



MODUL PRAKTIKUM

PEMROGRAMAN WEB

TEKNIK INFORMATIKA

UIN MAULANA MALIK IBRAHIM MALANG

TAHUN AJARAN 2022-2023

Modul 1 : Instalasi Software

1.1. Waktu Pelaksanaan Praktikum

Durasi kegiatan praktikum adalah **170 menit**, dengan rincian sebagai berikut.

- a. 15 menit untuk pengerjaan tes awal atau wawancara tugas pendahuluan.
- b. 60 menit untuk penyampaian materi.
- c. 45 menit untuk pengerjaan jurnal, tes akhir, atau tugas.
- d. 50 menit **pengayaan**

1.2. Tujuan

Setelah mengikuti praktikum ini, mahasiswa diharapkan dapat:

1. Melakukan instalasi *web server*.
2. Melakukan konfigurasi sederhana terhadap *web server*.
3. Mengoperasikan *database tool* untuk membuat *database* dan tabel.

1.3. Dasar Teori

a) Web Server, PHP Interpreter, dan Database Server

Teknologi *web* menggunakan arsitektur *client-server*. Karena itu, untuk dapat mulai membangun sebuah aplikasi *web*, kita harus memfungsikan komputer kita sebagai *web server*. Untuk itu, kita harus menginstal *software* khusus *web server*. Setelah *software* tersebut terinstal, komputer kita dapat berfungsi sebagai *web server* yang dapat melayani *request* yang dikirim oleh *client*. *Web server software* yang akan kita gunakan adalah salah satu *web server software* yang paling populer yaitu Apache Web Server.

Selain itu, karena pada mata kuliah ini kita akan menggunakan bahasa pemrograman PHP, maka kita juga harus menginstal *software* yang dapat mengeksekusi kode PHP yang biasa disebut dengan PHP *interpreter*. PHP *interpreter* yang akan kita gunakan adalah PHP *interpreter* standar. Kemudian kita juga perlu menginstal sebuah *database management system* (DBMS) karena sebuah *website* berfungsi untuk mengolah yang disimpan di *database*. DBMS *software* yang akan kita gunakan adalah MariaDB yang merupakan alternatif dari MySQL.

Kita dapat *download* dan menginstal ketiga *software* tersebut secara terpisah. Namun karena ketiga *software* tersebut seringkali digunakan bersamaan oleh pengembang *website*, ada beberapa *software* pihak ketiga yang menggabungkan ketiga *software* tersebut menjadi satu *software*. Tujuannya adalah agar pengembang dapat melakukan instalasi dan konfigurasi ketiga *software* tersebut dengan lebih cepat. Beberapa *software* pihak ketiga tersebut yang dapat kita gunakan adalah Laragon, XAMPP, WinNMP, WampServer, MAMP, dan lain-lain. Pada praktikum ini, kita akan menggunakan Laragon.

b) *Web Browser*

Setelah kita menginstal *web server*, kita membutuhkan *web browser* untuk menampilkan aplikasi *web* yang kita buat. Disarankan untuk menggunakan *web browser* modern seperti Google Chrome, Mozilla Firefox, atau Opera.

c) *Text Editor*

Untuk menulis kode PHP, kita dapat menggunakan *text editor* apapun, mulai dari yang paling sederhana seperti Notepad sampai IDE yang kompleks seperti NetBeans IDE. Tidak ada keharusan terhadap pemilihan *text editor* ini. Beberapa *text editor* yang sering digunakan adalah Notepad++, Sublime Text, Atom, dan Visual Studio Code.

d) *Database Tool*

Secara *default*, umumnya *software* DBMS hanya menyediakan antarmuka berupa *console*. Untuk mengolah data, antarmuka ini seringkali kurang nyaman dalam penggunaannya. Untuk itu, ada beberapa *software* yang berfungsi sebagai *tool* untuk mengolah data yang tersimpan di DBMS. Beberapa *tool* yang sering digunakan adalah PHPMyAdmin, HeidiSQL, DBVisualizer, Adminer, dan DBeaver. Beberapa dari *tool-tool* tersebut seperti PHPMyAdmin dan Adminer berbasis *web* sehingga membutuhkan *web server* sedangkan yang lainnya berbasis *desktop*.

1.4. Prosedur Praktikum

a) *Download dan Instalasi*

Download satu *software* dari masing-masing kategori berikut lalu lakukan instalasi.

Web Server, PHP Interpreter, dan Database Server	
Software	Website
Laragon	https://laragon.org/download/
XAMPP	https://www.apachefriends.org/download.html
WinNMP	https://winnmp.wtriple.com/
WampServer	http://www.wampserver.com
MAMP	https://www.mamp.info

Web Browser	
Software	Website
Google Chrome (disarankan)	https://www.google.com/chrome/browser/desktop/
Mozilla Firefox	https://www.mozilla.org/firefox/
Opera	http://www.opera.com/download

Text Editor	
Software	Website
Notepad++	https://notepad-plus-plus.org/
Sublime Text	https://www.sublimetext.com/
Atom	https://atom.io/
Visual Studio Code	https://code.visualstudio.com/

Database Tool	
Software	Website
PHPMYAdmin	https://www.phpmyadmin.net/ , sudah tercakup di Laragon/XAMPP
HeidiSQL (disarankan)	https://www.heidisql.com/ . Ada di Laragon
DBVisualizer	http://www.dbvis.com/
Adminer	https://www.adminer.org/ , sudah tercakup di WinNMP
DBeaver	https://dbeaver.jkiss.org/

b) Percobaan Web Server

1. Jalankan (*start*) *web server* yang telah terinstal.
2. Buka *browser* lalu buka URL "<http://localhost>". Pastikan muncul tampilan *default* dari *web server*.
3. Buka Notepad menggunakan akun *administrator* lalu buka *file* "C:\Windows\System32\drivers\etc\hosts". Tambahkan baris berikut di akhir *file* tersebut kemudian simpan.

```
# ganti "mywebsite.id" dengan string apapun
127.0.0.1 mywebsite.id
```

4. Buka browser dengan URL "<http://mywebsite.id>". Apakah yang terjadi? Apakah fungsi dari *file* "hosts"? Beri penjelasan.

5. Buka browser dengan URL "<http://mywebsite.id:81>". Apakah yang terjadi?
6. Buka Notepad lalu buka direktori *web server* yang telah terinstal (misal: "D:\xampp") kemudian buka *file* "apache\conf\httpd.conf". Cari baris "Listen 80" dan ubah menjadi "Listen 81" kemudian simpan.
7. *Restart (stop lalu start)* *web server* lalu buka URL "<http://mywebsite.id:81>". Apa yang terjadi? Buka kembali URL "<http://mywebsite.id>". Apa yang terjadi? Apakah sebenarnya angka 80 dan 81 tersebut? Beri penjelasan.

8. Kembalikan konfigurasi ke "Listen 80" lalu *restart* kembali *web server*.

c) Percobaan Database Server

- a. Buka Control Panel Laragon lalu pilih HeidiSQL.

- b. Buat database baru dengan nama “pemweb” lalu buat tabel berikut dengan nama “user”.

Field	Type Data
id	INT(10) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY
username	VARCHAR (50) NOT NULL UNIQUE
password	VARCHAR (255) NOT NULL
name	VARCHAR (255) NULL DEFAULT NULL

Modul 2 : HTML

2.1. Waktu Pelaksanaan Praktikum

Durasi kegiatan praktikum adalah **170 menit**, dengan rincian sebagai berikut.

- 15 menit untuk pengerjaan tes awal atau wawancara tugas pendahuluan
- 60 menit untuk penyampaian materi
- 45 menit untuk pengerjaan jurnal, tes akhir, atau tugas
- 50 menit **pengayaan**

2.2. Tujuan

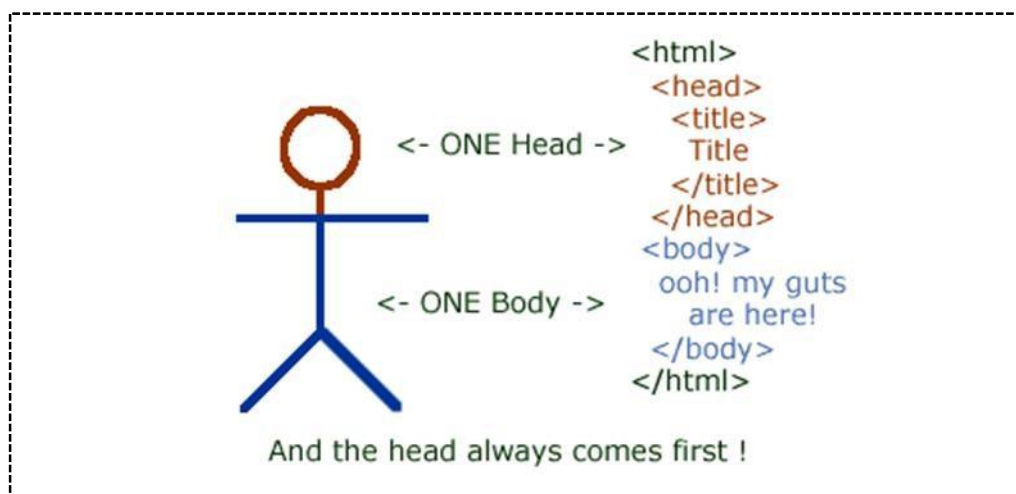
Setelah mengikuti praktikum ini, mahasiswa diharapkan dapat:

1. Memahami konsep dasar HTML.
2. Memahami kerangka dokumen HTML.
3. Memahami fungsi *tag-tag* HTML.
4. Mengaplikasikan CSS dalam dokumen HTML.
5. Memahami konsep JavaScript.
6. Menerapkan JavaScript di halaman *web*.

2.3. Dasar Teori

a) HTML

HTML atau *Hypertext Markup Language* adalah sebuah **bahasa markup** yang digunakan untuk membuat sebuah halaman *web*. Meskipun sering diartikan sebagai bahasa pemrograman dasar yang digunakan untuk membuat sebuah *web*, HTML pada dasarnya adalah bahasa *markup* (penanda) berbasis teks atau bisa juga disebut sebagai *formatting language* (bahasa untuk memformat). HTML menjadi dasar dalam pembuatan **website statis**.



Gambar 2.1 Struktur dasar dokumen HTML

Sintaks kode HTML diatur dan disusun dengan menggunakan **tag**. Berikut ini contoh kerangka HTML sederhana.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Judul Web</title>
5   </head>
6   <body>
7     // Teks (isi web) diletakkan di dalam tag body
8   </body>
9 </html>
```

Ada berbagai *tag* HTML yang dapat digunakan untuk membangun halaman HTML. Tiap-tiap *tag* HTML memiliki kegunaan yang berbeda-beda.

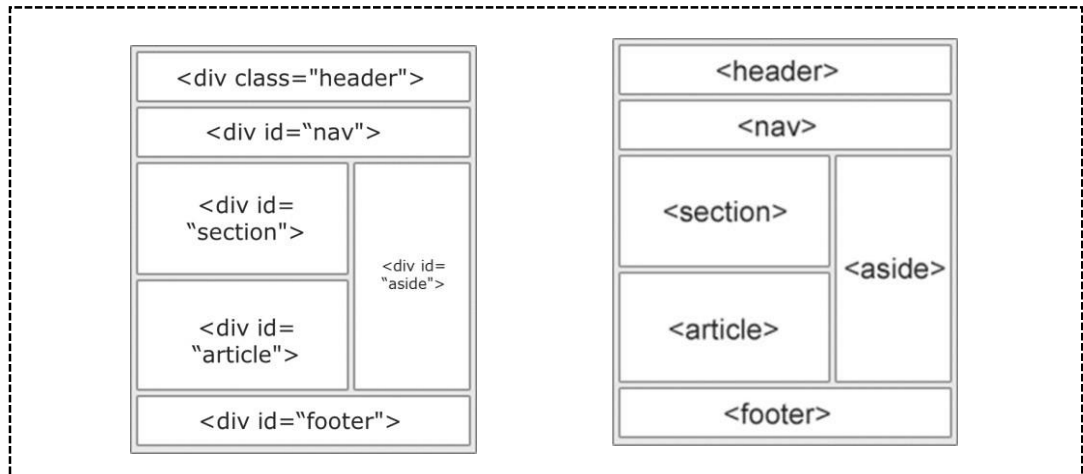
b) **HTML 5**

HTML5 adalah versi terbaru dari HTML. Versi terakhir HTML sebelum HTML versi 5 adalah HTML 4.01 dan XHTML 1.1. HTML5 menawarkan elemen-elemen baru yang dapat digunakan untuk pengembangan web. Kini, HTML5 telah diperbarui lagi menjadi HTML versi 5.1.

HTML5 memiliki keunggulan sebagai berikut.

1. Fitur baru yang didasarkan pada integrasi HTML, CSS, DOM, dan JavaScript.
2. Mengurangi kebutuhan untuk *plugin* eksternal (seperti Adobe Flash).
3. Penanganan kesalahan yang lebih baik.
4. Lebih banyak *markup element* yang dapat digunakan untuk menggantikan *scripting*.
5. HTML5 dimungkinkan menjadi perangkat mandiri.

Salah satu keunggulan HTML5 adalah merevolusi cara penulisan *tag-tag* dalam versi HTML sebelumnya menjadi lebih sederhana secara semantik. Selain itu penulisan *tag-tag* dengan menggunakan pola penulisan HTML4 masih dapat dilakukan dalam HTML5.



Gambar 2.2 Perbandingan HTML4 dan HTML5

3.3. Latihan

HTML Headings :

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Heading</title>
5 </head>
6 <body>
7 <h1>This is heading 1</h1>
8 <h2>This is heading 2</h2>
9 <h3>This is heading 3</h3>
10 <h4>This is heading 4</h4>
11 <h5>This is heading 5</h5>
12 <h6>This is heading 6</h6>
13 </body>
14 </html>

```

HTML Paragraf :

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Paragraf</title>
5 </head>
6 <body>
7 <p>This is first paragraph.</p>
8 <p>This is second paragraph.</p>
9 <p>This is third paragraph.</p>
10 </body>
11 |
12 <p>
13 My Bonnie lies over the ocean.
14
15 My Bonnie lies over the sea.
16 My Bonnie lies over the ocean.
17 Oh, bring back my Bonnie to me.
18 </p>
19 <p>Note that your browser ignores the layout!</p>
20 </body>
21 </html>
22

```


HTML Link:

```
link.html x
1 <html>
2 <body>
3 <a href="http://www.uin-malang.ac.id">UIN Malang</a>
4 | <a href="link2.html">Link 2</a>
5 </body>
6 </html>
```

```
link2.html x
1 <html>
2 <body>
3
4 <p>Create a link of an image:
5 <a href="link.html">
6 
7 </a></p>
8
9 <p>No border around the image, but still a link:
10 <a href="link2.html">
11 
12 </a></p>
13
14 </body>
15 </html>
```

HTML Image:

```
image.html x
1 <html>
2 <body>
3
4 <p>An image
5 
6 with align="bottom".</p>
7
8 <p>An image
9 
10 with align="middle".</p>
11
12 <p>An image
13 
14 with align="top".</p>
15
16 <p><b>Tip:</b> align="bottom" is default!</p>
17
18 <p>
19 An image before the text.</p>
20
21 <p>An image after the text.
22 </p>
23
24 </body>
25 </html>
```

HTML Formatting Text:

```
formatting.html x
1 <html>
2 <body>
3 <p><b>This text is bold</b></p>
4 <p><strong>This text is strong</strong></p>
5 <p><big>This text is big</big></p>
6 <p><i>This text is italic</i></p>
7 <p><em>This text is emphasized</em></p>
8 <p><code>This is computer output</code></p>
9 <p>This is<sub> subscript</sub> and <sup>superscript</sup></p>
10 </body>
11 </html>
```

2.4. Tugas

1. Jelaskan perbedaan dan persamaan `
` dengan `<p></p>`!!buat contohnya.
2. Buatlah 2 atau lebih file `*.html` yang dihubungkan dengan HTML Link untuk masing-masing file tersebut. Kemudian setiap file `*.html` jika di buka akan 'new tab'.
3. Jelaskan perbedaan dan persamaan `` dengan ``!! buat contohnya.

Modul 3

Cascading Style Sheet (CSS)&Java Scripts

3.1. Tujuan

- Mahasiswa dapat mengoperasikan struktur CSS dan Java Scripts
- Mahasiswa dapat memakai object dan form pada Java Scripts

3.2. Materi

- Cascading Style Sheet (CSS)*
- Java Scripts
- Pemrograman
- Form

3.3. Teori

Cascading Style Sheet (CSS)

CSS atau *Cascading Style Sheet* adalah standar pembuatan dan pemakaian *style* untuk dokumen HTML. CSS merupakan aturan untuk mengatur beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam. CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran *border*, warna *border*, warna *hyperlink*, warna *mouse over*, spasi antar paragraf, spasi antar teks, *margin* kiri, kanan, atas, bawah, dan parameter lainnya. Versi CSS terbaru untuk saat ini adalah CSS3.

Berikut ini merupakan contoh sintaks dasar penulisan CSS.

