| 2.6. List di HTML .....                         | 40        |
| 2.7. Pindah Baris di HTML.....                  | 42        |
| 2.8. Garis Horizontal di HTML.....              | 43        |
| 2.9. Gambar di HTML.....                        | 45        |
| 2.10. Tabel di HTML.....                        | 51        |
| 2.11. Tautan (HyperLink) di HTML.....           | 57        |
| 2.12. Frame di HTML .....                       | 66        |
| 2.13. Tag Div (Division) di HTML.....           | 71        |
| <b>BAB 3 CASCADING STYLE SHEETS (CSS) .....</b> | <b>74</b> |
| 3.1. Pengenalan CSS .....                       | 74        |
| 3.2. Kelebihan CSS .....                        | 74        |
| 3.3. Sintaks CSS.....                           | 75        |
| 3.4. Penempatan CSS di HTML .....               | 76        |
| 3.5. Properti Dasar CSS di HTML.....            | 80        |
| 3.6. Penerapan Tag Division di CSS.....         | 86        |
| <b>BAB 4 JAVASCRIPT .....</b>                   | <b>89</b> |
| 4.1. Pengenalan Javascript .....                | 89        |
| 4.2. Sintaks Penulisan Javascript .....         | 89        |
| 4.3. Letak Javascript Dalam HTML.....           | 90        |
| 4.4. Penerapan Javascript Dalam HTML.....       | 91        |
| <b>BAB 5 FORMULIR (FORM) DI HTML.....</b>       | <b>97</b> |
| 5.1. Pengenalan Formulir .....                  | 97        |

|  |            |
|--|------------|
| 5.2. Elemen Form .....   | 97         |
| <b>BAB 6 PENGENALAN PHP .....</b>                                      | <b>110</b> |
| 6.1. Sejarah PHP .....   | 110        |
| 6.2. Manfaat PHP .....   | 111        |
| 6.3. Kelebihan dan Kekurangan PHP .....                                | 111        |
| 6.4. Modal Dasar Mempelajari PHP dan Aplikasi Yang<br>Diperlukan ..... | 115        |
| 6.5. Instalasi PHP, MySQL, & Apache dengan AMPPS .....                 | 117        |
| 6.6. Dasar-dasar PHP .....   | 122        |
| 6.7. Menyisipkan kode PHP di Halaman Web .....                         | 123        |
| <b>BAB 7 SINTAKS-SINTAKS DASAR PHP.....</b>                            | <b>124</b> |
| 7.1. Sintaks Default PHP .....   | 124        |
| 7.2. Tanda Semi Colon (;).....   | 125        |
| 7.3. Pindah Spasi.....   | 126        |
| 7.4. Variabel.....   | 128        |
| 7.4.1. Variabel string .....   | 128        |
| 7.4.2. Variabel Numeric .....  | 129        |
| 7.4.3. Variabel Array .....  | 130        |
| 7.5. Aturan Penamaan Variabel .....                                    | 133        |
| 7.6. Echo .....  | 133        |
| 7.6.1. Menampilkan Nilai Variabel dengan Echo .....                    | 135        |
| 7.7. Komentar dalam PHP .....  | 136        |
| 7.8. Operator .....  | 136        |
| 7.8.1. Operator Aritmatik .....  | 137        |

|   |            |
|---|------------|
| 7.8.2. Operator Perbandingan.....                           | 137        |
| 7.8.3. Kombinasi Operator Aritmatik dan Assignment<br>..... | 138        |
| 7.8.4. Operator Logika.....                                 | 139        |
| <b>BAB 8 MODULARITAS .....</b>                              | <b>142</b> |
| 8.1. Include().....   | 142        |
| 8.2. Recuire() .....  | 144        |
| 8.3. Statement Kontrol .....                                | 146        |
| 8.3.1. Statement IF .....                                   | 147        |
| 8.3.2. Statement Else.....                                  | 148        |
| 8.3.3. Statement Elself .....                               | 150        |
| 8.3.4. Statement SWITCH .....                               | 151        |
| 8.4. LOOPING/PERULANGAN.....                                | 153        |
| 8.4.1. Statement WHILE .....                                | 153        |
| 8.4.2. Statement FOR .....                                  | 155        |
| 8.4.3. Statement Foreach .....                              | 156        |
| 8.5. FUNCTION.....  | 158        |
| 8.5.1. Lingkup Variabel .....                               | 159        |
| 8.6. OPERASI FILE.....                                      | 160        |
| 8.6.1. Membuka File .....                                   | 161        |
| 8.6.2. Menutup File .....                                   | 161        |
| 8.6.3. Menulis ke File.....                                 | 162        |
| 8.6.4. Membaca Isi File.....                                | 162        |
| 8.6.5. Menghapus File .....                                 | 164        |

|  |     |
|--|-----|
| 8.6.6. Menambah Isi File .....                             | 164 |
| 8.7. BEKERJA DENGAN COOKIES.....                           | 165 |
| 8.7.1. Membuat Cookies .....                               | 165 |
| 8.7.2. Mengambil Informasi dari Cookie .....               | 166 |
| 8.8. BEKERJA DENGAN SESSION .....                          | 167 |
| 8.8.1. Memulai PHP – Session .....                         | 169 |
| 8.8.2. Menyimpan Data ke dalam Session .....               | 169 |
| 8.8.3. Memanggil Data yang Tersimpan dalam<br>Session..... | 170 |
| 8.8.4. Menghapus Data Session .....                        | 171 |
| 8.9. BEKERJA DENGAN FORM.....                              | 172 |
| 8.10. Method Post dan Get .....                            | 174 |
| BAB 9 PHP-MYSQL .....                                      | 176 |
| 9.1. Koneksi PHP ke MySQL.....                             | 177 |
| 9.2. Memberikan Query ke MySQL .....                       | 178 |
| 9.3. Mengambil Data dari MySQL.....                        | 180 |
| BAB 10 MEMBUAT CRUD (Create, Read, Update, Delete) .....   | 183 |
| 10.1. Membuat File Config .....                            | 184 |
| 10.2. Membuat File Index.....                              | 185 |
| 10.3. Membuat File ADD .....                               | 186 |
| 10.4. Membuat File Edit .....                              | 188 |
| 10.5. Membuat File Hapus .....                             | 190 |
| Daftar Pustaka .....                                       | 192 |

# BAB 1

## PENGANTAR PEMROGRAMAN WEB

Bab ini menjelaskan tentang pengetahuan dasar pemrograman berbasis web khususnya teknologi PHP dan MYSQL.

### 1.1. Pengertian Website

Website merupakan Kumpulan halaman website yang dibuat menggunakan bahasa HTML (*Hypertext Markup Language*) yang berisikan data dan informasi baik berupa gambar, teks, suara, animasi dan video yang bisa di akses menggunakan web browser melalui internet oleh semua orang di seluruh dunia (Abdulloh, 2018).

Website, secara umum di bagi menjadi 3 jenis antara lain :

1. Website Statis

Website statis merupakan website yang isinya tidak dilakukan perbaharuan secara berkala sehingga isinya akan selalu tetap (tidak *uptodate*). Website ini biasanya digunakan untuk menampilkan profil seperti profil organisasi atau perusahaan.

2. Website Dinamis

Website dinamis merupakan website yang isi (*content*) nya mengalami pembaruan secara berkala oleh pengelola web. Seperti website perusahaan atau perorangan yang bisnisnya berkaitan dengan internet misalnya website berita atau blog.

3. Website Interaktif

Website interaktif termasuk dalam kategori website dinamis, hanya saja informasi yang di sajikan tidak hanya di ubah oleh pemilik website tetapi juga oleh pengguna website itu sendiri. Contohnya website jejaring sosial atau website marketplace.

## **1.2. Mengenal Pemrograman Berbasis Web**

Pemrograman web terdiri dari 2 kata yaitu pemrograman dan web. Pemrograman merupakan proses atau cara pembuatan program. Sedangkan web adalah jaringan komputer yang terdiri dari kumpulan situs-situs internet yang terdiri atas teks, grafik, suara dan sumber daya animasi melalui hypertext transfer protokol (HTML). Jadi, pemrograman web adalah kegiatan pembuatan program atau aplikasi berbasis web untuk memproses data yang menghasilkan informasi sesuai dengan kehendak pemilik website yang di akses melalui jaringan internet.

## **1.3. Kelebihan Pemrograman Berbasis Web**

Pemrograman web atau aplikasi berbasis web memiliki kelebihan di bandingkan dari pada aplikasi berbasis dekstop karena :

- a. Akses Informasi lebih mudah karena bisa darimana saja melalui jaringan baik intranet maupun internet.
- b. Pemrograman web merupakan pemrograman server side jadi Jika terdapat perubahan aplikasi atau program maka client tidak perlu melakukan update cukup di lakukan di server.
- c. Data disimpan di server sehingga akses data untuk client dapat di atur sesuai dengan kebutuhan.
- d. Aplikasi dapat diakses melalui komputer dengan berbagai sistem operasi (*cross-platform*) yang memiliki browser.

Disamping kelebihan pemrograman web juga mempunyai kekurangan yaitu dari sisi performa aplikasi yang sangat bergantung pada jaringan yang digunakan. Pada jaringan yang lambat, kecepatan akses dan proses data juga akan lambat. Selain itu, karena aplikasi dapat diakses dimana saja maka rentan terhadap serangan yang

dilakukan oleh pihak yang tidak bertanggungjawab sehingga aplikasi harus mempunyai perlindungan yang baik.

#### **1.4. Bahasa Pemrograman Web**

Dalam pemrograman web terdapat beberapa bahasa utama yang digunakan dalam pembuatan website dinamis yang masing-masing bahasa mempunyai fungsi sendiri, sebagai berikut :

- a. HTML merupakan bahasa pemrograman yang berperan sebagai pembentuk struktur halaman website sesuai dengan layout yang diinginkan.
- b. CSS untuk membentuk desain website agar website tampil menarik.
- c. PHP untuk memproses data pada server sesuai dengan permintaan dari client menjadi informasi yang dibutuhkan client, PHP juga sebagai penghubung antara aplikasi web dengan basis data.
- d. SQL merupakan bahasa untuk mengatur pengolahan data antara aplikasi dengan basis data sebagai tempat penyimpanan data. Basis data yang digunakan yaitu MySQL, PostgreSQL, Oracle SQL Server dan lain sebagainya.
- e. Javascript digunakan sebagai bahasa untuk memproses data yang ada di client serta untuk memanipulasi HTML dan CSS.

#### **1.5. Aplikasi Pengembang Web**

Dalam membuat aplikasi web di butuhkan berbagai aplikasi yang membantu programmer dalam merancang dan menulis skrip-skrip program maupun menampilkan hasil program. Berikut ini beberapa aplikasi yang digunakan dalam pembuatan program web :

- a. Text Editor Berfungsi untuk mengetik skrip program. Yang termasuk dalam text editor yaitu Notepad, Notepad++, Sblime Text, Bracket, Atom dan lain sebagainya.
- b. Web Browser digunakan untuk menampilkan hasil program.
- c. Web Server merupakan perangkat lunak untuk menerima permintaan melalui protokol HTTP atau HTTPS dari client kemudian mengirimkan kembali dalam bentuk halaman-halaman web. salah satu web server adalah Apache yang dalam penggunaannya biasanya satu paket dengan PHP dan MySQL. Contohnya adalah Xampp dan Appserv.

## **1.6. Istilah Dalam Pemrograman Web**

Berikut ini beberapa istilah yang berhubungan dengan pemrograman web antara lain :

- a. Server merupakan komputer yang digunakan untuk memproses data dan menyimpan file-file website yang mempunyai kecepatan dan kapasitas penyimpanan data di atas komputer biasa.
- b. Client merupakan komputer yang digunakan untuk mengakses halaman website baik berupa komputer personal, Laptop atau perangkat mobile.
- c. Domain adalah nama yang mewakili alamat IP server website biasanya berupa kata yang mudah di ingat sehingga lebih mudah dalam mengakses website di internet.
- d. Hosting merupakan tempat untuk menyimpan file-file website pada server di internet agar bisa di akses di mana saja oleh pengguna internet melalui komputer, laptop atau mobile.
- e. URL (*Uniform Resources Locator*) merupakan nama unik untuk mengidentifikasi letak sumber daya website yaitu file HTML,

gambar, video yang akan tampil pada address bar beowser saat website tersebut di akses.

- f. HTTP (*Hypertext Transfer Protocol*) merupakan protokol yang mengatur pengiriman data dari client ke server atau sebaliknya yang akan menterjemahkan informasi agar bisa dibaca oleh pengguna webiste.

## **BAB 2 HTML**

Bab ini menjelaskan tentang dasar-dasar HTML dan tag-tag yang ada di HTML.

### **2.1. HTML**

HTML merupakan singkatan dari *Hypertext Markup Language* yaitu bahasa standar web yang dikelola penggunaannya 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.

File HTML di simpan dengan ekstensi .html. HTML merupakan sistem standar untuk menandai file teks untuk mencapai efek font, warna, grafik, dan hyperlink pada halaman World Wide Web.

### **2.2. Struktur Dasar HTML**

File HTML disimpan dengan ekstensi \*.html, jika di dalamnya tidak mengandung skrip PHP. Namun jika di dalamnya mengandung skrip PHP maka file disimpan dengan ekstensi \*.PHP.

Penulisan skrip HTML ada beberapa tag yang harus dituliskan dengan struktur yang sudah ditentukan, dimana setiap file HTML harus menerapkan struktur tersebut sebagai berikut :

```
<!DOCTYPE html>
<html>
<head><title>.....</title>
</head>
<body>
```

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

Keterangan :

1. <!DOCTYPE html> Menandakan bahwa dokumen yang sedang dibuka adalah HTML
2. Tag <html> adalah sebagai penanda dokumen yang dibuat adalah dokumen web.
3. Tag <head> adalah kepala halaman web.
4. Tag <title> adalah judul untuk setiap halaman web yang ditampilkan di tile bar browser atau di tab browser.
5. Tag <body> digunakan untuk meletakkan isi website.

## **2.3. Unsur HTML**

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

### **2.3.1. Tag HTML**

Tag merupakan simbol HTML yang berupa dua karakter "<" dan ">" yang mengapit suatu teks sebagai nama tag. Contohnya tag <html> adalah tag dengan nama html. Tag memiliki pembuka dan penutup yang berfungsi membatasi pengaruh dari tag tersebut. Tag penutup ditambah simbol "/" setelah tanda "<", misalkan </html> yang merupakan penutup tag<html>.

- a. Tag HTML bersifat incase sensitive sehingga penulisan bisa menggunakan huruf besar, kecil atau campuran. Misalkan <HTML>, <html> atau <Html> tidak akan berpengaruh pada hasil eksekusi program. namun, untuk standarnya lebih baik menggunakan huruf kecil.

- b. Tag di HTML bisa berisi tag lain, dimana tag pertama di buka dahulu ditutup terakhir sedangkan tag yang dibuka terakhir ditutup pertama. Namun, kesalahan dalam struktur ini tidak mengakibatkan program error.
- c. Penulisan tag HTML sebagai berikut :

```
<tag pembuka>  
objek yang dikenai perintah tag  
</tag penutup>
```

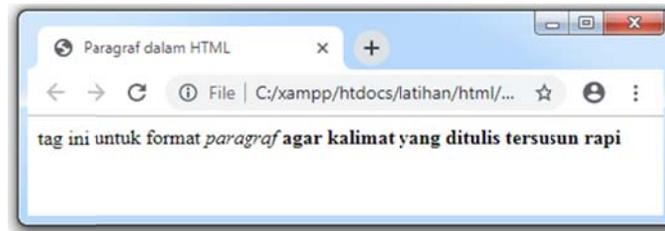
- d. Contoh penerapan tag  
Membuat format paragraf di web menggunakan tag <p> penerapannya dalam dokumen HTML sebagai berikut :

```
<p> tag ini untuk format paragraf </p>
```

Apabila lupa memberikan tag penutup, maka browser akan mengabaikan kesalahan tersebut dan akan tetap menampilkan hasilnya seolah-olah ada tag penutup. Namun hal tersebut dapat membingungkan programmer untuk membuat tag lain terutama tag didalam tag. Sebagai contoh :

```
<!DOCTYPE html>  
<html>  
<head>  
<title>Paragraf dalam HTML</title>  
<body>  
<p> tag ini untuk format <i>paragraf</i><b>  
agar kalimat yang ditulis tersusun rapi </b>  
</p>  
</body>  
</html>
```

Hasil kode HTML tersebut, jika diterjemahkan atau ditampilkan oleh browser menjadi :



Tag `<i>` pada kode HTML memberikan perintah ke browser untuk menampilkan format teks miring (*italic*) dan tag `<b>` untuk menebalkan teks (*bold*).

### 2.3.2. Elemen di HTML

Elemen adalah skrip HTML yang terdiri atas tag pembuka, isi tag, tag penutup dan atribut yang dimiliki oleh tag tersebut (apabila ada). Jika sebuah element ditampilkan pada browser, maka hanya isi element yang tampil di browser. Misal pada skrip berikut ini :

```
<p> tag ini untuk format paragraf </p>
```

Pada contoh skrip HTML di atas, elemen dari p adalah "`<p> tag ini untuk format </p>`". Elemen HTML tidak hanya berupa teks tetapi bisa tag lain yang ada di dalam tag lain. Misalkan :

```
<p> tag ini untuk format <i> paragraf </i></p>
```

Elemen dari p adalah "`<p> tag ini untuk format <i> paragraf </i></p>`".

### 2.3.3. Atribut di HTML

Atribut adalah property atau informasi tambahan yang digunakan untuk mengatur bagaimana elemen dari suatu tag akan ditampilkan. Bisa berupa pengaturan warna teks, ukuran huruf teks, dan lain sebagainya. Atribut mempunyai nama (*name*) dan nilai (*value*) yang ditulis di dalam simbol tag setelah nama tag di pisah dengan tanda spasi. Nilai suatu atribut ditulis di dalam tanda petik ganda ("..") di pisahkan dengan simbol sama dengan (=) dari nama atribut.

Contoh pada skrip HTML :

```
<a href="https://umsida.ac.id">link website umsida </a>
```

Pada contoh skrip HTML diatas, href="https://umsida.ac.id" adalah atribut. Href merupakan nama atribut sedangkan https://umsida.ac.id adalah nilai atau value dari atribut.

Atribut mempunyai karakteristik sebagai berikut :

- a. Atribut baik nama dan nilai hanya ditulis di tag pembuka.
- b. Setiap tag memiliki atribut yang berbeda-beda yang dapat diterapkan pada beberapa tag tersebut.
- c. Beberapa atribut dapat diterapkan ke semua tag atau disebut *global attribute*, misalnya class, id, style.
- d. Sebuah tag bisa mempunyai satu atribut, banyak atribut atau tanpa atribut sama sekali.
- e. Jika tag ditulis dengan beberapa atribut, masing-masing atribut harus di pisahkan dengan spasi.

Di bawah ini adalah beberapa atribut yang sering digunakan dalam HTML :

| Atribut  | Deskripsi   |
|----------|---|
| href     | Menentukan URL (alamat web) untuk tautan ( <i>link</i> )                      |
| src      | Menentukan URL (alamat web) untuk gambar                                      |
| alt      | Menentukan teks alternatif untuk gambar, saat gambar tidak dapat ditampilkan  |
| disabled | Menentukan bahwa elemen input harus dinonaktifkan                             |
| style    | Menentukan atribut pada style CSS inline untuk elemen                         |
| id       | Menentukan id unik untuk elemen   |
| title    | Menentukan informasi tambahan tentang elemen (ditampilkan sebagai a tool tip) |

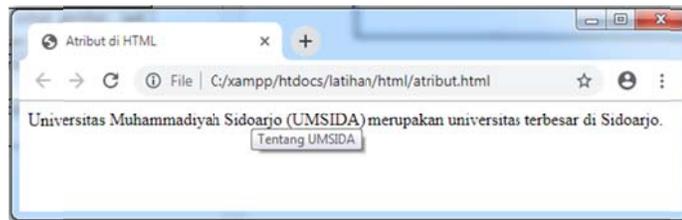
Contoh pada skrip HTML menggunakan atribut title :

```

<!DOCTYPE html>
<html>
<head>
<title>Atribut di HTML</title>
<body>
<p title="Tentang UMSIDA">
Universitas Muhammadiyah Sidoarjo (UMSIDA)
merupakan universitas terbesar di
Sidoarjo.
</p>
</body>
</html>

```

Hasil kode HTML tersebut, jika diterjemahkan atau ditampilkan oleh browser menjadi :



## 2.4. Komentar di HTML

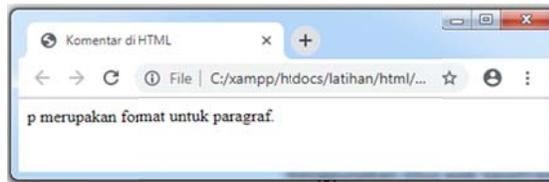
Tag komentar digunakan untuk memasukkan komentar dalam kode HTML. Komentar tidak ditampilkan oleh browser, tetapi komentar dapat membantu mendokumentasikan kode pada skrip HTML yang dibuat. Komentar juga berfungsi untuk debugging HTML, karena dapat memberikan keterangan pada setiap baris kode HTML untuk mencari kesalahan. Tag komentar di HTML adalah sebagai berikut :

```
<!-- tulis komentar di sini -->
```

Perhatikan bahwa ada tanda seru (!) pada tag pembuka, namun tidak dalam tag penutup. Seperti dalam skrip HTML berikut ini :

```
<!DOCTYPE html>
<html>
<head><title>Komentar di HTML</title>
</head>
<body>
<!-- Ini adalah sebuah komentar -->
<p> p merupakan format untuk paragraf.</p>
<!-- komentar tidak ditampilkan di browser -->
</body></html>
```

Hasil kode HTML tersebut, jika diterjemahkan atau ditampilkan oleh browser menjadi :



## 2.5. Format Teks di HTML

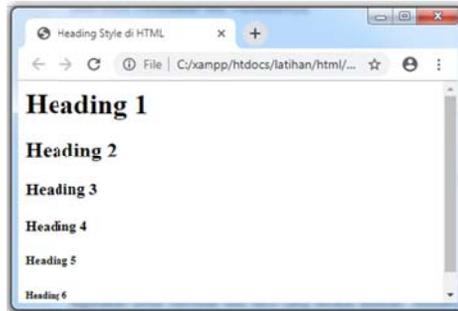
Format Teks di HTML digunakan untuk membuat teks yang ada dalam dokumen html menjadi kelihatan menarik, memperindah tampilan web.

### 2.5.1. Heading Style HTML

Heading merupakan judul atau sub judul dalam dokumen HTML. Tag heading berbeda dengan tag <title>, karena bisa muncul dalam halaman web. Dalam HTML ada 6 tingkatan heading, dimana heading 1 merupakan yang paling besar dan heading 6 yang paling kecil yang dapat dicontohkan dalam skrip HTML berikut ini :

```
<!DOCTYPE html>
<html>
<head><title>Heading Style di HTML</title>
</head>
<body>
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
</body>
</html>
```

Hasil kode HTML tersebut, jika diterjemahkan atau ditampilkan oleh browser menjadi :



Heading sangat penting untuk web terutama tampilan web. Mesin pencari menggunakan heading untuk mengindeks struktur dan isi dari halaman web. Pengguna melihat halaman web berdasarkan heading sehingga sangat penting menggunakan heading untuk menampilkan struktur dokumen. Heading `<h1>...</h1>` digunakan untuk heading utama, diikuti oleh heading `<h2>...</h2>`, jika kurang penting menggunakan heading `<h3>...</h3>` dan seterusnya.

## 2.5.2. Physical Style di HTML

Physical style di HTML merupakan jenis format yang digunakan untuk mengubah teks tanpa tergantung jenis elemen dasar teks tersebut. Dimana elemen-elemen teks yang digunakan mempunyai kesamaan dengan saat mengetik dokumen di Microsoft Word seperti memiringkan teks, menebalkan huruf, menggaris bawah huruf dan lain-lain.

Physical style di HTML yang sering digunakan sebagai berikut :

| Physical Style                        | Deskripsi   |
|---------------------------------------|---|
| <b>.. <b>&lt;/b&gt;</b>               | Untuk menebalkan huruf pada teks  |
| <strong>.. <b>&lt;/strong&gt;</b>     | Untuk menebalkan huruf pada teks yang penting di web                        |
| <i>.. <i>&lt;/i&gt;</i>               | Untuk memiringkan huruf pada teks   |
| <u>.. <u>&lt;/u&gt;</u>               | Untuk <u>menggaris bawah</u> hi teks  |
| <s>.. <del>&lt;/s&gt;</del>           | Untuk <del>memberi coretan</del> pada teks                                  |
| <strike>.. <del>&lt;/strike&gt;</del> | Untuk <del>memberi coretan</del> pada teks                                  |
| <tt>.. <b>&lt;/tt&gt;</b>             | Untuk menampilkan teks dalam format font typewriter                         |
| <blink>.. <del>&lt;/blink&gt;</del>   | Untuk membuat animasi teks berkedip (tidak semua browser mendukung tag ini) |
| <big>.. <b>&lt;/big&gt;</b>           | Untuk membesarkan ukuran teks   |
| <small>.. <b>&lt;/small&gt;</b>       | Untuk mengecilkan ukuran teks   |
| <sub>.. <sub>&lt;/sub&gt;</sub>       | Membuat teks <sub>subscript</sub>   |
| <sup>.. <sup>&lt;/sup&gt;</sup>       | Membuat teks <sup>superscript</sup>   |

Berikut ini penerapannya dalam skrip HTML :

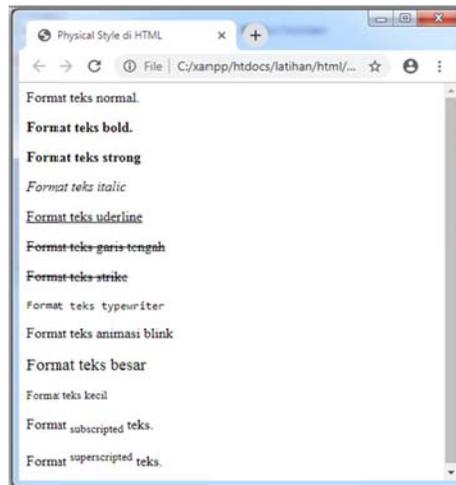
```

<!DOCTYPE html>
<html>
<head><title>Physical Style di HTML</title>
</head>
<body>
<p>Format teks normal.</p>
<p><b>Format teks bold.</b></p>
<p><strong>Format teks strong</strong></p>
<p><i>Format teks italic</i></p>
<p><u>Format teks uderline</u></p>
<p><s>Format teks garis tengah</s></p>
<p><strike>Format teks strike</strike></p>

```

```
<p><tt>Format teks typewriter</tt></p>
<p><blink>Format teks animasi blink</blink></p>
<p><big>Format teks besar</big></p>
<p><small>Format teks kecil</small></p>
<p>Format <sub>subscripted</sub> teks.</p>
<p>Format <sup>superscripted</sup> teks.</p>
</body>
</html>
```

Hasil kode HTML tersebut, jika diterjemahkan atau ditampilkan oleh browser menjadi :



### 2.5.3. Font Style di HTML

Tag font merupakan tag yang digunakan untuk menentukan ukuran huruf (size), warna huruf (color) dan jenis huruf (face) dalam HTML. Bentuk sintak font adalah sebagai berikut :

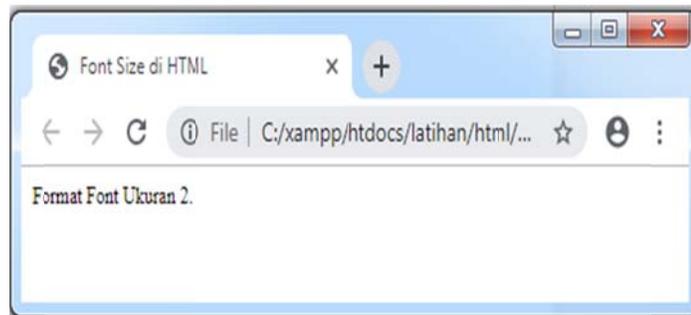
```
<font>...</font>
```

a. Format Ukuran Huruf (Font Size)

Font size merupakan sebuah atribut yang digunakan untuk mengubah ukuran huruf (font) dalam HTML. Contoh penerapan dalam skrip sebagai berikut :

```
<!DOCTYPE html>
<html>
<head><title>Font Size di HTML</title>
</head>
<body>
<font size="2">Format Font Ukuran 2.</font>
</body>
</html>
```

Hasil kode HTML tersebut, jika diterjemahkan atau ditampilkan oleh browser menjadi :



b. Format Warna Huruf (Font Color)