```
1 h1 {
2   color: #0789de;
3 }
```

Bagian pertama sebelum tanda “{ }” disebut *selector*, sedangkan yang diapit oleh tanda “{ }” disebut *declaration* yang terdiri dari dua unsur, yaitu *property* dan *value*. *Selector* dalam pernyataan di atas adalah h1, sedangkan color adalah *property*, dan #0789de adalah *value*.

Ada 3 cara penulisan CSS, yaitu:

1. *Inline Style Sheet*

Mengetikkan langsung dalam tag html sebagai atribut.

```
1 <HTML>
2 <HEAD>
3 <TITLE>centranet</TITLE>
4 </HEAD>
5 <BODY style="color: white;
6   background: green;
7   font-family : arial;" >
8 <H1>TI_UIN MALANG</H1>
9 <P>MATERI 3 Pemrograman Web adalah CSS </P>
10 </BODY>
11 </HTML>
```



2. *Embedded Style Sheet*

Menggunakan tag style di dalam tag head.

```

1 <HTML>
2 <HEAD>
3 <TITLE>centranet</TITLE>
4 <STYLE type="text/css">
5 body {
6 color: white;
7 background: green;
8 font-family : arial;
9 }
10 </STYLE>
11 </HEAD>
12 <BODY>
13 <H1>Teknik Informatiak</H1>
14 <P>Teknik Informatika UIN Malang berdiri tahun 2004</P>
15 </BODY>
16 </HTML>

```



3. Linked/External Style Sheet

Menyimpan informasi *style* ke dalam sebuah *file* dengan ekstensi *.css* dan memanggil *file* CSS dalam HTML dengan *tag* `<link>` yang diletakkan dalam *tag* `<head>`.

```

1 <HTML>
2 <HEAD>
3 <TITLE>Teknik Informatika</TITLE>
4 <LINK REL="STYLESHEET"
5 | TYPE="text/css" HREF="efek.css">
6 </HEAD>
7 <BODY>
8 <H1>UIN Malang</H1>
9 <P>Teknik Informatika UIN Malang </P>
10 </BODY>
11 </HTML>

```



Sekilas tentang JavaScript

JavaScript adalah bahasa skrip yang ditempelkan pada kode HTML dan diproses di sisi klien. Dengan adanya bahasa ini, kemampuan dokumen HTML menjadi semakin luas. Sebagai contoh, dengan menggunakan JavaScript dimungkinkan untuk memvalidasi masukan-masukan pada formulir sebelum formulir dikirimkan ke server. JavaScript bukanlah bahasa Java dan merupakan dua bahasa yang berbeda. JavaScript diinterpretasikan oleh klien (kodenya bisa dilihat pada sisi klien), sedangkan kode Java dikompilasi oleh pemrogram dan hasil kompilasinya yang dijalankan oleh klien.

Struktur JavaScript

Struktur dari JavaScript adalah sbb :

```

<SCRIPT LANGUAGE = "JavaScript">
<!-- -
Penulisan kode javascript
// - - >
</SCRIPT>

```

Keterangan

Kode `<!-- - // - - >`

umumnya disertakan dengan tujuan agar sekiranya browser tidak mengenali JavaScript maka browser akan memperlakukannya sebagai komentar sehingga tidak ditampilkan pada jendela browser.

JavaScript sebagai bahasa berorientasi pada obyek

Properti

Properti adalah atribut dari sebuah objek. Contoh, objek mobil punya properti warna mobil.

Penulisan :

```

Nama_objek.nama_properti = nilai
window. defaultStatus = "Selamat Belajar JavaScript";

```

Metode

Metode adalah suatu kumpulan kode yang digunakan untuk melakukan sesuatu tindakan terhadap objek.

Penulisan :

```
Nama_objek.nama_metode(parameter)  
document.write ("Hallo")
```

Letak JavaScript dalam HTML

Skrip Javascript dalam dokumen HTML dapat diletakkan pada :

1. Bagian Head
2. Bagian Body (jarang digunakan).

Latihan :

DASAR-DASAR JAVASCRIPT

1. Pemakaian alert sebagai property window

```
1 <HTML>  
2 <HEAD>  
3 <TITLE>Alert Box</TITLE>  
4 </HEAD>  
5 <BODY>  
6 <SCRIPT LANGUAGE = "JavaScript">  
7 //<!-  
8 window.alert("Ini merupakan pesan untuk Anda");  
9 //->  
10 </SCRIPT>  
11 </BODY>  
12 </HTML>  
13
```

2. Pemakaian metode dalam objek.

```
1 <HTML>  
2 <HEAD>  
3 <TITLE>Skrip JavaScript</TITLE>  
4  
5 </HEAD>  
6 <BODY>  
7 Percobaan memakai JavaScript:<BR>  
8  
9 <SCRIPT LANGUAGE = "JavaScript">  
10 //<!-  
11  
12 document.write("Selamat Mencoba JavaScript<BR>"); document.write("Semoga sukses!");  
13 //->  
14 </SCRIPT>  
15  
16 </BODY>  
17 </HTML>
```

3. Pemakaian prompt

```
1 <HTML>  
2  
3 <HEAD>  
4 <TITLE>Pemasukan Data</TITLE>  
5 </HEAD>  
6  
7 <BODY>  
8 <SCRIPT LANGUAGE = "JavaScript">  
9 //<!-  
10  
11 var nama = prompt("Siapa nama Anda?","Masukkan nama anda"); document.write("Hai, " + nama);  
12 //->  
13  
14 </SCRIPT>  
15 </BODY>  
16 </HTML>
```

Pemrograman di Java Scripts

Operasi dasar aritmatika

```
1 <HTML>
2 <HEAD>
3
4 <TITLE>Contoh Program Javascript</TITLE>
5 </HEAD>
6 <SCRIPT language="Javascript">
7
8   function pesan(){
9
10    alert ("memanggil javascript lewat body onload")
11  }
12
13 </SCRIPT>
14 <BODY onload=pesan()>
15 </BODY>
16
17 </HTML>
```

```
1
2 <HTML>
3 <HEAD>
4
5 <TITLE>Contoh Program Javascript</TITLE>
6 </HEAD>
7 <SCRIPT language="Javascript">
8
9   function test (val1,val2)
10  {
11    document.write("<br>"+"Perkalian : val1*val2 "+"<br>")
12
13    document.write(val1*val2)
14    document.write("<br>"+"Pembagian : val1/val2 "+"<br>")
15    document.write(val1/val2)
16
17    document.write("<br>"+"Penjumlahan : val1+val2 "+"<br>")
18
19    document.write(val1+val2)
20    document.write("<br>"+"Pengurangan : val1-val2 "+"<br>")
21
22    document.write(val1-val2)
23    document.write("<br>"+"Modulus : val1%val2 "+"<br>")
24    document.write(val1%val2)
25
26  }
27 </SCRIPT>
28 <BODY>
29
30 <input type="button" name="button1" value="arithmetic"
31 onclick=test(9,4)>
32 </BODY>
33
34 </HTML>
```

Operasi Relational

Seleksi kondisi (if..else)

```
1 <HTML>
2 <HEAD>
3 <TITLE>Contoh Program Javascript</TITLE>
4 </HEAD>
5 <SCRIPT LANGUAGE = "JavaScript">
6 //<-
7 var nilai = prompt("Nilai (0-100): ", 0);
8 var hasil = "";
9 if (nilai >= 60)
10  hasil = "Lulus";
11 else
12  hasil = "Tidak Lulus";
13 document.write("Hasil: " + hasil);
14 //->
15 </SCRIPT>
16 </BODY>
17 </HTML>
```

```

1 <HTML>
2 <HEAD>
3 <TITLE>Contoh Program Javascript</TITLE>
4 </HEAD>
5 <SCRIPT language="Javascript">
6 function test () {
7 val1=window.prompt("Nilai I :");
8 val2=window.prompt("Nilai II :");
9 document.write("<br>"+val1==val2+"<br>");
10 document.write(val1==val2)
11 document.write("<br>"+val1!=val2+"<br>");
12 document.write(val1!=val2)
13 document.write("<br>"+val1>val2+"<br>");
14 document.write(val1>val2)
15 document.write("<br>"+val1<val2+"<br>");
16 document.write(val1<val2) }
17 </SCRIPT>
18 <BODY>
19 <input type="button" name="button1" value="relational" onclick=test()>
20 </BODY>
21 </HTML>

```

Penggunaan operator switch untuk seleksi kondisi *pemakaian looping <for >*

```

1 <HTML>
2 <HEAD>
3 <TITLE>Contoh Program Javascript</TITLE>
4 </HEAD>
5 <SCRIPT language="Javascript">
6 function test ()
7 {
8
9 val1=window.prompt("Input Nilai (1-5):");
10 switch (val1)
11 {
12
13 case "1" :
14 document.write("bilangan satu")
15 break
16
17 case "2" :
18 document.write("bilangan dua")
19 break
20
21 case "3" :
22 document.write("bilangan tiga")
23 break
24
25 case "4" :
26 document.write("bilangan empat")
27 break
28
29 case "5" :
30 document.write("bilangan lima")
31 break
32
33 default :
34 document.write("bilangan lainnya")
35 }
36 }
37 </SCRIPT>
38 <BODY>
39 <input type="button" name="button1" value="switch"
40 onclick=test()>
41 </BODY>
42 </HTML>

```

```

1 <HTML>
2 <HEAD>
3
4 <TITLE>Contoh Program Javascript</TITLE>
5 </HEAD>
6 <BODY>
7
8 <SCRIPT language="Javascript">
9 /<!--
10 for (x=0;x<=10;x++)
11
12 document.write(x+"<br>")
13
14
15 // --> </SCRIPT>
16 </BODY>
17 </HTML>

```

Pemakaian looping < do..while >

Pemakaian looping < while >

```

1 <HTML>
2
3 <HEAD>
4 <TITLE>Contoh Program Javascript</TITLE>
5 </HEAD>
6
7 <BODY>
8 <SCRIPT language="Javascript">
9 //<!--
10
11 var x=0
12 do{
13 document.write(x+"<br>")
14
15 x++;
16 }
17 while (x<=10)
18
19 // -> </SCRIPT> </BODY>
20 </HTML>

```

Form

1. Form input :

```

1 <HTML>
2 <HEAD>
3 <TITLE>Objek document</TITLE>
4 </HEAD>
5 <BODY>
6 <SCRIPT LANGUAGE = "JavaScript">
7
8 //<!--
9 function ubahWarnaLB(warna) {
10 document.bgColor = warna;
11
12 }
13 function ubahWarnaLD(warna) {
14 document.fgColor = warna;
15 }
16 //-->
17 </SCRIPT>
18
19 <H1>TES</H1>
20 <FORM>
21 <INPUT TYPE = "BUTTON"
22
23 VALUE = "Latar Belakang Hijau"
24 onClick = "ubahWarnaLB('GREEN')">
25 <INPUT TYPE = "BUTTON"
26
27 VALUE = "Latar Belakang Putih"
28 onClick = "ubahWarnaLB('WHITE')">
29 <INPUT TYPE = "BUTTON"
30
31 VALUE = "Teks Kuning"
32 onClick = "ubahWarnaLD('YELLOW')">
33 <INPUT TYPE = "BUTTON"
34
35 VALUE = "Teks Biru"
36 onClick = "ubahWarnaLD('BLUE')">
37 </FORM></BODY></HTML>

```

2. Form button :

```

1 <html>
2 <head> </head>
3 <SCRIPT language="Javascript">
4
5 function test () {
6 var val1=document.kirim.T1.value
7 if (val1%2==0)
8
9 document.kirim.T2.value="bilangan genap"
10 else
11 document.kirim.T2.value="bilangan ganjil"
12
13 }
14 </SCRIPT>
15 <body>
16
17 <form method="POST" name="kirim">
18
19 <p>BIL <input type="text" name="T1" size="20"> MERUPAKAN BIL <input type="text" name="T2" size="20"> </p>
20
21
22 <p><input type="button" value="TEBAK" name="B1" onclick=test()> </p>
23
24 </form>
25 </body>
26 </html>

```


1.6 Latihan

1. Tulis kode dibawah ini

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <title>Hello World</title>
6   </head>
7   <body>
8     <h1>Hello World</h1>
9     <p>Belajar HTML</p>
10  </body>
11 </html>
```

Jalankan kode di atas tanpa menggunakan server. Apa yang terjadi? Kemudian hapus tag <body>. Amati apa yang terjadi lalu simpulkan.

2. Tulis kode dibawah ini

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <title>Hello World</title>
6     <style type="text/css">
7       h1 {
8         color:orange;
9         text-align:center;
10      }
11      p {
12        font-family:"Times New Roman";
13        font-size:40px;
14      }
15    </style>
16  </head>
17  <body>
18    <h1>Hello World</h1>
19    <p>Belajar HTML</p>
20  </body>
21 </html>
```

- Jalankan kode tersebut kemudian amati yang terjadi. Apakah ada perbedaan daripada sebelumnya? Beri penjelasan.
- Termasuk cara apakah penulisan CSS pada kode di atas?
- Ubah cara penulisan CSS ini dalam tipe linked (external) style sheet.

3. Tulis kode dibawah ini

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <body>
4     <h1>Operasi JS</h1>
5     <p id="demo"></p>
6     <script>
7       var x = 5;
8       var y = 2;
9       var z = x + y;
10      document.getElementById("demo").innerHTML = z;
11    </script>
12  </body>
13 </html>
```

Jalankan kode tersebut dan amati apa yang terjadi. Apa fungsi dari baris ke-10?

4. Tulis kode dibawah ini

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <script>
5       function convert(degree) {
6         if (degree == 'C') {
7           F = document.getElementById('c').value * 9 / 5 + 32;
8           document.getElementById('f').value = Math.round(F);
9         } else {
10          C = (document.getElementById('f').value - 32) * 5 / 9;
11          document.getElementById('c').value = Math.round(C);
12        }
13      }
14    </script>
15  </head>
16  <body>
17    <p>Masukkan angka pada masukan berikut:</p>
18    <input id="c" onkeyup="convert('C')"> derajat Celcius <br> equals<br>
19    <input id="f" onkeyup="convert('F')"> derajat Fahrenheit
20  </body>
21 </html>
```

- a. Jalankan kode di atas dengan server dan amati apa hasil dari kode di atas.
- b. Kemudian jalankan kembali kode diatas tanpa menggunakan server. Amati apakah ada perbedaan dengan sebelumnya?
- c. Simpulkan bagaimana sebenarnya prinsip kerja dari JavaScript.

5. Buat halaman html untuk mengkonversi nilai angka menjadi nilai huruf dengan menggunakan javascript .

Konversi : 0-40 =E

41-55=D

56-60=C

61-65=BC

66-70=B

71-80=AB

81-100=A

6. Buat halaman html untuk menampilkan aplikasi program kalkulator sederhana dengan menggunakan javascript.

Contoh tampilan :

Bil 1 dan Bil 2 merupakan text box, dapat diisi angka, bila tombol + atau - atau x atau / ditekan, maka akan keluar bilangan pada text box hasil, dimana bilangan ini merupakan operasi arithmetic sesuai dengan tombol yang ditekan.

Modul 4 : Dasar PHP

4.1 Waktu Pelaksanaan Praktikum

Durasi kegiatan praktikum adalah **170 menit**, dengan rincian sebagai berikut.

- 15 menit untuk pengerjaan Tes Awal atau wawancara Tugas Pendahuluan
- 60 menit untuk penyampaian materi
- 45 menit untuk pengerjaan jurnal, tes akhir atau tugas
- 50 menit **pengayaan**

4.2 Tujuan

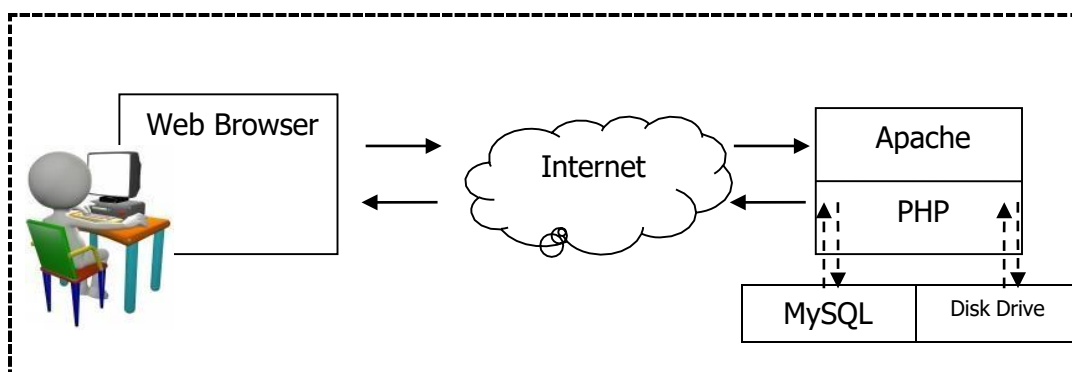
Setelah mengikuti praktikum ini, mahasiswa diharapkan dapat:

1. Memahami gambaran umum aplikasi web.
2. Memahami kerangka program PHP.
3. Memahami tipe data, variabel, dan operator pada PHP.
4. Memahami sintaks untuk mengatur alur program PHP.

4.3 Dasar Teori

a) PHP

PHP adalah salah satu bahasa pemrograman yang dirancang untuk membangun aplikasi **web dinamis**. PHP juga disebut sebagai **server-side programming/scripting** yang berarti *script/program* yang berjalan di sisi *server*. Kode PHP di-*parse* di *web server* oleh **interpreter** dan diterjemahkan ke dalam bentuk HTML (*Hypertext Markup Language*) dan pada akhirnya dikirimkan dan ditampilkan pada *browser* yang dibuka oleh *user*. Gambar 4.1 memberikan gambaran singkat alur kerja PHP dalam *web server*.



Gambar 4.1 Alur request-response PHP

Sintaks kode PHP biasanya diletakkan di dalam HTML, walaupun bisa juga dibuat dokumen dengan PHP murni. Kode PHP diawali dengan **tag pembuka PHP** “`<?php`” dan diakhiri dengan **tag penutup**

PHP “<?>” atau bisa juga dengan bentuk yang lebih singkat yaitu dengan awalan “<?” dan diakhiri “>”. Kode PHP harus disimpan di dalam file berekstensi .php. Jika tidak, maka kode PHP tidak akan dieksekusi oleh *web server* walaupun ditulis di antara *tag* pembuka dan penutup PHP. Berikut satu contoh kode PHP sederhana.

```
1 <?php
2     echo "Hello World";
3     echo 10000;
```

Perintah `echo` berfungsi untuk mencetak nilai baik teks maupun numerik ke *browser*. *Keyword-keyword* perintah PHP seperti `echo` dan juga tag pembuka dan penutup PHP (`<?php` dan `>`) tidak akan ikut tercetak di *browser*. Selain `echo`, kita juga bisa menggunakan perintah `print`:

```
1 <?php
2     print "Hello World";
```

Jika suatu file PHP hanya berisi kode PHP saja tanpa HTML atau kode PHP terletak di akhir *file*, maka *tag* penutup PHP (`>`) tidak perlu ditulis. Contoh:

```
1 <?php
2     print "Hello World";
3     <!-- kode HTML -->
4 </html>
5
6 <?php
7     print "Hello World";
8     // akhir file
```

Setiap perintah atau statemen di dalam kode PHP harus diakhiri dengan tanda titik koma (“;”).

Ada beberapa cara untuk meletakkan kode PHP dalam sebuah dokumen, yaitu dengan menyisipkan PHP dalam HTML atau HTML dalam PHP. Berikut contoh PHP dalam HTML.

```
1 <html>
2 <head>
3     <title>PHP dalam HTML</title>
4 </head>
5 <body>
6     <?php echo "<p>Teks dari kode PHP</p>"; ?>
7 </body>
8 </html>
```

Berikut contoh HTML dalam PHP.

```
1 <?php
2     echo "<html>";
3     echo "<head>";
4     echo "<title>HTML dalam PHP</title>";
5     echo "</head>";
6     echo "<body>";
7     echo "<p>Paragraf 2: Teks dari kode PHP</p>";
8     echo "</body>";
```

b) Tipe Data, Variabel, dan Operator

PHP mendukung delapan tipe data. Empat nilai skalar (*single-value*): *integer*, *float*, *string*, dan *boolean*; nilai koleksi: *array*. Sisanya adalah tipe khusus yaitu *resource* dan *null*.

Integer	<i>Integer</i> adalah tipe data yang merepresentasikan bilangan utuh atau bulat, positif maupun negatif dan bukan pecahan. Rentang dari bilangan integer adalah -2.147.483.648 sampai 2.147.483.647.
Float	<i>Float</i> disebut sebagai bilangan pecahan atau sering juga disebut sebagai bilangan <i>real</i> yang mengandung angka desimal di belakang koma. Contohnya adalah 3,14; 2,54; dan 9,99.
String	PHP mendukung tipe data <i>string</i> yaitu tipe data karakter yang biasanya diekspresikan dengan diapit oleh petik ganda atau petik tunggal. Contoh: <code>'ini adalah string'</code> atau <code>"ini adalah string"</code>
Boolean	Tipe <i>boolean</i> adalah tipe data yang merepresentasikan nilai kebenaran (<i>truth value</i>). Nilai yang mungkin pada tipe <i>boolean</i> adalah <i>true</i> dan <i>false</i> . Penulisan nilai <i>true</i> dan <i>false</i> bersifat <i>case-insensitive</i> sehingga <code>"true"</code> , <code>"True"</code> , dan <code>"TRUE"</code> adalah sama dan <code>"false"</code> , <code>"False"</code> , dan <code>"FALSE"</code> adalah sama. Contoh: <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre><?php \$found = true; if (\$found) { echo "Data telah ditemukan"; }</pre> </div> Maksud dari kode diatas adalah jika variabel <code>\$found</code> bernilai <i>true</i> , maka dilakukan kode di dalam blok <code>if</code> akan dieksekusi.
Array	<i>Array</i> adalah variabel yang menyimpan sekelompok nilai yang dapat diidentifikasi berdasarkan posisinya atau indeksnya. Contoh: <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre>\$person = array('Dadang', 'Jajang', 'Udin', 'Rendra');</pre> </div> Penulisan <i>array</i> juga dapat dilakukan dengan sintaks yang lebih ringkas. Contoh: <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre>\$person = ['Dadang', 'Jajang', 'Udin', 'Rendra'];</pre> </div>

Pendefinisian Variabel

Variabel di PHP ditulis dengan awalan karakter dolar (`$`). Contoh:

```
1 $username = "Hendro";
2 $usia = 25;
```

PHP tidak seperti bahasa pemrograman lain yang mengharuskan kita untuk memberikan atau menuliskan tipe variabel. Artinya PHP mempunyai kemampuan untuk mendeteksi secara otomatis apa tipe variabel yang kita tulis.

Operator

Dalam PHP operator dibagi menjadi 4 yaitu operator *assignment*, operator aritmetika, operator relasional, dan operator logika. Operator *assignment* adalah operator yang digunakan untuk memberikan sebuah nilai yang akan dimasukkan ke dalam variabel. Operator *assignment* ditulis dengan karakter sama dengan (“=”). Operator aritmetika digunakan untuk operasi aritmetika dengan daftar berikut.

Operator	Keterangan
+	Operator penjumlahan
-	Operator pengurangan
*	Operator perkalian
/	Operator pembagian
%	Operator modulus (sisa dari pembagian)
++	Operator untuk menaikkan nilai variabel sebesar 1
--	Operator untuk menurunkan nilai variabel sebesar 1

Operator relasional adalah operator yang digunakan untuk membandingkan 2 buah nilai. Hasil operasi hanya ada dua yaitu *true* dan *false*. Berikut adalah contohnya.

Operator	Keterangan
$\$a == \b	Memeriksa apakah \$a sama dengan \$b
$\$a != \b	Memeriksa apakah \$a tidak sama dengan \$b
$\$a < \b	Memeriksa apakah \$a kurang dari \$b
$\$a > \b	Memeriksa apakah \$a lebih besar dari \$b
$\$a <= \b	Memeriksa apakah \$a kurang dari sama dengan \$b
$\$a >= \b	Memeriksa apakah \$a lebih dari sama dengan \$b

Sedangkan operator logika adalah operator yang digunakan untuk mengoperasikan dua buah *operand* yang bertipe *boolean*. Contoh dari operator logika adalah sebagai berikut.

Operator	Keterangan
&&	Logika AND
	Logika OR
!	Logika NOT

c) Alur

Alur adalah sintaks yang didefinisikan untuk mengatur alur program. Pembahasan ini dibagi menjadi dua, yaitu:

- Kondisi
- Perulangan

Kondisi

Kondisi digunakan untuk menentukan kode mana yang akan dijalankan. Biasanya kondisi diawali dengan pengecekan beberapa parameter untuk diseleksi. Pada kondisi terdapat beberapa sintak yang bisa kita gunakan. Yang pertama adalah kondisi `if`. Berikut adalah contoh kodenya.

```
1  if (kondisi) {
2      statemen yang dieksekusi
3  }
```

Jika kita tidak menemukan hasil pada seleksi pertama maka kita bisa menambahkan `else` seperti contoh berikut.

```
1  if (kondisi) {
2      statemen yang dieksekusi
3  } else {
4      statemen lainnya
5  }
```

Atau kita bisa gunakan model seperti di bawah ini.

```
1  if (kondisi1){
2      // statemen untuk kondisi 1;
3  } elseif (kondisi2){
4      // statemen untuk kondisi 2;
5  } elseif (kondisi3){
6      // statemen untuk kondisi 3;
7  }
```

Selain `if-else`, kita juga bisa menggunakan kondisional `switch`, kondisional ini digunakan ketika kita membutuhkan banyak kondisi. Contohnya adalah sebagai berikut.

```
1  <?php
2      switch ($bulan){
3          case 1:
4              echo "Januari";
5              break;
6          case 2:
7              echo "Februari";
8              break;
9          case 3:
10             echo "Maret";
11             break;
12          case 4:
13             echo "April";
14             break;
15          case 5:
16             echo "Mei";
17             break;
18          case 6:
19             echo "Juni";
20             break;
21          case 7:
22             echo "Juli";
23             break;
24          case 8:
25             echo "Agustus";
26             break;
```

```

27     case 9:
28         echo "September";
29         break;
30     case 10:
31         echo "Oktober";
32         break;
33     case 11:
34         echo "November";
35         break;
36     case 12:
37         echo "Desember";
38         break;
39     default:
40         echo "Ini default statemen";
41     }
42     ?>

```

Perulangan

Dalam pembuatan sebuah aplikasi *web*, seringkali kita harus melakukan pengulangan terhadap statemen-statemen tertentu. PHP menyediakan kode yang dapat kita gunakan untuk mengulang statemen, antara lain `while`, `do-while`, `for`, dan `foreach`. Berikut sintaks masing-masing kode.

While

```

1 while(ekspresi) {
2     Statemen;
3 }

```

Do-while

```

1 do {
2     statemen
3 } while (ekspresi);

```

For

```

1 for (inisialisasi; ekspresi; increment/decrement) {
2     statemen
3 }

```

Foreach

```

1 foreach ($array as $current) {
2     statemen
3 }

```

4.4 Prosedur Praktikum

a) Latihan 1

Tulis kode di bawah ini.


```
1 <?php
2     $hello = "Hello World!";
3     $a_number = 4;
4     $anotherNumber = 8.9;
5     $bool = true;
```

Hapus notasi \$, lalu jalankan kode di atas. Apa yang terjadi? Lalu simpulkan.

b) Latihan 2

Tulis kode di bawah ini.

```
1 <?php
2     echo "<h1>Variabel</h1><br>";
3
4     $angka = 99;
5     echo "Ini adalah angka = $angka <br>";
6
7     $angka_2 = $angka + 1;
8     echo "Ini hasilnya = $angka_2 <br>";
```

Amati apa hasil dari kode di atas lalu simpulkan.

c) Latihan 3

Tulis kode di bawah ini.

```
1 <?php
2     $addition = 2 + 4;
3     $subtraction = 6 - 2;
4     $multiplication = 5 * 3;
5     $division = 15 / 3;
6     $modulus = 5 % 2;
7
8     echo "Penambahan: 2 + 4 = $addition <br>";
9     echo "Pengurangan: 6 - 2 = $subtraction <br>";
10    echo "Perkalian: 5 * 3 = $multiplication <br>";
11    echo "Pembagian: 15 / 3 = $division <br>";
12    echo "Pembagian sisa: 5 % 2 = $modulus";
```

Amati apa hasil dari kode di atas lalu simpulkan.

d) Latihan 4

Tulis kode di bawah ini.

```
1 <?php
2     $x = 4;
3     $x += 3;
4     echo "Hasil dari operasi tersebut adalah = $x";
```

Gantikan operator "+=" dengan:

1. "-="
2. "*="
3. "/="
4. "%="
5. ".="

Amati apa hasil dari operator tersebut lalu simpulkan.

e) Latihan 5

Tulis kode di bawah ini.

```
1 <?php
2     $my_name = "anotherguy";
3
4     if ($my_name == "someguy") {
5         echo "Your name is someguy!<br>";
6     }
7
8     echo "Welcome to my homepage!";
```

Isikan variabel \$my_name dengan nilai yang lain dan simpulkan.

f) Latihan 6

Tulis kode di bawah ini.

```
1 <?php
2 $destination = "Tokyo";
3 echo "Traveling to $destination<br>";
4
5 switch ($destination){
6     case "Las Vegas":
7         echo "Bring an extra $500";
8         break;
9     case "Amsterdam":
10        echo "Bring an open mind";
11        break;
12    case "Egypt":
13        echo "Bring 15 bottles of SPF 50 Sunscreen";
14        break;
15    case "Tokyo":
16        echo "Bring lots of money";
17        break;
18    case "Caribbean Islands":
19        echo "Bring a swimsuit";
20        break;
21 }
```

Gantikan nilai dari variabel `$destination` dengan "Amsterdam". Apa yang terjadi? Jelaskan alur kerja dari alur `switch`.

g) Latihan 7

Tulis kode di bawah ini.

```
1 <?php
2 $brush_price = 5;
3
4 echo "<table border=\"1\" align=\"center\">";
5 echo "<tr><th>Quantity</th>";
6 echo "<th>Price</th></tr>";
7
8 for ( $counter = 10; $counter <= 100; $counter += 10) {
```

```
9      echo "<tr><td>";
10     echo $counter;
11     echo "</td><td>";
12     echo $brush_price * $counter;
13     echo "</td></tr>";
14 }
15
16 echo "</table>";
```

Ganti inkremen dengan `$counter += 5`, apa yang terjadi? Ganti looping `for` dengan menggunakan `while`, dan `do-while`? Jelaskan bagaimana alur kerja dari looping `for`, `while`, dan `do-while`?

Modul 5 : Fungsi

5.1 Waktu Pelaksanaan Praktikum

Durasi kegiatan praktikum adalah **170 menit**, dengan rincian sebagai berikut.

- a. 15 menit untuk pengerjaan Tes Awal atau wawancara Tugas Pendahuluan
- b. 60 menit untuk penyampaian materi
- c. 45 menit untuk pengerjaan jurnal, tes akhir atau tugas
- d. 50 menit **pengayaan**

5.2 Tujuan

Setelah mengikuti praktikum ini, mahasiswa diharapkan dapat:

1. Memahami prinsip dan karakteristik fungsi di PHP.
2. Membuat fungsi sederhana di PHP.
3. Membuat fungsi dengan jumlah parameter bebas di PHP.
4. Membuat fungsi yang bekerja secara rekursif di PHP.
5. Memahami perbedaan antar variabel lokal dan variabel global.

5.3 Dasar Teori

a) Fungsi

Fungsi adalah sekumpulan baris kode yang dapat diberi nama dan melakukan suatu *task* tertentu. Suatu fungsi dapat menerima input yang disebut dengan **parameter** dan memberikan output yang disebut dengan **return value**. Penggunaan fungsi dapat memperbaiki struktur kode secara keseluruhan dan membuat dapat kode lebih mudah untuk dibaca.

Ada fungsi-fungsi yang telah ada secara *built-in* di PHP dan kita juga dapat membuat fungsi kita sendiri. Contoh fungsi *built-in* adalah `strpos()` dan `unlink()`. Sintaks pemanggilan fungsi adalah seperti berikut.