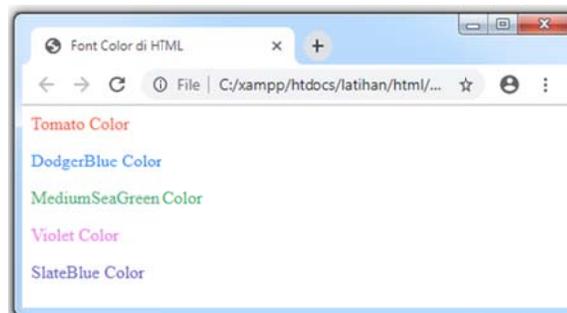
Font color merupakan sebuah atribut yang digunakan untuk mengubah warna huruf dalam HTML. Nilai (value) dari atribut color berupa nama warna langsung atau bisa juga berupa kode *hexadecimal*. Contoh skrip HTML menggunakan value nama warna sebagai berikut :

```
<!DOCTYPE html>
<html>
<head><title>Font Color di HTML</title>
</head>
<body>
<p><font color="Tomato">Tomato
Color</font></p>
<p><font color="DodgerBlue">DodgerBlue
Color</font></p>
<p><font
color="MediumSeaGreen">MediumSeaGreen
Color</font></p>
<p><font color="Violet">Violet
Color</font></p>
<p><font color="SlateBlue">SlateBlue
Color</font></p>
</body>
</html>
```

Contoh font color menggunakan value hexadecimal :

```
<!DOCTYPE html>
<html>
<head><title>Font Color di HTML</title>
</head>
<body>
<p><font color="#ff6347">Tomato
Color</font></p>
<p><font color="#1e90ff">DodgerBlue
Color</font></p>
<p><font color="#3cb371">MediumSeaGreen
Color</font></p>
<p><font color="#ee82ee">Violet
Color</font></p>
<p><font color="#6a5acd">SlateBlue
Color</font></p>
</body>
</html>
```

Hasil kode HTML tersebut, jika diterjemahkan atau ditampilkan oleh browser menjadi :



c. Format Jenis Huruf (Font Face)

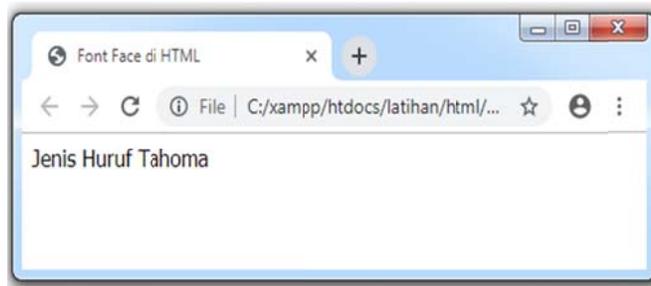
Font face digunakan untuk mengubah jenis huruf (font) yang digunakan dalam HTML.

```

<!DOCTYPE html>
<html>
<head><title>Font Face di HTML</title>
</head>
<body>
<font face="Tahoma">Jenis Huruf
Tahoma</font>
</body>
</html>

```

Hasil tampilan di browser :



#### 2.5.4. Karakter dan Simbol Khusus di HTML

Dalam HTML bisa juga disisipkan atau ditambahkan karakter khusus untuk mendukung informasi yang diberikan website. Berikut ini karakter khusus dan simbol yang ada dalam HTML.

| Kode Nomor HTML | Kode Nama HTML | Karakter /Simbol | Deskripsi |
|-----------------|----------------|------------------|-----------|
| &#32;           |                |                  | Spasi     |

|       |        |    |               |
|-------|--------|----|---------------|
| &#33; |        | !  | Tanda seru    |
| &#34; | &quot; | "  | Kutip ganda   |
| &#35; |        | #  | Simbol pagar  |
| &#36; |        | \$ | Simbol dollar |
| &#37; |        | %  | Simbol persen |
| &#38; | &amp;  | &  | Simbol dan    |
| &#39; |        | '  | Kutip satu    |
| &#40; |        | (  | Kurung buka   |
| &#41; |        | )  | Kurung tutup  |
| &#42; |        | *  | Tanda bintang |
| &#43; |        | +  | Tanda tambah  |
| &#44; |        | ,  | Tanda koma    |
| &#45; |        | -  | Tanda kurang  |
| &#46; |        | .  | Tanda titik   |
| &#47; |        | /  | Garis miring  |
| &#48; |        | 0  | Nol           |
| &#49; |        | 1  | Satu          |
| &#50; |        | 2  | Dua           |

|       |      |   |             |
|-------|------|---|-------------|
| &#51; |      | 3 | Tiga        |
| &#52; |      | 4 | Empat       |
| &#53; |      | 5 | Lima        |
| &#54; |      | 6 | Enam        |
| &#55; |      | 7 | Tujuh       |
| &#56; |      | 8 | Delapan     |
| &#57; |      | 9 | Sembilan    |
| &#58; |      | : | Titik dua   |
| &#59; |      | ; | Titik koma  |
| &#60; | &lt; | < | Kurang dari |
| &#61; |      | = | Sama dengan |
| &#62; | &gt; | > | Lebih dari  |
| &#63; |      | ? | Tanda tanya |
| &#64; |      | @ | Simbol [at] |
| &#65; |      | A |             |
| &#66; |      | B |             |
| &#67; |      | C |             |
| &#68; |      | D |             |

|       |  |   |  |
|-------|--|---|--|
| &#69; |  | E |  |
| &#70; |  | F |  |
| &#71; |  | G |  |
| &#72; |  | H |  |
| &#73; |  | I |  |
| &#74; |  | J |  |
| &#75; |  | K |  |
| &#76; |  | L |  |
| &#77; |  | M |  |
| &#78; |  | N |  |
| &#79; |  | O |  |
| &#80; |  | P |  |
| &#81; |  | Q |  |
| &#82; |  | R |  |
| &#83; |  | S |  |
| &#84; |  | T |  |
| &#85; |  | U |  |
| &#86; |  | V |  |

|        |  |   |   |
|--------|--|---|---|
| &#87;  |  | W |   |
| &#88;  |  | X |   |
| &#89;  |  | Y |   |
| &#90;  |  | Z |   |
| &#91;  |  | [ | Kurung siku buka                        |
| &#92;  |  | \ | Garis miring terbalik/ <i>backslash</i> |
| &#93;  |  | ] | Kurung siku tutup                       |
| &#94;  |  | ^ | Tanda sisipan/sirkumfleksa              |
| &#95;  |  | _ | Garis bawah                             |
| &#96;  |  | ` | Tanda petik                             |
| &#97;  |  | a |   |
| &#98;  |  | b |   |
| &#99;  |  | c |   |
| &#100; |  | d |   |
| &#101; |  | e |   |
| &#102; |  | f |   |
| &#103; |  | g |   |

|        |  |   |  |
|--------|--|---|--|
| &#104; |  | h |  |
| &#105; |  | i |  |
| &#106; |  | j |  |
| &#107; |  | k |  |
| &#108; |  | l |  |
| &#109; |  | m |  |
| &#110; |  | n |  |
| &#111; |  | o |  |
| &#112; |  | p |  |
| &#113; |  | q |  |
| &#114; |  | r |  |
| &#115; |  | s |  |
| &#116; |  | t |  |
| &#117; |  | u |  |
| &#118; |  | v |  |
| &#119; |  | w |  |
| &#120; |  | x |  |
| &#121; |  | y |  |

|        |          |   |                                       |
|--------|----------|---|---------------------------------------|
| &#122; |          | z |                                       |
| &#123; |          | { | Kurung kurawal buka                   |
| &#124; |          |   | Garis tegak/ <i>pipe</i>              |
| &#125; |          | } | Kurung kurawal tutup                  |
| &#126; |          | ~ | Simbol ekuivalen                      |
| &#160; | &nbsp;   |   | <i>Non-breaking space</i>             |
| &#161; | &iexcl;  | ¡ | Tanda seru terbalik                   |
| &#162; | &cent;   | ¢ | Simbol sen                            |
| &#163; | &pound;  | £ | Simbol pondsterling                   |
| &#164; | &curren; | ¤ | Simbol mata uang                      |
| &#165; | &yen;    | ¥ | Simbol yen                            |
| &#166; | &brvbar; | ¦ | Garis tegak terputus                  |
| &#167; | &sect;   | § | Simbol bagian/seksi                   |
| &#168; | &uml;    | ¨ | Titik dua di atas huruf vokal         |
| &#169; | &copy;   | © | Simbol hak cipta ( <i>Copyright</i> ) |
| &#170; | &ordf;   | ª | Indikator ordinal feminin             |
| &#171; | &laquo;  | « | Panah kiri ganda                      |

|        |          |   |   |
|--------|----------|---|---|
| &#172; | &not;    | ¬ | Simbol bukan                            |
| &#173; | &shy;    |   | Tanda penghubung halus                  |
| &#174; | &reg;    | ® | Simbol merek dagang terdaftar           |
| &#175; | &macr;   | ¯ | Garis atas                              |
| &#176; | &deg;    | ° | Simbol derajat                          |
| &#177; | &plusmn; | ± | Simbol lebih kurang                     |
| &#178; | &sup2;   | ² | Pangkat dua, kuadrat, persegi           |
| &#179; | &sup3;   | ³ | Pangkat tiga, kubik                     |
| &#180; | &acute;  | ´ | Aksen akut                              |
| &#181; | &micro;  | μ | Simbol mikro                            |
| &#182; | &para;   | ¶ | Simbol <i>pilcrow</i> , simbol paragraf |
| &#183; | &middot; | · | Titik tengah, koma Georgia              |
| &#184; | &cedil;  | ¸ | Spasi <i>cedilla</i>                    |
| &#185; | &sup1;   | ¹ | Pangkat satu                            |
| &#186; | &ordm;   | º | Indikator ordinal maskulin              |
| &#187; | &raquo;  | » | Panah kanan ganda                       |

|        |          |   |                           |
|--------|----------|---|---------------------------|
| &#188; | &frac14; | ¼ | Satu per empat            |
| &#189; | &frac12; | ½ | Satu per dua,<br>setengah |
| &#190; | &frac34; | ¾ | Tiga per empat            |
| &#191; | &iquest; | ¿ | Tanda tanya terbalik      |
| &#192; | &Agrave; | À |                           |
| &#193; | &Aacute; | Á |                           |
| &#194; | &Acirc;  | Â |                           |
| &#195; | &Atilde; | Ã |                           |
| &#196; | &Auml;   | Ä |                           |
| &#197; | &Aring;  | Å |                           |
| &#198; | &AElig;  | Æ |                           |
| &#199; | &Ccedil; | Ç |                           |
| &#200; | &Egrave; | È |                           |
| &#201; | &Eacute; | É |                           |
| &#202; | &Ecirc;  | Ê |                           |
| &#203; | &Euml;   | Ë |                           |
| &#204; | &lgrave; | Ì |                           |
| &#205; | &Iacute; | Í |                           |

|        |          |   |                  |
|--------|----------|---|------------------|
| &#206; | &Icirc;  | î |                  |
| &#207; | &luml;   | ï |                  |
| &#208; | &ETH;    | Ð |                  |
| &#209; | &Ntilde; | Ñ |                  |
| &#210; | &Ograve; | Ò |                  |
| &#211; | &Oacute; | Ó |                  |
| &#212; | &Ocirc;  | Ô |                  |
| &#213; | &Otilde; | Õ |                  |
| &#214; | &Ouml;   | Ö |                  |
| &#215; | &times;  | × | Simbol perkalian |
| &#216; | &Oslash; | Ø |                  |
| &#217; | &Ugrave; | Ù |                  |
| &#218; | &Uacute; | Ú |                  |
| &#219; | &Ucirc;  | Û |                  |
| &#220; | &Uuml;   | Ü |                  |
| &#221; | &Yacute; | Ý |                  |
| &#222; | &THORN;  | Þ |                  |
| &#223; | &szlig;  | ß |                  |

|        |          |   |  |
|--------|----------|---|--|
| &#224; | &agrave; | à |  |
| &#225; | &aacute; | á |  |
| &#226; | &acirc;  | â |  |
| &#227; | &atilde; | ã |  |
| &#228; | &auml;   | ä |  |
| &#229; | &aring;  | å |  |
| &#230; | &aelig;  | æ |  |
| &#231; | &ccedil; | ç |  |
| &#232; | &egrave; | è |  |
| &#233; | &eacute; | é |  |
| &#234; | &ecirc;  | ê |  |
| &#235; | &euml;   | ë |  |
| &#236; | &igrave; | ì |  |
| &#237; | &iacute; | í |  |
| &#238; | &icirc;  | î |  |
| &#239; | &iuml;   | ï |  |
| &#240; | &eth;    | ð |  |
| &#241; | &ntilde; | ñ |  |

|        |          |   |                  |
|--------|----------|---|------------------|
| &#242; | &ograve; | ò |                  |
| &#243; | &oacute; | ó |                  |
| &#244; | &ocirc;  | ô |                  |
| &#245; | &otilde; | õ |                  |
| &#246; | &ouml;   | ö |                  |
| &#247; | &divide; | ÷ | Simbol pembagian |
| &#248; | &oslash; | ø |                  |
| &#249; | &ugrave; | ù |                  |
| &#250; | &uacute; | ú |                  |
| &#251; | &ucirc;  | û |                  |
| &#252; | &uuml;   | ü |                  |
| &#253; | &yacute; | ý |                  |
| &#254; | &thorn;  | þ |                  |
| &#255; | &yuml;   | ÿ |                  |
| &#338; | &OElig;  | Œ |                  |
| &#339; | &oelig;  | œ |                  |
| &#352; | &Scaron; | Š |                  |
| &#353; | &scaron; | š |                  |

|        |           |   |         |
|--------|-----------|---|---------|
| &#376; | &Yuml;    | ÿ |         |
| &#402; | &fnof;    | f |         |
| &#710; | &circ;    | ^ |         |
| &#732; | &tilde;   | ~ |         |
| &#913; | &Alpha;   | A | Alpha   |
| &#914; | &Beta;    | B | Beta    |
| &#915; | &Gamma;   | Γ | Gamma   |
| &#916; | &Delta;   | Δ | Delta   |
| &#917; | &Epsilon; | E | Epsilon |
| &#918; | &Zeta;    | Z | Zeta    |
| &#919; | &Eta;     | H | Eta     |
| &#920; | &Theta;   | Θ | Theta   |
| &#921; | &Iota;    | I | Iota    |
| &#922; | &Kappa;   | K | Kappa   |
| &#923; | &Lambda;  | Λ | Lambda  |
| &#924; | &Mu;      | M | Mu      |
| &#925; | &Nu;      | N | Nu      |
| &#926; | &Xi;      | Ξ | Xi      |

|        |           |   |         |
|--------|-----------|---|---------|
| &#927; | &Omicron; | O | Omicron |
| &#928; | &Pi;      | Π | Pi      |
| &#929; | &Rho;     | Ρ | Rho     |
| &#931; | &Sigma;   | Σ | Sigma   |
| &#932; | &Tau;     | Τ | Tau     |
| &#933; | &Upsilon; | Υ | Upsilon |
| &#934; | &Phi;     | Φ | Phi     |
| &#935; | &Chi;     | Χ | Chi     |
| &#936; | &Psi;     | Ψ | Psi     |
| &#937; | &Omega;   | Ω | Omega   |
| &#945; | &alpha;   | α | alpha   |
| &#946; | &beta;    | β | beta    |
| &#947; | &gamma;   | γ | gamma   |
| &#948; | &delta;   | δ | delta   |
| &#949; | &epsilon; | ε | epsilon |
| &#950; | &zeta;    | ζ | zeta    |
| &#951; | &eta;     | η | eta     |
| &#952; | &theta;   | θ | theta   |

|        |           |   |              |
|--------|-----------|---|--------------|
| &#953; | &iota;    | ι | iota         |
| &#954; | &kappa;   | κ | kappa        |
| &#955; | &lambda;  | λ | lambda       |
| &#956; | &mu;      | μ | mu           |
| &#957; | &nu;      | ν | nu           |
| &#958; | &xi;      | ξ | xi           |
| &#959; | &omicron; | ο | omicron      |
| &#960; | &pi;      | π | pi           |
| &#961; | &rho;     | ρ | rho          |
| &#962; | &sigmaf;  | ς | sigmaf       |
| &#963; | &sigma;   | σ | sigma        |
| &#964; | &tau;     | τ | tau          |
| &#965; | &upsilon; | υ | upsilon      |
| &#966; | &phi;     | φ | phi          |
| &#967; | &chi;     | χ | chi          |
| &#968; | &psi;     | ψ | psi          |
| &#969; | &omega;   | ω | omega        |
| &#977; | &thetasym | ϑ | Simbol theta |

|         |          |   |                              |
|---------|----------|---|------------------------------|
|         | ;        |   |                              |
| &#978;  | &upsih;  | Υ | Simbol epsilon               |
| &#982;  | &piv;    | ϖ | Simbol pi                    |
| &#8194; | &ensp;   |   | Spasi en                     |
| &#8195; | &emsp;   |   | Spasi em                     |
| &#8201; | &thinsp; |   | Spasi tipis                  |
| &#8204; | &zwnj;   |   | <i>Zero width non-joiner</i> |
| &#8205; | &zwj;    |   | <i>Zero width joiner</i>     |
| &#8206; | &lrm;    |   | Tanda kiri ke kanan          |
| &#8207; | &rlm;    |   | Tanda kanan ke kiri          |
| &#8211; | &ndash;  | – | Garis en                     |
| &#8212; | &mdash;  | — | Garis em                     |
| &#8216; | &lsquo;  | ‘ | Kutip kiri tunggal           |
| &#8217; | &rsquo;  | ’ | Kutip kanan tunggal          |
| &#8218; | &sbquo;  | , | Kutip rendah tunggal         |
| &#8220; | &ldquo;  | “ | Kutip kiri ganda             |
| &#8221; | &rdquo;  | ” | Kutip kanan ganda            |
| &#8222; | &bdquo;  | „ | Kutip rendah ganda           |

|         |          |     |                                      |
|---------|----------|-----|--------------------------------------|
| &#8224; | &dagger; | †   | Salib                                |
| &#8225; | &Dagger; | ‡   | Salib ganda                          |
| &#8226; | &bull;   | •   | Peluru                               |
| &#8230; | &hellip; | ... | Ellipsis horisontal                  |
| &#8240; | &permil; | ‰   | Per mil                              |
| &#8242; | &prime;  | '   | Menit                                |
| &#8243; | &Prime;  | ''  | Detik                                |
| &#8249; | &lsaquo; | ‹   | Panah kiri tunggal                   |
| &#8250; | &rsaquo; | ›   | Panah kanan tunggal                  |
| &#8254; | &oline;  | —   | Garis atas                           |
| &#8364; | &euro;   | €   | Simbol Euro                          |
| &#8482; | &trade;  | ™   | Merek dagang                         |
| &#8592; | &larr;   | ←   | Panah kiri                           |
| &#8593; | &uarr;   | ↑   | Panah atas                           |
| &#8594; | &rarr;   | →   | Panah kanan                          |
| &#8595; | &darr;   | ↓   | Panah bawah                          |
| &#8596; | &harr;   | ↔   | Panah kiri kanan                     |
| &#8629; | &crarr;  | ↵   | Simbol Enter/ <i>Carriage return</i> |

|         |          |                |  |
|---------|----------|----------------|--|
| &#8704; | &forall; | $\forall$      |  |
| &#8706; | &part;   | $\partial$     |  |
| &#8707; | &exist;  | $\exists$      |  |
| &#8709; | &empty;  | $\emptyset$    |  |
| &#8711; | &nabla;  | $\nabla$       |  |
| &#8712; | &isin;   | $\in$          |  |
| &#8713; | &notin;  | $\notin$       |  |
| &#8715; | &ni;     | $\ni$          |  |
| &#8719; | &prod;   | $\prod$        |  |
| &#8721; | &sum;    | $\sum$         |  |
| &#8722; | &minus;  | $-$            |  |
| &#8727; | &lowast; | $*$            |  |
| &#8730; | &radic;  | $\sqrt{\quad}$ |  |
| &#8733; | &prop;   | $\propto$      |  |
| &#8734; | &infin;  | $\infty$       |  |
| &#8736; | &ang;    | $\angle$       |  |
| &#8743; | &and;    | $\wedge$       |  |
| &#8744; | &or;     | $\vee$         |  |

|         |          |               |  |
|---------|----------|---------------|--|
| &#8745; | &cap;    | $\cap$        |  |
| &#8746; | &cup;    | $\cup$        |  |
| &#8747; | &int;    | $\int$        |  |
| &#8756; | &there4; | $\therefore$  |  |
| &#8764; | &sim;    | $\sim$        |  |
| &#8773; | &cong;   | $\cong$       |  |
| &#8776; | &asymp;  | $\approx$     |  |
| &#8800; | &neq;    | $\neq$        |  |
| &#8801; | &equiv;  | $\equiv$      |  |
| &#8804; | &leq;    | $\leq$        |  |
| &#8805; | &geq;    | $\geq$        |  |
| &#8834; | &sub;    | $\subset$     |  |
| &#8835; | &sup;    | $\supset$     |  |
| &#8836; | &nsup;   | $\not\supset$ |  |
| &#8838; | &sube;   | $\subseteq$   |  |
| &#8839; | &supe;   | $\supseteq$   |  |
| &#8853; | &oplus;  | $\oplus$      |  |
| &#8855; | &otimes; | $\otimes$     |  |

|         |          |   |  |
|---------|----------|---|--|
| &#8869; | &perp;   | ⊥ |  |
| &#8901; | &sdot;   | · |  |
| &#8968; | &lceil;  | ⌈ |  |
| &#8969; | &rceil;  | ⌋ |  |
| &#8970; | &lfloor; | ⌊ |  |
| &#8971; | &rfloor; | ⌋ |  |
| &#9674; | &loz;    | ◊ |  |
| &#9824; | &spades; | ♠ |  |
| &#9827; | &clubs;  | ♣ |  |
| &#9829; | &hearts; | ♥ |  |
| &#9830; | &diams;  | ♦ |  |

Contoh penerapan dalam skrip HTML :

```

<!DOCTYPE html><html>
<head><title>Simbol Di HTML</title>
</head>
<body>
<h3>Contoh Simbol di HTML</h3>
Umsida &#8482;<br>
Umsida &reg;<br>

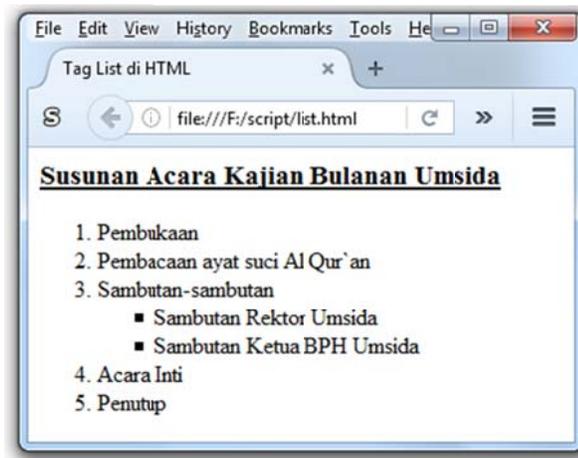
```



</li>...</li> yang digunakan untuk membuat list item pada setiap baris list. Contoh penerapan list di skrip HTML adalah sebagai berikut :

```
<!DOCTYPE html>
<html>
<head><title>Tag List di HTML</title>
</head>
<body>
<h3><u>Susunan Acara Kajian Bulanan
Umsida</u></h3>
<ol type="a">
<li>Pembukaan</li>
<li>Pembacaan ayat suci Al Qur`an</li>
<li>Sambutan-sambutan
<ul type="square">
<li>Sambutan Rektor Umsida</li>
<li>Sambutan Ketua BPH Umsida</li>
</ul>
</li>
<li>Acara Inti</li>
<li>Penutup</li>
</ol>
</body>
</html>
```

Hasil tampilan apabila skrip dijalankan di browser adalah sebagai berikut :

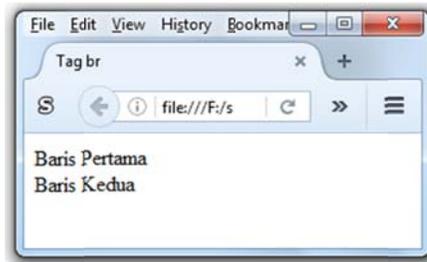


## 2.7. Pindah Baris di HTML

Dalam HTML tag untuk membuat baris baru atau pindah baris menggunakan break `<br>`. Tag `<br>` tidak mempunyai tag penutup, sehingga tak tersebut tidak sesuai dengan ketentuan salah satu aturan dasar penulisan HTML versi baru (XHTML yang berbasis XML) yang mempunyai ketentuan semua tag HTML harus di tutup. Maka penulisan tag penutup untuk pembuatan baris baru adalah `<br/>`.

```
<!DOCTYPE html>
<html>
<head><title>Tag br</title>
</head>
<body>
Baris Pertama<br>
Baris Kedua <br/>
</body>
</html>
```

Hasil tampilan skrip di browser sebagai berikut :



## 2.8. Garis Horizontal di HTML

Dalam pembuatan web terutama untuk membuat tampilan web yang menarik di tambahkan garis horizontal. Tag untuk garis horizontal adalah `<hr>`. Dimana tag ini mempunyai atribut sebagai berikut :

- a. Size digunakan untuk menentukan ketebalan garis.
- b. Width untuk menentukan lebar garis horizontal.
- c. Align digunakan untuk menentukan posisi atau letak garis.
- d. Noshade digunakan untuk mengatur agar baris tidak mempunyai bayangan (*shade*).