```
1 $return_value = nama_fungsi([parameter, ...]);
```

Tanda kurung kotak dan titik-titik di bagian *parameter* menunjukkan bahwa *parameter* suatu fungsi bersifat opsional dan dapat berjumlah berapapun. Contoh pemanggilan fungsi `strpos()` dan `unlink()` adalah seperti berikut.

```
1 $pos = strpos('abc', 'a');  
2 unlink('myfile.txt');
```

Urutan penulisan parameter harus sesuai dengan urutan *parameter* di mana fungsi tersebut ditulis.

b) Membuat Fungsi

Berikut adalah sintaks penulisan fungsi.

```
1 function [&] nama_fungsi([parameter, ...]) {
2     statemen-statemen
3 }
```

Nama fungsi diawali dengan huruf atau karakter *underscore* (`_`) diikuti dengan huruf, karakter *underscore*, atau angka. Nama fungsi bersifat *case-insensitive*. Artinya, fungsi `strpos()` dapat dipanggil dengan `strpos()`, `StrPos()`, atau `STRPOS()`. Namun begitu, berdasarkan konvensi penamaan PHP, fungsi sebaiknya diberi nama menggunakan huruf kecil semua.

Untuk mengembalikan suatu nilai sebagai *return value*, gunakan statemen `return` diikuti dengan nilai yang ingin dikembalikan. Sebagai contoh, fungsi berikut menerima dua parameter berupa *string*, menggabungkan kedua *string* tersebut, kemudian mengembalikan *string* gabungan tersebut sebagai *return value*.

```
1 function concat($string1, $string2) {
2     $concatenated = $string1 . $string2;
3
4     return $concatenated;
5 }
```

Fungsi tersebut dapat ditulis dengan lebih ringkas seperti berikut.

```
1 function concat($string1, $string2) {
2     return $string1 . $string2;
3 }
```

Fungsi tersebut kemudian dapat dipanggil seperti contoh berikut.

```
1 $str1 = 'FILKOM';
2 $str2 = ' UB';
3 $str3 = concat($str1, $str2);
4 echo concat($str2, $str3);
```

c) Variabel Lokal dan Global

Variabel-variabel yang dideklarasikan di dalam sebuah fungsi memiliki lingkup di dalam fungsi itu saja. Variabel-variabel tersebut tidak dapat diakses di luar fungsi tersebut. Contoh berikut menjelaskan karakteristik tersebut.

```
1 $a = 2;
2
3 function add() {
4     $a += 3;
5 }
6
7 add();
```

```
8 echo $a;
9
10 // Output: 2
```

Variabel `$a` yang berada di dalam dan di luar fungsi `add()` adalah dua variabel yang hanya namanya saja yang sama namun sebenarnya merupakan dua variabel yang berbeda dan terpisah. Penambahan nilai variabel `$a` (statemen `$a += 3`) yang dilakukan di dalam fungsi `add()` tidak berpengaruh terhadap nilai dari variabel `$a` yang berada di luar fungsi tersebut. Oleh karena itu, variabel `$a` yang berada di luar fungsi tetap bernilai 3. Variabel yang dideklarasikan di dalam sebuah fungsi seringkali disebut dengan **variabel lokal** sedangkan variabel yang dideklarasikan di luar suatu fungsi seringkali disebut dengan **variabel global**.

Agar suatu variabel global dapat dikenali di dalam sebuah fungsi, digunakan *keyword* `global`. Contoh:

```
1 $a = 2;
2
3 function add() {
4     global $a;
5
6     $a += 3;
7 }
8
9 add();
10 echo $a;
11
12 // Output: 5
```

d) Variabel Statis

Perhatikan kode berikut.

```
1 function add() {
2     $a = 0;
3
4     $a += 3;
5
6     echo "$a\n";
7 }
8
9 add();
10 add();
11 add();
12
13 // Output:
14 3
15 3
16 3
```

Pada kode di atas, terdapat sebuah fungsi yaitu `add()` dan di dalam fungsi tersebut, dideklarasikan sebuah variabel yaitu `$a` yang diinisialisasi dengan nilai nol. Fungsi tersebut dipanggil tiga kali. Di setiap pemanggilan tersebut, variabel `$a` diinisialisasi dengan nilai nol sehingga pemanggilan fungsi tersebut sebanyak tiga kali memberikan *output* yang sama, yaitu 3.

Variabel `$a` tersebut dapat mempertahankan nilainya yang terakhir dengan mendeklarasikannya menggunakan *keyword* `static`. Contoh:

```
1 function add() {
2     static $a = 0;
3
4     $a += 3;
5
6     echo "$a\n";
7 }
8
9 add();
10 add();
11 add();
12
13 // Output:
14 3
15 6
16 9
```

Dengan *keyword* `static`, sebuah variabel akan mempertahankan nilai yang terakhir disimpannya sehingga pada kode di atas, pemanggilan fungsi `add()` sebanyak tiga kali akan menambah nilai dari variabel `$a` sebanyak tiga kali pula dan menghasilkan output 3, 6, dan 9.

e) *Pass by Value* dan *Pass by Reference*

Saat memanggil suatu fungsi, kita bisa mengirimkan satu atau lebih *parameter*. Ada dua cara untuk mengirimkan parameter tersebut: *pass by value* dan *pass by reference*. Kode berikut memberikan contoh *pass by value*.

```
1 function add($b) {
2     $b += 3;
3 }
4
5 $a = 0;
6 add($a);
7 echo $a;
8
9 // Output:
10 0
```

Pada kode di atas, dideklarasikan sebuah fungsi yaitu `add()` yang memiliki sebuah parameter yaitu `$b` dan sebuah variabel yaitu `$a` yang diinisialisasi dengan nilai nol. Variabel `$a` yang bernilai nol dikirimkan sebagai parameter saat pemanggilan fungsi `add()`. Di dalam fungsi `add()`, variabel `$b` ditambah nilainya dengan nilai 3. Tetapi, karena menggunakan *pass by value*, yang dikirimkan sebagai parameter ke fungsi `add()` hanya nilai dari variabel `$a` saja sehingga variabel `$a` dan `$b` menjadi dua variabel yang terpisah dan tidak saling memengaruhi. Oleh karena itu, variabel `$a` tetap bernilai 3 setelah pemanggilan fungsi `add()`.

Perhatikan kode berikut yang memberikan contoh *pass by reference*.

```
1 function add(&$b) {
2     $b += 3;
3 }
```



```

4
5 $a = 0;
6 add($a);
7 echo $a;
8
9 // Output:
10 3

```

Perhatikan karakter *ampersand* (&) pada parameter di fungsi `add()` yang membuat parameter fungsi tersebut bersifat *pass by reference*. Dengan *pass by reference*, saat pemanggilan fungsi `add()`, yang dikirimkan sebagai parameter (variabel `$a`) adalah alamat dari variabel `$a`. Oleh karena itu, saat dilakukan penambahan nilai (statemen `$b += 3`), nilai dari variabel `$a` juga terpengaruh sehingga variabel `$a` di akhir kode bernilai 3.

f) *Default Parameters*

Penulisan parameter saat memanggil suatu fungsi dapat dibuat opsional. Perhatikan contoh berikut.

```

1 function greet($name = 'world') {
2     echo "Hello $name!";
3 }
4
5 greet();
6 greet('bro');
7
8 // Output:
9 Hello world!
10 Hello bro!

```

Pada kode di atas, fungsi `greet()` dideklarasikan dengan satu parameter yang bersifat *default*, yaitu `$name`. Fungsi tersebut kemudian dapat dipanggil dengan ataupun tanpa parameter. Jika dipanggil tanpa parameter, maka nilai *default* dari parameter tersebut, yaitu *string* "world", yang akan digunakan.

g) *Parameter Bebas*

Sebuah fungsi dapat memiliki jumlah parameter yang bebas. Fungsi tersebut dapat dipanggil dengan jumlah parameter berapapun. Untuk membuat fungsi tersebut, PHP menyediakan tiga fungsi *built-in*, yaitu:

- `func_get_args()` untuk mendapatkan semua parameter yang digunakan
- `func_num_args()` untuk mendapatkan banyaknya parameter yang digunakan
- `func_get_arg(n)` untuk mendapatkan parameter ke-n

Karena banyaknya parameter adalah bebas sesuai dengan parameter yang digunakan saat pemanggilan fungsi, fungsi tersebut ditulis dengan parameter kosong. Sebagai contoh, fungsi pada kode berikut bertujuan untuk menjumlahkan semua parameternya yang berupa angka. Banyaknya parameter tersebut bebas.

```

1 function sum() {
2     if (func_num_args() == 0) {

```

```

3     return false;
4 } else {
5     $sum = 0;
6
7     foreach (func_get_args() as $arg) {
8         $sum += $arg;
9     }
10
11     return $sum;
12 }
13 }
14
15 echo sum(1, 2, 3);
16 echo sum(2, 8);
17 echo sum(15);
18
19 // Output:
20 6
21 10
22 15

```

Fungsi `sum()` di atas dapat dipanggil dengan berapapun banyaknya parameter kemudian menjumlahkan semua parameter tersebut.

5.4 Prosedur Praktikum

a) Fungsi

Tulis kode berikut ini.

```

1 <?php
2     function mySum($numX, $numY){
3         $total = $numX + $numY;
4         return $total;
5     }
6
7     $myNumber = 0;
8     echo "Sebelum fungsi dijalankan, myNumber = ". $myNumber . "<br>";
9
10    $myNumber = mySum(3, 4);
11    echo "Setelah fungsi dijalankan, myNumber = ". $myNumber . "<br>";

```

Amati apa hasil dari kode di atas lalu simpulkan.

b) Return Value

Tulis kode berikut.

```
1 <?php
2     function myfunction(): int {
3         return 10;
4     }
5
6     echo myfunction();
```

Jalankan kode tersebut. Lalu ubah menjadi seperti berikut.

```
1 <?php
2     function myfunction(): int {
3         return 'ten';
4     }
5
6     echo myfunction();
```

Amati apa yang terjadi. Ubah lagi menjadi seperti berikut.

```
1 <?php
2     function myfunction(): string {
3         return 'ten';
4     }
5
6     echo myfunction();
```

Apa yang sebenarnya terjadi? Apakah tujuan dari bagian kode “: int” dan “: string” di belakang nama fungsi? Beri penjelasan.

c) Parameter Bebas

Buat sebuah fungsi yang menerima beberapa parameter dengan tipe *integer* dan menentukan angka terbesar dari parameter-parameter tersebut. Gunakan parameter bebas (point g di dasar teori) pada fungsi tersebut sehingga fungsi tersebut dapat dipanggil seperti berikut.

```
1 <?php
2     function get_max() {
3         // ...
4     }
5
6     echo get_max(10, 20);           // Output: 20
7     echo get_max(10, 20, 30);     // Output: 30
```

8

```
echo get_max(10, 20, 30, 40); // Output: 40
```

d)

Rekursi

Buatlah sebuah fungsi untuk menyelesaikan permasalahan berikut.

- 1 Perhitungan Fibonacci dengan sistem rekursif di mana input pada fungsi digunakan untuk menunjukkan bilangan ke sekian dari deret tersebut.
- 2 Perhitungan nilai pangkat, dengan input x (bilangan yang akan dikuadratkan) dan input y (bilangan pangkat).

Modul 6 : Array

6.1 Waktu Pelaksanaan Praktikum

Durasi kegiatan praktikum adalah **170 menit**, dengan rincian sebagai berikut.

- a. 15 menit untuk pengerjaan Tes Awal atau wawancara Tugas Pendahuluan
- b. 60 menit untuk penyampaian materi
- c. 45 menit untuk pengerjaan jurnal, tes akhir atau tugas
- d. 50 menit **pengayaan**

6.2 Tujuan

Setelah mengikuti praktikum ini, mahasiswa diharapkan dapat:

1. Mengolah *array* menggunakan PHP.
2. Melakukan pengurutan data di *array* menggunakan PHP.
3. Mengolah *array* multidimensi di PHP.

6.3 Dasar Teori

a) *Array*

Array adalah sekumpulan nilai yang disimpan secara berderet. Di PHP, *array* terdiri dari dua jenis: *array* terindeks dan *array* asosiatif. *Array* terindeks dapat diakses menggunakan indeks berupa angkaterurut yang dimulai dari nol. *Array* asosiatif tidak menggunakan indeks berupa angka, melainkan berupa *string*. Di kedua jenis *array* tersebut, tidak ada lebih dari satu indeks yang sama. Contoh:

```
1 echo $terindeks[10];
2 echo $asosiatif['nama'];
```

b) Pembuatan *Array*

Pembuatan *array* bisa dilakukan dengan melakukan *assignment* secara langsung seperti berikut.

```
1 // array terindeks
2 $email[0] = 'admin@filkom.ub.ac.id';
3 $email[1] = 'dekan@filkom.ub.ac.id';
4 $email[2] = 'kemahasiswaan@filkom.ub.ac.id';
5
6 // array asosiatif
7 $harga['sapu'] = 30000;
8 $harga['botol'] = 10000;
9 $harga['gelas'] = 9000;
```

Pembuatan *array* juga bisa dilakukan menggunakan sintaks `array()` seperti contoh berikut.

```

1 // array terindeks
2 $email = array('admin@filkom.ub.ac.id',
3               'dekan@filkom.ub.ac.id',
4               'kemahasiswaan@filkom.ub.ac.id');
5
6 // array asosiatif
7 $harga = array('sapu' => 30000,
8               'botol' => 10000,
9               'gelas', => 9000);

```

Tersedia juga sintaks alternatif yang lebih ringkas untuk melakukan hal yang sama:

```

1 // array terindeks
2 $email = ['admin@filkom.ub.ac.id',
3          'dekan@filkom.ub.ac.id',
4          'kemahasiswaan@filkom.ub.ac.id'];
5
6 // array asosiatif
7 $harga = ['sapu' => 30000,
8          'botol' => 10000,
9          'gelas', => 9000];

```

Kode berikut membuat *array* kosong.

```

1 $empty = array();
2 $empty = [];

```

Menambahkan elemen baru ke akhir suatu *array* dapat dilakukan menggunakan sintaks “[*i*]”. Contoh:

```

1 $hari = ['Senin', 'Selasa'];
2 $hari[] = 'Rabu';
3 // $hari = ['Senin', 'Selasa', 'Rabu'];

```

Untuk membuat *array* dengan elemen-elemen yang memiliki urutan tertentu, fungsi `range()` dapat digunakan. Contoh:

```

1 $angka = range(1, 5);
2 // $angka = [1, 2, 3, 4, 5];
3
4 $huruf = range('a', 'z');
5 // $huruf = ['a', 'b', 'c', 'd', 'e', 'f', ...]

```

Panjang dari sebuah *array* bisa didapatkan menggunakan fungsi `count()` atau `sizeof()`. Kedua fungsi tersebut identik. Contoh:

```

1 $angka = range(1, 5);
2 echo count($angka);
3 echo sizeof($angka);
4
5 // Output:
6 5
7 5

```

c) Konversi Antara *Array* dan Variabel

Array dapat dikonversi menjadi variabel dan sebaliknya. PHP menyediakan fungsi `extract()` untuk membuat beberapa variabel dari sebuah *array*. Contoh:

```
1 $arr = ['name' => 'Joko', 'age' => 56];
2 extract($arr);
3
4 echo $name;
5 echo $age;
6
7 // Output:
8 Joko
9 56
```

Fungsi `compact()` digunakan untuk membuat sebuah *array* berdasarkan beberapa variabel. Contoh:

```
1 $name = 'Joko';
2 $age = 56;
3 $arr = compact($name, $age);
4 // $arr = ['name' => 'Joko', 'age' => 56];
```

d) Perulangan Terhadap *Array*

Seringkali kita perlu melakukan sesuatu terhadap setiap elemen yang ada pada suatu *array*. Kita melakukannya dengan menggunakan perulangan (*looping*). Ada beberapa cara untuk melakukan hal tersebut.

Bentuk `for` adalah salah satu cara yang paling banyak digunakan untuk memproses *array*. Sintaks `for` di PHP mengadopsi sintaks `for` di bahasa pemrograman C sehingga formatnya sama. Contoh:

```
1 $emails = ['admin@filkom.ub.ac.id', 'admin@ub.ac.id'];
2
3 foreach ($i = 0; $i < sizeof($emails); $i++) {
4     $email = $emails[$i];
5     echo "Mengirim email ke $email";
6 }
7
8 // Output:
9 Mengirim email ke admin@filkom.ub.ac.id
10 Mengirim email ke admin@ub.ac.id
```

Bentuk `foreach` merupakan cara yang lebih ringkas daripada `for` dalam memproses *array* karena tidak melibatkan variabel iterasi (`$i`). Contoh:

```
1 $emails = ['admin@filkom.ub.ac.id', 'admin@ub.ac.id'];
2
3 foreach ($emails as $email) {
4     echo "Mengirim email ke $email";
5 }
6
7 // Output:
8 Mengirim email ke admin@filkom.ub.ac.id
```

```
9 Mengirim email ke admin@ub.ac.id
```

Bentuk `foreach` juga dapat digunakan pada *array* asiatif. Contoh:

```
1 $arr = ['name' => 'Joko', 'age' => 56];
2
3 foreach ($arr as $key => $value) {
4     echo "$key is $value";
5 }
6
7 // Output
8 name is Joko
9 age is 56
```

e) *Array* Multidimensi

Array multidimensi dapat dibuat seperti membuat *array* seperti biasa namun dengan elemen-elemen yang merupakan *array* juga. Contoh:

```
1 $row0 = array(1, 2, 3);
2 $row1 = array(4, 5, 6);
3 $row2 = array(7, 8, 9);
4 $multidimensi = array($row0, $row1, $row2);
```

Pengaksesan elemen-elemen dalam *array* juga harus dilakukan dengan indeks sebanyak dimensi dari *array* tersebut. Contoh:

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

6.4 Prosedur Praktikum

a) Pembuatan *Array*

Eksekusi kode di bawah ini dan tampilkan isi dari *array* tersebut.

```
1 $hari = [1 => 'Senin', 'Selasa', 'Rabu', 'Kamis', 'Jumat', 'Sabtu',
'Minggu'];
```

Jelaskan tentang kode tersebut.

Ubah kode tersebut menjadi seperti berikut.


```
1 $hari = ['Sen' => 'Senin', 'Selasa', 'Rabu', 'Kamis', 'Jumat', 'Sabtu',  
  'Minggu'];
```

Jelaskan tentang kode tersebut.

b) Perulangan Terhadap *Array*

Diberikan *array* sebagai berikut.

```
1 $info = [  
2   'komputer mekanik' => 'Charles Babbage',  
3   'world wide web' => 'Tim Berners-Lee',  
4   'mesin uap' => 'James Watt'  
5 ];
```

Dengan menggunakan `foreach`, buat tampilan sebagai berikut.

```
Charles Babbage menemukan komputer mekanik  
Tim Berners-Lee menemukan world wide web  
James Watt menemukan mesin uap
```


c) Pengurutan *Array*

Tulis dan eksekusi kode berikut.

```
1 $kode = [  
2   'Afghanistan' => 93,  
3   'Brazil' => 55,  
4   'Indonesia' => 62,  
5   'Malaysia' => 60,  
6   'Palestina' => 970  
7 ];  
8  
9 asort($kode);  
10 print_r($kode);  
11  
12 arsort($kode);  
13 print_r($kode);
```

```
14
15 ksort($kode);
16 print_r($kode);
17
18 krsort($kode);
19 print_r($kode);
```

Perhatikan output dari kode tersebut lalu jelaskan perbedaan antara fungsi `asort()`, `arsort()`, `ksort()`, dan `krsort()`.

d) Mengubah Urutan Array

Tulis dan eksekusi kode berikut.

```
1 $kode = [
2     'Afghanistan' => 93,
3     'Brazil' => 55,
4     'Indonesia' => 62,
5     'Malaysia' => 60,
6     'Palestina' => 970
7 ];
8
9 $array1 = array_reverse($kode);
10 $array2 = array_flip($kode);
11 shuffle($kode);
12
13 print_r($array1);
14 print_r($array2);
15 print_r($kode);
```

Perhatikan output dari kode tersebut lalu jelaskan tujuan dari fungsi `array_reverse()`, `array_flip()`, dan `shuffle()`.

e) Mencari Nilai dalam Array

Buatlah sebuah fungsi untuk mengecek apakah suatu nilai terdapat pada suatu *array*. Fungsi tersebut harus mengembalikan nilai *true* jika nilai yang dicari ada pada *array*, dan *false* jika tidak ada. Bentuk fungsinya adalah sebagai berikut.

```
1 function cari($array, $cari):bool {
2     // lakukan pencarian
3     // jika ditemukan, return true
4     // jika tidak, return false
5 }
```


Modul 7 : Form

7.1 Waktu Pelaksanaan Praktikum

Durasi kegiatan praktikum adalah **170 menit**, dengan rincian sebagai berikut.

- a. 15 menit untuk pengerjaan Tes Awal atau wawancara Tugas Pendahuluan
- b. 60 menit untuk penyampaian materi
- c. 45 menit untuk pengerjaan jurnal, tes akhir atau tugas
- d. 50 menit **pengayaan**

7.2 Tujuan

Setelah mengikuti praktikum ini, mahasiswa diharapkan dapat:

1. Memahami mekanisme *form* di HTML.
2. Membuat *form* menggunakan HTML.
3. Mengolah data yang terkirim dari *form* HTML menggunakan PHP.
4. Menentukan metode *form* yang cocok untuk berbagai kasus.

7.3 Dasar Teori

a) Form

Form dalam HTML adalah suatu bagian yang berfungsi sebagai input atau masukan dari pengguna yang kemudian akan diproses atau diolah untuk dapat digunakan sesuai dengan kebutuhan. *Form* adalah antarmuka pengguna (*user interface*) agar pengguna dapat berkomunikasi dengan sistem aplikasi. Setiap permintaan dari pengguna disebut *request* dan setiap hasil yang dari pemrosesanoleh *server* disebut *response*.

Form pada HTML dibuat dengan menggunakan elemen *form*. Elemen *form* harus membungkus seluruh elemen-elemen masukan form lain untuk memastikan masukan informasi pengguna dapat dibaca oleh aplikasi *web*.

```
1 <form action="process.php" method="post">
2     ....
3 </form>
```

Pada *form* terdapat dua atribut yang wajib dimiliki oleh elemen *form*. Atribut pertama adalah "**action**" yang berfungsi untuk memberitahukan *browser* alamat pengiriman dari data yang diisikan pengguna di dalam *form*. Parameter ini diisikan dengan sebuah URL pada *server* yang melakukan pemrosesan data. Atribut kedua yang wajib diisikan adalah atribut "**method**". Atribut ini memberitahukan *browser* dengan metode apa data akan dikirimkan ke *server* yang alamatnya diisikan pada atribut *action*. Terdapat dua nilai yang dapat diisikan pada atribut ini, yaitu "**GET**" dan "**POST**".

b) *Form Control*

Sebuah form berisi satu atau beberapa *form control*, antara lain *text field*, *text area*, *checkbox*, dan lain-lain. Setiap *form control* tersebut harus diberi nama agar dapat dibaca di sisi *server*. Tanpa nama, data yang dikirim tidak dapat karena setiap *form control* akan dibaca sebagai variabel yang berbeda di sisi *server*.

Pemberian nama tersebut dilakukan menggunakan atribut "name" pada masing-masing *form control*. Berikut contoh pemberian nama pada sebuah *text field*.

```
1 <input type="text" name="username">
```

Ada beberapa jenis *form control*, yaitu:

- Input teks

Kategori input ini terdiri dari beberapa jenis, yaitu:

- Input teks satu baris

```
1 <input type="text" name="...">
```

- Input teks banyak baris (*text area*)

```
1 <textarea name="..."></textarea>
```

- Input teks *password*

Dengan form control ini, setiap karakter yang diketikkan diganti dengan karakter bintang (*) atau *bullet* (•) sehingga tidak terbaca oleh pengguna.

```
1 <input type="password" name="...">
```

- Input teks pencarian (HTML5)

```
1 <input type="search" name="...">
```

- Input teks *email* (HTML5)

```
1 <input type="email" name="...">
```

- Input teks telepon (HTML5)

```
1 <input type="tel" name="...">
```

- Input teks URL (HTML5)

```
1 <input type="url" name="...">
```

Input teks pencarian, *email*, telepon, dan URL memiliki tampilan yang sama dengan input teks satu baris. Perbedaannya adalah pada tampilan *keyboard* saat dibuka di perangkat *mobile*. Saat menampilkan input teks pencarian, keyboard pada perangkat *mobile* akan

menampilkan tombol bertuliskan "Search". Saat menampilkan input teks *email*, tombol karakter "@" akan ditampilkan. Saat menampilkan input telepon, keyboard hanya akan memberikan tampilan angka. Saat menampilkan input teks URL, akan tampil tombol bertuliskan ".com". Hal tersebut dilakukan untuk mempermudah pengguna memasukkan data pada form.

Selain itu, *browser* juga dapat melakukan validasi terhadap format teks yang diisikan. Sebagai contoh, jika pada input teks email, dimasukkan alamat *email* yang tidak valid, maka *browser* akan memberikan pesan kesalahan pada saat *form* di-submit dan mencegah data dikirim sebelum semua data yang dimasukkan memiliki format yang valid.

- Tombol *submit* dan *reset*

Tombol *submit* digunakan untuk mengirim data yang telah diisikan di *form*. Tombol *reset* digunakan untuk mengosongkan isian pada setiap *form control* yang ada di *form* tersebut. Perlu diperhatikan bahwa saat ini tombol reset ini jarang sekali digunakan karena alasan pengalaman pengguna (*user experience*, UX). Pengguna yang telah mengisikan data pada suatu *form* seringkali secara tidak sengaja menekan tombol *reset* padahal sebenarnya ingin menekan tombol *submit* karena kedua tombol tersebut biasanya diletakkan berdekatan.

Teks pada kedua tombol tersebut dapat diganti dengan menambahkan atribut "value".

Contoh:

1	<code><input type="submit" value="Simpan"></code>
2	<code><input type="reset" value="Kosongi"></code>

- Tombol radio dan *checkbox*

Tombol radio bertujuan agar pengguna dapat memilih paling banyak satu pilihan dari beberapa pilihan yang tersedia. Sedangkan *checkbox* bertujuan agar pengguna dapat memilih lebih dari satu pilihan dari beberapa pilihan yang tersedia. Beberapa tombol radio dan beberapa *checkbox* dapat dikelompokkan dengan memberi nilai yang sama pada atribut *name*. Berikut contoh pembuatan tombol radio pada form pemilihan usia di mana pengguna hanya dapat memilih satu jenis usia.

1	<code><input type="radio" name="usia" value="17">Di bawah 17</code>
2	<code><input type="radio" name="usia" value="25">17-25</code>
3	<code><input type="radio" name="usia" value="35">25-35</code>
4	<code><input type="radio" name="usia" value="50">35-50</code>
5	<code><input type="radio" name="usia" value="17">Di atas 50</code>

Berikut contoh pembuatan *checkbox* pada *form* pemilihan hobi pengguna di mana pengguna dapat membuat beberapa opsi.

1	<code><input type="checkbox" name="hobi" value="membaca">Membaca</code>
2	<code><input type="checkbox" name="hobi" value="menulis">Menulis</code>
3	<code><input type="checkbox" name="hobi" value="olahraga">Olah raga</code>
4	<code><input type="checkbox" name="hobi" value="berkebun">Berkebun</code>

- *Dropdown menu*

Dropdown menu memiliki fungsi yang sama dengan tombol radio, yaitu memungkinkan pengguna untuk memilih satu dari beberapa pilihan yang tersedia. Namun, *dropdown menu*

lebih cocok digunakan apabila pilihan yang tersedia berjumlah banyak. Form control ini dibuat menggunakan tag `<select>` dan `<option>`. Berikut contoh pembuatan *dropdown menu*.

```
1 <select name="band">
2   <option value="dmasiv">
3   <option value="dewa">
4   <option value="thegroove">
5   <option value="hivi">
6   <option value="nidji">
7 </select>
```

- **File upload**

Form control ini memungkinkan pengguna dapat memilih satu file untuk kemudian di-*upload*. Contoh:

```
1 <input type="file" name="profilepic">
```

Agar dapat melakukan proses *upload*, pada tag `<form>`, harus ditambahkan atribut `enctype` seperti berikut.

```
1 <form action="..." method="post" enctype="multipart/form-data">
2   ...
3 </form>
```

- **Input tersembunyi**

Form control ini bertujuan untuk mengirimkan data yang tidak berasal dari input oleh pengguna. Salah satu contoh kasus penggunaan *form control* ini adalah untuk mengirimkan URL yang ingin dituju (*redirect*) saat user berhasil melakukan proses *login*. Contoh:

```
1 <input type="hidden" name="redirect" value="http://filkom.ub.ac.id">
```

- **Tanggal dan waktu (HTML5)**

Ini termasuk fitur baru yang ditambahkan secara *native* di *browser* di HTML5. Sebelum HTML5, *form control* ini hanya dapat dibuat menggunakan JavaScript. Ada beberapa jenis dari input teks ini, yaitu:

- Tanggal

```
1 <input type="date">
```

- Waktu

```
1 <input type="time">
```

- Tanggal dan waktu

```
1 <input type="datetime">
```

- Tanggal dan waktu lokal

```
1 <input type="datetime-local">
```

- Bulan

```
1 <input type="month">
```

- Pekan

```
1 <input type="week">
```

- Input numerik (HTML5)

Ada dua jenis dari input numerik, yaitu:

- Angka

```
1 <input type="number">
```

- Rentang

```
1 <input type="range">
```

- Input warna (HTML5)

Dengan *form control* ini, *browser* dapat menampilkan *pop-up* untuk memilih warna. Contoh:

```
1 <input type="color" name="favorit">
```

c) Metode GET dan POST

Data dari *form* yang menggunakan metode *get* dikirimkan ke *server* melalui URL pada bagian **query string**. Pada URL, nama *domain* dan *query string* dipisahkan oleh karakter tanda tanya (?). *Query string* memiliki format:

```
field1=nilai1&field2=nilai2&...
```

Sebagai contoh, perhatikan data berikut.

Field	Nilai
nama	Joko
usia	56
alamat	Jakarta

Data tersebut dapat dituliskan dalam format *query string* seperti berikut.

```
nama=Joko&usia=56&alamat=Jakarta
```

Sehingga bentuk *query string* tersebut pada URL dengan nama *domain* "domain.com" adalah:

```
http://domain.com?nama=Joko&usia=56&alamat=Jakarta
```

Karena data dari *form* yang menggunakan metode *get* dikirimkan melalui URL, maka pengguna dapat membaca data tersebut melalui *address bar* pada *browser*. Oleh karena itu, untuk alasan keamanan, metode ini tidak baik untuk digunakan untuk mengirimkan data yang bersifat sensitif. Form dengan

data yang sensitif, seperti *password*, harus dikirimkan menggunakan metode POST. Selain itu, metode GET memiliki keterbatasan pada kapasitas jika dibandingkan dengan metode POST.

Salah satu contoh kasus penggunaan metode GET dalam sebuah *form* adalah pada *form* pencarian. *Keyword* yang dikirimkan dari sebuah *form* pencarian umumnya tidak berisi informasi yang sensitif. Selain itu, keuntungan penggunaan metode GET pada *form* pencarian adalah URL dari hasil pencarian tersebut dapat dibagi (*share*) ke orang lain dan juga dapat di-*bookmark* di *browser* pengguna.

Berikut contoh *form* dalam HTML dengan metode GET.

```
1 <form action="search.php" method="get">
2   <input type="search" name="keyword">
3   <input type="submit" value="Search">
4 </form>
```

Data dari *form* yang menggunakan metode POST dikirimkan ke *server* melalui HTTP *request header* tanpa mengubah URL sehingga URL hanya berisi nama *domain* saja sehingga data yang dikirimkan tidak terbaca oleh pengguna.

Berikut contoh form dalam HTML dengan metode POST.