Penerapan tag `<hr>` dalam srip HTML adalah sebagai berikut :

```
<!DOCTYPE html>
<html>
<head><title>Tag HR Di HTML</title>
</head>
<body>
<h3>Universitas Muhammadiyah Sidoarjo</h3>
<HR align="left" width="65%" size="10"
noshade>
Jl. Majapahit 666 B Telp. 031-8944454 <br>
Sidoarjo - Jawa Timur <br>
Indonesia
</body>
</html>
```

Hasil tampilan di browser :



## 2.9. Gambar di HTML

Website yang di bangun dapat ditambahkan gambar sehingga meningkatkan desain dan tampilan halaman web. Dalam HTML, gambar didefinisikan dengan tag <img>. Tag <img> hanya berisi atribut yaitu :

- a. Atribut src untuk menentukan URL (alamat web) gambar dan tidak memiliki tag penutup. Sintaks gambar sebagai berikut :

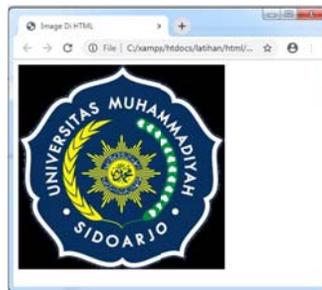
```

```

Penerapan tag <img> dalam skrip HTML :

```
<!DOCTYPE html><html>  
<head><title>Image Di HTML</title>  
</head>  
<body>  
  
</body>  
</html>
```

Hasil tampilan jika di jalankan di browser adalah sebagai berikut:

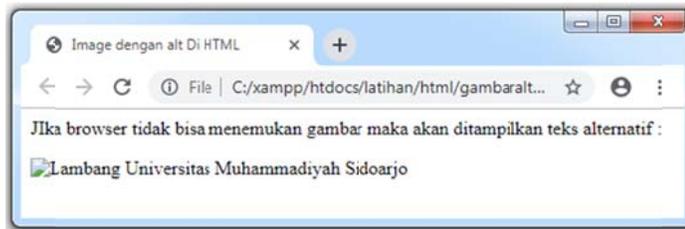


- b. Atribut alt menyediakan teks alternatif untuk gambar apabila karena alasan tertentu gambar tidak bisa tampil misalkan karena koneksi internet lambat, kesalahan dalam atribut src atau jika pengguna menggunakan screen rider. Contoh penerapan atribut alt di skrip HTML :

```
<!DOCTYPE html>
<html>
<head><title>Image dengan alt Di
HTML</title></head>
<body>
<p>Jika browser tidak bisa menemukan gambar
maka akan ditampilkan teks alternatif :</p>

</body>
</html>
```

Hasil tampilan jika di jalankan di browser adalah sebagai berikut :



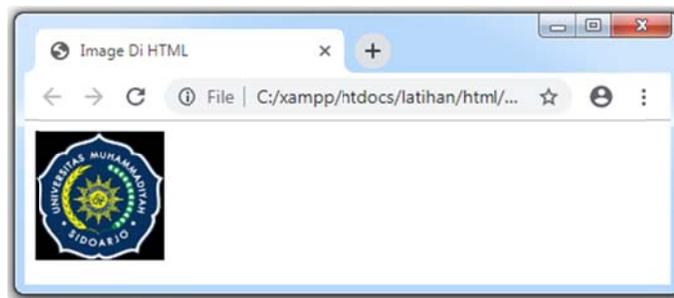
- c. Atribut width dan height digunakan untuk mengubah ukuran gambar (dalam pixel) yang akan ditampilkan di web. Tag width untuk mengatur lebar gambar sedangkan height digunakan

untuk mengatur tinggi gambar. Contoh penerapan dalam skrip HTML :

```
<!DOCTYPE html>
<html>
<head><title>Image Di HTML</title>
</head>
<body>

</body>
</html>
```

Hasil skrip jika di jalankan di browser adalah sebagai berikut :



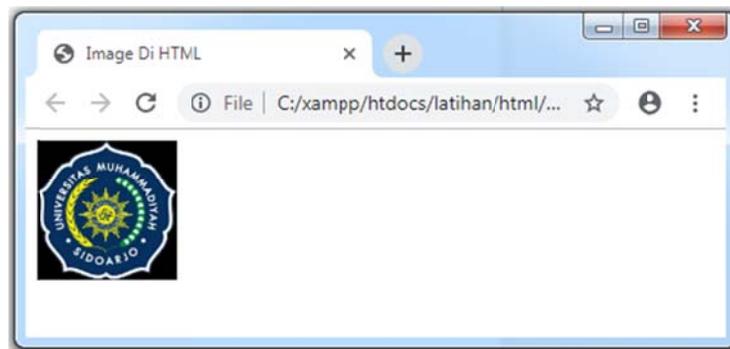
d. Memanggil gambar di folder lain

Agar pemeliharaan website terutama skrip yang dibuat mudah, maka usahakan gambar di simpan dalam folder tersendiri atau dalam sub folder. Untuk memanggil gambar dalam sub folder harus di masukkan nama folder tersebut ke dalam atribut src yang di contohkan sebagai berikut :

```
<!DOCTYPE html>
<html>
<head><title>Image Di HTML</title>
</head>
<body>

</body>
</html>
```

Hasil tampilan skrip jika dijalankan di browser adalah sebagai berikut :



- e. Memanggil gambar dari server lain  
Di HTML juga memungkinkan untuk mengakses gambar dari server lain. Contohnya sebagai berikut :

```
<!DOCTYPE html>
<html>
<head><title>Image Di HTML</title>
</head>
<body>
<h2>Gambar dari Server Lain</h2>

</body>
</html>
```

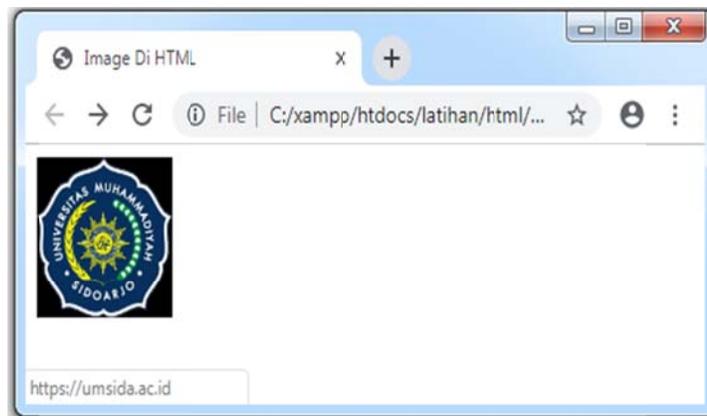
Hasil tampilan skrip jika di jalankan di browser adalah sebagai berikut :



- f. Gambar sebagai tautan (link)  
Untuk menggunakan gambar sebagai tautan (*link*) dengan memasukkan tag <img> di dalam tag <a>:

```
<!DOCTYPE html>
<html>
<head><title>Image Di HTML</title>
</head>
<body>
<a href="https://umsida.ac.id"></a>
</body>
</html>
```

Hasil tampilan skrip jika dijalankan di browser adalah sebagai berikut :



- g. Gambar sebagai Background  
Gambar juga dapat digunakan sebagai latar belakang (background) website dengan memasukkan ke elemen body di skrip HTML. Seperti contoh berikut ini :

```
<!DOCTYPE html>
<html>
<head><title>Image Di HTML</title>
</head>
<body background="umsida_logo.jpg">
</body>
</html>
```

Hasil tampilan skrip jika di jalankan di browser adalah sebagai berikut :



## 2.10. Tabel di HTML

Table merupakan elemen yang mewakili data tabular, yaitu informasi yang disajikan dalam tabel dua dimensi yang terdiri dari baris dan kolom sel yang berisi data. Tabel HTML didefinisikan dengan tag <table>. Setiap baris tabel ditentukan dengan tag <tr>. Sedangkan header tabel didefinisikan dengan tag <th> dan data tabel (sel) didefinisikan dengan tag <td> yang dapat berisi segala

macam elemen HTML; teks, gambar, daftar, tabel lain, dan lain sebagainya. Sintaks dari table di HTML adalah sebagai berikut :

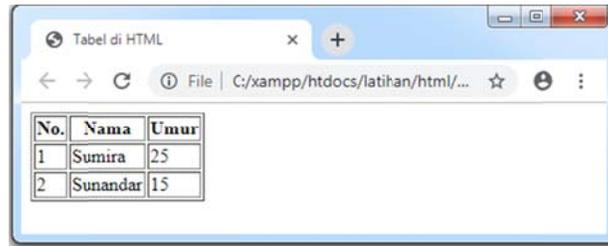
```
<!DOCTYPE html>
<html>
<head><title>Tabel di HTML</title></head>
<body>
<table>
<tr><th>..</th></tr>
<tr><td>..</td></tr>
</table></body>
</html>
```

Contoh penerapan table dalam skrip HTML :

```
<!DOCTYPE html>
<html>
<head><title>Tabel di HTML</title>
</head>
<body>
<table border="1">
<tr>
<th>No.</th>
<th>Nama</th>
<th>Umur</th>
</tr>
<tr>
<td>1</td>
<td>Sumira</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>Sunandar</td>
<td>15</td>
</tr>
```

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

Hasil tampilan skrip apabila di jalankan pada browser sebagai berikut :



Atribut yang di miliki oleh tabel adalah sebagai berikut :

- a. Atribut Align digunakan untuk mengatur posisi tabel. Ada tiga nilai (*value*) align yaitu left untuk tabel ditampilkan di sisi kiri dokumen, center untuk posisi tengah dokumen dan right untuk posisi sebelah kanan dokumen. Sintaks untuk cellpadding sebagai berikut :

```
<table align="nilai">
```

- b. Atribut cellpadding digunakan untuk untuk pengaturan teks terhadap tepi kiri tabel. Jika tabel tidak ditentukan paddingnya, maka sel-sel tabel akan ditampilkan tanpa padding. Sintaks untuk cellpadding sebagai berikut :

```
<table cellpadding="nilai">
```

Dimana nilai atau value yang dimasukkan berupa angka numerik (bilangan bulat).

c. Atribut cellpadding

untuk pengaturan jarak bagian sel terhadap tepi dalam bingkai tabel. Jika tabel tidak ditentukan spacingnya, maka sel-sel tabel akan ditampilkan tanpa spacing. Sintaks untuk cellpadding sebagai berikut :

```
<table cellpadding="nilai">
```

Sama dengan cellpadding nilai atau value yang dimasukkan adalah berupa angka numerik (bilangan bulat).

d. Atribut bgcolor digunakan menentukan warna latar belakang tabel. Tetapi atribut bgcolor pada <table> tidak didukung dalam HTML5 sehingga harus menggunakan CSS. Sintaks untuk atribut ini adalah sebagai berikut :

```
<table bgcolor="nama_warna|warna_hexadecimal">
```

Dimana :

- Nama\_warna adalah menentukan warna latar belakang dengan nama warna (seperti "red").
  - Warna\_hexadecimal adalah menentukan warna latar belakang dengan kode hex (seperti "# ff0000")
- e. Atribut Height dan Width digunakan untuk mengatur tinggi dan lebar tabel. Nilai (*value*) yang di pakai bisa numerik atau persentase (%). Sintaks untuk atribut ini adalah sebagai berikut :

```
<table height="value" width="value">
```

- f. Atribut Rowspan dan Colspan digunakan untuk menggabungkan beberapa sel dalam tabel. Rowspan digunakan untuk menggabungkan beberapa baris sel sedangkan colspan digunakan untuk menggabungkan beberapa kolom dalam satu baris. Sintaks untuk atribut ini adalah sebagai berikut :

```
<table>
<tr>
<td rowspan="nilai"></td>
<td colspan="nilai"></td>
</tr>
</table>
```

Berikut ini contoh penerapan <table> dalam skrip HTML :

```
<!DOCTYPE html>
<html>
<head><title>Atribut Tabel dalam HTML</title>
</head>
<body>
<h3 align="left">Tabel Hobi Mahasiswa</h3>
<table border="1" cellspacing="2"
cellpadding="2" bgcolor="Blue sea"
align="left" height="1" width="25%">
<tr>
<th>No</th>
<th>Nama </th>
<th>Hobi</th>
</tr>
<tr>
<td rowspan="2">1</td>
```

```

<td>Sepak Bola</td>
</tr>
<tr>
<td>Panahan</td>
</tr><tr>
<td rowspan="2">2</td>
<td rowspan="2">Sakura</td>
<td>Renang</td>
</tr>
<tr>
<td>Baca Buku</td>
</tr>
</table>
</body>
</html>

```

Hasil tampilan skrip jika dijalankan di browser adalah sebagai berikut :

The screenshot shows a web browser window with the title 'Atribut Tabel dalam HTML'. The address bar shows the file path 'C:/xampp/htdocs/latihan/htm'. The main content area displays a table with the following structure:

| No | Nama   | Hobi       |
|----|--------|------------|
| 1  | Romano | Sepak Bola |
|    |        | Panahan    |
| 2  | Sakura | Renang     |
|    |        | Baca Buku  |

## 2.11. Tautan (*HyperLink*) di HTML

*Link* atau *Hyperlink* ditemukan di hampir semua halaman web. *Hyperlink* pada web digunakan untuk menghubungkan web dari halaman satu ke halaman lainnya baik dalam website itu sendiri atau website lainnya. Tag untuk link dalam HTML adalah sebagai berikut :

```
<a href="URL">Label Hyperlink</a>
```

Dimana :

- URL (*Uniform Resource Locator*) dapat berupa alamat suatu dokumen web, file grafik, atau menyatakan suatu protokol lain (FTP).
- Sedangkan Label Hyperlink merupakan bagian yang terlihat yang akan di pilih atau di klik oleh pengguna web.

Selain itu URL ada dua jenis yaitu :

- URL Relatif adalah URL yang menyebutkan nama-nama berkas secara langsung. Contohnya sebagai berikut :  

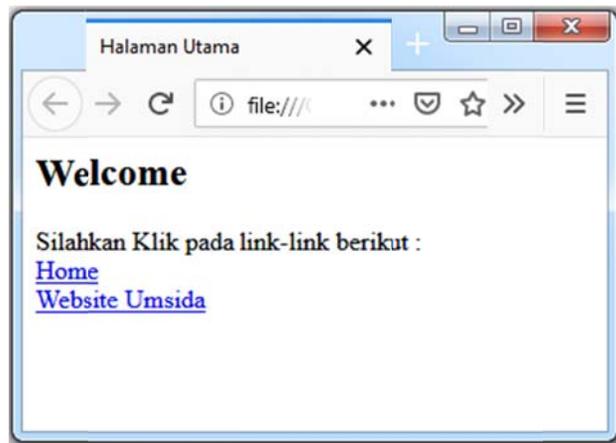
```
<a href="home.html">Home</a>
```
- URL Absolut memiliki ciri – ciri diawali dengan : http://, ftp://. Kelebihan url absolut adalah dalam hal pembuatannya yang sangat cepat. Dapat dicontohkan sebagai berikut :  

```
<a href="https://umsida.ac.id">Website Umsida</a>
```

Contoh penerapan hyperlink dalam HTML adalah sebagai berikut :

```
<!DOCTYPE html>
<html>
<head><title>Halaman Utama</title>
</head>
<body>
<h2>Halaman Utama</h2>
Silahkan Klik pada link-link berikut
:<br>
<a href="home.html">Home</a><br>
<a href="https://umsida.ac.id">Website
Umsida</a><br>
</body>
</html>
```

Hasil tampilan skrip jika di jalankan di browser adalah sebagai berikut :



Link mempunyai beberapa atribut yaitu :

a. Link Color

Label yang menjadi penghubung ke halaman web yang lain secara bawaan diberi warna tertentu :

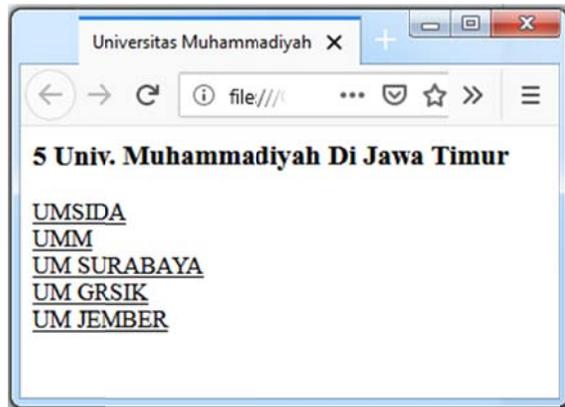
- Link yang belum diklik pemakai berwarna biru dan bergaris bawah.
- Link yang sudah diklik pemakai berwarna ungu dan bergaris bawah
- Link aktif berwarna merah dan bergaris bawah

Warna tersebut dapat diubah dengan menggunakan atribut LINK, ALINK dan VLINK yang terdapat pada tag <BODY> dimana ALINK digunakan untuk Untuk menentukan link aktif, yaitu link yang telah diklik pemakai tapi halaman belum selesai dibuka sedangkan VLINK Untuk menentukan warna link yang halaman Web terkaitnya telah dikunjungi seperti contoh berikut ini :

```
<!DOCTYPE html>
<html><head><title>Universitas
Muhammadiyah</title></head>
<body link="black" alink="green" vlink="pink">
<h3>5 Univ. Muhammadiyah Di Jawa Timur</h3>
<a href="https://umsida.ac.id/">UMSIDA</a><br>
<a href="http://www.umm.ac.id/">UMM</a><br>
<a href="http://www.um-surabaya.ac.id/">UM
SURABAYA</a><br>
<a href="http://umg.ac.id/">UM GRESIK</a><br>
<a
href="http://unmuhjember.ac.id/index.php/id/">
UM JEMBER</a>
</body>
</html>
```

Hasil tampilan jika dijalankan di browser sebagai berikut :

- Hasil tampilan sebelum link di klik



- Hasil tampilan sesudah link di klik



- b. Atribut link target berfungsi untuk menentukan tempat untuk membuka dokumen yang ditautkan. Atribut target mempunyai nilai (*value*) antara lain :
- `_blank` berfungsi untuk membuka dokumen yang ditautkan di jendela atau tab baru.

- `_self` berfungsi Buka dokumen yang ditautkan di jendela / tab yang sama dengan yang diklik (hakaman default).

Contoh penerapan atribut target di link dalam HTML :

```
<!DOCTYPE html>
<html>
<head><title>Universitas
Muhammadiyah</title>
</head>
<body link="black" alink="green"
vlink="pink">
<h3>2 Univ. Muhammadiyah Di Jawa
Timur</h3>
<a href="http://www.umm.ac.id/"
target="_blank">UMM</a><br>
<a href="http://www.um-surabaya.ac.id/"
target="_self">UM SURABAYA</a><br>
</body>
</html>
```

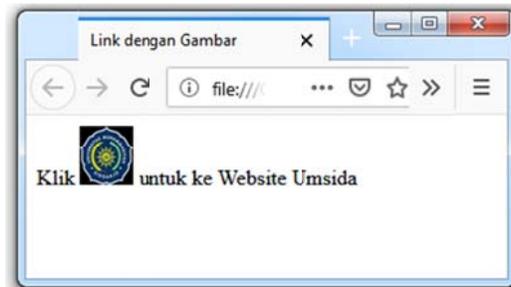
Hasil tampilan jika dijalankan di browser sebagai berikut :



- c. Sebuah gambar dapat dijadikan link ke halaman lain, dengan menyebutkan tag <img> diantara pasangan <a>...</a> yang di contohkan sebagai berikut :

```
<!DOCTYPE html>
<html>
<head><title>Link dengan Gambar</title>
</head>
<body>
Klik <a href="https:\\umsida.ac.id"></a>
untuk ke Website Umsida
</body>
</html>
```

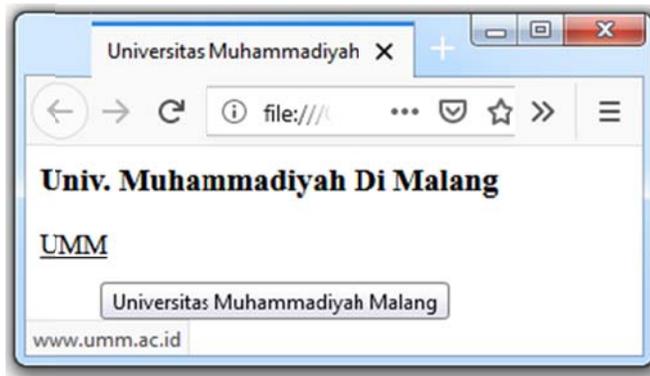
Hasil tampilan sebagai berikut :



- d. Atribut title digunakan untuk menentukan informasi tambahan tentang suatu elemen. Informasi ini paling sering ditampilkan sebagai teks tooltip ketika mouse bergerak di atas elemen seperti conoth berikut ini :

```
<!DOCTYPE html>
<html>
<head><title>Universitas
Muhammadiyah</title>
</head>
<body link="black" alink="green"
vlink="pink">
<h3>Univ. Muhammadiyah Di Malang</h3>
<a href="http://www.umm.ac.id/"
target="_blank" title="Universitas
Muhammadiyah Malang">UMM</a><br>
</body>
</html>
```

Hasil tampilan skrip apabila dijalankan di browser adalah sebagai berikut :



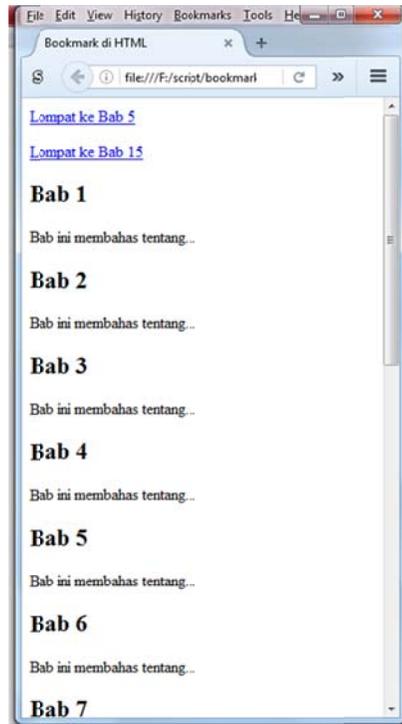
- e. Link Bookmark merupakan link yang mengacu pada sejumlah link di halaman itu sendiri. Bookmark HTML digunakan untuk memungkinkan pembaca melompat ke bagian tertentu dari

halaman Web. Bookmark dapat bermanfaat jika halaman web sangat panjang. Untuk membuat bookmark, harus terlebih dahulu membuat bookmark, kemudian menambahkan link di bookmark tersebut. Ketika link diklik, halaman tersebut akan menuju ke bagian bookmark tersebut. Atribut link yang digunakan di bookmark adalah id. Penerapan bookmark bisa dilihat pada contoh berikut ini :

```
<!DOCTYPE html>
<html>
<head><title>Bookmark di HTML</title>
</head>
<body>
<p><a href="#B5">Lompat ke Bab 5</a></p>
<p><a href="#B15">Lompat ke Bab 15</a></p>
<h2>Bab 1</h2>
<p>Bab ini membahas tentang...</p>
<h2>Bab 2</h2>
<p>Bab ini membahas tentang...</p>
<h2>Bab 3</h2>
<p>Bab ini membahas tentang...</p>
<h2>Bab 4</h2>
<p>Bab ini membahas tentang...</p>
<h2 id="B5">Bab 5</h2>
<p>Bab ini membahas tentang...</p>
<h2>Bab 6</h2>
<p>Bab ini membahas tentang...</p>
<h2>Bab 7</h2>
<p>Bab ini membahas tentang...</p>
```

```
<h2>Chapter 8</h2>
<p>Bab ini membahas tentang...</p>
<h2>Chapter 9</h2>
<p>Bab ini membahas tentang...</p>
<h2>Chapter 10</h2>
<p>Bab ini membahas tentang...</p>
<h2>Chapter 11</h2>
<p>Bab ini membahas tentang...</p>
<h2>Chapter 12</h2>
<p>Bab ini membahas tentang...</p>
<h2>Chapter 13</h2>
<p>Bab ini membahas tentang...</p>
<h2 id="B15">Chapter 15</h2>
<p>Bab ini membahas tentang...</p>
</body>
</html>
```

Hasil tampilan dari skrip jika dijalankan di browser adalah sebagai berikut :



## 2.12. Frame di HTML

Frame adalah teknik yang digunakan untuk membagi window menjadi beberapa bagian. Setiap bagian dapat diisi dengan sebuah halaman web yang sesuai. Keuntungan dari penggunaan frame adalah mudahnya pengaturan hubungan antar satu halaman dengan halaman lainnya. Untuk membuat halaman web yang menggunakan frame perlu membuat beberapa file html, yaitu:

- a. Sebuah pendefinisi file. File tersebut digunakan untuk menampilkan halaman web dengan bentuk frame.
- b. Beberapa file pengisi frame, yang jumlahnya tergantung dari berapa bagian yang akan dibuat pada frame tersebut.

- c. Tag yang digunakan `<FRAMESET>.....</FRAMESET>` pada frame ini tidak dipelukan tag `<body>....</body>`, struktur dasar pendefinisi Frame sbb :

```
<!DOCTYPE html>
<html>
<head>
<title></title>
</head>
<frameset>
*** bagian frame ***
</frameset>
</html>
```

- d. Pada file pendefinisi frame ini, setiap akan membagi menjadi beberapa bagian, maka digunakan tag `<FRAMESET></FRAMESET>`. Atribut-atribut yang dapat menyertai tag ini adalah:
- Cols digunakan untuk membuat frame vertikal (kolom).
  - Rows digunakan untuk membuat frame horizontal (baris).
  - Border digunakan untuk Mengatur ketebalan garis pemisah antar Frame.
- e. Jika Frame tidak akan dibagi lagi, maka selanjutnya definisikan frame tersebut dengan tag `<FRAME>..</FRAME>`. Atribut-atribut yang dapat menyertai tag `<frame>` adalah:
- Src digunakan untuk memanggil atau menunjukkan file yang mengisi frame tersebut.
  - Scrolling =“pilihan” digunakan untuk Menggunakan scrolling bar di samping atau di bawah halaman, pilihan inputnya adalah "YES, NO atau AUTO".

- Noresize digunakan agar Frame tidak bisa diubah ketika ditampilkan di browser.
  - Name digunakan untuk memberi identitas/nama bagi frame tersebut. Fungsinya adalah ketika frame tersebut dijadikan target untuk menampilkan suatu halaman.
- f. Yang perlu diperhatikan saat membuat file pendefinisi frame adalah bahwa file-file yang digunakan untuk mengisi Frame telah dibuat.
- g. Struktur Dasar sebagai berikut :

```
<!DOCTYPE html>
<html>
<head>
<title></title>
</head>
<frameset>
<frame src="">
</frame src="">
</frameset>
</html>
```

Berikut contoh penerapan frame kedalam skrip HTML :

- Membuat file header.html untuk diletakkan di header web

```
<!DOCTYPE html>
<html>
<head>
<title></title>
</head>
<body bgcolor="#006699">
<h2><font color="#FFFFFF">Logo dan nama
(header web) Anda</font></h2>
</body>
</html>
```

- Kemudian membuat file untuk isi web yaitu file halaman utama dan profil web.

File halaman utama web welcome.html

```
<!DOCTYPE html>
<html>
<head>
<title></title>
</head>
<body bgcolor="#edf5d2">
<h2>Welcome to My Website</h2>
<h4>Isi Website Anda</h4>
</body>
</html>
```

File profil web simpan dengan nama profil.html

```
<!DOCTYPE html>
<html>
<head>
<title></title>
```

```
</head>
<body bgcolor="#edf5d2">
<h2>Profil Website</h2>
<h4>Profil Website Anda</h4>
</body>
</html>
```

- Membuat juga file menu.html untuk navigasi web

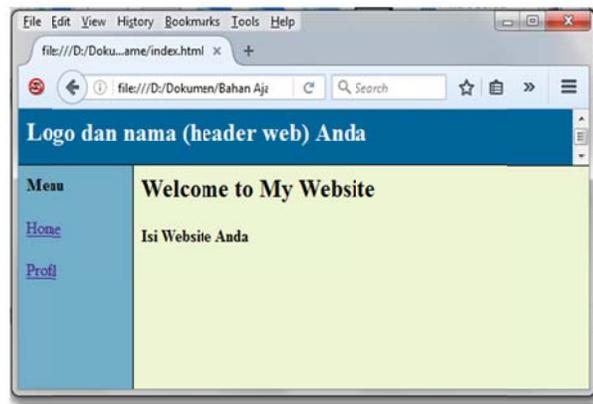
```
<!DOCTYPE html>
<html>
<head>
<title></title>
</head>
<body bgcolor="#71aec7">
<h4>Menu</h4>
<a href="welcome.html"
target="isi">Home</a><br><br>
<a href="profil.html"
target="isi">Profil</a>
</body>
</html>
```

- Membuat halaman index.html yang digunakan untuk menyimpan frame yang dibuat.

```
<!DOCTYPE html>
<html>
<head>
<title></title>
</head>
<frameset rows="20%,*" border="1">
```

```
<frame name="atas" src="header.html"
noresize>
<frameset cols="20%,*">
<frame scrolling="no" src="menu.html"
noresize>
<frame src="welcome.html" name="isi"
noresize>
</frameset>
</frameset>
</html>
```

- Hasil tampilan frame jika dijalankan di browser adalah sebagai berikut :



### 2.13. Tag Div (Division) di HTML

Tag <div> digunakan untuk menentukan divisi atau bagian dalam dokumen HTML. Elemen <div> sering digunakan sebagai tempat untuk elemen HTML lainnya untuk CSS style atau JavaScript. Elemen

<div> juga sering digunakan untuk membuat layout halaman web biasanya terdiri dari :

- Header (kepala website) yang berisi logo dan nama website,
- Content merupakan isi website yang berisi informasi terbaru atau tentang website tersebut.
- Sidebar digunakan untuk menu atau link ke web lain.
- bagian kaki web (footer) yang berisi tentang pembuatan dan tahun pembuatan website tersebut.

Dengan menggunakan tag ini web lebih terstruktur karena terbagi menjadi bagian yang lebih spesifik. Contoh penerapan <div> dalam HTML sebagai berikut :

```
<div id="header">
</div>

<div id="sidebar">
</div>

<div id="content">
</div>

<div id="footer">
</div>
```

Tag div mempunyai dua atribut yaitu :

a. Atribut ID

Atribut ID digunakan untuk penamaan elemen dalam HTML yang memiliki struktur berbeda. Elemen ID yang digunakan dalam satu struktur tidak boleh ada yang sama seperti contoh berikut ini :

```
<div id="sidebar">
  <ul id="menu">
    <li>Beranda</li>
    <li>Profil</li>
  </ul>
</div>
```

b. Atribut Class

Atribut class pada <div> digunakan untuk penamaan elemen yang mempunyai struktur yang sama dan dapat digunakan berulang kali dalam kode HTML. Berikut contoh penerapan class dalam HTML :

```
<ul id="menu">
  <li class="biru">Beranda</li>
  <li>Profil</li>
  <li class="biru">Gallery</li>
  <li>Kontak kami</li>
</ul>
```

## **BAB 3**

### **CASCADING STYLE SHEETS (CSS)**

Bab ini membahas tentang Cascading Style Sheet baik selector, property, declaration dan penerapannya dalam HTML agar tampilan website lebih menarik dan memudahkan dalam pemeliharaan website.

#### **3.1. Pengenalan CSS**

Cascading Style Sheet (CSS) merupakan salah satu bahasa pemrograman web untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. CSS menjelaskan bagaimana elemen HTML akan ditampilkan pada layar, kertas, atau di media lain. CSS digunakan untuk menciptakan suatu kumpulan style yang terkadang digunakan untuk memperluas kemampuan HTML.