```
1 <form action="proses.php" method="post">
2   <input type="text" name="nama">
3   <input type="submit" value="Go">
4 </form>
```

Untuk mengakses data yang telah diisi oleh pengguna melalui sebuah *form*, PHP menyediakan tiga *array* secara built-in: `$_GET`, `$_POST`, dan `$_FILES`. Ketiga *array* tersebut merupakan *array* asosiatif dengan indeks sesuai dengan nilai pada atribut `name` di masing-masing *form control*. Sebagai contoh, perhatikan *form* berikut.

```
1 <form action="login.php" method="post">
2   <input type="text" name="username">
3   <input type="password" name="password">
4   <input type="submit" value="Login">
5 </form>
```

Saat *form* di atas di-*submit*, data *username* dan *password* yang diisi oleh pengguna dapat dibaca di kode PHP menggunakan kode PHP berikut.

```
1 <?php
2   $username = $_POST['username'];
3   $password = $_POST['password'];
```

7.4 Prosedur Praktikum

a) Metode GET dan POST

Tulis kode berikut.

```
1 <body>
2   <form action="proses.php" method="post">
3     <input type="text" name="nama">
4     <input type="submit" value="Go">
5   </form>
6 </body>
```

Jalankan kode tersebut di *browser* lalu ganti metodenya dengan `GET`. Lihat perbedaannya pada URL di *browser*. Beri kesimpulan.

b) Menentukan Metode

Untuk beberapa kasus berikut, tentukan metode apakah yang cocok untuk digunakan dan beri alasan.

1. *Form* untuk *login*.
2. *Form* untuk mengirim gambar desain poster ke *website* percetakan.
3. *Form* untuk mencari suatu artikel di situs berita.
4. *Form* untuk *search engine*.

c) Form

Tulis kode di bawah ini.

```
1 <form action="proses02.php" method="post" name="input">
2   Nama Anda: <input type="text" name="nama" required>
```

```
3 <br>
4 <input type="submit" name="input" value="Input">
5 </form>
```

Kemudian tulis kode di bawah ini.

```
1 <?php
2     if (isset($_POST['input'])) {
3         $nama = $_POST['nama'];
4         echo "Nama Anda: <b>$nama</b>";
5     }
```

- a. Jalankan kode di atas kemudian isikan data input dan tekan tombol input. Apa yang terjadi?
- b. Ulangi dengan mengosongkan data input dan amati apa yang terjadi.
- c. Jelaskan alur pengiriman data dari *form* yang ada di kode tersebut!

d) Required

Ubah kembali kode pada Latihan 3 dengan menghapus tulisan `required` pada baris 2. Jalankan kode dan kosongi data nama kemudian tekan tombol Input. Apa yang terjadi? Jelaskan apa fungsi dari kode `required`.

e) Upload

Tulis kode berikut.

```
1 <form enctype="multipart/form-data" action="upload.php" method="post">
2     Choose a file to upload:
3     <input name="uploadedfile" type="file" /> <br>
4     <input type="submit" value="Upload File" />
5 </form>
```

Tulis kode berikut.

```
1 <?php
```

```
2 $target_path = "uploads/";
3 $target_path = $target_path . basename($_FILES['uploadedfile']['name']);
4
5     if(move_uploaded_file($_FILES['uploadedfile']['tmp_name'],
6 $target_path)) {
7         echo "The file ". basename( $_FILES['uploadedfile']['name']). " has
8 been uploaded";
9     } else {
10        echo "There was an error uploading the file, please try again!";
11    }
```

Di manakah posisi file setelah proses upload berhasil?

Apakah hasil output dari kode tersebut setelah dijalankan?

Modul 8 : String

8.1 Waktu Pelaksanaan Praktikum

Durasi kegiatan praktikum adalah **170 menit**, dengan rincian sebagai berikut.

- a. 15 menit untuk pengerjaan Tes Awal atau wawancara Tugas Pendahuluan
- b. 60 menit untuk penyampaian materi
- c. 45 menit untuk pengerjaan jurnal, tes akhir atau tugas
- d. 50 menit **pengayaan**

8.2 Tujuan

Setelah mengikuti praktikum ini, mahasiswa diharapkan dapat:

1. Memahami interpolasi variabel di dalam *string*.
2. Mengolah *string* menggunakan PHP.
3. Mengubah kapital *string* di PHP sesuai kebutuhan.
4. Memahami perbedaan operator pembandingan *string* `==` dan `===`.

8.3 Dasar Teori

a) String

Banyak data yang diolah di sebuah aplikasi, termasuk aplikasi web, berbentuk *string*. Contohnya data nama, *password*, alamat, nomor kartu kredit, histori pembelian, dan lain-lain. Karena itu, PHP menyediakan banyak fungsi untuk bekerja dengan *string*.

Ada tiga cara untuk menulis *literal* bertipe *string*, yaitu dengan tanda petik tunggal, tanda petik ganda, dan format *here documents* (*heredoc*).

b) Interpolasi Variabel

Interpolasi adalah proses mengganti nama variabel di dalam suatu *string* dengan nilai dari variabel tersebut. Jika kita menulis *string* menggunakan tanda petik ganda atau format *heredoc*, maka akan terjadi interpolasi variabel. Contoh:

```
1 $var = 'world'
2 echo "Hello $var";
3 // output: Hello world
```

Namun jika kita menggunakan tanda petik tunggal, maka tidak akan terjadi interpolasi variabel. Contoh:

```
1 $var = 'world'
2 echo 'Hello $var';
```

```
3 // output: Hello $var
```

c) *Here Documents*

Format *here documents* (*heredoc*) digunakan untuk menulis *literal* bertipe *string* lebih dari satu baris. Sintaks ini menggunakan operator “<<<” diikuti dengan sebuah penanda yang menandai awal dan akhir *string*. Penanda tersebut dapat berupa *string* apapun asalkan penanda awal dan akhir adalah *string* yang sama. Contoh:

```
1 $var = <<< END
2 Fakultas Ilmu Komputer
3 Universitas Brawijaya
4 Malang
5 END;
```

d) *Mencetak String*

Ada beberapa fungsi untuk mencetak *string* di PHP: `echo`, `print()`, `printf()`, `print_r()`, dan `var_dump()`. Fungsi `echo` merupakan fungsi yang paling sering digunakan untuk mencetak *string* di PHP. Fungsi ini dapat digunakan dengan ataupun tanpa tanda kurung. Contoh:

```
1 echo 'Hello world!';
2 echo('Hello world!');
```

Fungsi `echo` dapat mencetak beberapa nilai *string* sekaligus. Contoh:

```
1 echo 'Hello', ' ', 'world', '!';
```

Tetapi tanda kurung tidak dapat digunakan untuk mencetak beberapa nilai *string* sekaligus. Contoh:

```
1 // Error!
2 echo('Hello', ' ', 'world', '!');
```

Fungsi `print()` sama dengan `echo`. Hanya saja, penggunaannya harus dengan tanda kurung. Contoh:

```
1 print('Hello world!');
```

Fungsi `printf()` bertujuan untuk mencetak *string* terformat. Fungsi ini mengadopsi dari fungsi `printf()` yang ada di bahasa pemrograman C. Format yang ingin digunakan dituliskan dengan awalan tanda persen (%). Contoh:

```
1 printf('Pi is %.2f', 3.1415);
2 // Output:
3 Pi is 3.14
```

Fungsi `print_r()` sering digunakan untuk untuk mencetak *array* dan objek dengan format yang mudah dibaca. Contoh:

```
1 $arr = ['name' => 'Joko', 'age' => 56];
2 $obj = class C {
3     var $name = 'Joko';
4 }
5 $inst = new C;
6
7 print_r($arr);
8 print_r($inst);
9
10 Output:
11 Array
12 (
13     [name] => Joko,
14     [age] => 56)
15
16 Object
17 (
18     [name] => Joko)
```

e) Mengakses Karakter

Karakter-karakter di suatu *string* dapat diakses menggunakan tanda kurung kurawal (“{” dan “}”).
Contoh:

```
1 $str = 'PHP';
2
3 for ($i = 0; $i < strlen($str); $++) {
4     echo $str{$i}, "\n";
5 }
6
7 // Output:
8 P
9 H
10 P
```

f) Menghapus *Whitespace*

Whitespace adalah karakter-karakter kosong yang terdiri dari spasi, *tab*, karakter baris baru, dan beberapa karakter kosong lainnya. Karakter-karakter tersebut menjadi tidak bermakna bila berada di luar *string*. PHP menyediakan tiga fungsi untuk menghapus karakter-karakter tersebut. Fungsi `ltrim()` menghapus *whitespace* yang berada di kiri *string*, fungsi `rtrim()` menghapus *whitespace* yang berada di kanan *string*, dan fungsi `trim()` menghapus *whitespace* yang berada di kanan dan kiristring. Contoh:

```
1 $str = "  PHP \n ";echo
2 ltrim($str); echo
3 rtrim($str); echo
4 trim($str);
5
6 // Output:
7 "PHP \n "
8 "PHP"
9 "PHP"
```

g) Mengubah Huruf Kapital

PHP menyediakan beberapa fungsi untuk mengubah huruf kapital. Contoh:

```
1 <?php
2 $str1 = 'TI UIN Maulana Malik Ibrahim Malang';
3 $str2 = 'ti uin maulana malik ibrahim malang';
4 echo strtolower($str1)."<br>";
5 echo strtoupper($str1)."<br>";
6 echo ucfirst($str2)."<br>";
7 echo ucwords($str1)."<br>";
8 echo ucwords($str2)."<br>";
9 ?>
```

// Output:
ti uin maulana malik ibrahim malang
TI UIN MAULANA MALIK IBRAHIM MALANG
Ti uin maulana malik ibrahim malang
TI UIN Maulana Malik Ibrahim Malang
Ti Uin Maulana Malik Ibrahim Malang

h) Perbandingan String

PHP menyediakan dua operator dan enam fungsi untuk membandingkan *string*. Operator `==` dan `===` dapat digunakan untuk membandingkan dua variabel dengan sedikit perbedaan antara kedua operator tersebut. Operator `==` hanya membandingkan isi dari kedua variabel sedangkan operator `===` membandingkan isi dan tipe data dari kedua variabel tersebut. Contoh:

```
1 $var1 = 13;
2 $var2 = '13';
3
4 if ($var1 == $var2) {
5     echo '== sama';
6 } else {
7     echo '== tidak sama';
8 }
9
10 if ($var2 === $var2) {
11     echo '=== sama';
12 } else {
13     echo '=== tidak sama';
14 }
15
16 // Output:
17 == sama
18 === tidak sama
```

8.4 Prosedur Praktikum

a) Mengubah Kapital

Tulis kode untuk mengubah *string* berikut:

```
'BAPAK PRESIDEN JOKO WIDODO'
```

menjadi:


```
'Bapak Presiden Joko Widodo'
```

Jelaskan langkah-langkahnya.

b) Memotong String

Tulis kode untuk memisahkan suatu *string* menjadi suatu *array* bertipe *string* berdasarkan karakter titik. Contoh:

```
'Program Studi.Teknik.Informatika.UIN Maliki Malang'
```

menjadi:

```
['Program Studi', 'Teknik', 'Informatika', 'UIN Maliki Malang']
```

Jelaskan langkah-langkahnya.

c) Posisi String

Tulis kode di bawah ini.

```
1 <?php
2     $numberedString = "1234567890123456789012345678901234567890";
3
4     $fivePos = strpos($numberedString, "5");
5     echo "The position of 5 in our string was $fivePos";
6
7     $fivePos2 = strpos($numberedString, "5", $fivePos + 1);
8     echo "<br>The position of the second 5 was $fivePos2";
```

- a. Apa maksud dari `$fivePos + 1`?
- b. Modifikasilah kode diatas sehingga keluaran program sama dengan keluaran program awal namun tanpa menggunakan keyword `strpos`!

d) Mengganti *String*

Tulis kode di bawah ini.

```
1 <?php
2 //string that needs to be customized
3 $rawstring = "Welcome Birmingham parents. Your replaceme is a pleasure
4 to have!";
5
6 //male string
7 $malestr = str_replace("replaceme", "son", $rawstring);
8
9 //female string
10 $femalestr = str_replace("replaceme", "daughter", $rawstring);
11
12 echo "Son: ". $malestr . "<br>";
13 echo "Daughter: ". $femalestr;
```

Apa hasil dari kode di atas? Simpulkan.

e) Mengganti *String*

Tulis kode berikut.

```
1 <?php
2 //string that needs to be customized
3 $rawstring = "Welcome Birmingham parent! <br>
4 Your offspring is a pleasure to have!
5 We believe pronoun is learning a lot.<br>
6 The faculty simple adores pronoun2 and you
7 can often hear them say \"Attah sex!\"<br>";
8
9 //placeholders array
10 $placeholders = array('offspring', 'pronoun', 'pronoun2', 'sex');
11
12 //male replace values array
13 $malevals = array('son', 'he', 'him', 'boy');
14
15 //female replace values array
16 $femalevals = array('daughter', 'she', 'her', 'girl');
```

```
17
18 //male string
19 $malestr = str_replace($placeholders, $malevals, $rawstring);
20
21 //female string
22 $femalestr = str_replace($placeholders, $femalevals, $rawstring);
23
24 echo "Son: ". $malestr . "<br>";
25 echo "Daughter: ". $femalestr;
```

Apa hasil dari kode di atas? Apa maksud dari keyword `str_replace($placeholders, $malevals, $rawstring)`? Simpulkan.

f) Memotong String

Tulis kode berikut.

```
1 <?php
2 $someWords = "Please don't blow me to pieces.";
3
4 $wordChunks = explode(" ", $someWords);
5
6 for($i = 0; $i < count($wordChunks); $i++){
7     echo "Piece $i = $wordChunks[$i] <br>";
8 }
```

Apa hasil dari kode di atas? Jelaskan maksud baris keempat.

Modul 9 : Operasi File

9.1 Waktu Pelaksanaan Praktikum

Durasi kegiatan praktikum adalah **170 menit**, dengan rincian sebagai berikut.

- 15 menit untuk pengerjaan Tes Awal atau wawancara Tugas Pendahuluan
- 60 menit untuk penyampaian materi
- 45 menit untuk pengerjaan jurnal, tes akhir atau tugas
- 50 menit **pengayaan**

9.2 Tujuan

Setelah mengikuti praktikum ini, mahasiswa diharapkan dapat:

1. Memahami tahap-tahap dalam membuka *file*.
2. Membaca isi file menggunakan PHP.
3. Menulis suatu konten ke file menggunakan PHP.
4. Menggunakan berbagai fungsi utilitas yang tersedia di PHP.

9.3 Dasar Teori

a) Operasi File

File dapat dijadikan alternatif untuk menyimpan data-data berukuran kecil yang dimasukkan user melalui *form*. File juga banyak digunakan untuk menyimpan data-data yang akan digunakan untuk keperluan konfigurasi sistem. PHP menyediakan fasilitas untuk mengakses *file* (membaca, mengubah dan menyimpan).

Sebelum mengakses sebuah *file*, maka kita harus membuka dulu *file* yang akan kita proses. PHP menyediakan fungsi *built-in* untuk membuka *file* yaitu `fopen()`. Bentuk penggunaan fungsi `fopen()` adalah sebagai berikut.

```
$file_handler = fopen(nama_file, mode);
```

Parameter `mode` bisa kita isikan dengan kode seperti daftar di bawah ini.

Mode	Keterangan
r	Membuka <i>file</i> untuk proses pembacaan (<i>reading</i>)
r+	Membuka <i>file</i> untuk proses pembacaan (<i>reading</i>) dan penulisan
w	Membuka <i>file</i> untuk proses penulisan (<i>writing</i>)
w+	Membuka <i>file</i> untuk proses penulisan dan pembacaan.
a	Membuka <i>file</i> untuk proses penambahan data (<i>appending</i>)
a+	Membuka <i>file</i> untuk proses pembacaan dan penambahan data.

Setelah membuka *file* dan selesai diproses dengan mode yang ada pada daftar tersebut, maka kita harus menutup sebagai tahapan terakhir dari proses. PHP menyediakan fungsi yang berfungsi untuk menutup *file*, yaitu `fclose()`.

```
fclose($file_handler);
```

Contoh:

```
1 <?php
2     $filename = 'test.txt';
3     $file_handle = fopen($filename, 'w') or die('Can't open file');
4     // operasi file
5     fclose($file_handle);
```

b) Membaca File

PHP menyediakan satu fungsi yang merupakan fungsi ringkas untuk membaca isi suatu *file*, yaitu `file_get_contents()`. Berikut contoh penggunaannya.

```
1 <?php
2     $filename = 'data.txt';
3     $content = file_get_contents($filename);
```

c) Menulis File

Menulis ke suatu file dilakukan seperti contoh berikut.

```
1 <?php
2     $filename = 'test.txt';
3     $file_handle = fopen($filename, 'a+') or die('Can't open file');
4     $content = 'This is the file content';
5
6     if (flock($file_handle, LOCK_EX)) {
7         if (!fwrite($file_handle, $content)) {
8             echo "Cannot write to file $filename";
9         }
10
11     flock($file_handle, LOCK_UN);
12 }
```

Saat menulis ke suatu *file*, ada kemungkinan file yang sedang ditulis juga sedang ditulis oleh proses yang lain. Kemungkinan tersebut menjadi semakin tinggi seiring bertambahnya pengguna yang mengakses aplikasi tersebut. Jika hal tersebut terjadi, maka *file* tersebut dapat mengalami kesalahan (*error*) pada proses penulisan sehingga isinya menjadi tidak sesuai. Untuk mengatasi hal tersebut, ada mekanisme penguncian (*locking*) yang memastikan suatu *file* hanya diakses secara eksklusif oleh satu proses pada suatu waktu. Jika ada proses lain yang mencoba mengakses *file* tersebut pada waktu yang sama, maka proses tersebut harus menunggu hingga proses sebelumnya selesai.

Di PHP, fungsi `flock()` digunakan untuk melakukan penguncian (*locking*) *file* yang ingin ditulis agar tidak terjadi konflik dengan proses yang lain. Fungsi tersebut menerima dua parameter, yaitu *file*

handle dan sebuah *flag*. Untuk mengunci *file* yang akan ditulis, digunakan *flag* `LOCK_EX`. Untuk melepas kunci yang telah dibuat sebelumnya, digunakan *flag* `LOCK_UN`.

PHP juga menyediakan satu fungsi yang merupakan fungsi ringkas untuk menulis ke suatu *file*, yaitu `file_put_contents()`. Berikut contoh penggunaannya.

```
1 <?php
2     $filename = 'data.txt';
3     $content = 'PHP is fun!';
4     $content = file_put_contents($filename, $content);
```

Jika *file* yang akan ditulis belum ada, maka fungsi tersebut akan membuat *file* baru sesuai dengan nama yang diberikan. Jika *file* telah ada sebelumnya, maka fungsi akan menggantikan isi *file* yang lama dengan isi *file* yang baru. Jika isi *file* yang baru ingin ditambahkan di akhir *file* yang lama tanpa menghapus isi *file* yang lama, kita dapat menggunakan *flag* `FILE_APPEND` sebagai parameter ketiga. Contoh:

```
1 <?php
2     $filename = 'data.txt';
3     $content = 'PHP is fun!';
4     $content = file_put_contents($filename, $content, FILE_APPEND);
```

Secara *default*, fungsi tersebut tidak mengunci *file* yang akan ditulis. Untuk melakukan penguncian *file*, kita dapat menggunakan *flag* `LOCK_EX`. Contoh:

```
1 <?php
2     $filename = 'data.txt';
3     $content = 'PHP is fun!';
4     $content = file_put_contents($filename, $content, LOCK_EX);
```

Beberapa *flag* dapat digunakan sekaligus dengan memisahkan beberapa *flag* tersebut dengan karakter *pipe* (`|`). Contoh:

```
1 <?php
2     $filename = 'data.txt';
3     $content = 'PHP is fun!';
4     $content = file_put_contents($filename, $content, FILE_APPEND | LOCK_EX);
```

d) Fungsi Utilitas

PHP menyediakan beberapa fungsi untuk beragam tujuan yang seringkali diperlukan, yaitu:

- `mkdir()`. Untuk membuat direktori baru di *server*.
- `file_exists()`. Untuk mengecek keberadaan suatu *file* atau direktori di *server*.
- `filesize()`. Untuk mengetahui ukuran suatu *file*.

9.4 Prosedur Praktikum

a) Latihan 1

Buatlah file txt dengan nama `testFile.txt`, dan berisi:

Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang

Tulis kode berikut.

```
1 <?php
2     $myFile = "testFile.txt";
3     $fh = fopen($myFile, 'r');
4     $theData = fread($fh, 5);
5     fclose($fh);
6     echo $theData;
```

- Apa yang dihasilkan dari kode tersebut? Lalu ganti `fread($fh, 5);` dengan `fread($fh, filesize($myFile));` dan bandingkan serta jelaskan!
- Jelaskan mengapa ada huruf `r` pada baris ketiga dan jelaskan pula artinya!

b) Latihan 2

Tulis kode berikut.

```
1 <?php
2     $myFile = "testFile.txt";
3     $fh = fopen($myFile, 'r');
4     $theData = fgets($fh);
5     fclose($fh);
6     echo $theData;
```

- Bandingkan dengan kode di Latihan 1, dan buatlah sebuah kesimpulan terhadap fungsi `fread()`.
- Modifikasi kode diatas sehingga dapat menampilkan isi baris kedua dari `testFile.txt`!

c) Latihan 3

Tulis kode berikut.

```
1 <?php
2 $myFile = "testFile.txt";
3 $fh = fopen($myFile, 'w') or die("can't open file");
4 $stringData = "Bobby Bopper\n";
5 fwrite($fh, $stringData);
6 $stringData = "Tracy Tanner\n";
7 fwrite($fh, $stringData);
8 fclose($fh);
```

Apa isi dari file testFile.txt? Jelaskan.

d) Latihan 4

Tulis kode berikut.

```
1 <?php
2 $myFile = "testFile.txt";
3 $fh = fopen($myFile, 'a') or die("can't open file");
4 $stringData = "New Stuff 1\n";
5 fwrite($fh, $stringData);
6 $stringData = "New Stuff 2\n";
7 fwrite($fh, $stringData);
8 fclose($fh);
```

- a. Apa isi dari file testFile.txt?
- b. Bandingkan dengan kode di soal sebelumnya. Jelaskan apa beda dan maksudnya.

e) Latihan 5

- Buatlah file `guestbook.txt` yang isinya kosong.

- Buatlah form dengan isian `nama` dan `komentar`. Form ini akan mengirimkan data nama dan komentar user untuk selanjutnya akan disimpan dalam file `guestbook.txt`. Simpan dengan pemisah “;” antara nama dan komentar.
- Buatlah kode untuk menampilkan hasil penyimpanan pada file `guestbook.txt`. Gunakan fungsi `explode()` dengan pemisah “;” untuk menampilkan data.

MODUL 10

KONEKSI PHP DENGAN MariaDB

A. TUJUAN:

1. Memahami langkah-langkah koneksi PHP dengan MariaDB.
2. Memahami perbedaan pengambilan record dari database.

B. DASAR TEORI

Langkah-langkah koneksi PHP- MariaDB

1. Membuka koneksi ke server MariaDB. **mysqli_connect()**

Digunakan untuk melakukan uji dan koneksi kepada server database MariaDB.

Sintaks:

```
$conn = mysqli_connect ("host","username","password");
```

\$conn adalah nama variabel penampung status hasil koneksi kepada database.
host adalah nama host atau alamat server database MariaDB.
username adalah nama user yang telah diberi hak untuk dapat mengakses server database.
password adalah kata sandi untuk username untuk dapat masuk ke dalam database.

2. Mengambil sebuah query dari sebuah database. **mysqli_query()**

Digunakan untuk melakukan eksekusi perintah SQL untuk memanipulasi database yang berhasil dilakukan koneksinya menggunakan **mysqli_connect()**.

Sintaks:

```
$hasil = mysqli_query("SQL Statement");
```

\$hasil akan berupa record set apabila SQL Statement berupa perintah select.

Catatan: *mysqli_query ()* bisa diganti dengan nama variable dari *mysqli_connect ()*.

Contoh :

```
$koneksi=mysqli_connect ();
```

```
$a=$koneksi->query();
```

```
$a=mysqli_query();
```

Dua variabel \$a diatas nilai sama. Hanya cara pemanggilannya (sintaks) nya berbeda.

3. Mengambil record dari table:

a. mysqli_fetch_array()

Digunakan untuk melakukan pemrosesan hasil query yang dilakukan dengan perintah **mysqli_query()**, dan memasukkannya ke dalam array asosiatif, array numeris atau keduanya.

Sintaks :

```
$row = mysqli_fetch_array($hasil);
```

\$row adalah array satu record dari record \$hasil yang diproses nomor record sesuai dengan nomor urut dari proses **mysqli_fetch_array** yang sedang dilakukan.

\$hasil adalah record set yang akan diproses.

b. mysqli_fetch_assoc()

Fungsi ini hampir sama dengan fungsi **mysqli_fetch_array()**, hanya saja array yang dihasilkan hanya array asosiatif.

Sintaks:

```
$row = mysqli_fetch_assoc($hasil);
```

c. mysqli_fetch_row()

Fungsi ini hampir sama dengan fungsi **mysqli_fetch_array()**, hanya saja array yang dihasilkan hanya array numeris.

Sintaks:

```
$row = mysqli_fetch_row($hasil);
```

d. mysqli_num_rows()

Fungsi ini digunakan untuk menghitung jumlah record yang ada pada database.

Sintaks:

```
$jml = mysqli_num_rows($hasil);
```

\$jml akan memiliki nilai sesuai dengan jumlah record yang ada.

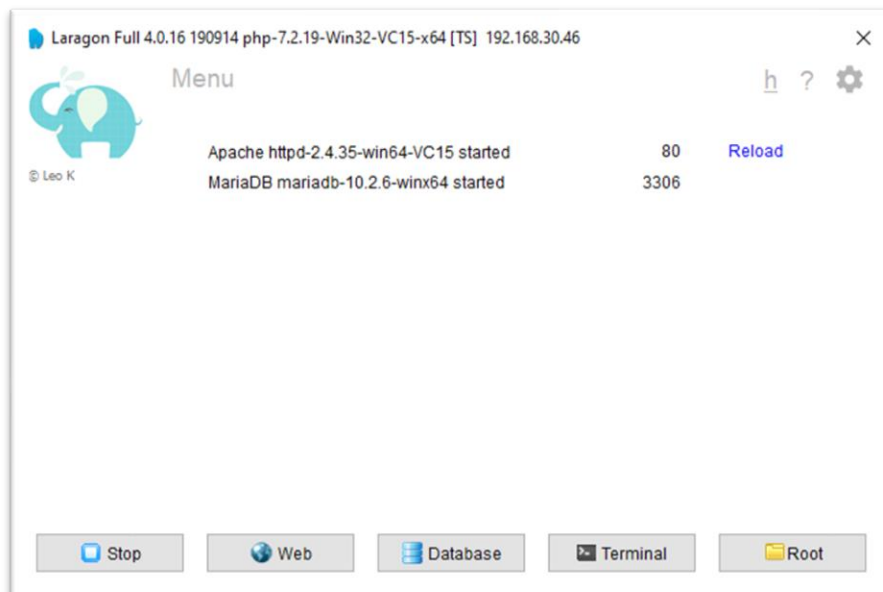
C. LATIHAN – LATIHAN:

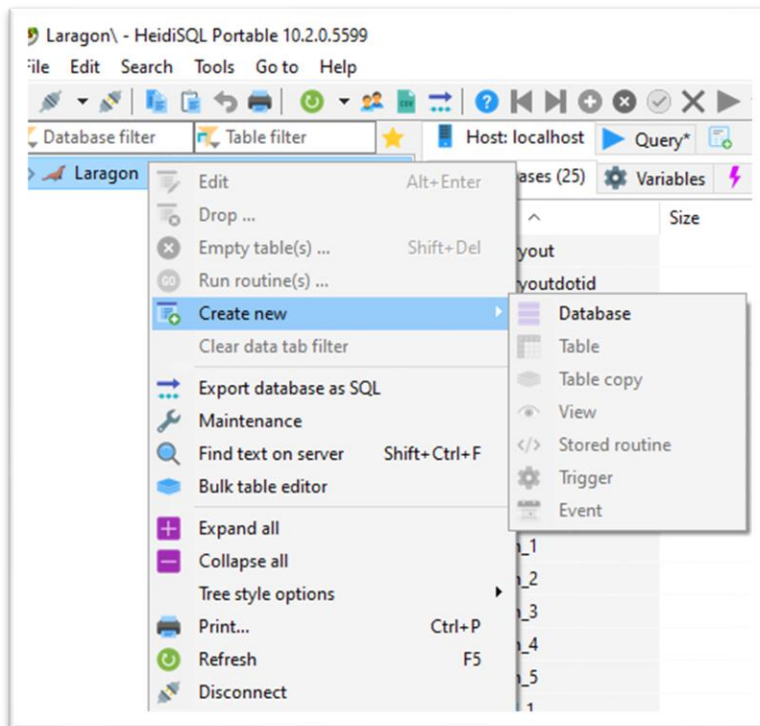
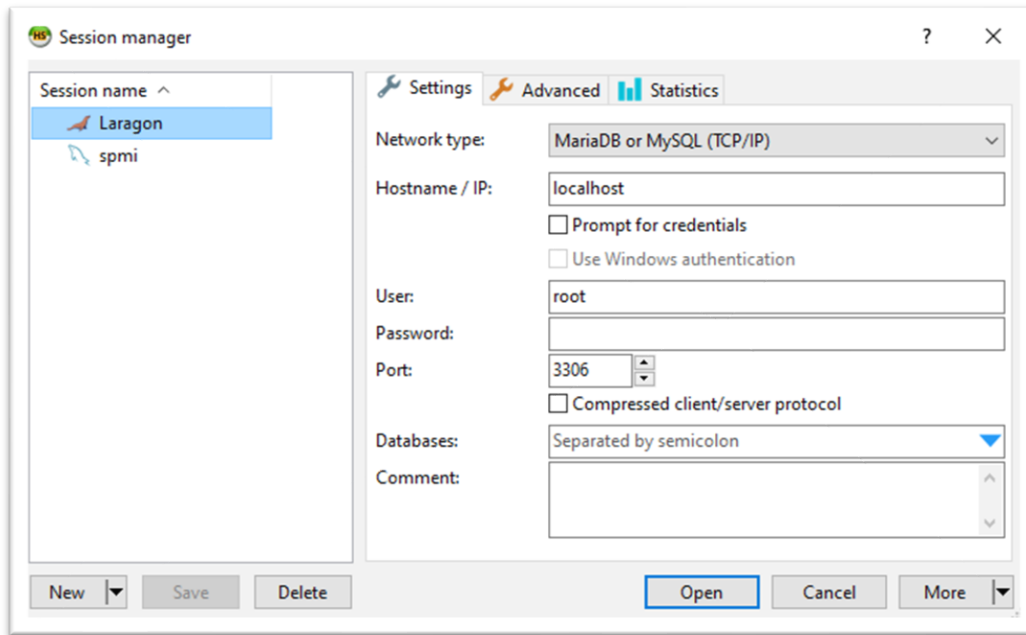
1. Menguji koneksi PHP dengan MariaDB (nama file koneksi.php)

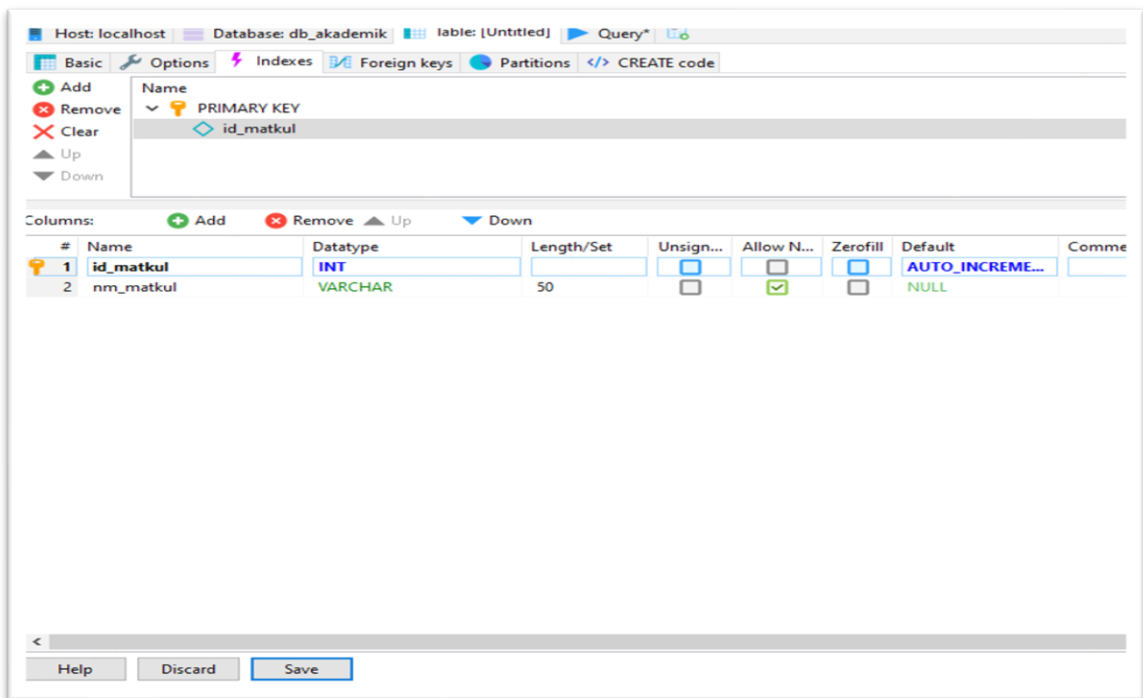
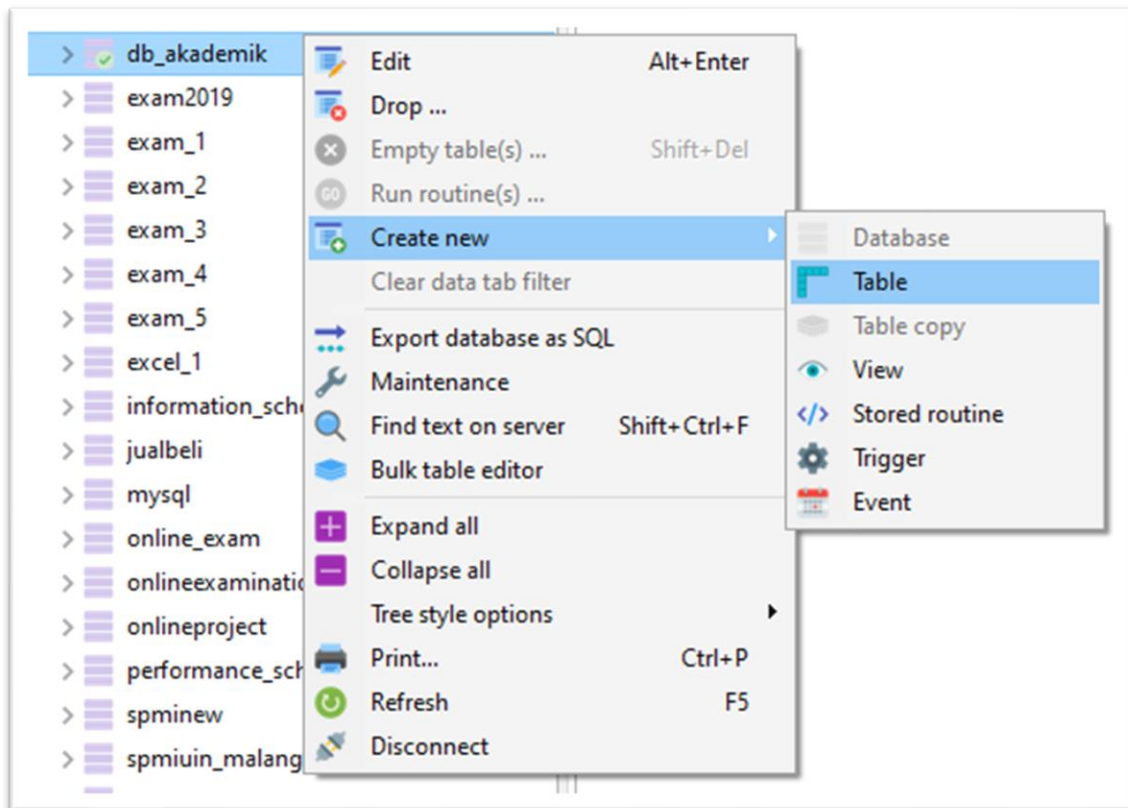
```
1 <html>
2 <head>
3 <title>Koneksi Database MySQL</title>
4 </head>
5 <body>
6
7 <h1>Demo koneksi database MySQL</h1> <?
8   $host = "localhost";//untuk host
9   $username = "root";//untuk username
10  $password = ""; //untuk password
11  $database = "db_akademik";//untuk nama database
12  $koneksi = mysqli_connect($host, $username, $password, $database);
13  if ($koneksi) {
14    echo "OK";
15  } else {
16    echo "Server not connected";
17  }
18  ?>
19 </body>
20 </html>
```

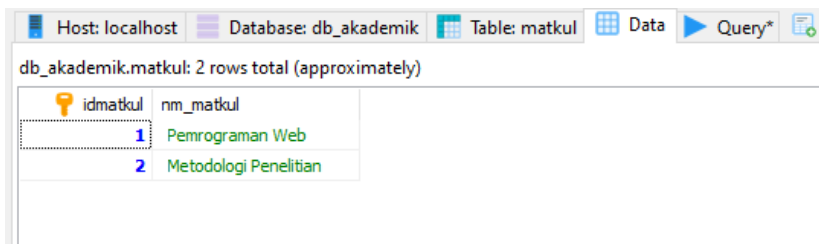
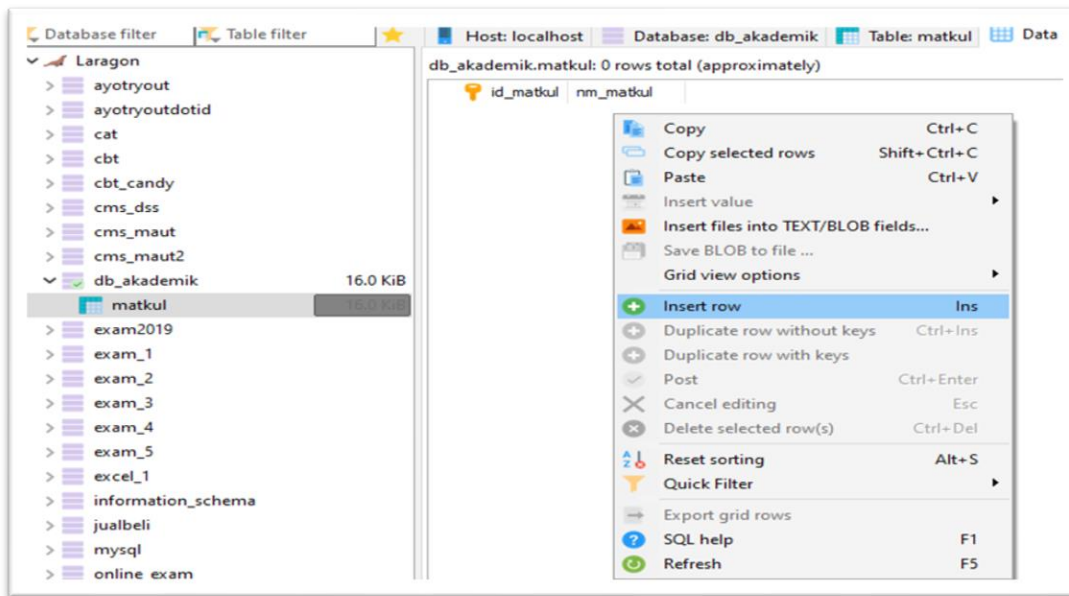
2. Melihat perbedaan antara *mysqli_fetch_array()*, *mysqli_fetch_assoc()*, *mysqli_fetch_row()*.

Heidisql dibuat khusus untuk mengelola database MariaDB, Aplikasi ini *free* dan sudah bawaan otomatis dari Laragon. Bahkan, dengan menginstal Laragon, maka secara otomatis Apache, MariaDB, PHP, HeidiSQL akan terinstal. Untuk membuka aplikasi HeidiSQL, harus menjalankan terlebih dahulu *server web* Apache, dan *server database* MariaDB, yaitu melalui Laragon control panel, seperti pada gambar dibawah ini:









Akses databases menggunakan *mysqli_fetch_array()*

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title></title>
5  </head>
6  <body>
7  <table border="1">
8      <tr>
9          <thead>
10             <td>No.</td>
11             <th>Nama Mata Kuliah</th>
12          </thead>
13      </tr>
14      <tbody>
15      <?php
16          $no=1;
17          include "koneksi.php";
18          $a="SELECT * FROM matkul";
19          $b=$koneksi->query($a);
20          while ($c=$b->fetch_array()) { ?>
21              <tr>
22                  <th><?php echo $no++; ?></th>
23                  <th><?php echo $c['nm_matkul']; ?></th>
24              </tr>
25      <?php
26          }
27      ?>
28      </tbody>
29  </table>
30 </body>
31 </html>

```

Akses databases menggunakan *mysqli_fetch_row()*

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title></title>
5 </head>
6 <body>
7 <table border="1">
8     <tr>
9         <thead>
10            <td>No.</td>
11            <th>Nama Mata Kuliah</th>
12        </thead>
13    </tr>
14    <tbody>
15    <?php
16    $no=1;
17    include "koneksi.php";
18    $a="SELECT * FROM matkul";
19    $b=$koneksi->query($a);
20    while ($c=$b->fetch_row()) { ?>
21    <tr>
22        <th><?php echo $no++; ?></th>
23        <th><?php echo $c[1]; ?></th>
24    </tr>
25    <?php
26    }
27    ?>
28    </tbody>
29    </table>
30 </body>
31 </html>
```


TUGAS MEMBUAT BUKU TAMU

1. Buatlah flowchart untuk sistem Buku Tamu.
2. Buatlah database dengan nama **db_akademik** kemudian buat tabel **bukutamu** yang memiliki 3 field : id_bukutamu (int 11 PK AI), nama (Varchar 100), email (Varchar 100), komentar (TEXT)
3. Buatlah file koneksi dengan nama **koneksi.php** dan file untuk tampil data dengan **namaview.php**

MODUL

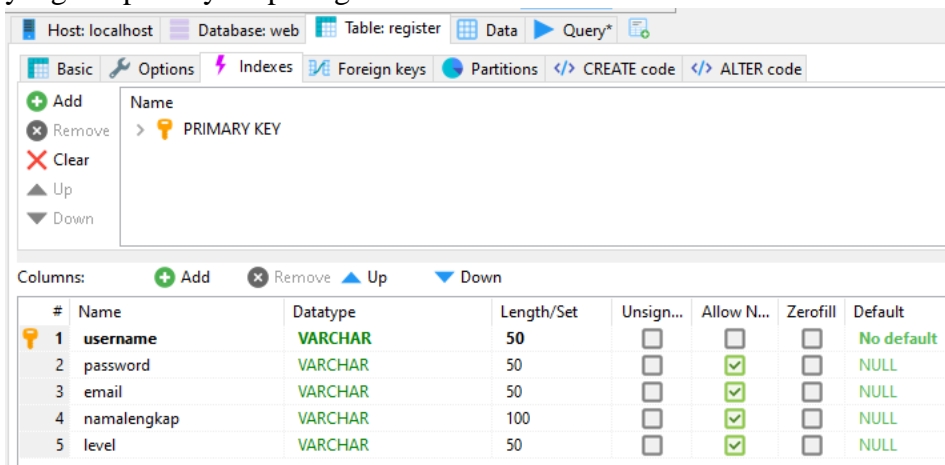
LOGIN MULTI USER dan DOWNLOAD UPLOAD

Tujuan

1. Mahasiswa mampu membuat sistem dengan menggunakan user yang lebih dari 1
2. Mahasiswa mampu membuat sistem upload download file/dokument

Praktikum :

1. Buatlah suatu sistem register yang digunakan untuk login pada suatu sistem informasi, yang tampilannya seperti gambar dibawah ini :



Host: localhost Database: web Table: register Data Query*

Basic Options Indexes Foreign keys Partitions </> CREATE code </> ALTER code

Columns: + Add - Remove ^ Up v Down

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default
1	username	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default
2	password	VARCHAR	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
3	email	VARCHAR	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
4	namalengkap	VARCHAR	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
5	level	VARCHAR	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL

REGISTER

Username :


Password :

Ulangi Password :

Nama Lengkap :

Email :

Level :

 Please fill out this field.

REGISTER

Username :

Password :

Ulangi Password :

Nama Lengkap :

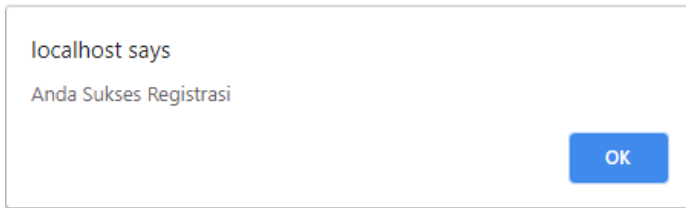
Email :

Level :

Admin
Mahasiswa
Dosen

localhost says

Ulangi, Password Anda tidak sama



Host: localhost Database: web Table: register Data Query*

web.register: 1 rows total (approximately) >> Next

username	password	email	namalengkap	level
agungtwa	e10adc3949ba59abbe56e057f20f883e	agung.twa@gmail.com	Agung Teguh WA	Admin

2. Buat form Login dan aksi login:

```
formlogin.html x
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Form Login</title>
5 </head>
6 <body>
7   <form action="act_login.php?op=in" method="GET">
8     <table>
9       <tr>
10        <td>Username :</td>
11        <td><input type="text" name="username"></td>
12      </tr>
13      <tr>
14        <td>Password :</td>
15        <td><input type="password" name="psw"></td>
16      </tr>
17      <tr>
18        <td></td>
19        <td><input type="submit" value="Login"></td>
20      </tr>
21    </table>
22  </form>
23 </body>
24 </html>
```

```
act_login.php
1 <?php
2 session_start();
3 include "koneksi.php";
4 $user = $_POST['username'];
5 $psw = md5($_POST['psw']);
6
7 $op = $_GET['op'];
8
9 if($op=="in"){
10     $query_1="SELECT * from register where username='$user' AND password='$psw'";
11     $h_1 = $koneksi->query($query_1);
12     if(mysqli_num_rows($h_1)==1){
13         $d_1 = $h_1->fetch_array();
14         $_SESSION['username'] = $d_1['username'];
15         $_SESSION['level'] = $d_1['level'];
16         if($d_1['level']=="Admin"){
17             header("location:home.php");
18             //echo "admin";
19         }else if($d_1['level']=="Mahasiswa"){
20             header("location:home.php");
21             // echo "mahasiswa";
22         }else if($d_1['level']=="Dosen"){
23             header("location:home.php");
24             // echo "Dosen";
25         }else{
26             die("password salah <a href=\"javascript:history.back()\">kembali</a>");
27         }
28     }
29 }else if($op=="out"){
30     unset($_SESSION['username']);
31     unset($_SESSION['level']);
32     header("location:formlogin.html");
33 }
34 ?>
```

```
home.php
1 <?php
2 session_start();
3 include "koneksi.php";
4 error_reporting(E_ALL^(E_NOTICE|E_WARNING));
5 if(!isset($_SESSION['username'])){
6     die("Anda belum login");
7 }
8 $username=$_SESSION['username'];
9 $level=$_SESSION['level'];
10 if ($level=='Admin') {
11     echo "Anda Sebagai ".$level;
12 } elseif ($level=='Mahasiswa') {
13     echo "Anda Sebagai ".$level;
14 } elseif ($level=='Dosen') {
15     echo "Anda Sebagai ".$level;
16 }
17
18 ?>
```

3. Upload dan download dokumen/file

Host: localhost Database: web Table: document Data Query* Query #2* X

Basic Options Indexes Foreign keys Partitions CREATE code ALTER code

Name: document

Comment:

Columns: + Add x Remove ▲ Up ▼ Down

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	C
1	kode_doc	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME...	
2	tgl	VARCHAR	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	
3	deskripsi	TEXT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	
4	file	MEDIUMBLOB		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	
5	nama_file	VARCHAR	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	
6	tipe_file	VARCHAR	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	
7	ukuran_file	VARCHAR	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	