#### **3.2. Kelebihan CSS**

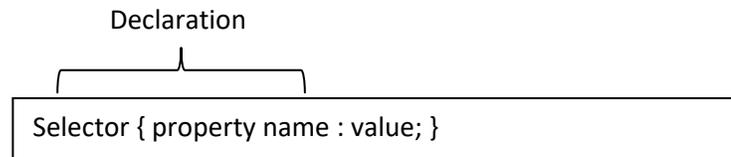
1. CSS menghemat waktu : file CSS dapat ditulis sekali dan menggunakan kembali file yang sama di beberapa halaman HTML.
2. Kemudahan *Maintainence* : untuk membuat perubahan global hanya mengubah style saja, dan semua elemen di semua halaman web akan diperbarui secara otomatis.
3. Mesin telusur : CSS dianggap sebagai teknik pengkodean yang bersih, yang berarti mesin telusur tidak perlu kesulitan dalam "membaca" kontennya.

4. Style unggul untuk HTML : CSS memiliki *array* yang jauh lebih luas dari atribut dari HTML, sehingga tampilan HTML jauh lebih baik dibandingkan dengan atribut HTML.
5. *Offline browsing* : CSS dapat menyimpan aplikasi web secara lokal dengan bantuan *cache offline* sehingga situs website bisa di akses secara *offline*.

### 3.3. Sintaks CSS

Sintaks dasar pada CSS terdiri dari dua bagian, yaitu bagian Selector dan bagian blok Declaration. Di dalam Declaration terdapat Property dan Value yang dapat dijabarkan sebagai berikut :

1. Selector adalah elemen HTML yang akan menjadi target pemformatan CSS.
2. Declaration adalah sintaks CSS yang berisi format-format (styles) yang terdiri atas Property dan Value.
3. Sebuah baris kode CSS bisa terdapat satu atau banyak blok declaration.



Dimana aturan Penulisan CSS adalah sebagai berikut :

1. Selector ditulis pada bagian awal baris kode CSS.
2. Declaration ditulis dalam tanda kurung kurawal ({... }).
3. Penulisan Property dan Value pada Declaration dibatasi dengan tanda titik dua (:).
4. Setiap Declaration diakhiri dengan tanda titik koma (;) sebelum Declaration baru.

5. Penerapan CSS dalam skrip HTML dapat di contohkan sebagai berikut :

```
p {color:blue;text-align:center;}
```

Atau bisa juga sebagai berikut :

```
p {  
color:blue;  
text-align:center;  
}
```

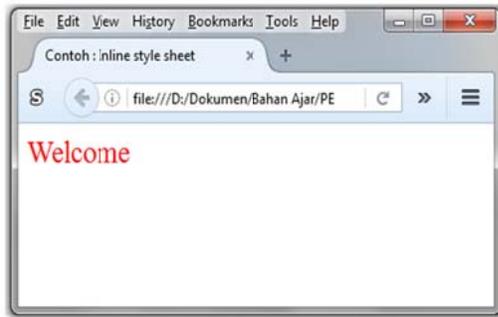
### 3.4. Penempatan CSS di HTML

CSS dapat ditambahkan ke elemen HTML dengan 3 cara:

1. *Inline* yaitu dengan menggunakan atribut style dalam setiap tag atau elemen HTML. Penerapan dalam skrip dapat di contohkan sebagai berikut :

```
<!DOCTYPE html>  
<html>  
<head>  
<title>Contoh : Inline style  
sheet</title>  
</head>  
<body>  
<p style='color:red; font-  
size:25px'>Welcome</p>  
</body>  
</html>
```

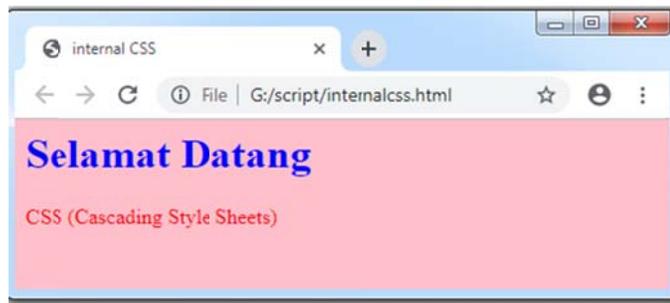
Hasil tampilan skrip apabila dijalankan di browser adalah sebagai berikut :



2. *Internal* atau *Embedded* yaitu dengan menggunakan `<style></style>` yang diletakkan di bagian `<head>` HTML. Penerapan dalam skrip dapat di contohkan sebagai berikut :

```
<!DOCTYPE html>
<html>
<head><title>internal CSS</title>
<style>
body {background-color: pink;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
<body>
<h1>Selamat Datang</h1>
<p>CSS (Cascading Style Sheets)</p>
</body>
</html>
```

Hasil tampilan skrip apabila dijalankan di browser adalah sebagai berikut :



3. *Eksternal* yaitu dengan menggunakan file CSS eksternal yang dipanggil di bagian head HTML. Dengan CSS eksternal, tampilan seluruh situs web dapat dapat diubah hanya dengan mengubah satu file saja. File untuk CSS harus di simpan dengan ekstensi .css. Sintak untuk eksternal CSS adalah sebagai berikut :

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

Penerapan eksternal dalam skrip HTML adalah sebagai berikut :

- File eksternal style.css

```
body {  
  background-color: pink;  
}  
h1 {  
  color: blue;  
}  
p {  
  color: red;  
}
```

- File web yang menggunakan css

```
<!DOCTYPE html>
<html>
<head><title>eksternal CSS</title>
<link rel="stylesheet"
href="style.css">
</head>
<body>
<h1 >Selamat Datang</h1>
<p>CSS (Cascading Style Sheets)</p>
</body>
</html>
```

Hasil tampilan skrip jika dijalankan di browser :



Cara yang paling umum untuk menambahkan CSS, adalah dengan memisahkan antara file CSS dengan file HTML. Namun, style inline dan internal lebih mudah diterapkan di HTML.

### 3.5. Properti Dasar CSS di HTML

CSS juga mempunyai properti dasar antara lain :

1. Properti Teks (Text) yang berfungsi untuk mengatur CSS teks dalam HTML. Berikut ini Properti teks :

| Properti        | Deskripsi                           | Nilai   |
|-----------------|-------------------------------------|---|
| color           | Mengatur warna teks                 | RGB, hex, keyword (nama warna)                          |
| line-height     | Mengatur jarak antar garis          | normal, angka, length (pixel,pt,cm,dll), persentase (%) |
| letter-spacing  | Jarak antar karakter                | normal, length  |
| text-align      | letak atau posisi teks              | left, right, center, justify                            |
| text-decoration | Menambahkan dekorasi ke teks        | none, underline, overline, line-through                 |
| text-indent     | Mengidentifikasi baris teks pertama | length, %   |
| text-transform  | Mengontrol bentuk huruf             | none, capitalize, uppercase, lowercase                  |

2. Properti Daftar (List) yang berfungsi untuk mengatur CSS List dalam HTML. Berikut ini Properti CSS List :

| <b>Properti</b>     | <b>Deskripsi</b>                                  | <b>Nilai</b>  |
|---------------------|---|---|
| list-style          | Mengatur semua properti list dalam satu deklarasi | list-style-type, list-style-position, list-style-image,inherit  |
| list-style-image    | Menentukan gambar sebagai tanda list-item         | URL, none, inherit  |
| list-style-position | Menentukan posisi penempatan penanda item list    | inside, outside, inherit  |
| list-style-type     | Menentukan jenis penanda list-item                | none, disc, circle, square, decimal, decimal-leading-zero, armenian, georgian, lower-alpha, upper-alpha, lower-greek, lower-latin, upper-latin, lower-roman, upper-roman, inherit |

3. Properti Batas (Border) yang berfungsi untuk mengatur CSS border dalam HTM. Berikut ini Properti border CSS:

| <b>Properti</b>     | <b>Deskripsi</b>  | <b>Nilai</b>  |
|---------------------|---|---|
| Border              | Mengatur semua properti border dalam satu deklarasi                   | border-width, border-style, border-color                      |
| border-bottom       | Mengatur semua batas bawah (border-bottom) dalam satu deklarasi       | border-bottom-width, border-bottom-style, border-bottom-color |
| border-bottom-color | Mengatur warna batas bawah  | border-color  |
| border-bottom-style | Mengatur style batas bawah  | border-color  |
| border-bottom-width | Atur lebar batas bawah  | border-width  |
| border-color        | Mengatur warna batas  | color_name, hex_number, rgb_number, transparent, inherit      |
| border-left         | Mengatur semua properti batas kiri (border-left) dalam satu deklarasi | border-left-width, border-left-style, border-left-color       |
| border-left-color   | Mengatur warna batas kiri   | border-color  |
| border-left-style   | Mengatur style batas kiri   | border-style  |

|                    |  |  |
|--------------------|--|--|
| border-left-width  | Atur lebar batas kiri                    | border-width   |
| border-right-color | Mengatur warna batas kanan               | border-color   |
| border-right-style | Mengatur style batas kanan               | border-style   |
| border-right-width | Mengatur lebar batas kanan               | border-width   |
| border-top         | Mengatur batas atas dalam satu deklarasi | none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset, inherit |
| border-top-color   | Mengatur warna batas atas                | border-color   |
| border-top-style   | Mengatur style batas atas                | border-style   |
| border-top-width   | Mengatur lebar batas atas                | thin, medium, thick, length, inherit   |

4. Properti Huruf (Font) yang berfungsi untuk mengatur CSS Font dalam HTM. Berikut ini Properti font dalam CSS :

| Properti | Deskripsi                           | Nilai   |
|----------|-------------------------------------|---|
| Font     | Mengatur huruf dalam satu deklarasi | font-style, font-variant, font-weight, font-size/line-height, |

|              |   |   |
|--------------|---|---|
|              |   | font-family, caption, icon, menu, message-box, small-caption, status-bar, inherit               |
| font-family  | Menentukan kelompok font untuk teks                                   | family-name, generic-family, inherit  |
| font-style   | Menentukan style huruf  | normal, italic, oblique, inherit  |
| font-size    | Menentukan ukuran huruf   | xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, length, %, inherit |
| font-variant | Menentukan apakah teks harus ditampilkan dalam huruf kecil atau tidak | normal, small-caps, inherit   |
| font-weight  | Menentukan ketebalan font   | normal, bold, bolder, lighter, inherit  |

Penerapan properti CSS dalam skrip HTML dapat di contohkan sebagai berikut :

```

<!DOCTYPE html>
<html>
<head><title>CSS Pertamaku</title>
<style type="text/CSS">
<!--
H1 { font-size: 14pt;
      font-style: italic;
      color: green; }

H2 { font-size: 11pt;
      font-style: none;
      color: red; }

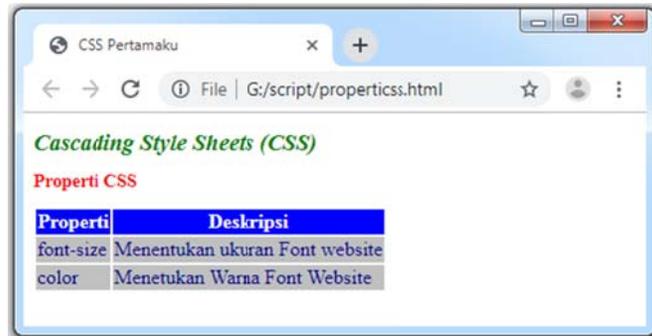
TH { font-size: 12pt;
      font-wight: bold;
      color: white;
      background-color: blue;}

TR { font-size: 12pt;
      font-wight: bold;
      color: darkblue;
      background-color: silver;}
-->
</style>
</head>
<body>
<H1>Cascading Style Sheets (CSS) </H1>
<H2>Properti CSS</H2>
<table>
<th>Properti</th><th>Deskripsi</th>
<tr>
<td>font-size</td><td>Menentukan      ukuran
Font website</td>
</tr>
<tr>

```



Hasil tampilan skrip jika dijalankan dibrowser :



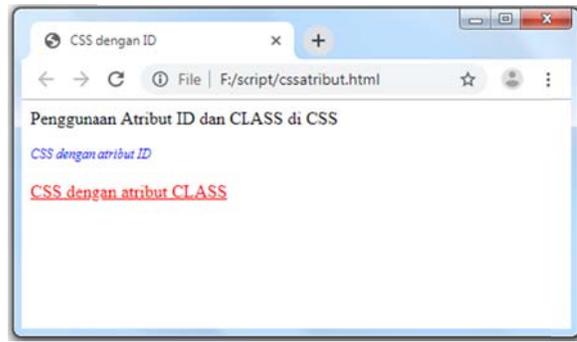
### 3.6. Penerapan Tag Division di CSS

Tag division (div) dalam CSS digunakan untuk memanggil properti CSS dengan menggunakan atribut id dan class. Jika CSS menggunakan atribut id maka diawali dengan tanda # tetapi jika menggunakan atribut class maka CSS diawali dengan tanda titik(.). Berikut ini contoh penerapannya dalam skrip HTML :

```
<!DOCTYPE html>
<html>
<head><title>CSS dengan ID</title>
<style type="text/CSS">
<!--
#miring { font-size: 12px;
font-wight: bold;
font-style: italic;
color: blue; }

.garis_bawah { color: red;
                text-decoration: underline;
}
-->
</style>
</head>
<body>
<p>Penggunaan Atribut ID dan CLASS di CSS</p>
<div id="miring">CSS dengan atribut
ID</div><br>
<div class="garis_bawah">CSS dengan atribut
CLASS</div>
</body>
</html>
```

Hasil tampilan jika skrip dijalankan pada browser adalah sebagai berikut :



## BAB 4

# JAVASCRIPT

Bab ini membahas tentang penggunaan Javascript dalam HTML. Mulai dari Alert box, konfirmasi website, validasi dan print halaman.

### 4.1. Pengenalan Javascript

Javascript adalah skrip yang ditempelkan pada kode HTML dan diproses di sisi klien. Javascript bukan bahasa Pemrograman Java (2 bahasa yang berbeda). Javascript diinterpretasikan oleh klien ( skrip program dapat dilihat pada sisi klien atau pada browser) sedangkan Java dikompilasi oleh pemrogram dan hasil kompilasinya yang dijalankan oleh klien. Contoh penggunaan Javascript pada HTML yaitu untuk memvalidasi masukan-masukan pada formulir sebelum formulir dikirimkan ke server.

### 4.2. Sintaks Penulisan Javascript

Sintak penulisan javascript sebagai berikut :

```
<script>
  <!-- -
                Penulisan kode Javascript
  //-- ->
</script>
```

Dimana :

kode <!-- -> umumnya digunakan jika browser tidak mengenali Javascript maka browser akan memperlakukannya sebagai komentar sehingga tidak ditampilkan pada browser.

### 4.3. Letak Javascript Dalam HTML

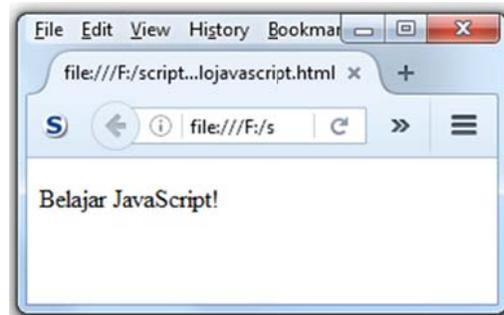
Skript Javascript dalam dokumen HTML dapat diletakkan pada :

1. Bagian Head (<head>....</head>)
2. Bagian body (<body>.....</body>) tetapi jarang digunakan

Untuk memilih elemen HTML, JavaScript paling sering menggunakan metode `document.getElementById()`. Berikut ini contoh skrip penggunaan javascript dalam HTML :

```
<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML =
"Belajar JavaScript!";
</script>
</body>
</html>
```

Hasil skrip apabila dijalankan di browser :



## 4.4. Penerapan Javascript Dalam HTML

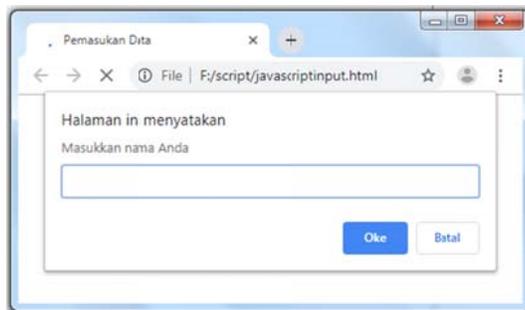
Javascript dapat diterapkan dalam beberapa elemen di HTML antara lain :

### 1. Pemasukan Data

Memasukkan data ke website juga bisa dilakukan dengan menggunakan javascript seperti contoh berikut ini :

```
<!DOCTYPE html>
<html>
<head>
<title>Pemasukan Data</title>
</head>
<body>
<script>
<!--
var nama = prompt ("Masukkan nama Anda ");
document.write("Selamat Datang :" + nama);
//-->
</script>
</body>
</html>
```

Hasil skrip apabila di jalankan di browser adalah sebagai berikut :

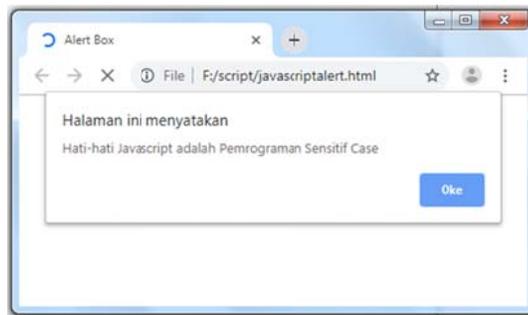


## 2. Alert Box

Javascript juga bisa digunakan untuk memberi peringatan atau himbauan ke pengguna sesuatu yang penting yang harus dilakukan atau tidak boleh dilakukan pada website tersebut.

```
<!DOCTYPE html>
<html>
<head>
<title>Alert Box</title>
</head>
<body>
<script>
<!--
window.alert("Hati-hati Javascript adalah
Pemrograman Sensitif Case");
//-->
</script>
</body>
</html>
```

Hasil skrip apabila di jalankan di browser adalah sebagai berikut :

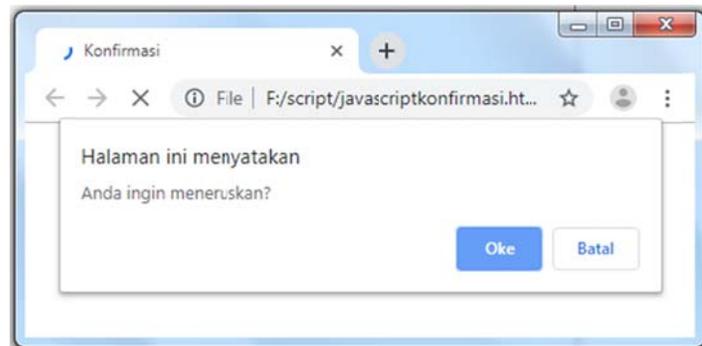


### 3. Konfirmasi

Penggunaan konfirmasi akan sesuai pada website dapat menggunakan javascript sebagai berikut :

```
<!DOCTYPE html>
<html>
<head>
<title>Konfirmasi</title>
</head>
<body>
<script>
<!--
var jawaban = window.confirm ("Anda ingin
meneruskan? ");
document.write("Jawaban Anda :" + jawaban);
//-->
</script>
</body>
</html>
```

Hasil skrip apabila di jalankan di browser adalah sebagai berikut :



#### 4. Validasi form

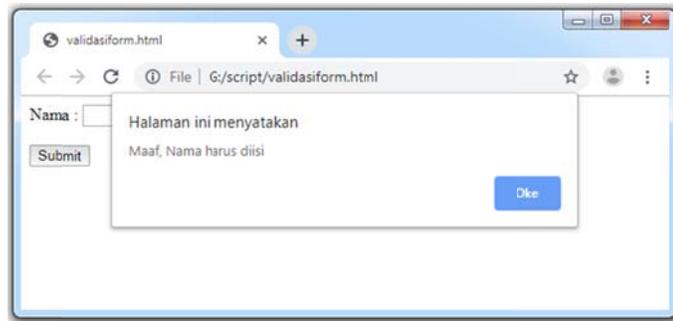
Untuk menjaga data yang dimasukkan pengguna valid maka perlu dilakukan validasi sebelum data disimpan ke website. Validasi data melalui form dapat menggunakan javascript sebagai berikut :

```
<!DOCTYPE html>
<html>
<head>
<script>
function validateForm() {
    var x =
document.forms["myForm"]["fname"].value;
    if (x == "") {
        alert("Maaf, Nama harus diisi");
        return false;
    }
}
</script>
</head>
<body>

<form name="myForm" action="/action_page.php"
onsubmit="return validateForm()"
method="post">
Nama : <input type="text"
name="fname"><br><br>
<input type="submit" value="Submit">
</form>

</body>
</html>
```

Hasil skrip apabila di jalankan di browser adalah sebagai berikut :

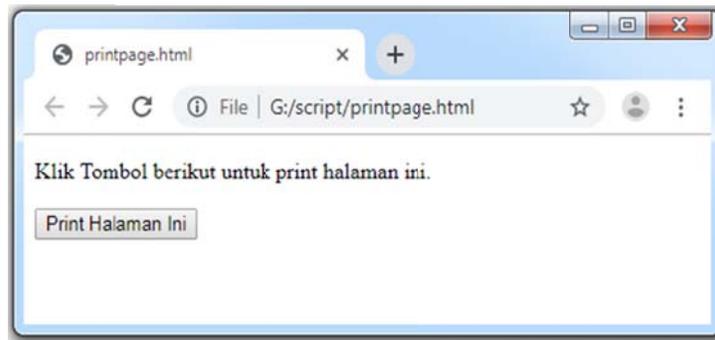


#### 5. Print page

Untuk memudahkan dalam mencetak laporan atau data di website dapat menggunakan javascript sebagai berikut :

```
<!DOCTYPE html>
<html>
<body>
<p>Klik Tombol berikut untuk print Halaman
ini.</p>
<button onclick="myFunction()">Print Halaman
ini</button>
<script>
function myFunction() {
    window.print();
}
</script>
</body>
</html>
```

Hasil skrip apabila di jalankan di browser adalah sebagai berikut :



## **BAB 5**

### **FORMULIR (FORM) DI HTML**

Bab ini membahas tentang formulir untuk memasukkan data pengguna ke website. Mulai dari elemen-elemen untuk memasukkan data, elemen tombol dan atribut aksi dalam formulir.

#### **5.1. Pengenalan Formulir**

Formulir atau form digunakan untuk memasukkan data pada website. Website yang dinamis memungkinkan pengguna untuk bisa berinteraksi salah satunya yaitu bisa memasukkan data dan menampilkan informasi sesuai yang dibutuhkan oleh pengguna dan pemilik website. Sintaks dari form adalah sebagai berikut :

```
<!DOCTYPE html>
<html>
<head><title>..</title>
</head>
<body>
<form>
form elements
</form>
</body>
</html>
```

#### **5.2. Elemen Form**

Ada beberapa jenis elemen form antara lain elemen <label>, elemen <input>, elemen <select>, elemen <textarea>, elemen <button>, elemen <datalist> dan elemen <output>. Elemen form

yang paling penting adalah elemen `<input>`. Elemen `<input>` dapat ditampilkan dalam beberapa cara, tergantung pada tipe atribut `<atribut type>` yang digunakan.

#### 1. Label Form

Label di form digunakan untuk memberikan keterangan pada setiap input yang ada di form. Sintaks untuk label adalah sebagai berikut :

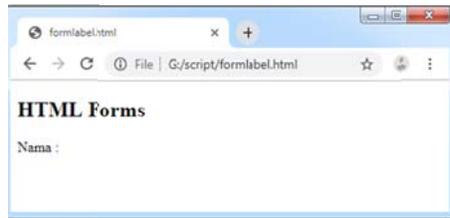
```
<label for="name">Keterangan Input</label>
```

Dimana atribut "for" pada label diisi sama dengan atribut name pada kontrol yang diberi label.

Penerapan label dalam skrip HTML adalah sebagai berikut :

```
<!DOCTYPE html>
<html>
<body>
<h2>HTML Forms</h2>
<form action="action_page.php">
  <label for="nama"> Nama : </label>
</form>
</body>
</html>
```

Hasil skrip apabila di jalankan di browser adalah sebagai berikut :



## 2. Elemen Input pada Form

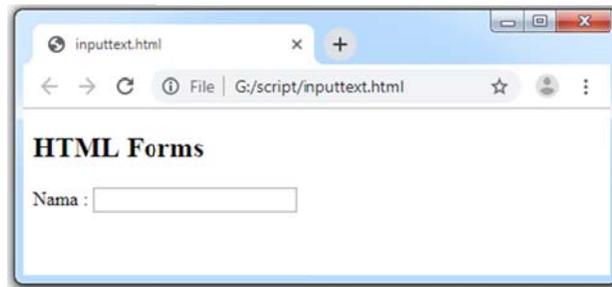
Elemen input pada form mempunyai beberapa tipe (*type*) yang mempunyai fungsi berbeda yaitu antara lain :

### a. Tipe Text

Inputan Tipe text biasanya digunakan untuk memasukkan satu baris kata atau yang tidak terlalu panjang seperti nama, pencarian dan sebagainya. Contoh penerapan input text dalam skrip HTML adalah sebagai berikut :

```
<!DOCTYPE html>
<html>
<body>
<h2>HTML Forms</h2>
<form action="action_page.php">
  <label for="nama"> Nama : </label>
  <input type="text" name="nama">
</form>
</body>
</html>
```

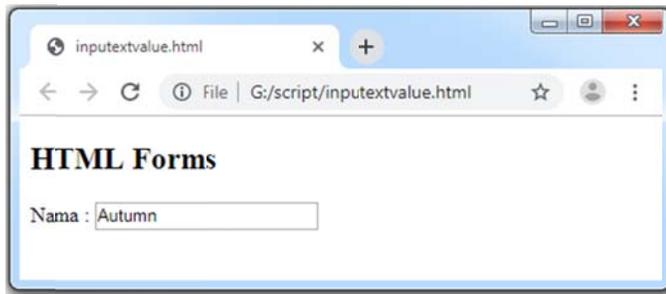
Hasil skrip apabila di jalankan di browser adalah sebagai berikut :



Apabila input text dapat langsung menampilkan nilai (*value*), maka nilai tersebut di masukkan dalam atribut value sehingga jika halaman form dijalankan maka otomatis input text akan terisi dengan nilai yang sudah dimasukkan ke dalam atribut value.

```
<!DOCTYPE html>
<html>
<body>
<h2>HTML Forms</h2>
<form action="action_page.php">
  <label for="nama"> Nama : </label>
  <input type="text" name="nama"
value="Autumn">
</form>
</body>
</html>
```

Hasil skrip apabila di jalankan di browser adalah sebagai berikut :

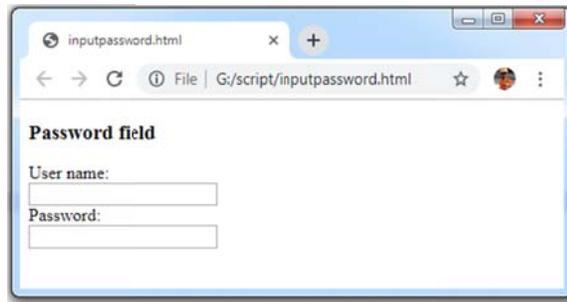


b. Tipe Password

Input ini digunakan untuk memasukkan password ke website. Kata yang dimasukkan dalam field ini akan diubah ke tanda bintang atau lingkaran sehingga tidak bisa di baca oleh pengguna yang lain. Berikut ini penerapan di skrip HTML :

```
<!DOCTYPE html>
<html>
<body>
<form action="">
User name:<br>
<input type="text" name="userid">
<br>
User password:<br>
<input type="password" name="psw">
</form>
</body>
</html>
```

Hasil skrip apabila di jalankan di browser adalah sebagai berikut :



c. Tipe Radio

Tombol radio memungkinkan pengguna memilih hanya satu dari sejumlah pilihan yang ada. Berikut contoh penerapan radio button dalam skrip HTML :

```
<!DOCTYPE html>
<html>
<body>
<h2>Radio Buttons</h2>
<label for="kelamin">Jenis Kelamin
:</label>
<form action="/action_page.php">
  <input type="radio" name="kelamin"
value="L" checked> Laki-laki<br>
  <input type="radio" name="kelamin"
value="P"> Perempuan<br>
</form>
</body>
</html>
```

Hasil skrip apabila di jalankan di browser adalah sebagai berikut :

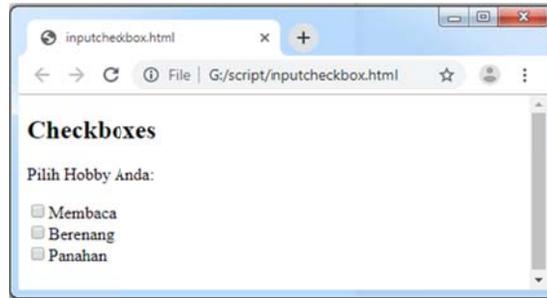


d. Tipe Ceckbox

Input ini memungkinkan pengguna tidak melakukan pilihan atau melakukan pilihan lebih dari satu berdasarkan pilihan yang ada.

```
<!DOCTYPE html>
<html>
<body>
<h2>Checkboxes</h2>
<p>Pilih Hobby Anda:</p>
<form action="/action_page.php">
<input type="checkbox" name="hobi1"
value="Membaca">Membaca
<br>
<input type="checkbox" name="hobi2"
value="Berenang">Berenang
<br>
<input type="checkbox" name="hobi3"
value="Panahan">Panahan
<br><br>
</form>
</body>
</html>
```

Hasil skrip apabila di jalankan di browser adalah sebagai berikut :



e. Tipe Submit

Input ini berupa tombol yang bisa diklik yang berfungsi untuk mengirimkan data formulir ke form-handler. Dimana Form-handler biasanya adalah halaman server dengan skrip untuk memproses data dari form input ke basis data. Form-handler ditentukan dalam atribut aksi form. Penerapan tipe submit dalam skrip HTML sebagai berikut :

```
<!DOCTYPE html>
<html>
<head><title>Submit Button</title></head>
<body>
<h2>HTML Forms</h2>
<form action="action_page.php">
  <label for="nama"> Nama : </label>
  <input type="text" name="nama"
value=" "><br><br>
  <input type="submit" value="Simpan">
</form>
</body>
</html>
```

Hasil skrip apabila di jalankan di browser adalah sebagai berikut :

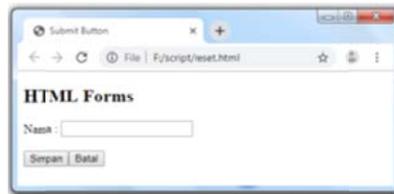


f. Input Tipe Reset

Tipe input ini digunakan untuk mengatur ulang semua nilai formulir ke nilai standarnya. Apabila mengubah nilai input kemudian mengklik tombol "Reset", formulir-data akan diatur ulang ke nilai default. Penerapan tipe reset pada skrip HTML adalah sebagai berikut :

```
<!DOCTYPE html>
<html>
<head><title>Submit Button</title></head>
<body>
<h2>HTML Forms</h2>
<form action="action_page.php">
  <label for="nama"> Nama : </label>
  <input type="text" name="nama"
value=""><br><br>
  <input type="submit"
value="Simpan"><input type="reset"
value="Batal">
</form>
</body>
</html>
```

Hasil skrip apabila di jalankan di browser adalah sebagai berikut :

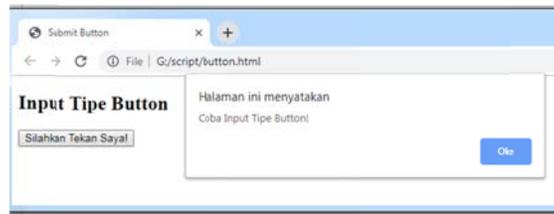


g. Input Tipe Button

Tipe input ini untuk membuat tombol baik itu di dalam form atau diluar form. Input tombol juga menggunakan dengan javascript untuk membuat fungsi lain yang akan dijalankan ketika tombol tersebut di klik. Contoh penerapan dalam HTML sebagai berikut :

```
<!DOCTYPE html>
<html>
<head><title>Submit Button</title></head>
<body>
<h2>Input Tipe Button</h2>
<input type="button" onclick="alert('Coba
Input Tipe Button!')" value="Silahkan
Tekan Saya! ">
</body>
</html>
```

Hasil skrip apabila di jalankan di browser adalah sebagai berikut :

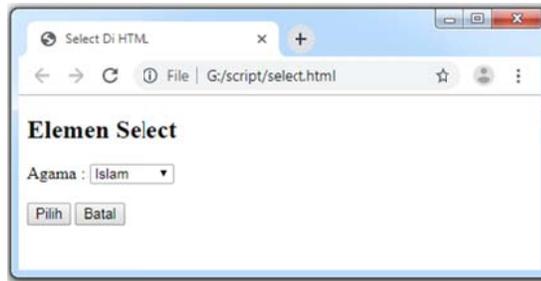


### 3. Select di Form

Elemen `<select>` merupakan *drop-down list* digunakan untuk memilih daftar pilihan. Berikut ini penerapannya dalam HTML :

```
<!DOCTYPE html>
<html><head><title>Select Di
HTML</title></head>
<body>
<h2>Elemen Select</h2>
<form action="/action_page.php">
<label for="agama">Agama : </label>
<select name="agama">
<option value="Islam">Islam</option>
<option value="Katholik">Katholik</option>
<option value="Protestan">Protestan</option>
<option value="Hindu">Hindu</option>
<option value="Budha">Budha</option>
</select>
<br><br>
<input type="submit"
value="Pilih">&nbsp;<input type="reset"
value="Batal">
</form>
</body>
</html>
```

Hasil skrip apabila di jalankan di browser adalah sebagai berikut :



#### 4. Textarea di form

Tag <textarea> digunakan untuk memasukkan teks lebih dari satu baris (multi-baris). Textarea dapat menampung jumlah karakter yang tidak terbatas, dan teks ditampilkan dalam font dengan lebar tetap. Ukuran Textarea dapat ditentukan oleh atribut cols and row, atau pada CSS dengan properti height dan width. Berikut ini contoh penerapannya dalam HTML :

```
<!DOCTYPE html>
<html><head><title>Textarea</title></head>
<body>
<h2>Textarea</h2>
<p>Kritik/Saran :</p>
<form action="/action_page.php">
<textarea name="message" rows="10"
cols="30"></textarea>
<br>
<input type="submit"
value="Kirim">&nbsp;<input type="reset"
value="Batal">
</form>
</body>
</html>
```

Hasil skrip apabila di jalankan di browser adalah sebagai berikut :



## **BAB 6**

### **Pengenalan PHP**

#### **6.1. Sejarah PHP**

Demokratisasi teknologi mengacu pada proses dimana teknologi tersedia untuk segmen populasi yang semakin besar. Mungkin tidak ada penghalang yang lebih kuat dari demokratisasi seperti itu selain Internet, yang telah menjadi platform bagi pengembang di seluruh dunia untuk membangun dan berbagi perangkat lunak *open source*. Perangkat lunak ini pada gilirannya digunakan untuk memberi daya pada jutaan situs web yang dimiliki oleh perusahaan-perusahaan Fortune 50, pemerintahan, lembaga pendidikan, *startup*, organisasi, dan individu.

Meskipun ada banyak platform yang dikembangkan secara kolaboratif yang muncul dalam teknologi Internet (server web Apache, dan bahasa Perl, Python, dan Ruby), mungkin tidak ada yang memiliki pengguna yang begitu banyak seperti bahasa PHP. Dibuat pada pertengahan 1990-an oleh pengembang perangkat lunak Kanada / Denmark bernama Rasmus Lerdorf untuk meningkatkan kemampuan situs pribadinya, dia memulai membuat kode dan menulis untuk digunakan secara umum, pada awalnya proyek tersebut dinamakan Personal Home Page, atau PHP.

Kemudian PHP menjadi singkatan dari “PHP: Hypertext Preprocessor”, yaitu bahasa pemrograman disisi server yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML. Ketika Anda mengakses sebuah URL, maka web browser akan melakukan request ke sebuah web server.

Setiap rilis banyak sekali jumlah kontributor yang mengirimkan peningkatan dan perbaikan bug yang ditemukan, di samping kenaikan yang signifikan dalam jumlah pengguna, dari hanya beberapa lusin pada pertengahan 1990-an untuk memainkan peran dalam 244 juta situs yang mengagumkan (termasuk Facebook, Wikipedia, Cisco WebEx, dan IBM, dan banyak lainnya), menurut survei Netcraft Januari 2013. PHP terlihat di banyak lingkungan dimana pengembang web memiliki begitu banyak pilihan kualitas untuk dipilih, hal ini benar-benar pencapaian yang menakjubkan. Pada tahun 2016 dan 2017 telah dilaporkan bahwa PHP digunakan pada 82% dari server web dibandingkan dengan bahasa pemrograman terkenal lainnya. Versi terbaru dan stabil dari bahasa pemrograman PHP saat ini adalah versi 7.0.16 dan 7.1.2 yang resmi dirilis pada tanggal 17 Februari 2017.

## **6.2. Manfaat PHP**

Dengan sifatnya yang *open source* tidak menjadikan PHP terbatas dalam pemanfaatannya, banyak sekali yang dapat dilakukan dengan PHP diantaranya:

1. Mengurangi waktu untuk membuat web berskala besar
2. Mampu menciptakan web interaktif
3. Menciptakan berbagai tool untuk keperluan online
4. Mendukung e-commerce (shopping carts)
5. Dll

## **6.3. Kelebihan dan Kekurangan PHP**

Bahasa pemrograman PHP mempunyai kelebihan dan kekurangan. Kelebihan PHP di antaranya yaitu :

1. Memiliki Komunitas yang besar

Programmer Web mana sih yang tidak mengetahui PHP, semua web programmer paling tidak pasti pernah mencoba PHP. Banyak sekali website yang menggunakan PHP sebagai bahasa pemrograman untuk membuat aplikasi web atau website nya. Facebook, Yahoo, Wikipedia, WordPress adalah contoh website terkenal yang menggunakan PHP. Forum yang membahas mengenai PHP pun juga banyak tersedia.

2. Mudah Dipelajari

PHP mudah di install dan dikonfigurasi. membuatnya menjadi bahasa pemrograman tingkat entry level yang mudah dipelajari bagi seseorang yang baru memulai belajar pengembangan web. Tutorial untuk memulai belajar pemrograman PHP dapat diperoleh dengan mudah secara online diantaranya [www.w3schools.com](http://www.w3schools.com), [www.codepolitan.com](http://www.codepolitan.com), dan masih banyak yang lainnya, di toko buku, ataupun di lembaga bimbingan kursus pengembangan website.

3. Pengembangan Cepat

Membuat Aplikasi web menggunakan PHP jauh lebih cepat dan mudah daripada mengembangkan aplikasi web menggunakan bahasa pemrograman lain. banyak sekali alat bantu yang tersedia secara *open source* untuk bahasa pemrograman PHP. hal ini mempercepat proses dari start sampai dengan finish sebuah projek pembuatan aplikasi web.

4. Ringkas

Programmer web yang pernah mencoba bahasa pemrograman ASP maupun java pasti mengerti betul satu kelebihan ini. Mulai dari proses install yang tidak perlu setting yang rumit, konfigurasi dengan database yang mudah. hingga proses pengembangan yang tidak memerlukan waktu kompilasi. membuat PHP terasa sangat ringkas dan praktis berbeda dengan bahasa pemrograman lain yang membutuhkan proses kompilasi

untuk dapat melihat website yang telah diselesaikan oleh pembuatan kodenya.

5. Maintenance Mudah

Sekali web yang menggunakan PHP berjalan, programmer dapat dengan mudah melakukan update dari software PHP dengan mudah jika memang diperlukan. Aplikasi web yang dibuat dengan menggunakan PHP dapat dengan mudah diupgrade versi PHP tanpa harus melakukan kompilasi ulang source code namun terkadang ada sintaks yang harus disesuaikan. berbeda sekali dengan bahasa pemrograman lain yang membutuhkan kompilasi ulang jika melakukan upgrade versi dari bahasa pemrograman. PHP juga dapat berjalan pada berbagai macam web server seperti apache, nginx, dan IIS.

6. Open Source

PHP merupakan sebuah projek Open source dengan license yang dikeluarkan oleh PHP group yaitu PHP license V3.01. Inti dari license ini adalah setiap pengguna program PHP bebas menggunakan PHP secara gratis tanpa harus memberikan royalty apapun ke pihak PHP group namun tetap wajib mencantumkan licensi atas PHP yang dimiliki PHP Group. Dengan kata lain selama pemakai program PHP tidak mengakui produk PHP adalah buatannya maka perjual belian program yang menggunakan PHP diperbolehkan tanpa harus membayar licensi apapun.

7. Perkembangan yang Pesat

Karena sifat PHP yang open source, banyak sekali bermunculan projek projek open source besar yang menggunakan PHP seperti OsCommerce, WordPress, Drupal, dan lain lain. Hal ini menjadi keunggulan yang sangat besar bagi orang yang menguasai pemrograman PHP. Dengan sangat luasnya perkembangan PHP,

maka kesempatan untuk bisnis ataupun kerja pada bidang pemrograman PHP sangatlah luas.

Sedangkan kekurangan yang di miliki oleh PHP adalah sebagai berikut :

1. Banyak Kompetisi

Komunitas yang banyak tentu membawa kompetisi yang ketat. Para web developer yang menguasai PHP tiap hari semakin bertambah. Namun kekurangan ini seharusnya menjadi pemacu bagi para pebisnis yang menginginkan produk IT untuk menggunakan PHP sebagai bahasa pemrograman yang digunakan untuk mengembangkan aplikasi bisnisnya karena terbukanya para programmer PHP yang sangat kompetitif dan tiap hari semakin banyak.

2. Mudah Dibajak

Karena sifatnya yang open source membuat aplikasi yang dibangun menggunakan PHP sangat rawan dibajak, source code dari aplikasi PHP dapat dengan mudah di ubah, hal ini membuat PHP tidak cocok untuk digunakan mengembangkan aplikasi jika pemilik aplikasi memiliki source code yang ingin dijaga kerahasiaannya.

3. Terkesan Kurang Prestigious

PHP mudah dipelajari oleh *programmer* pemula membuat bahasa pemrograman PHP terkesan kurang prestisius jika dibandingkan dengan bahasa pemrograman web lain yang terkesan lebih sulit untuk digunakan. Aplikasi web yang dihasilkan dari penggunaan bahasa pemrograman PHP terkesan kurang aman dan memiliki celah keamanan. Namun sebenarnya hal ini disebabkan karena faktor *developer* yang mungkin belum mempelajari secara penuh bagaimana standar dan cara membuat aplikasi yang benar dengan menggunakan PHP. Dari

sisi performa, pemrograman PHP dan pemrograman lainnya jika digunakan dengan standar dan penerapan yang benar, akan menghasilkan sebuah aplikasi web yang berkualitas.

## **6.4. Modal Dasar Mempelajari PHP dan Aplikasi Yang Diperlukan**

Sebelum mempelajari lebih lanjut bahasa pemrograman PHP, sebaiknya kita harus memiliki dasar – dasar pemrograman (C/C++ lebih baik) serta telah menguasai sintak-sintak HTML sehingga akan lebih mudah nantinya mempelajari bahasa pemrograman PHP dan mengimplementasikan dalam membuat aplikasi berbasis web.

Adapun aplikasi yang perlu disiapkan sebelum mempelajari bahasa pemrograman PHP adalah sebagai berikut:

1. Web Server (Apache, IIS, Personal Web Server/PWS), web server merupakan salah satu kebutuhan yang harus dipenuhi ketika kita mempelajari dan mengimplementasikan bahasa pemrograman PHP, web server adalah sebuah software yang memberikan layanan berbasis data dan berfungsi menerima *request* dari HTTP atau HTTPS pada sisi klien yang dikenal dan biasanya kita kenal dengan nama web browser (Mozilla Firefox, Google Chrome, Opera, safari dll) dan untuk mengirimkan kembali yang hasilnya dalam bentuk beberapa halaman web dan pada umumnya akan berbentuk dokumen HTML.

Salah satu Web Server yang sering dipakai adalah Apache. Apache (Apache Web Server – The HTTP Web Server) merupakan web server yang paling banyak dipergunakan di Internet. Program ini pertama kali didesain untuk sistem operasi lingkungan UNIX namun untuk saat ini apache bersifat *multi platform* artinya dapat berjalan di banyak sistem operasi.

Apache mempunyai program pendukung yang cukup banyak. Hal ini memberikan layanan yang cukup lengkap bagi penggunanya.

Beberapa dukungan Apache :

- Kontrol Akses  
Kontrol ini dapat dijalankan berdasarkan nama host atau nomor IP CGI (Common Gateway Interface) Yang paling terkenal untuk digunakan adalah perl (Practical Extraction and Report Language), didukung oleh Apache dengan menempatkannya sebagai modul (mod\_perl).
- PHP (Personal Home Page/PHP Hypertext Processor)  
Program dengan metode semacam CGI, yang memproses teks dan bekerja di server. Apache mendukung PHP dengan menempatkannya sebagai salah satu modulnya (mod\_php). Hal ini membuat kinerja PHP menjadi lebih baik
- SSI (Server Side Includes)

Web server Apache mempunyai kelebihan dari beberapa pertimbangan di atas :

- a. Apache termasuk dalam kategori freeware sehingga bebas digunakan tanpa membayar lisensi.
  - b. Apache mudah sekali proses instalasinya dan tidak membutuhkan waktu yang lama serta ringan.
  - c. Mampu beroperasi pada berbagai platform sistem operasi.
  - d. Mudah mengatur konfigurasinya. Apache mempunyai hanya empat file konfigurasi.
  - e. Mudah dalam menambahkan peripheral lainnya ke dalam platform web servernya.
2. PHP server, selain web server tentunya kita memerlukan aplikasi PHP agar sintak-sintak yang kita tuliskan dapat berjalan

- a. Database Server (MySQL, Interbase, MS SQL, Oracle, PostgreSQL, dll), bahasa pemrograman dapat mudah terkoneksi ke banyak database salah satunya database MySQL.
- b. Web Editor, ada banyak sekali web editor yang bisa kita pakai untuk menuliskan sintak-sintak PHP diantaranya notepad, notepad ++, SubLime, Atom dan lainnya

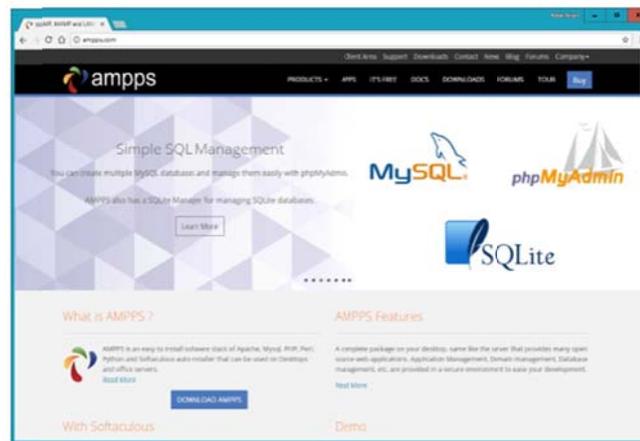
Dari beberapa aplikasi diatas nomor 1(satu) sampai dengan 3(tiga) telah tersedia dalam bentuk aplikasi *bundle* beberapa diantaranya adalah AMPPS, XAMPP, APPSERV, WAMP dan lainnya pada buku ini penulis akan menjelaskan bagaimana proses instalasi salah satu program *bundle* php mysql yaitu XAMPP.

## **6.5. Instalasi PHP, MySQL, & Apache dengan AMPPS**

AMPPS adalah aplikasi web *stack* yang terdiri dari Apache, Mysql, MongoDB, PHP, Perl, Python and Softaculous auto-installer, hal ini menjadikan AMPPS salah satu aplikasi web server *bundle* terbaik. Web application stack ini dibuat oleh Softaculous Ltd dan dapat diinstal di platform Windows, Linux dan MacOS. Antarmuka AMPPS sangat mirip dengan antarmuka Softaculous. Bila Anda terbiasa menggunakan softaculous, Anda tidak akan kesulitan menggunakan AMPPS ini.

Langkah-langkah Instalasi AMPPS sebagai berikut :

1. Download aplikasi secara gratis di <http://ampps.com/>.



Penulis menyarankan untuk selalu mengunduh rilis stabil terbaru (saat menulis ini, yang di unduh adalah versi 3,7, yang berukuran kurang lebih 200MB). Tersedia berbagai pilihan installer mulai dari Windows, OS X, dan Linux pada halaman unduhan.

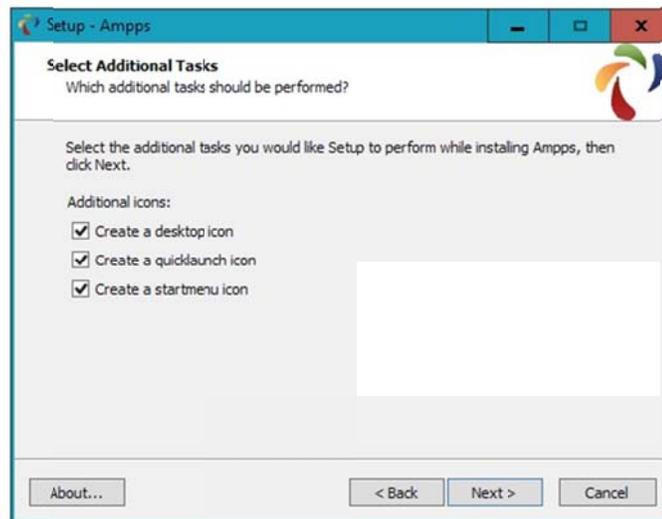
Beberapa tampilan dan pilihan yang ditampilkan dalam panduan berikut mungkin saja berubah ketika anda mengunduh versi terbaru. Jika demikian, cukup ikuti petunjuk atau langkah-langkah pada saat proses instalasi.

2. Setelah mengunduh installer, jalankan dengan cara double klik untuk memunculkan jendela ampps. Namun, sebelumnya, jika Anda menggunakan program anti-virus atau mengaktifkan Kontrol Akun Pengguna di Windows, Anda mungkin akan diperlihatkan satu atau lebih pemberitahuan, dan harus mengklik Ya dan / atau OK untuk melanjutkan proses instalasi.

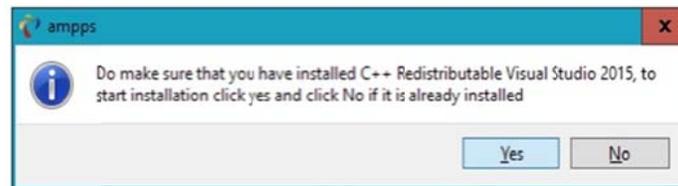


3. Setelah itu klik tombol next sampai pada halaman konfirmasi tempat pemasangan, yang mungkin akan disarankan sebagai berikut tergantung huruf drive yang anda gunakan, anda dapat ubah sesuai keinginan.

4. Setelah Anda memutuskan di mana akan menginstal Ampps, klik Next, pilih nama folder menu Start, dan klik Next. Anda dapat memilih ikon mana yang ingin Anda instal. Pada layar berikutnya, klik tombol Instal untuk memulai prosesnya.



5. Hal terakhir yang harus Anda lakukan adalah menginstal C++ Redistributable Visual Studio, jika belum terinstall. Visual Studio adalah lingkungan di mana Anda akan melakukan pekerjaan pengembangan. Sebuah jendela akan muncul untuk meminta Anda Instal C ++ Visual Studio Redistributable.
6. konfirmaKlik Ya untuk memulai instalasi atau Tidak jika Anda yakin sudah menginstalnya.



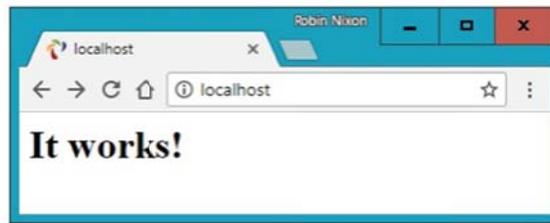
7. Jika Anda memilih untuk melanjutkan dan menginstal, Anda harus menyetujui syarat dan ketentuan di jendela pop-up yang

muncul, dan kemudian klik Instal. Pemasangan ini cukup cepat. Klik Tutup untuk menyelesaikan.

8. Setelah Amps diinstal, akan muncul jendela kontrol di kanan bawah desktop Anda. Jendela ini juga dapat dipanggil menggunakan aplikasi Amps di menu Start, atau di desktop jika Anda mengizinkan ikon ini dibuat.



9. Untuk melakukan tes apakah web server sudah berjalan atau belum, silahkan mencoba untuk akses lewat browser dengan mengetik alamat localhost atau 127.0.0.1, jika webserver sudah aktif maka akan muncul tampilan "It works!" di browser Anda.



## 6.6. Dasar-dasar PHP

Pada saat ini, Anda sudah terbiasa dengan latar belakang dan riwayat PHP, dan Anda telah mempelajari aplikasi apa saja yang dibutuhkan serta proses instalasi web server. Apa yang telah Anda pelajari sejauh ini menetapkan tahapan untuk bahan yang tersisa dalam buku ini: membuat situs web berbasis PHP yang kuat! Bab ini mengawali diskusi kita dengan memperkenalkan garis besar fitur dasar bahasa PHP. Secara khusus, Anda akan belajar bagaimana melakukan hal berikut:

1. Menyisipkan kode PHP ke halaman web Anda.
2. Kode komentar menggunakan berbagai metodologi yang mengadopsi dari bahasa scripting Unix shell, C, dan C++.
3. Mengeluarkan data ke browser menggunakan pernyataan `echo()`, `print()`, `printf()`, dan `sprintf()`.
4. Menggunakan tipe data, variabel, operator, dan pernyataan PHP untuk membuat skrip canggih.
5. Manfaatkan struktur dan pernyataan kontrol utama, termasuk *if-else-elseif*, *while*, *foreach*, *include*, *require*, *break*, *continue*, dan *declare*.

Pada akhir bab ini, Anda tidak hanya memiliki pengetahuan yang diperlukan untuk membuat aplikasi PHP dasar, tetapi juga pemahaman tentang apa yang diperlukan untuk memanfaatkan sebagian besar materi yang dibahas dalam bab-bab selanjutnya.

## 6.7. Menyisipkan kode PHP di Halaman Web

Salah satu keunggulan PHP adalah Anda dapat menanamkan kode PHP secara langsung di dalam HTML. Agar kode dapat berjalan, halaman harus diteruskan ke mesin PHP untuk interpretasi. Tetapi server web tidak hanya memproses setiap halaman, melainkan, ia hanya memproses halaman-halaman yang diidentifikasi oleh ekstensi file tertentu (biasanya .php). Namun, bahkan secara selektif jika hanya mengirimkan halaman tertentu ke server web akan sangat tidak efisien bagi mesin untuk melakukan *scanning* setiap baris sebagai perintah PHP.

Oleh karena itu, mesin memerlukan beberapa cara untuk segera menentukan area halaman mana yang diaktifkan PHP. Ini secara logis dilakukan dengan memasukkan kode dalam tag PHP. Tag PHP biasanya didefinisikan sebagai `<?php` untuk tag awal kode dan `?>` Untuk tag akhir kode.

Setiap file dapat berisi satu blok kode PHP atau beberapa blok yang tertanam di seluruh file. Saat file berisi satu blok kode PHP, sudah umum dilakukan untuk mengecualikan tag `?>`. Ini akan menghilangkan konten apa pun, biasanya spasi putih di akhir file yang dikirim ke klien sebagai bagian dari output.

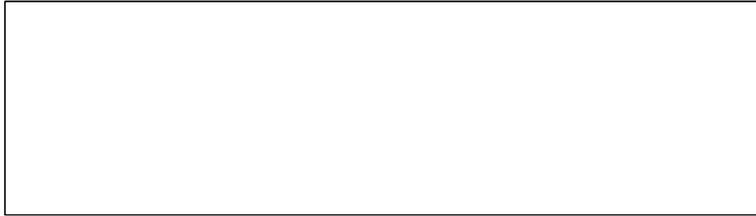
## BAB 7

### SINTAKS-SINTAKS DASAR PHP

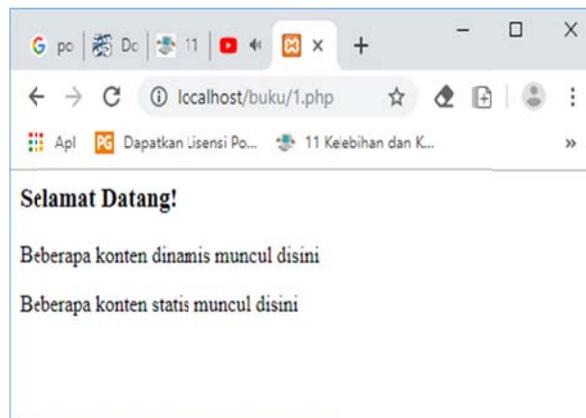
PHP adalah bahasa yang cukup sederhana dengan bersumber pada bahasa C dan Perl, namun lebih mirip Java. Ini juga sangat fleksibel, tetapi ada beberapa aturan yang perlu Anda pelajari tentang sintaks dan strukturnya.

#### 7.1. Sintaks Default PHP

Sintaks *default* pembatas dibuka dengan `<?php` dan diakhiri dengan `?>`, Seperti ini:



Hasil tampilan apabila di jalankan di browser adalah sebagai berikut :



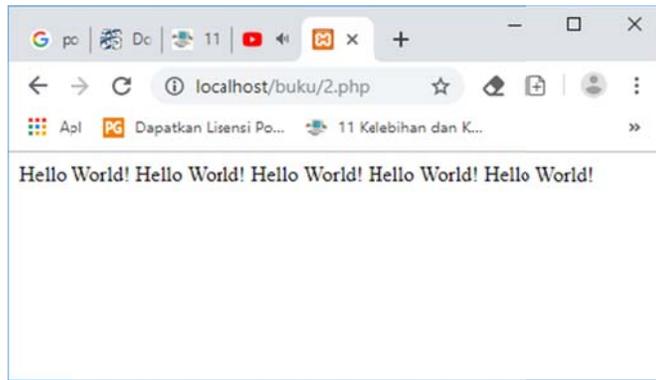
Apabila Anda memiliki kode PHP yang disisipkan dalam HTML dalam suatu file dan menginginkan web server dapat menjalankannya, maka file tersebut harus disimpan dalam ekstensi .php. Apabila Anda menyimpannya dengan ekstensi .html atau .htm, maka kode PHP tersebut tidak akan diproses dan akan ditampilkan dalam web browser seperti apa adanya (berupa kode-kode). Sintak-sintaks dasar PHP antara lain :

## 7.2. Tanda Semi Colon (;)

Apabila Anda perhatikan contoh sebelumnya, maka terdapat tanda titik koma (semicolon) pada akhir perintah echo. Tanda semicolon merupakan penanda akhir dari statement PHP dan harus ada. Sebagai contoh perhatikan sintak berikut:

```
<!DOCTYPE html>
<html>
<head> <title>Halaman PHP pertamaku</title>
</head>
<body>
<?php
echo "Hello World! ";
echo "Hello World! ";
echo "Hello World! ";
echo "Hello World! ";
echo "Hello World! ";
?>
</body>
</html>
```

Hasil tampilan skrip apabila dijalankan dengan browser adalah sebagai berikut :



### 7.3. Pindah Spasi

Seperti halnya HTML, pergantian spasi dalam PHP tidak akan mempengaruhi tampilan hasilnya. Dengan kata lain, pergantian spasi akan diabaikan oleh PHP. Perhatikan contoh berikut ini. Pada contoh tersebut diberikan tiga bentuk penulisan kode PHP yang berbeda namun akan dihasilkan tampilan yang sama dalam web browser.

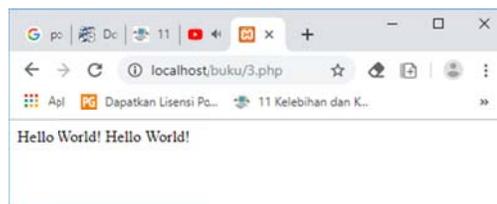
```
<!DOCTYPE html>
<html>
<head>
<title>Halaman PHP pertamaku</title>
</head>
<body>
<?php
echo "Hello World! ";
echo "Hello World! ";
?>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<title>My First PHP Page</title>
</head>
<body>
<?php echo "Hello World! "; echo "Hello World!";
?>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<title>Halaman PHP pertamaku</title>
</head>
<body>
<?php
echo "Hello World! ";

echo "Hello World! ";
?>
</body>
</html>
```

Hasil tampilan dari skrip yang dijalankan di browser adalah sebagai berikut :

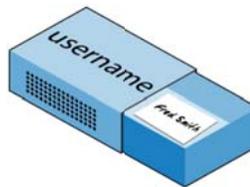


## 7.4. Variabel

Misalkan dalam PHP kita akan menyimpan suatu nilai berupa angka atau string dalam suatu variabel, caranya adalah membuat nama variabel terlebih dahulu kemudian diberikan suatu assignment pada nilai yang diinginkan. Perhatikan sintaks berikut ini :



Ada analogi sederhana yang akan membantu Anda memahami apa itu variabel PHP. Anggap saja kotak korek api kecil (atau besar)!, korek api yang telah Anda lukis dan tuliskan nama.



### 7.4.1. Variabel string

Bayangkan Anda memiliki kotak korek api di mana Anda telah menulis kata *username*. Anda kemudian menulis Fred Smith di selembar kertas dan meletakkannya di dalam kotak (lihat Gambar xx33). Ya, itu proses yang sama dengan menetapkan nilai string ke variabel, seperti ini:

```
$username = "Fred Smith";
```

Tanda petik menunjukkan bahwa "Fred Smith" adalah serangkaian karakter (string). Anda harus melampirkan setiap string dalam tanda petik (tanda petik tunggal), meskipun ada sedikit

perbedaan antara kedua jenis kutipan, yang dijelaskan kemudian. Ketika Anda ingin melihat apa yang ada di dalam kotak, Anda membukanya, mengeluarkan selebar kertas, dan membacanya. Di PHP, untuk melakukan hal seperti ini (menampilkan konten variabel ke layar) dapat menggunakan perintah berikut:

```
echo $username;
```

Contoh penerapan variabel

```
<?php
    $hello = "Hello World!";
    $sebuah_bilangan = 4;
    $bilanganYangLain = 8;
?>
```

Dari contoh di atas tampak bahwa dalam PHP, nama variabel tidak perlu dideklarasikan terlebih dahulu seperti halnya bahasa Pascal atau C/C++.

#### **7.4.2. Variabel Numeric**

Variabel tidak hanya mengandung string, variabel juga bisa mengandung angka. Jika kita kembali ke analogi kotak korek api, untuk menyimpan angka 17 dalam variabel \$count, persamaannya akan menempatkan, katakanlah, 17 manik-manik dalam kotak korek api Anda dapat menulis kata count:

```
$count = 17;
```

Anda juga bisa menggunakan angka desimal, sintaksnya sama:

```
$count = 17.5;
```

Untuk membaca isi kotak korek api, Anda cukup membukanya dan menghitung manik-manik. Dalam PHP, Anda akan menetapkan nilai \$count ke variabel lain atau mungkin hanya menampilkan ke browser web.

### 7.4.3. Variabel Array

Jadi apa itu array? Nah, Anda bisa menganggapnya sebagai beberapa kotak korek api yang direkatkan. Misalnya, katakanlah kami ingin menyimpan nama pemain untuk tim sepak bola yang terdiri dari lima orang dalam larik yang disebut \$tim. Untuk melakukan ini, kita bisa merekatkan lima kotak korek api berdampingan dan menuliskan nama semua pemain di selebar kertas terpisah, dan menempatkan satu nama di setiap kotak korek api.

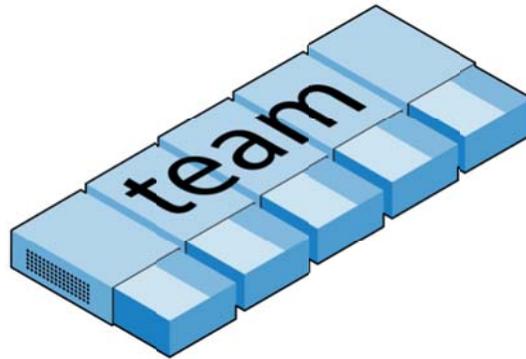
Di seluruh bagian atas rakitan kotak korek api kami akan menulis kata tim (lihat Gambar ). Jika ditulis di PHP adalah sebagai berikut:

```
$team = array('Bill', 'Mary', 'Mike', 'Chris', 'Anne');
```

Array dibangun dengan konstruksi *array()*;

dengan lima string di dalamnya. Setiap string terlampir dalam tanda petik, dan string harus dipisahkan dengan koma. Jika kita ingin tahu siapa pemain 4, kita bisa menggunakan perintah ini:

```
echo $ team [3]; // Menampilkan nama Chris
```

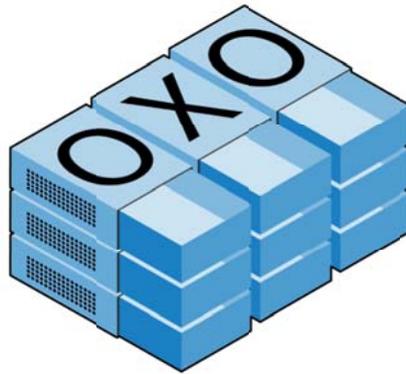


Gambar array yang diumpamakan deretan korek api

Mengapa untuk menampilkan pemain 4 menggunakan angka 3, bukan 4, adalah karena elemen pertama dari array PHP sebenarnya adalah elemen nol, sehingga jumlah pemain akan menjadi 0 hingga 4.

### ***Array Dua Dimensi***

Ada banyak lagi yang dapat Anda lakukan dengan array. Misalnya, alih-alih menjadi garis kotak korek api satu dimensi, mereka dapat berupa matriks dua dimensi atau bahkan dapat memiliki lebih banyak dimensi. Sebagai contoh array dua dimensi, katakanlah kita ingin melacak permainan tic-tactoe, yang membutuhkan struktur data sembilan sel yang disusun dalam kotak  $3 \times 3$ . Untuk mewakili ini dengan kotak korek api, bayangkan sembilan dari mereka direkatkan satu sama lain dalam matriks tiga baris dengan tiga kolom (lihat Gambar).



Gambar Analogi array multi dimensi

Anda sekarang dapat menempatkan selembur kertas dengan x atau o di kotak korek api yang benar untuk setiap gerakan yang dimainkan. Untuk melakukan ini dalam kode PHP, Anda harus mengatur array yang berisi tiga array lagi, seperti dalam Contoh berikut, di mana array diatur dengan permainan yang sudah dalam proses.

Contoh:

```
<?php
$oxo = array(array('x', ' ', 'o'),
              array('o', 'o', 'x'),
              array('x', 'o', ' '));
?>
```

Sintaks kali ini Nampak lebih sulit dibanding sintak array sebelumnya, tetapi mudah dimengerti jika Anda memahami sintaks dasar array. Ada tiga konstruksi array() yang bersarang di dalam array(). Kami telah mengisi setiap kolom atau setiap baris dengan larik yang hanya terdiri dari satu karakter, x, a, atau spasi kosong. (Kami menggunakan ruang kosong sehingga semua sel akan memiliki lebar yang sama saat ditampilkan.)

Untuk kembali elemen ketiga di baris kedua array ini, Anda akan menggunakan perintah PHP berikut, yang akan menampilkan x:

```
echo $oxo[1][2];
```

**Ingat** bahwa indeks array (pointer pada elemen dalam array) mulai dari nol, bukan satu, jadi [1] pada perintah sebelumnya merujuk ke yang kedua dari tiga array, dan [2] mereferensikan posisi ketiga dalam array tersebut. Ini akan mengembalikan isi kotak korek api baris ke dua kolom ke tiga yakni “x”.

Seperti yang disebutkan sebelumnya, kita dapat mendukung array dengan dimensi lebih banyak hanya dengan membuat lebih banyak array dalam array. Namun, pada buku ini tidak dibahas mengenai array dengan dimensi lebih banyak.

## 7.5. Aturan Penamaan Variabel

Berikut ini adalah beberapa aturan penulisan nama variabel:

- Nama variabel harus diawali dengan huruf atau underscore (\_)
- Nama variabel hanya boleh dituliskan dengan alpha numeric a-z, A-Z, 0-9 dan underscore
- Nama variabel yang terdiri lebih dari satu kata, dapat dipisahkan dengan underscore
- Nama variabel *case sensitive* dengan huruf besar-kecil. Variabel \$High\_Score tidak sama dengan variabel \$high\_score.

## 7.6. Echo

Seperti yang Anda lihat pada contoh-contoh kode PHP sebelumnya, bahwa perintah echo digunakan untuk menampilkan teks ke dalam browser. Suatu teks atau string dapat dituliskan di browser dengan langsung dituliskan dalam echo yang diapit oleh dua tanda petik ganda (quotes) atau menyimpan string atau teks terlebih

dahulu dalam suatu variabel kemudian dituliskan dalam echo. Berikut ini adalah contohnya:

Contoh:

```
<?php
  $StringKu = "Hello!";
  echo $StringKu;
  echo "<h5>I love using PHP!</h5>";
?>
```

Perlu diperhatikan kita harus hati-hati dalam penulisan suatu string yang di dalamnya terdapat tanda petik ganda (quotes) menggunakan echo. Dalam echo, tanda quotes merupakan penanda awal dan akhir teks/string yang akan ditulis dengan echo, sehingga Anda harus memperhatikan hal-hal berikut ini

- Jangan menggunakan tanda quotes di dalam teks yang akan ditulis dengan echo
- Apabila Anda tetap ingin menuliskan tanda quotes dalam teks yang akan ditulis dengan echo, maka berikan tanda slash “\” di depan quotes tersebut.
- Selain itu, dapat pula Anda gunakan tanda petik tunggal (apostrophes) untuk menggantikan tanda quotes pada teks.

Contoh:

```
<?php echo "<font face="verdana" size="4">I love
using PHP!</font>";    ?>

<?php echo "<font face=\"verdana\" size=\"4\">I
love using PHP!</font>";    ?>

<?php echo "<font face='verdana' size='4'>I love
using PHP!</h5>";    ?>
```

Pada contoh kode pertama di atas akan terjadi error karena dalam teks yang ditulis dalam echo terdapat tanda quotes. Sedangkan untuk kode kedua dan ketiga tidak terdapat error dan akan dihasilkan output yang sama di browser.

### 7.6.1. Menampilkan Nilai Variabel dengan Echo

Nilai variabel dapat dengan mudah ditampilkan dengan menggunakan echo, baik nilai yang berupa bilangan maupun string. Berikut ini adalah contoh-contoh di antaranya:

```
<?php
$string_ku = "Hello.. Nama saya: ";
$bilangan_ku = 4;
$huruf_ku = "a";
echo $string_ku;
echo $bilangan_ku;
echo $huruf_ku;
?>
```

Catatan: Untuk menampilkan nilai variabel dengan echo tanpa menggunakan tanda quotes.

Berikut ini contoh menampilkan gabungan suatu nilai dari variabel yang berupa string.

```
<?php
$string_ku = "Hello. Nama saya: ";
$baris_baru = "<br>";
echo $string_ku."Ari".
$baris_baru;
echo "Hi, Nama saya Ari. Kamu siapa?
".$string_ku.$baris_baru;
echo "Hi, Nama saya Ari. Kamu siapa?
".$string_ku."Amalia";
?>
```

Untuk menggabungkan beberapa string menjadi satu digunakan operator dot (.)

## 7.7. Komentar dalam PHP

Seperti halnya bahasa pemrograman yang lain, komentar dalam suatu kode PHP tidak akan dieksekusi. Terdapat dua cara memberikan komentar dalam PHP, yaitu

- Diberikan tanda // di depan teks komentar. Perintah ini hanya bisa berlaku untuk komentar dalam satu baris
- Diberikan tanda /\* di depan teks komentar dan diakhiri dengan \*/. Perintah ini dapat digunakan untuk komentar yang terdiri lebih dari satu baris.

Contoh:

```
<?php
echo "Hello World!"; // Ini akan mencetak Hello
World!
echo "<br>Psst...You can't see my PHP
comments!"; // echo "nothing"; //
echo "Namaku Faza!";
?>
```

```
<?php
/* Berikut ini adalah perintah
untuk menuliskan Hello World */
echo "Hello World!";
/* echo "My name is Humperdinkle!";
echo "No way! My name is Ari-PHP Programmer!";
*/
```

## 7.8. Operator

Secara umum dalam bahasa pemrograman untuk memanipulasi atau melakukan proses perhitungan suatu nilai menggunakan

sebuah operator. Sampai sini anda sudah mencoba operator “.” dan operator “=”, selain dua operator tersebut masih ada beberapa operator lagi dalam PHP yaitu.

- Operator aritmatik
- Operator perbandingan
- Gabungan operator aritmatik dan assignment
- Operator Logika

### 7.8.1. Operator Aritmatik

Operator aritmatika digunakan untuk melakukan operasi matematika. Anda dapat menggunakannya untuk empat operasi utama (penjumlahan, pengurangan, perkalian, dan pembagian) serta untuk menemukan modulus (sisanya setelah pembagian) dan untuk menambah (*increment*) atau mengurangi(*decrement*) nilai seperti pada tabel berikut ini.

Tabel Operator Aritmatik

| Operator | Deskripsi   | Contoh     |
|----------|-------------|------------|
| +        | Penjumlahan | $\$j + 1$  |
| -        | Pengurangan | $\$j - 6$  |
| *        | Perkalian   | $\$j * 11$ |
| /        | Pembagian   | $\$j / 4$  |
| %        | Modulus     | $\$j \% 9$ |
| ++       | Increment   | $++\$j$    |
| --       | Decrement   | $--\$j$    |
| !        | Not         | $!\$j$     |

### 7.8.2. Operator Perbandingan

Operator pembanding umumnya digunakan di dalam sebuah konstruksi seperti pernyataan if di mana Anda perlu membandingkan dua item. Misalnya, Anda mungkin ingin tahu

apakah suatu variabel yang telah Anda tambahkan telah mencapai nilai tertentu, atau apakah variabel lain kurang dari nilai yang ditetapkan, dan seterusnya (lihat Tabel).

**Perhatian!!** perbedaan antara = dan ==. Yang pertama adalah operator penugasan, dan yang kedua adalah operator perbandingan. Bahkan pemrogram tingkat lanjut kadang-kadang dapat salah dalam penggunaan keduanya, jadi hati-hati.

Misal \$j=10, maka berikut ini adalah beberapa contoh penggunaan operator perbandingan dan hasilnya.

Tabel Operator Perbandingan

| Operator | Deskripsi                         | Contoh    | Hasil |
|----------|-----------------------------------|-----------|-------|
| ==       | sama dengan                       | \$j == 4  | False |
| !=       | tidak sama dengan                 | \$j != 21 | True  |
| <        | lebih kecil dari                  | \$j < 3   | False |
| >        | lebih besar dari                  | \$j > 100 | False |
| <=       | lebih kecil atau sama dengan dari | \$j <= 15 | True  |
| >=       | lebih besar atau sama dengan dari | \$j >= 15 | False |

### 7.8.3. Kombinasi Operator Aritmatik dan Assignment

Dalam pemrograman seringkali dijumpai proses yang melibatkan proses increment. Misalkan kita menginginkan proses increment dengan tingkat kenaikan 1, maka perintah yang dituliskan dapat berupa

```
$count=$count + 1;
```

Pernyataan diatas dapat disingkat didalam pemrograman PHP seperti berikut:

```
$count +=1;
```

Dari contoh di atas tampak bahwa operator yang digunakan (+=) merupakan gabungan dari operator aritmatik dan assignment. Berikut ini adalah bentuk-bentuk operator lain jenis ini.

Tabel Gabungan Operator Aritmatik dan Assigment

| Operator | Contoh                         | Operasi yang Ekuivalen                      |
|----------|--------------------------------|---|
| +=       | <code>\$x += 2;</code>         | <code>\$x = \$x + 2;</code>                 |
| -=       | <code>\$x -= 4;</code>         | <code>\$x = \$x - 4;</code>                 |
| *=       | <code>\$x *= 3;</code>         | <code>\$x = \$x * 3;</code>                 |
| /=       | <code>\$x /= 2;</code>         | <code>\$x = \$x / 2;</code>                 |
| %=       | <code>\$x %= 5;</code>         | <code>\$x = \$x % 5;</code>                 |
| .=       | <code>\$my_str="hello";</code> | <code>\$my_str = \$my_str . "hello";</code> |

#### 7.8.4. Operator Logika

Jika Anda belum pernah menggunakannya sebelumnya, operator logika mungkin awalnya terlihat sulit. Tapi pikirkan saja cara Anda menggunakan logika dalam bahasa Inggris. Misalnya, Anda mungkin berkata, "Jika waktunya lebih dari 12 malam. dan lebih awal dari jam 2 siang, makan siang." Dalam PHP, kode untuk ini mungkin terlihat seperti berikut:

```
if ($hour > 12 && $hour < 14) dolunch();
```

dari sintaks diatas, kita telah memindahkan serangkaian instruksi untuk pergi makan siang ke fungsi yang harus kita buat nanti yang disebut fungsi *dolunch*.

Seperti yang ditunjukkan contoh sebelumnya, Anda biasanya menggunakan operator logika untuk menggabungkan hasil dari dua operator pembandingan yang diperlihatkan di bagian sebelumnya. Operator logika juga dapat menjadi inputan ke operator logika lain: "Jika waktunya lebih dari 12 p.m. dan lebih awal dari jam 2 malam, atau jika bau daging panggang menembus lorong dan ada piring di atas meja." Sebagai aturan, jika sesuatu memiliki nilai *TRUE* atau *FALSE*, itu dapat dimasukkan ke operator logika. Operator logika mengambil dua inputan benar atau salah dan menghasilkan pernyataan benar atau salah. Tabel berikut ini memperlihatkan operator logika.

Tabel Operator Logika

| Operator | Deskripsi                  | Contoh                                    |
|----------|----------------------------|---|
| &&       | dan                        | <code>\$j == 3 &amp;&amp; \$k == 2</code> |
| and      | prioritas rendah dari dan  | <code>\$j == 3 and \$k == 2</code>        |
|          | atau                       | <code>\$j &lt; 5    \$j &gt; 10</code>    |
| or       | prioritas rendah dari atau | <code>\$j &lt; 5 or \$j &gt; 10</code>    |
| !        | tidak                      | <code>! (\$j == \$k)</code>               |
| xor      | Exclusive OR               | <code>\$j xor \$k</code>                  |

Perhatikan bahwa "&&" biasanya dapat dipertukarkan dengan "and"; hal yang sama berlaku untuk "||" dan "or". Namun, karena "and" dan "or" memiliki prioritas yang lebih rendah Anda benar-benar harus menghindari penggunaannya, kecuali jika itu satu-satunya pilihan, seperti dalam pernyataan berikut, yang harus menggunakan operator "or":

```
$html = file_get_contents($site) or die("Tidak dapat download dari $site");
```

Yang paling jarang digunakan atau tidak biasa dari operator logika adalah xor, yang berarti *exclusive or*, logika ini mengembalikan nilai *TRUE* jika salah satu bernilai *TRUE*, tetapi akan menghasilkan nilai *FALSE* jika kedua input *TRUE* atau kedua input *FALSE*. Untuk

memahami hal ini, bayangkan Anda ingin membuat pembersih sendiri untuk barang-barang rumah tangga. Amonia merupakan pembersih yang baik, demikian juga pemutih, jadi Anda ingin pembersih Anda memiliki salah satunya. Tetapi pembersih tidak harus memiliki keduanya, karena kombinasi ini berbahaya. Di PHP, Anda bisa mewakili ini sebagai berikut:

```
$komposisi = $amonia xor $pemutih;
```

Pada contoh, jika salah satu diantara \$amonia dan \$pemutih bernilai *TRUE*, \$komposisi juga akan bernilai ke *TRUE*. Tetapi jika keduanya *TRUE* atau keduanya *FALSE*, \$komposisi akan bernilai *FALSE*

## BAB 8

# MODULARITAS

Modularisasi dalam bahasa pemrograman umum dilakukan guna mempermudah debugging dan pengembangan program. program yang besar dipecah menjadi program program yang kecil yang selanjutnya disebut modul. Setiap modul dikembangkan untuk tujuan dan fungsi khusus, modul-modul kecil tersebut dapat dipanggil sewaktu-waktu diperlukan. Dalam PHP juga mendukung konsep tersebut yang selanjutnya diberinama modularitas. Ada dua perintah modularitas dalam PHP yang dapat digunakan yaitu *include* dan *require*.

### 8.1. Include()

Statement `include()` akan menyertakan isi suatu file tertentu. Include dapat diletakkan dalam suatu looping misal dalam *statement for* atau *while*. Untuk memudahkan pemahaman, diberikan contoh berikut. Misalkan kita akan membuat menu link sejumlah 4 buah yaitu `index`, `about`, `links`, dan `contact` pada setiap halaman web yang kita buat. Teknik yang digunakan adalah membuat menu link dalam suatu modul tersendiri kemudian modul tersebut dipanggil pada setiap halaman web yang diinginkan terdapat menu link di dalamnya.

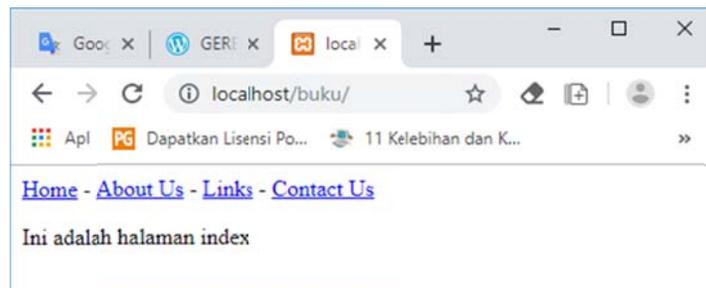
#### **menu.php**

```
<html>
<body>
  <a href="index.php">Home</a> -
  <a href="about.php">About Us</a> -
  <a href="links.php">Links</a> -
  <a href="contact.php">Contact Us</a> <br>
</body>
</html>
```

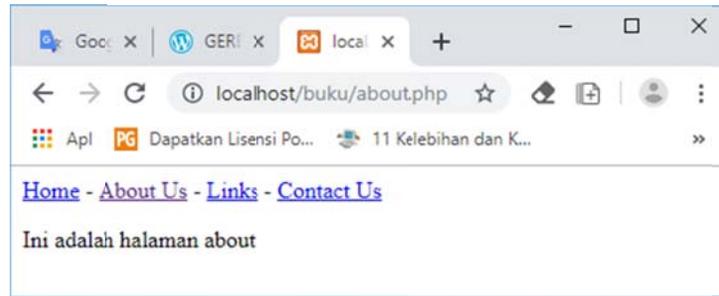
## index.php

```
<html>
<body>
<?php
    include("menu.php");
?>
<p>Ini adalah halaman index</p>
</body>
</html>
```

Dari teknik di atas tampak adanya kemudahan dalam membuat halaman web. Dalam hal ini, kita tidak perlu membuat menu link di setiap halaman web yang ada. Bayangkan seandainya kita punya halaman web sejumlah 100 buah yang kesemuanya ingin diberi menu link tanpa menggunakan teknik di atas, tentu hal tersebut sangat merepotkan. dari empat buah halaman diatas jika di jalankan (*localhost/nama\_folder*) pada browser maka akan nampak seperti gambar berikut:



Gambar halaman pertama yang dijalankan(index.php)



Gambar halaman about.php

## 8.2. Require()

Statement require digunakan untuk membaca nilai variable dan fungsi-fungsi dari sebuah file lain. Cara penulisan:

```
require(namafile);
```

berbeda dengan include, require tidak dapat dimasukkan dalam struktur perulangan/looping, misalnya while atau for, karena hanya memperbolehkan pemanggilan file yang sama tersebut hanya sekali saja. Selain tidak dapat dimasukkan kedalam perulangan lalu apalagi perbedaan dengan include? untuk itu perhatikan contoh sintak berikut:

```
<?php
include("noFileExistsHere.php");
echo "Hello World!";
?>
```

Asumsi bahwa file noFileExistsHere.php tidak ada maka dengan menggunakan include akan menghasilkan tampilan:

```
warning:  main(noFileExistsHere.php):  failed
to open stream: No such file or directory in
include.php on line 2
```

```
warning:      main():      Failed      opening
'noFileExistsHere.php' for inclusion
(include_path='./usr/lib/php:/usr/local/lib/
php') in include.php on line 2
```

Hello World

Selanjutnya mari kita coba menggunakan require().

```
<?php
require("noFileExistsHere.php");
echo "Hello World!";
?>
```

Apabila dijalankan maka menghasilkan error sebagai berikut :

```
Warning:  main(noFileExistsHere.php):  failed
to open stream: No such file or directory in
require.php on line 2
```

```
Fatal error: main(): Failed opening required
'noFileExistsHere.php'
(include_path='./usr/lib/php:/usr/local/lib/
php') in require.php on line 2
```

Dari kedua sintak tersebut diatas, pada include(), error yang dihasilkan hanya berupa Warning saja dan statement berikutnya masih dapat dijalankan. Hal ini terlihat bahwa teks Hello World! Masih ditampilkan di browser. Sedangkan pada require(), error yang

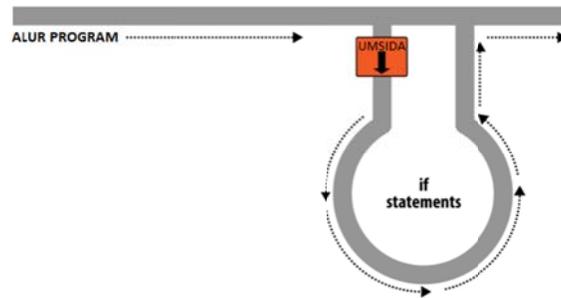
dihasilkan berupa Fatal Error. Dengan demikian statement selanjutnya tidak akan dijalankan. Disarankan agar Anda menggunakan require() dengan harapan bahwa kode PHP yang Anda buat tidak akan diproses apabila terdapat file yang hilang atau tidak ada.

### **8.3. Statement Kontrol**

Setiap skrip PHP dibuat dari serangkaian pernyataan. Pernyataan dapat berupa penugasan, pemanggilan fungsi, perulangan, pernyataan bersyarat atau bahkan pernyataan yang tidak melakukan apa-apa (pernyataan kosong). Pernyataan biasanya diakhiri dengan tanda titik koma. Selain itu, pernyataan dapat dikelompokkan menjadi kelompok pernyataan dengan merangkum sekelompok pernyataan dengan kurung kurawal. Kelompok pernyataan adalah pernyataan dengan sendirinya juga. Berbagai jenis pernyataan dijelaskan dalam bab ini.

Statement kontrol adalah pernyataan yang mempunyai kemampuan untuk mengubah alur dari jalannya suatu pemrograman seperti mengulangi eksekusi dari beberapa pernyataan, dan melewati suatu kondisi. Salah satu cara berpikir tentang aliran program adalah dengan membayangkannya sebagai jalan raya satu jalur yang Anda lewati. Itu cukup garis lurus, tetapi sekarang dan kemudian Anda menemukan berbagai tanda yang memberi tahu Anda ke mana harus pergi. Dalam hal pernyataan if, Anda bisa membayangkan menemukan tanda jalan memutar yang harus Anda ikuti jika kondisi tertentu BENAR.

Jika demikian, Anda pergi dan ikuti jalan memutar sampai kembali ke tempat dimulainya dan kemudian melanjutkan perjalanan ke arah semula. Atau, jika kondisinya tidak BENAR, Anda mengabaikan jalan memutar dan terus mengemudi (lihat Gambar).



Gambar Alur program seperti jalan raya satu jalur

### 8.3.1. Statement IF

Sintaks dari statement if dalam PHP adalah sebagai berikut:

```

if (syarat) {
    statement1;
    statement2;
    .
    .
}

```

Untuk menyatakan syarat, biasanya digunakan operator perbandingan seperti yang telah dibahas sebelumnya. Apabila syarat bernilai TRUE maka statement-statement yang diapit dengan tanda kurung kurawal akan dijalankan. Bentuk lain dari sintaks IF adalah

```

if (syarat) {
    statement1;
    statement2;
} else {
    statement3;
    statement4;
}

```

Untuk sintaks kedua di atas, statement3, statement4, dst akan dijalankan apabila syarat bernilai FALSE

### Contoh

```
<?php
    if ($bank_balance < 100000)
    {
        $money          = 1000000;
        $bank_balance += $money;
    }
?>
```

Dalam contoh ini, Anda memeriksa saldo Anda untuk melihat apakah kurang dari 100 ribu. Jika demikian, Anda membayar sendiri Rp. 1.000.000 dan kemudian menambahkannya ke saldo. (Andai saja menghasilkan uang sesederhana itu! ☺). Jika saldo bank adalah 100 ribu rupiah atau lebih besar, pernyataan kondisi diabaikan dan aliran program melompati ke baris berikutnya (tidak ditampilkan).

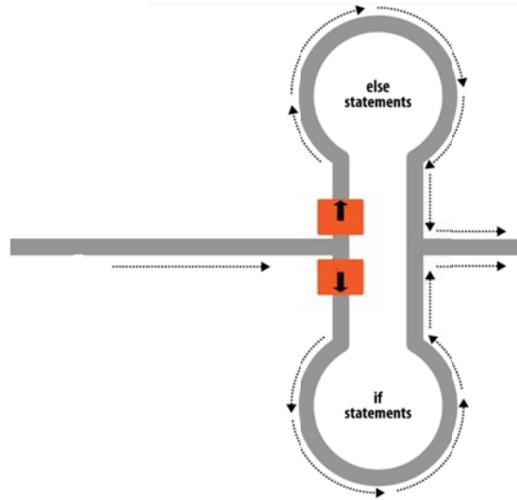
Dalam buku ini, membuka kurung kurawal umumnya dimulai pada baris baru. Beberapa orang suka menempatkan kurung kurawal pertama di sebelah kanan ekspresi kondisional; yang lain memulai baris baru dengannya. Salah satu dari ini baik-baik saja, karena PHP memungkinkan Anda untuk menetapkan karakter spasi putih (spasi, baris baru, dan tab) dengan cara apa pun yang Anda pilih. Namun, Anda akan menemukan kode Anda lebih mudah dibaca dan di-debug jika Anda membuat indentasi setiap tingkat dengan sebuah tab.

### 8.3.2. Statement Else

Terkadang ketika kondisional tidak BENAR, Anda mungkin tidak ingin langsung melanjutkan ke kode program utama tetapi mungkin ingin melakukan sesuatu yang lain sebagai gantinya. Di

sinilah pernyataan lain masuk. Dengan itu, Anda dapat mengatur jalan memutar kedua di jalan raya Anda, seperti pada Gambar.

Dengan pernyataan `if ... else`, pernyataan kondisional pertama dijalankan jika kondisinya benar. Tetapi jika itu SALAH, yang kedua dieksekusi. Salah satu dari dua pilihan harus dijalankan. Dalam keadaan apa pun tidak dapat (atau tidak keduanya) dieksekusi. Contoh menunjukkan penggunaan struktur `if ... else`.



Gambar Alur program seperti jalan raya dua jalur

Contoh Pernyataan `if ... else` dengan kurung kurawal

```
<?php
    if ($bank_balance < 100000)
    {
        $money = 1000000;
        $bank_balance += $money;
    }else{
        $savings += 50000;
        $bank_balance -= 50000;
    }
?>
```

Dalam contoh ini, sekarang setelah Anda memastikan bahwa Anda memiliki Rp. 100.000 atau lebih di bank, pernyataan lain dijalankan, di mana Anda menempatkan sebagian uang ini ke dalam rekening tabungan Anda.

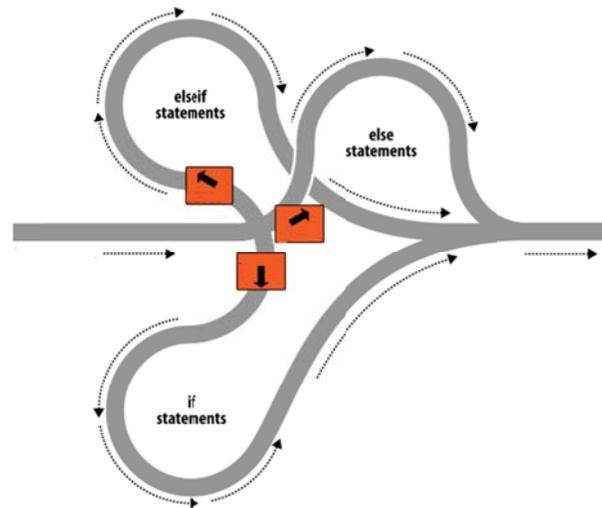
### 8.3.3. Statement Elseif

Ada juga saat-saat ketika Anda menginginkan sejumlah kemungkinan yang berbeda terjadi, berdasarkan urutan kondisi. Anda dapat mencapai ini menggunakan pernyataan elseif. Seperti yang mungkin Anda bayangkan, itu seperti pernyataan lain, kecuali bahwa Anda menempatkan ekspresi kondisional lebih lanjut sebelum kode kondisional. Dalam Contoh , Anda dapat melihat lengkap konstruksi if ... elseif ... else. Contoh Pernyataan if ... elseif ... else sebagai berikut :

```
<?php
    if ($bank_balance < 100000)
    {
        $money = 1000000;
        $bank_balance += $money;
    }elseif ($bank_balance > 200000)
    {
        $savings += 100000;
        $bank_balance -= 100000;
    }else{
        $savings += 50000;
        $bank_balance -= 50000;
    }
?>
```

Dalam contoh tersebut, pernyataan elseif telah dimasukkan antara pernyataan if dan else. Ini memeriksa apakah saldo bank Anda melebihi Rp.200.000 dan, jika demikian, memutuskan bahwa Anda mampu menyimpan Rp. 100.000 bulan ini.

Meskipun buku ini mulai menjelaskan agak terlalu rumit, Anda dapat membayangkan ini sebagai serangkaian jalan memutar (lihat Gambar).



Gambar analogi else if

#### 8.3.4. Statement SWITCH

Pernyataan switch berguna di mana satu variabel, atau hasil ekspresi, dapat memiliki beberapa nilai, yang masing-masing harus memicu aktivitas yang berbeda. Berikut konstruksi dari statement switch:

```
switch (variabel)
{
    case option1:
        statement11;
        statement12;
        .
        .
        break;
    case option2:
```

```

        statement21;
        statement22;
        .
        break;
    .
default:
    statementdefault1;
    statementdefault2;
    .
    break;
}

```

Pada sintaks di atas, nilai dari variabel akan dicek pada setiap option yang ada (terletak di bagian case). Jika ada option yang sama dengan nilai variabel, maka statement-statement di bawah option tersebutlah yang akan dijalankan. Bagian default adalah optional (boleh ada, boleh tidak). Contoh penerapan dalam skrip PHP adalah sebagai berikut :

```

<?php
    switch ($page)
    {
        case "Home":
            echo "You selected Home";
            break;
        case "About":
            echo "You selected About";
            break;
        case "News":
            echo "You selected News";
            break;
        case "Login":
            echo "You selected Login";
            break;
        case "Links":
            echo "You selected Links";
            break;
    }
}
?>

```

## Contoh

```
<?php
    switch ($page)
    {
        case "About":
            echo "You selected About";
            break;
        case "News":
            echo "You selected News";
            break;
        case "Login":
            echo "You selected Login";
            break;
        case "Links":
            echo "You selected Links";
            break;
        default:
            echo "You selected Home";
            break;
    }
?>
```

### 8.4. LOOPING/PERULANGAN

Salah satu hal hebat tentang komputer adalah dapat mengulangi tugas penghitungan dengan cepat dan tanpa capek. Seringkali Anda mungkin ingin sebuah program mengulang urutan kode yang sama berulang kali sampai sesuatu terjadi, seperti pengguna memasukkan nilai atau mencapai akhir yang alami. Struktur loop PHP menyediakan cara sempurna untuk melakukan ini.

#### 8.4.1. Statement WHILE

Statement ini digunakan untuk mengerjakan suatu statement secara berulang-ulang sampai suatu syarat dipenuhi. Sintaksnya adalah sebagai berikut :

```
while (syarat)
{
    statement;
    statement;
}
```

Pada sintaks di atas, selama syarat bernilai TRUE maka statement-statement di dalam while akan terus dijalankan secara berulang-ulang. Perulangan baru akan berhenti apabila syarat bernilai FALSE. Sebelum statement yang diulang-ulang dilakukan, terlebih dahulu akan dicek syarat nya apakah bernilai TRUE atau FALSE. Apabila TRUE maka statement akan dijalankan. Sedangkan apabila FALSE, perulangan akan langsung berhenti. Dengan kata lain, statement dalam WHILE bisa jadi tidak akan pernah dilakukan, yaitu apabila syaratnya langsung bernilai FALSE. Contoh penerapan dalam skrip :

```
<?php
    $fuel = 10;
    while ($fuel > 1)
    {
        // Keep driving ...
        echo "There's enough fuel";
    }
?>
```

Jika Anda mencoba contoh ini sendiri, perhatikan bahwa itu akan tetap mencetak string sampai Anda mengklik tombol Stop di browser Anda. Berbeda dengan contoh berikut:

```
<?php
    $fuel = 10;
    while ($fuel > 1)
    {
        // Keep driving ...
        echo "There's enough fuel";
        $fuel -=1;
    }
?>
```

Nah dengan menambahkan perintah decrement `$fuel -=1` maka program akan mencetak string sebanyak 9(sembilan) kali atau selama `$fuel` bernilai lebih dari 1, pada setiap looping nilai `$fuel` akan dikurangi 1.

#### 8.4.2. Statement FOR

Pengulangan dengan `for` adalah jenis pengulangan yang paling baik, karena menggabungkan kemampuan untuk mengatur variabel saat Anda memasukkan loop, menguji kondisi saat mengulang loop, dan memodifikasi variabel setelah setiap iterasi. Statement `FOR` mirip dengan `WHILE` yang memiliki sintaks berikut ini

```
for (inisialisasi counter; syarat;
increment/decrement counter)
{
    statement;
    .
    .
}
```

## Contoh

```
<?php
$harga_sikat = 1500;

for ($jumlah_sikat = 10; $jumlah_sikat <= 100;
    $jumlah_sikat+=10)
{
    echo "jumlah sikat".$jumlah_sikat." ";
    echo "Harga Sikat Rp. ".$harga_sikat *
    $jumlah_sikat."<br>";
}
?>
```

### 8.4.3. Statement Foreach

Misalkan suatu saat Anda mempunyai data berupa array asosiatif yang akan diproses secara berulang-ulang, maka PHP menyediakan statement foreach yang mudah digunakan.

Sintaksnya adalah:

```
foreach(variabelarray as kunci => value)
{
    statement;
    .
    .
    .
    .
    .
}
```

Contoh penerapan :

```
<?php
$links=array("www.umsida.ac.id","fst.umsida.ac.
id","informatika.umsida.ac.id");
echo "Alamat Web<br>";
foreach($links as $link) {
    echo "{$link}<br>";
}
?>
```

Variasi kedua sangat cocok untuk bekerja dengan kunci dan nilai array. Sintaksnya berikut:

```
foreach (array_expr as $key => $value)
{
    statement
}
```

Merevisi contoh sebelumnya, misalkan \$links adalah array asosiatif yang berisi tautan dan judul tautan yang sesuai:

```
$links = array("Web UMSIDA" => "www.umsida.ac.id",
              "Web Saintek" => "fst.umsida.ac.id",
              "Web Informatika" => "informatika.umsida.ac.id");
```

Setiap item array terdiri dari kunci dan nilai yang sesuai. Pernyataan foreach dapat dengan mudah mengupas setiap pasangan kunci / nilai dari array, seperti ini:

```
echo "Alamat Web<br>";
foreach($links as $title => $link) {
    echo "<a href=\"http://{ $link}\">{$title}</a>
    <br>";
}
```

## 8.5. FUNCTION

*Function* memisahkan bagian kode yang melakukan tugas tertentu. Misalnya, mungkin Anda sering perlu mencari tanggal dan mengembalikannya dalam format tertentu. Itu akan menjadi contoh yang baik untuk berubah menjadi sebuah *function*. Kode yang melakukannya mungkin hanya terdiri dari tiga baris, tetapi jika Anda harus menempelkannya ke dalam program Anda ribuan kali, Anda membuat program Anda menjadi besar dan rumit, kecuali jika Anda menggunakan suatu *function*. Dan jika Anda memutuskan untuk mengubah format data nanti, menempatkannya dalam suatu fungsi berarti harus mengubahnya hanya di satu tempat saja.

Menempatkan kode ke dalam fungsi tidak hanya mempersingkat program Anda dan membuatnya lebih mudah dibaca, tetapi juga menambahkan fungsionalitas, karena fungsi dapat melewati parameter untuk membuatnya bekerja secara berbeda. Mereka juga dapat mengembalikan nilai ke kode panggilan. Berikut contoh pembuatan sebuah *function*.

```
<?php
    function longdate($timestamp)
    {
        return date("l F jS Y",
            $timestamp);
    }
?>
```

Fungsi ini mengembalikan tanggal dalam format Minggu 30 Juni 2019. Sejumlah parameter dapat dilewatkan di antara tanda kurung awal; kami telah memilih untuk menerima hanya satu parameter. Dalam kurung kurawal melampirkan semua kode yang dieksekusi ketika Anda kemudian memanggil fungsi tersebut. Perhatikan bahwa huruf pertama dalam panggilan fungsi tanggal pada contoh sebelumnya adalah huruf kecil L, jangan disamakan dengan angka 1 (satu). Untuk menampilkan tanggal hari ini menggunakan fungsi ini, masukkan sintaks pemanggil berikut dalam kode Anda:

```
echo longdate (time ());
```

### **8.5.1. Lingkup Variabel**

Jika Anda memiliki program yang sangat panjang, sangat mungkin Anda dapat mulai kebingungan memilih nama variabel yang tepat, tetapi dengan PHP Anda dapat memutuskan ruang lingkup suatu variabel. Dengan kata lain, Anda dapat, misalnya, mengatakan bahwa Anda ingin variabel \$temp hanya digunakan di dalam fungsi tertentu. Ini adalah ruang lingkup default untuk variabel PHP. Sebagai alternatif, didalam PHP suatu variabel dapat memiliki cakupan global dan karenanya dapat diakses oleh setiap bagian lain dari program Anda.

#### **8.5.1.1. Variabel Lokal**

Variabel lokal adalah variabel yang dibuat di dalam, dan hanya dapat diakses oleh, suatu fungsi. Mereka umumnya adalah variabel sementara yang digunakan untuk menyimpan hasil yang diproses sebagian sebelum pengembalian fungsi. Satu set variabel lokal adalah daftar argumen untuk suatu fungsi. Di bagian sebelumnya, kami mendefinisikan fungsi yang menerima parameter bernama \$timestamp. Ini bermakna hanya di tubuh fungsi; Anda tidak bisa mendapatkan atau mengatur nilainya di luar fungsi.

#### **8.5.1.2. Variabel Global**

Adakalanya menemukan kasus ketika Anda membutuhkan variabel untuk memiliki cakupan global, karena Anda ingin semua kode Anda dapat mengaksesnya. Selain itu, beberapa data mungkin besar dan kompleks, dan Anda tidak ingin meneruskannya sebagai argumen untuk fungsi. Untuk mengakses variabel dari lingkup global, tambahkan kata kunci global.

Mari kita asumsikan bahwa Anda memiliki cara untuk login pengguna Anda ke situs web Anda dan ingin semua kode Anda tahu apakah itu berinteraksi dengan pengguna yang masuk atau tamu. Salah satu cara untuk melakukan ini adalah menggunakan kata kunci global sebelum variabel seperti `$is_logged_in`:

```
global $is_logged_in;
```

Sekarang fungsi login Anda hanya perlu mengatur variabel ke 1 pada login yang berhasil, atau 0 pada login yang gagal. Karena cakupan variabel diatur ke global, setiap baris kode dalam program Anda dapat mengaksesnya.

Anda harus menggunakan variabel yang diberikan akses global dengan hati-hati. Sangat disarankan Anda membuatnya hanya ketika Anda benar-benar tidak dapat menemukan cara lain untuk mencapai hasil yang Anda inginkan. Secara umum, program-program yang dipecah menjadi bagian-bagian kecil dan data terpilah lebih tidak bermasalah dan lebih mudah dipelihara. Jika Anda memiliki program seribu baris di mana Anda menemukan bahwa variabel global memiliki nilai yang salah di beberapa titik, berapa lama Anda akan menemukan kode yang salah?

Juga, jika Anda memiliki terlalu banyak variabel dengan cakupan global, Anda berisiko menggunakan salah satu dari nama-nama itu lagi secara lokal, atau setidaknya berpikir Anda telah menggunakannya secara lokal, padahal sebenarnya telah dinyatakan sebagai global. Segala macam bug aneh dapat muncul dari situasi seperti itu.

## **8.6. OPERASI FILE**

Pada bab ini akan dipaparkan bagaimana menggunakan perintah PHP untuk melakukan operasi file mulai dari proses membuka dan menutup file. Setelah itu dilanjutkan dengan proses membaca, menulis, menambah isi, menghapus dan meng-upload file

### 8.6.1. Membuka File

Secara umum terdapat 3 cara membuka file, yaitu membuka file hanya untuk dibaca (read: 'r'), hanya untuk ditulis baru (write: 'w'), dan hanya untuk ditambahi isinya (append: 'a'). Selain 3 cara membuka file tersebut, terdapat pula cara lain membuka file yaitu dapat dibaca dan ditulis (read/write: 'r+'), serta dapat dibaca dan ditambahi isinya (append: 'a+'). Berikut ini adalah contoh kode PHP untuk membuka file :

```
<?php
$nama_file = "test.txt";
$fh = fopen($nama_file, 'X') or die("File tidak
bisa dibuka");
fclose($fh);
?>
```

dengan 'X' dapat diganti dengan 'w', 'r', 'a', 'r+', 'a+'.

Apabila nama file yang akan dibuka ternyata salah, atau letak filenya yang tidak tepat, maka function die() yang akan dijalankan. Function die() akan menampilkan teks sebagai peringatan apabila proses membuka file gagal.

### 8.6.2. Menutup File

Setelah file dibuka, hendaknya file tersebut juga ditutup ketika pemrosesan selesai. File yang yang tidak ditutup kemungkinan dapat terjadi kerusakan pada strukturnya. Berikut ini adalah contoh kode PHP untuk menutup file yang telah dibuka

```
$NamaFile = "testFile.txt";
$FileHandle = fopen($NamaFile, 'w') or
die("File tidak bisa dibuka");
fclose($FileHandle);
```

File yang telah ditutup tidak bisa untuk dibaca, ditulis, dan ditambah. Untuk bisa melakukan hal itu kembali, file terlebih dahulu harus dibuka lagi seperti sebelumnya.

### 8.6.3. Menulis ke File

File yang telah dibuka dapat ditulis dengan data di dalamnya. Berikut ini adalah contoh kode PHP untuk menulis suatu string ke dalam file.

```
<?php
$FileKu = "testFile.txt";

$FileHandle = fopen($Fileku, 'w') or die("File
gagal dibuka");
$DataString = "Hallo Amalia... \n";
fwrite($FileHandle, $DataString);
$DataString = "Hallo Faza dan Nada... \n";
fwrite($FileHandle, $DataString);
fclose($FileHandle);
?>
```

Apabila Anda menggunakan mode 'w' pada fopen(), dan selanjutnya Anda menuliskan data pada file, maka isi file yang lama (jika sebelumnya terdapat isi pada file tersebut) akan terhapus dan diganti dengan isi yang baru. Sedangkan apabila Anda menginginkan data yang lama pada suatu file tidak dihapus, maka Anda gunakan mode append 'a+' atau 'a'.

### 8.6.4. Membaca Isi File

Misalkan kita memiliki file dengan nama test.txt yang isinya adalah sbb:

Selamat berjumpa lagi kawan!! Senang bertemu Anda.

Kita dapat menggunakan kode PHP untuk membaca file tersebut dan selanjutnya hasil pembacaan dapat ditampilkan di browser atau diproses lebih lanjut. Berikut ini adalah contoh kode pembacaannya.

```
<?php
$FileKu = "test.txt";
$FileHandle = fopen($FileKu, 'r');
$Data = fread($FileHandle, 5);
fclose($FileHandle);
echo $Data;
?>
```

Apabila kode di atas dijalankan, maka pada browser hanya akan menampilkan teks

Selam

Hal ini dikarenakan adanya nilai 5 pada fread(). Nilai tersebut menyatakan jumlah karakter awal yang dibaca dari file. Supaya seluruh isi file dapat dibaca, gunakan function filesize().

```
<?php
$FileKu = "test.txt";
$FileHandle = fopen($FileKu, 'r');
$Data = fread($FileHandle, filesize($FileKu));
fclose($FileHandle);
echo $Data;
?>
```

Setelah Anda lihat di browser, ternyata ganti baris pada isi file test.txt diabaikan. Meskipun teks dalam file tersebut ditulis dalam 2 baris, oleh perintah fread() akan dibaca dalam satu baris. Hal ini dikarenakan dalam file test.txt tidak terdapat tag html <br> untuk pindah baris.

### 8.6.5. Menghapus File

Dalam PHP, function untuk menghapus file adalah unlink(). Berikut ini contohnya.

```
<?php
  $myFile = "testFile.txt";
  unlink($myFile);
?>
```

### 8.6.6. Menambah Isi File

Maksud dari menambah isi file di sini adalah, menambah data baru pada file (diasumsikan data sudah ada sebelumnya). Untuk menambah isi file dalam PHP, mode pembukaan file nya menggunakan 'a' atau 'a+' dan fwrite() untuk menulis data ke dalam file. Berikut ini adalah contohnya.

Misalkan kita mempunyai file buah.txt yang isinya adalah sbb:

Jeruk  
Apel  
Mangga

Selanjutnya kita punya kode PHP sbb:

```
<?php
  $myFile = "buah.txt";
  $fh = fopen($myFile, 'a') or die("File tidak
  bisa dibuka");
  $buah1 = "Anggur\n";
  fwrite($fh, $buah1);
  $buah2 = "Nanas\n";
  fwrite($fh, $buah2);
  fclose($fh);
?>
```

Kode di atas akan menambahkan 2 buah data baru ke file buah.txt. Perintah \n (new line) digunakan untuk ganti baris pada file.

## 8.7. BEKERJA DENGAN COOKIES

Cookies telah lama digunakan dalam internet yang digunakan oleh administrator web untuk menyimpan informasi user atau pengunjung di komputer user tersebut.

### 8.7.1. Membuat Cookies

Untuk membuat cookies, caranya dengan menggunakan perintah `setcookie(name, value, expiration)`. Perintah ini memiliki 3 buah argumen, yaitu :

1. **name**, merupakan nama dari cookie. Nama cookie ini dapat dipanggil sewaktu-waktu untuk mendapatkan informasi. Jadi jangan sampai nama cookie ini lupa.
2. **value**, merupakan informasi atau data yang disimpan dalam cookie. Biasanya value ini berupa username atau tanggal pengaksesan suatu halaman web.
3. **expiration**, merupakan batas waktu penyimpanan cookie (dalam detik timestamp). Apabila lama penyimpanan sebuah cookie melebihi batas waktu ini, maka secara otomatis cookie tersebut akan terhapus.

Contoh:

Pada contoh ini akan dibuat perintah PHP untuk membuat cookie yang digunakan untuk menyimpan data waktu kunjungan terakhir seorang user yang mengakses suatu halaman web. Cookie ini diberi batas waktu sampai 2 bulan (60 hari) penyimpanan.

```
<?php
$duabulanlagi = time() + 60 * 24 * 3600;
setcookie(KunjunganTerakhir, date("G:i -
m/d/y"), $duabulanlagi);
?>
```

### 8.7.2. Mengambil Informasi dari Cookie

Apabila cookie belum terhapus, maka kita dapat mengambil informasi dari cookie. Untuk mengecek apakah suatu cookie sudah terhapus atau belum menggunakan perintah :

```
isset($_COOKIE['nama_cookie'])
```

Apabila cookie masih ada, maka perintah di atas menghasilkan TRUE. Sedangkan apabila sudah terhapus, akan menghasilkan nilai FALSE.

Sedangkan untuk mengambil informasi dari cookie, menggunakan perintah

```
$_COOKIE['nama_cookie']
```

Contoh:

Berikut ini contoh perintah PHP untuk menampilkan tanggal kunjungan terakhir user yang mengunjungi halaman web.

```
<?php
if(isset($_COOKIE['KunjunganTerakhir']))
{
    $visit = $_COOKIE['KunjunganTerakhir'];
    echo "Kunjungan Anda terakhir pada - ".
    $visit;
```

```
}
else
    echo "Anda sudah 2 bulan lebih tidak
    mengunjungi web ini";

?>
```

## 8.8. BEKERJA DENGAN SESSION

Session digunakan untuk mengirim data ke beberapa halaman web. Sebuah halaman web, secara normal tidak akan mengirim suatu data dari halaman yang satu ke halaman yang lain. Dengan kata lain, semua informasi tentang data tersebut akan hilang begitu halaman web direload. Berikut ini akan diberikan contoh tentang penjelasan di atas, mengenai perlunya menggunakan session.

Contoh:

### Form.php

```
<form action="submit.php" method="post">
Username <input type="text" name="username">
<input      type="submit"      name="submit"
value="Submit">
</form>
```

### Submit.php

```
<?php
$username = $_POST['username'];
echo "Nama user Anda adalah: $username<br><br>";
echo "<a href=hal1.php>Hal 1</a> <a
href=hal2.php>Hal 2</a> <a href=hal3.php>Hal
3</a>";
?>
```

### Hal1.php

```
<?php
    echo "Ini adalah halaman 1<br>";
    echo "Nama user Anda adalah:
$username<br><br>";
    echo "<a href=hal1.php>Hal 1</a> <a
href=hal2.php>Hal 2</a> <a href=hal3.php>Hal
3</a>";
?>
```

### Hal2.php

```
<?php
echo "Ini adalah halaman 2<br>";
echo "Nama user Anda adalah: $username<br><br>";
echo " <a href=hal1.php>Hal 1</a> <a
href=hal2.php>Hal 2</a> <a href=hal3.php>Hal
3</a>";
?>
```

### Hal3.php

```
<?php
echo "Ini adalah halaman 3<br>";
echo "Nama user Anda adalah: $username<br><br>";
echo " <a href=hal1.php>Hal 1</a> <a
href=hal2.php>Hal 2</a> <a href=hal3.php>Hal
3</a>";
?>
```

Dengan file-file di atas, user diminta melakukan login terlebih dahulu (memasukkan nama user) melalui form.php. Selanjutnya akan muncul submit.php yang menampilkan nama user yang dimasukkan tadi dan beberapa link ke halaman lain. Begitu user mengakses 3 buah halaman web yang ada tersebut, diharapkan nama user yang telah dimasukkan dalam form sebelumnya tetap ditampilkan dalam setiap halaman web yang diakses. Akan tetapi

apa yang terjadi? Ternyata untuk ketiga halaman web yang diakses tersebut tidak menampilkan nama user. Hal ini dikarenakan hilangnya data/informasi dari nama user yang dimasukkan sebelumnya.

Nah... di sinilah kita perlu adanya session. Dengan session, data dapat disimpan dan selanjutnya dapat diakses di beberapa halaman web. Data yang tersimpan dalam session bersifat temporary/ sementara. Biasanya akan terhapus secara otomatis begitu user menutup browser, atau melakukan logout.

### **8.8.1. Memulai PHP - Session**

Sebelum Anda menyimpan data dalam session, terlebih dahulu harus memulai session. Untuk memulai session, perintahnya adalah:

```
session_start()
```

### **8.8.2. Menyimpan Data ke dalam Session**

Untuk menyimpan data ke dalam session, digunakan perintah

```
$_SESSION['nama_session'] = data;
```

Contoh:

Berikut ini adalah code dalam file submit.php (contoh sebelumnya) yang telah dimodifikasi. Nama user akan disimpan dalam session.

```
<?php
session_start();
$username = $_POST['username'];
$_SESSION['namauser'] = $username;
echo "Nama user Anda adalah: $username<br><br>";
```

```
echo "<a href=hal1.php>Hal 1</a> <a  
href=hal2.php>Hal 2</a> <a href=hal3.php>Hal  
3</a>";  
?>
```

### 8.8.3. Memanggil Data yang Tersimpan dalam Session

Setelah data disimpan dalam session, selanjutnya dapat dipanggil kembali apabila diperlukan. Untuk memanggil data dalam session, caranya cukup menuliskan

```
$_SESSION['nama_session'].
```

Contoh :

Berikut ini adalah code dari file hal1.php, hal2.php, dan hal3.php sehingga dapat menampilkan nama user yang telah disimpan dalam session.

#### Hal1.php

```
<?php  
session_start();  
echo "Ini adalah halaman 1<br>";  
echo "Nama user Anda adalah:  
".$_SESSION['namauser']. "<br><br>";  
echo "<a href=hal1.php>Hal 1</a> <a  
href=hal2.php>Hal 2</a> <a href=hal3.php>Hal  
3</a>";  
?>
```

## Hal2.php

```
<?php
session_start();
echo "Ini adalah halaman 2<br>";
echo "Nama user Anda adalah:
".$_SESSION['namauser']. "<br><br>";
echo "<a href=hal1.php>Hal 1</a> <a
href=hal2.php>Hal 2</a> <a href=hal3.php>Hal
3</a>";
?>
```

## Hal3.php

```
<?php
session_start();
echo "Ini adalah halaman 3<br>";
echo "Nama user Anda adalah:
".$_SESSION['namauser']. "<br><br>";
echo "<a href=hal1.php>Hal 1</a> <a
href=hal2.php>Hal 2</a> <a href=hal3.php>Hal
3</a>";
?>
```

### 8.8.4. Menghapus Data Session

Untuk menghapus data yang tersimpan dalam suatu session, digunakan perintah :

```
unset($_SESSION['nama_session']);
```

Misalkan terdapat lebih dari satu session dan kita ingin menghapus semuanya, maka caranya dengan memberikan perintah :

```
session_destroy();
```

Biasanya menghapus data session yang terkait dengan username diterapkan pada proses logout. Dengan proses logout, maka username yang telah tersimpan dalam session akan terhapus.

Contoh:

Akan dibuat script untuk proses logout dari kasus di atas.

```
<?php
  session_start();
  unset($_SESSION[ 'namauser' ] );
?>
```

## 8.9. BEKERJA DENGAN FORM

Setelah kita belajar dasar-dasar perintah PHP, sekarang saatnya mengaplikasikannya pada aplikasi web. Biasanya PHP digunakan sebagai pengolah data yang diinputkan melalui form yang dibuat dengan HTML. Sebagai contoh, andaikan Anda memiliki toko virtual dalam web yang menjual alat-alat tulis seperti pensil, buku tulis, dan ballpoint. Berikut ini adalah salah satu bentuk kode HTML yang digunakan untuk membuat form pemesanan pembelian barang-barang tersebut.

```
<html>
<body>
<h2>Toko Alat Tulis Amalia</h2>
<form action="proses.php" method="post">
<select name="barangpesanan">
<option>Pensil</option>
<option>Buku Tulis</option>
<option>Ballpoint</option>
```

```
</select>
Jumlah pesanan: <input name="jumlah"
type="text">
<input type="submit" value="Submit">
</form>
</body>
</html>
```

Seperti yang Anda lihat pada kode HTML di atas, perintah `action="proses.php"` digunakan untuk mengarahkan ke file PHP yang digunakan untuk memproses barang pembelian ketika tombol submit ditekan. Dalam form yang dihasilkan dari kode di atas terdapat 2 buah komponen input yaitu berbentuk combobox dan textbox. Untuk combobox, diberi nama "barangpesanan" (perhatikan perintah `<select name="barangpesanan">`) dan textbox diberi nama "jumlah" (perhatikan perintah `<input name="jumlah" type="text">`).

Penjelasan mengenai `method="post"` akan dijelaskan pada sub bab berikutnya. Sedangkan berikut ini adalah salah satu contoh kode PHP untuk memproses input dari form di atas. Kode PHP ini disimpan dengan nama `proses.php`.

```
<html>
<body>
<?php
$jumlah = $_POST['jumlah'];
$barangpesanan = $_POST['barangpesanan'];

echo "Anda memesan ". $jumlah . " " .
$barangpesanan . ".<br>";
echo "Terima kasih atas kesediaan Anda memesan
barang dari kami!";
?>
</body>
</html>
```

Kalau Anda perhatikan, terdapat keterkaitan perintah `$_POST['xxx']`; dengan "xxx" pada `name = "xxx"` (nama komponen input).

## 8.10. Method Post dan Get

Pada contoh sebelumnya, kita mengirim data input dari form menuju ke file PHP untuk diproses menggunakan metode post. Selain metode tersebut, terdapat pula metode get. Lantas perbedaannya apa? Kapan kita gunakan metode post atau get? Itulah yang akan dibahas pada bagian ini. Untuk melihat perbedaan post dan get, kita akan sedikit mengubah file HTML form dan file PHP `proses.php` sebelumnya.

```
<html>
<body>
<h2>Toko Alat Tulis Amalia</h2>
<form action="proses.php" method="get">
<select name="barangpesanan">
<option>Pensil</option>
<option>Buku Tulis</option>
<option>Ballpoint</option>
</select>
Jumlah pesanan: <input name="jumlah"
type="text">
<input type="submit" value="Submit">
</form>
</body>
</html>
```

Dan pada file `proses.php` sebagai berikut:

```
<html>
<body>
<?php
$jumlah = $_GET['jumlah'];
$barangpesanan = $_GET['barangpesanan'];
```

```
echo "Anda memesan ". $jumlah . " " .  
$barangpesanan . ".<br>";  
echo "Terima kasih atas kesediaan Anda memesan  
barang dari kami!";  
>  
</body>  
</html>
```

Perbedaan kode HTML dan PHP di atas dengan sebelumnya adalah yang dicetak merah. Metode pengiriman data input dari form menggunakan get, dan dalam proses.php \$\_POST diganti dengan \$\_GET. Apabila aplikasi di atas dijalankan, maka secara sekilas hasil yang tampak sama dengan ketika digunakan metode post. Namun, coba perhatikan URL yang tampak ketika proses.php muncul. Pada URL tersebut terdapat tambahan ?barangpesanan=...&jumlah=... setelah nama file (proses.php). Titik-titik tersebut akan diisi dengan data sesuai dengan yang diinputkan pada form.

Coba bandingkan dengan URL ketika digunakan metode post. Data isian pada form tidak ditampilkan pada URL. Sehingga inilah perbedaan antara keduanya. Dengan demikian, hendaknya kita jangan menggunakan metode get ketika akan memproses data input melalui form. Bayangkan seandainya form tersebut digunakan untuk login atau untuk keperluan yang menyangkut privasi. Apabila Anda gunakan metode get, maka semua input data akan ditampilkan pada URL. Bisa-bisa password Anda akan kelihatan di URL (jika terdapat input password ketika login). Untuk metode get, biasanya digunakan untuk input data melalui link (bukan melalui form).

## BAB 9

### PHP-MYSQL

MySQL adalah mesin/alat basis data relasional yang memungkinkan pengembang untuk menggunakan sesuatu yang disebut Structured Query Language (SQL) untuk berinteraksi dengan database. SQL dapat digunakan untuk melakukan dua jenis tugas. Tipe pertama adalah membuat(*create*), mengubah (*alter*) atau menghapus (*drop*) objek dalam database. Objeknya adalah *table*, *view*, *procedures*, *indexes*, dll. Jenis perintah kedua digunakan untuk berinteraksi dengan data dengan memilih (*select*), menyisipkan (*insert*), memperbarui (*update*), atau menghapus (*delete*) baris dalam tabel. Tabel dapat dibandingkan dengan spreadsheet dengan baris dan kolom. Setiap kolom memiliki nama, tipe data, dan panjang serta bendera lain yang menentukan bagaimana data ditangani.

Meskipun SQL digunakan oleh banyak sistem basis data yang berbeda, SQL tidak semua mengikuti sintaksis yang sama atau mendukung fitur yang sama; Namun, kebanyakan mengikuti standar yang disebut SQL92 dengan sejumlah fitur khusus. Salah satu contohnya adalah opsi yang diajukan MySQL yang disebut `AUTO_INCREMENT`. Ketika opsi ini diterapkan ke kolom bilangan bulat dalam tabel, database akan secara otomatis memberikan nilai ke kolom setiap kali baris ditambahkan ke tabel kecuali pernyataan insert memberikan nilai untuk kolom. Database lain menggunakan `DEFAULT UNIQUE` (FrontBase) atau `IDENTITY ()` (SQL Server). Database Oracle membutuhkan penciptaan urutan yang kemudian digunakan untuk membuat nilai unik pada sisipan. Perbedaan ini membuatnya sulit untuk menulis kode yang berjalan pada sistem basis data yang berbeda.

PHP telah mendukung MySQL hampir sejak awal proyek, termasuk API dengan rilis versi 2. Bahkan, menggunakan MySQL dengan PHP pada akhirnya menjadi hal yang lumrah sehingga selama beberapa tahun ekstensi `php_mysql` diaktifkan secara default. Tapi mungkin bukti yang paling indikatif dari ikatan kuat antara dua kubu teknologi adalah rilis ekstensi MySQL yang diperbarui dengan PHP 5, yang dikenal sebagai MySQL Improved (dan biasanya disebut sebagai `mysqli`). Hal ini dimaksudkan untuk kepentingan keamanan pada koneksi php ke mysql.

## 9.1. Koneksi PHP ke MySQL

Sebelum kita melakukan koneksi ke MySQL ada beberapa parameter yang harus kita ketahui terlebih dahulu. Untuk melakukan koneksi, dibutuhkan:

- Server name, merupakan nama server atau no. IP server dimana MySQL tersebut diinstall
- Username, merupakan nama user yang diberikan wewenang untuk mengakses database dalam MySQL
- Password, merupakan password yang dimiliki username dalam rangka autentifikasi.
- Database name, merupakan nama database dalam MySQL yang ingin kita akses.

Untuk memperoleh informasi parameter di atas, dapat menghubungi server administrator. Sedangkan perintah PHP untuk melakukan koneksi ke MySQL adalah :

```
<?php
mysql_connect("localhost", "username",
"password") or die(mysql_error());
mysql_select_db("test") or
die(mysql_error());
echo "Koneksi ke MySQL Sukses<br>";
?>
```

Perintah di atas akan menampilkan Koneksi ke MySQL sukses apabila koneksi telah berhasil, sedangkan apabila gagal akan menampilkan pesan kesalahan. Saat ini kita disarankan untuk menggunakan koneksi dengan fungsi MySQLi untuk tingkat keamanan yang lebih baik, selain itu lebih cepat dibandingkan dengan MySQL, jadi intinya di PHP 5 keatas sudah tidak disarankan lagi menggunakan **mysql** tapi sudah menggunakan **mysqli** yang dicontohkan sebagai berikut :

```
$koneksi = mysqli_connect('localhost',
username', 'password', 'database'); // koneksi
ke database
mysqli_close($koneksi); // memutus koneksi
```

## 9.2. Memberikan Query ke MySQL

Setelah kita memilih database dimana kita akan bekerja, selanjutnya kita dapat memberikan perintah query seperti SELECT, DELETE, CREATE, UPDATE. Berikut ini contoh script PHP untuk membuat tabel example dalam database test.

```

<?php
$koneksi = mysqli_connect('localhost',
'root', '', 'test');
If (mysqli_connect_errno())
{
    echo "Failed to connect to MySQL: " .
mysqli_connect_error();
}
$sql = "create table example(
        id INT AUTO_INCREMENT,name
        VARCHAR(20) NOT NULL,primary key
        (id))";
    if(mysqli_query($koneksi, $sql)){
        echo "Tabel berhasil dibuat";
    } else {
        echo "Tabel Gagal dibuat";
    }
    mysqli_close($koneksi);

?>

```

Pada contoh di atas, dapat dilihat bahwa perintah PHP untuk menuliskan query ke MySQL adalah sebagai berikut :

```

mysqli_query("query");
Sedangkan berikut ini contoh script untuk
menyisipkan 2 buah record/data ke tabel example.
<?php
$koneksi = mysqli_connect('localhost', 'root',
'', 'test');
If (mysqli_connect_errno())
{
    echo "Failed to connect to MySQL: " .
mysqli_connect_error();
}

```

```

$data1="INSERT INTO example(id,
name)VALUES(1, 'budiman')";
$data2="INSERT INTO example(id,
name)VALUES(2, 'surti')";
    if(mysqli_query($koneksi, $data1)
&& mysqli_query($koneksi, $data2)){
        echo "Data berhasil dimasukkan";
    } else {
        echo "Data gagal dimasukkan";
    }
    mysqli_close($koneksi);

?>

```

### 9.3. Mengambil Data dari MySQL

Mengambil data di sini terkait dengan penggunaan query SELECT. Berikut ini contoh untuk menampilkan record pertama dari tabel example.

```

<?php
$koneksi = mysqli_connect('localhost', 'root',
'', 'test');
If (mysqli_connect_errno())
{
    echo "Failed to connect to MySQL: " .
mysqli_connect_error();
}
$sql = 'SELECT * FROM example';
    $result = mysqli_query($koneksi, $sql);
    $row = mysqli_fetch_assoc($result);
    echo "Id: " . $row["id"]. "<br>";

```

```
        echo "Name: " . $row["name"]. "<br>";
    mysqli_close($koneksi);
?>
```

Output script di atas adalah:

Id: 1

Name: Budiman

Perintah di atas hanya akan menampilkan record pertama dari tabel example. Lantas, bagaimana caranya untuk menampilkan record yang lebih dari satu? Untuk menampilkan record yang lebih dari satu, kita gunakan looping. Perhatikan contoh berikut ini.

```
<?php
$koneksi = mysqli_connect('localhost', 'root',
'', 'test');
If (mysqli_connect_errno())
{
    echo "Failed to connect to MySQL: " .
mysqli_connect_error();
}
$sql = 'SELECT * FROM example';
$result = mysqli_query($koneksi, $sql);

    if (mysqli_num_rows($result) > 0) {
        while($row =
mysqli_fetch_assoc($result)) {
            echo "Id: " . $row["id"].
"<br>";
            echo "Name: " . $row["name"]. "<br><br>";
        }
    }
}
```

```
} else {  
    echo "0 results";  
}  
mysqli_close($koneksi);  
  
?>
```

Looping while di atas akan terus berjalan selama record masih ada untuk dibaca. Hasil dari script di atas adalah

Id: 1  
Name: budiman

Id: 2  
Name: surti

## BAB 10

### MEMBUAT CRUD (Create, Read, Update, Delete)

CRUD adalah akronim dari Create, Read, Update, dan Delete. Operasi CRUD adalah manipulasi data dasar untuk database. Dalam buku ini kita akan membuat aplikasi PHP sederhana untuk melakukan semua operasi ini pada tabel database MySQL di satu tempat. Kita akan membuat sebuah aplikasi dengan tampilan seperti pada gambar di bawah dengan sebuah tabel yang menampilkan data dari database di sisi frontend. Terdapat beberapa tombol untuk mengubah data yang ada di database lewat antarmuka yang kita buat. Tabel pengguna yang kita buat akan berisi informasi pengguna seperti nama, nomor telepon, email, dll.

[Add New User](#)

| Name                    | Mobile     | Email                   | Update                                      |
|-------------------------|------------|-------------------------|---|
| Mochamad Alfian Rosid   | 0865247162 | alfanrosid@umsida.ac.id | <a href="#">Edit</a> <a href="#">Delete</a> |
| Ika Ratna Indra Astutik | 086532763  | ikaratna@umsida.ac.id   | <a href="#">Edit</a> <a href="#">Delete</a> |

#### Langkah-Langkah Pembuatan CRUD

Untuk membuat CRUD langkah pertama yang harus kita lakukan tentunya adalah membuat sebuah database dan tabel, untuk database kita akan menggunakan database yang telah kita buat sebelumnya yaitu tabel test, sekarang kita akan membuat sebuah tabel pada database test tersebut yaitu tabel users, gunakan phpmySQL untuk memudahkan pembuatan tabel tersebut.

```
CREATE TABLE `users` (  
  `id` int(11) NOT NULL auto_increment,  
  `name` varchar(100),  
  `email` varchar(100),  
  `mobile` varchar(15),  
  PRIMARY KEY (`id`)
```

);

Kemudian kita buat sebuah file untuk koneksi ke database, pada buku ini kami namakan file config.php

## 10.1. Membuat File Config

File config.php menyimpan informasi tentang database host, username dan password. Sebagian besar server lokal bekerja dengan detail yang diberikan. Anda dapat mengubahnya sesuai dengan detail host dan database Anda.

```
<?php
/**
 * using mysqli_connect for database connection
 */

$databaseHost = 'localhost';
$databaseName = 'test';
$databaseUsername = 'root';
$databasePassword = '';

$mysqli = mysqli_connect($databaseHost,
$databaseUsername, $databasePassword,
$databaseName);

?>
```

Untuk menghubungkan PHP dengan MySQL, kita menggunakan fungsi `mysqli_connect()` dengan alamat server sebagai parameter pertama, user database sebagai parameter kedua, password user sebagai parameter ketiga, dan nama database sebagai parameter keempat.

Silahkan isi password sesuai dengan yang dibuat di servernya. Jika tidak menggunakan password, maka isikan dengan String kosong saja.

## 10.2. Membuat File Index

Setelah file config berhasil dibuat langkah selanjutnya adalah membuat file index.php seperti berikut:

```
<?php
// Create database connection using config file
require("config.php");

// Fetch all users data from database
$result = mysqli_query($mysqli, "SELECT * FROM
users ORDER BY id DESC");
?>

<html>
<head>
    <title>Homepage</title>
</head>

<body>
<a href="add.php">Add New User</a><br/><br/>

    <table width='80%' border=1>

        <tr>
            <th>Name</th> <th>Mobile</th>
<th>Email</th> <th>Update</th>
        </tr>
        <?php
            while($user_data =
mysqli_fetch_array($result)) {
                echo "<tr>";
```

```

        echo "<td>".$user_data['name']. "</td>";
        echo
"<td>".$user_data['mobile']. "</td>";
        echo "<td>".$user_data['email']. "</td>";
        echo "<td><a
href='edit.php?id=$user_data[id]'">Edit</a> | <a
href='delete.php?id=$user_data[id]'">Delete</a></
td></tr>";
    }
    ?>
</table>
</body>
</html>

```

File index.php merupakan file utama yang menyertakan file konfigurasi untuk koneksi database. Kemudian menampilkan semua daftar pengguna menggunakan MySQL Select Query. Pengguna yang akan ditampilkan di dalam daftar perlu menambahkan terlebih dahulu menggunakan tautan 'Tambahkan Pengguna Baru'.

### 10.3. Membuat File ADD

Langkah selanjutnya adalah membuat file add.php yang akan digunakan untuk memasukkan data kedalam database, berikut adalah sintaks add.php:

```

<html>
<head>
    <title>Add Users</title>
</head>
<body>
    <a href="index.php">Go to Home</a>
    <br/><br/>
    <form action="add.php" method="post"
name="form1">
        <table width="25%" border="0">
            <tr>

```

```

<td>Name</td>
      <td><input type="text"
name="name"></td>
    </tr>
    <tr>
      <td>Email</td>
      <td><input type="text"
name="email"></td>
    </tr>
    <tr>
      <td>Mobile</td>
      <td><input type="text"
name="mobile"></td>
    </tr>
    <tr>
      <td></td>
      <td><input type="submit"
name="Submit" value="Add"></td>
    </tr>
  </table>
</form>

<?php

    // Check If form submitted, insert form data
into users table.
    if(isset($_POST['Submit'])) {
        $name = $_POST['name'];
        $email = $_POST['email'];
        $mobile = $_POST['mobile'];

        // include database connection file
        include_once("config.php");

        // Insert user data into table
        $result = mysqli_query($mysqli, "INSERT
INTO users(name,email,mobile)
VALUES('$name','$email','$mobile')");

        // Show message when user added

```

```
        echo "User added successfully. <a  
href='index.php'>View Users</a>";  
    }  
    ?>  
</body>  
</html>
```

Formulir HTML digunakan untuk menerima masukan data pengguna. Setelah data pengguna diserahkan, MySQL INSERT Query digunakan untuk memasukkan data pengguna ke dalam database.

## 10.4. Membuat File Edit

Langkah selanjutnya adalah membuat file edit.php, file ini digunakan untuk merubah data yang telah tersimpan didalam database. Untuk membuat file tersebut perhatikan sintaks berikut:

```
<?php  
// include database connection file  
require("config.php");  
  
// Check if form is submitted for user update,  
then redirect to homepage after update  
if(isset($_POST['update']))  
{  
    $id = $_POST['id'];  
    $name=$_POST['name'];  
    $mobile=$_POST['mobile'];  
    $email=$_POST['email'];  
    // update user data  
    $result = mysqli_query($mysqli, "UPDATE  
users SET  
name='$name',email='$email',mobile='$mobile'  
WHERE id=$id");
```

```

        // Redirect to homepage to display updated
user in list
        header("Location: index.php");
    }
?>
<?php
// Display selected user data based on id
// Getting id from url
$id = $_GET['id'];

// Fetech user data based on id
$result = mysqli_query($mysqli, "SELECT * FROM
users WHERE id=$id");

while($user_data = mysqli_fetch_array($result))
{
    $name = $user_data['name'];
    $email = $user_data['email'];
    $mobile = $user_data['mobile'];
}
?>
<html>
<head>
    <title>Edit User Data</title>
</head>

<body>
    <a href="index.php">Home</a>
    <br/><br/>

    <form name="update_user" method="post"
action="edit.php">
        <table border="0">
            <tr>
                <td>Name</td>
                <td><input type="text"
name="name" value=?php echo $name;?></td>
            </tr>
            <tr>

```

```

<td>Email</td>
      <td><input type="text"
name="email" value=<?php echo $email;?>></td>
    </tr>
    <tr>
      <td>Mobile</td>
      <td><input type="text"
name="mobile" value=<?php echo $mobile;?>></td>
    </tr>
    <tr>
      <td><input type="hidden"
name="id" value=<?php echo $_GET['id'];?>></td>
      <td><input type="submit"
name="update" value="Update"></td>
    </tr>
  </table>
</form>
</body>
</html>

```

File ini akan mengarahkan pengguna kembali ke homepage, setelah update sukses. Dengan perintah `header("Location: index.php");`

## 10.5. Membuat File Hapus

Dan yang terakhir adalah membuat file `delete.php` yang digunakan untuk menghapus data yang telah tersimpan dalam database.

```

<?php
// include database connection file
require("config.php");

// Get id from URL to delete that user
$id = $_GET['id'];

// Delete user row from table based on given id

```

|  |     |
|--|-----|
|  | ROM |
|  | est |

Setelah selesai membuat beberapa file tersebut silahkan coba untuk menjalankan di browser. Kami harap anda dapat melakukan perbaikan apabila terjadi error berdasarkan materi-materi yang telah dijelaskan pada buku ini.

**Contoh Penerapan di sistem informasi**



## DAFTAR PUSTAKA

- Abdulloh Rohi. 2018. *7 in 1 Pemrograman Wb untuk Pemula*. PT Elex Media Komputindo. Jakarta.
- Ariona Rian. 2018. *Belajar HTML dan CSS - Tutorial Fundamental dalam mempelajari HTML dan CSS*. Khasanah Buku Jogja. Yogyakarta.
- Frank M. Kromann. 2018. *Beginning PHP and MySQL\_ From Novice to Professional*. Apress.
- Gauchat, JD. *Introduction to HTML.Introduction to Web Development Using HTML 5*. Jones & Bartlett Publishers.Burlington, Massachusetts, Amerika
- Michaud Thomas. 2013. *Foundations of Web Design: Introduction to HTML & CSS*. New Riders. USA.
- Rerung Rante Rintho. 2018. *Pemrograman Web*. Deepublish Publisher. Yogyakarta.
- Robin Nixon. 2018. *Learning PHP, MySQL & JavaScript\_ With jQuery, CSS & HTML5 (Early Release 5th)*. O'Reilly Media.
- Yuana Rosihan Ari. 2010. *67 Trik & Ide Brilian Master PHP*. Penerbit Lokomedia. Yogyakarta.

<https://www.w3schools.com/>

## BIODATA PENULIS



**Ika Ratna Indra Astutik, S.Kom., M.T.** dilahirkan di Sidoarjo, 13 Mei 1981. Pada tahun 2005, penulis mendapatkan gelar Sarjana Teknik Informatika dari Universitas Muhammadiyah Sidoarjo. Penulis melanjutkan magister Jaringan Cerdas Multimedia di ITS dengan program beasiswa dari DIKTI. Tahun 2015, penulis secara resmi mendapatkan gelar M.T.

Penulis mengawali karirnya sebagai Dosen di prodi Teknik Informatika Universitas Muhammadiyah Sidoarjo. Penulis juga aktif terlibat dalam kegiatan penelitian dan pengabdian kepada masyarakat. Penelitian yang pernah dilakukan oleh penulis adalah tentang sistem informasi berbasis web, basis data dan sistem pengambilan keputusan.



**Mochamad Alfian Rosid, S.Kom., M.Kom.** lahir di Sidoarjo, 25 April 1986. Lulus Sarjana Komputer Universitas Muhammadiyah Sidoarjo tahun 2010. Penulis melanjutkan studi S2 di Prodi Teknologi Informasi Program Pascasarjana Sekolah Tinggi Teknik Surabaya lulus tahun 2014 dengan mendapatkan gelar M.Kom. Penulis mengawali

karirnya sebagai Dosen di prodi Teknik Informatika Universitas Muhammadiyah Sidoarjo. Penulis juga aktif terlibat dalam kegiatan penelitian dan pengabdian kepada masyarakat. Penelitian yang pernah dilakukan oleh penulis adalah tentang sistem informasi berbasis, basis data dan sistem pengambilan keputusan. Selain pendidikan dan pengajaran penulis juga terlibat dalam kegiatan penelitian dan pengabdian kepada masyarakat baik didanai oleh Ristekdikti maupun dana mandiri. Penulis juga aktif dalam mengikuti kegiatan-kegiatan penunjang akademik seperti seminar, workshop/ lokakarya, pelatihan serta pembimbingan tugas akhir dan kegiatan akademik.