



DIGITAL
TALENT
SCHOLARSHIP



VOCATIONAL SCHOOL GRADUATE ACADEMY

Junior Web Developer

Hari #3

Lokasi Pelatihan: ... (Tuliskan lokasi pelatihan)



KOMINFO



#JADIJAGOANDIGITAL

Badan Penelitian dan Pengembangan Sumber Daya Manusia

PROFIL PENGAJAR



Course Definition

Peserta mempelajari pengetahuan dan keterampilan untuk membuat pemrograman web sederhana yang mampu mendemonstrasikan keterampilan membuat aplikasi web sederhana meliputi:

- Menggunakan teknologi web sederhana (HTML, CSS, Bootstrap, PHP scripting)
- Membuat Struktur menu program dengan HTML, CSS
- Membuat pengelolaan data dengan aksi berupa lihat, input, edit, hapus dengan scripting PHP dan Database Mysql

Pada akhir pelatihan, peserta diharapkan mampu menghasilkan aplikasi sederhana berbasis web sebagai **bukti kompetensi hasil pembelajaran**.

Learning Objective

Peserta mampu memenuhi pengetahuan , keterampilan dan sikap kerja pada unit kompetensi sebagai berikut:

No	Kode Unit	Judul Unit	Standar Kompetensi
1	J.620100.005.02	Mengimplementasikan user interface	SKKNI Nomor 282 Tahun 2016 Bidang Software Development Subbidang Pemrograman
2	J.620100.010.01	Menerapkan perintah eksekusi bahasa pemrograman berbasis teks, grafik, dan multimedia	SKKNI Nomor 282 Tahun 2016 Bidang Software Development Subbidang Pemrograman
3	J.620100.015.01	Menyusun fungsi, file atau sumber daya pemrograman yang lain dalam organisasi yang rapi	SKKNI Nomor 282 Tahun 2016 Bidang Software Development Subbidang Pemrograman
4	J.620100.016.01	Menulis kode dengan prinsip sesuai guidelines dan best practices	SKKNI Nomor 282 Tahun 2016 Bidang Software Development Subbidang Pemrograman
5	J.620100.017.02	Mengimplementasikan pemrograman terstruktur	SKKNI Nomor 282 Tahun 2016 Bidang Software Development Subbidang Pemrograman
6	J.620100.019.02	Menggunakan library atau komponen pre-existing	SKKNI Nomor 282 Tahun 2016 Bidang Software Development Subbidang Pemrograman

KODE UNIT : J.620100.017.02

JUDUL UNIT : Mengimplementasikan Pemrograman Terstruktur

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

ELEMEN KOMPETENSI	KRITERIA UNTUK KERJA
1. Menggunakan tipe data dan <i>control program</i>	1.1 Tipe data yang sesuai standar ditentukan. 1.2 <i>Syntax program</i> yang dikuasai digunakan sesuai standar. 1.3 Struktur kontrol program yang dikuasai digunakan sesuai standar.
2. Membuat program sederhana	2.1 Program baca tulis untuk memasukkan data dari <i>keyboard</i> dan menampilkan ke layar monitor termasuk variasinya sesuai standar masukan/keluaran telah dibuat. 2.2 Struktur kontrol percabangan dan pengulangan dalam membuat program telah digunakan.
3. Membuat program menggunakan prosedur dan fungsi	3.1 Program dengan menggunakan prosedur dibuat sesuai aturan penulisan program. 3.2 Program dengan menggunakan fungsi dibuat sesuai aturan penulisan program. 3.3 Program dengan menggunakan prosedur dan fungsi secara bersamaan dibuat sesuai aturan penulisan program. 3.4 Keterangan untuk setiap prosedur dan fungsi telah diberikan.

4. Membuat program menggunakan <i>array</i>	4.1 Dimensi <i>array</i> telah ditentukan. 4.2 Tipe data <i>array</i> telah ditentukan. 4.3 Panjang <i>array</i> telah ditentukan. 4.4 Pengurutan <i>array</i> telah digunakan.
5. Membuat program untuk akses <i>file</i>	5.1 Program untuk menulis data dalam media penyimpan telah dibuat. 5.2 Program untuk membaca data dari media penyimpan telah dibuat.
ELEMEN KOMPETENSI	KRITERIA UNTUK KERJA
6. Mengkompilasi Program	6.1 Kesalahan program telah dikoreksi. 6.2 Kesalahan <i>syntax</i> dalam program telah dibebaskan.

Tag Dasar Pemrograman PHP

Tag diperlukan agar browser mengenali file yang akan dieksekusi. File PHP cukup diawali dengan tag `<?php` dan diakhiri dengan penutup `?>`. Diantara kedua tag tersebut berisi statement yang akan dikerjakan.

```
1  <?php → Tag Pembuka file PHP
2
3  echo "Saya sedang belajar pemrograman PHP";
4
5  ?> → Tag penutup file PHP
```

Cara Penulisan Syntax PHP - 1

Menggunakan Metode Full Tag (Direkomendasikan)

Cara penulisan open tag php yang pertama adalah menggunakan metode full tag. Penulisan open tag PHP pada metode ini diawali dengan `<?php` dan diakhiri dengan `?>`

```
1  <?php
2      echo 'Halo dunia!;
3  ?>
4
```

PHP

- ❖ Skrip PHP berkedudukan sebagai tag dalam bahasa HTML.
- ❖ Kode PHP diawali dengan **<?php** dan diakhiri dengan **?>**.
- ❖ Adapun kode berikut adalah contoh kode PHP yang berada di dalam kode HTML.

```
<HTML>
    <HEAD>
        <TITLE>Contoh</TITLE>
    </HEAD>
    <BODY>
Selamat Belajar PHP. <BR>
<?php
    Printf("Tgl. Sekarang : %s", Date("d F Y"));
?>
    </BODY>
</HTML>
```


Cara Penulisan Syntax PHP - 2

- ❖ Penulisan Open Tag PHP Menggunakan SGML Style
- ❖ Penulisan open tag php menggunakan style ini diawali dengan <? dan diakhiri dengan ?>, lebih singkat karena tanpa mencantumkan kata-kata php

```
1      <?
2      |      echo 'Halo dunia!';
3      ?>
4
5
```

PHP Case Sensitivity - 1

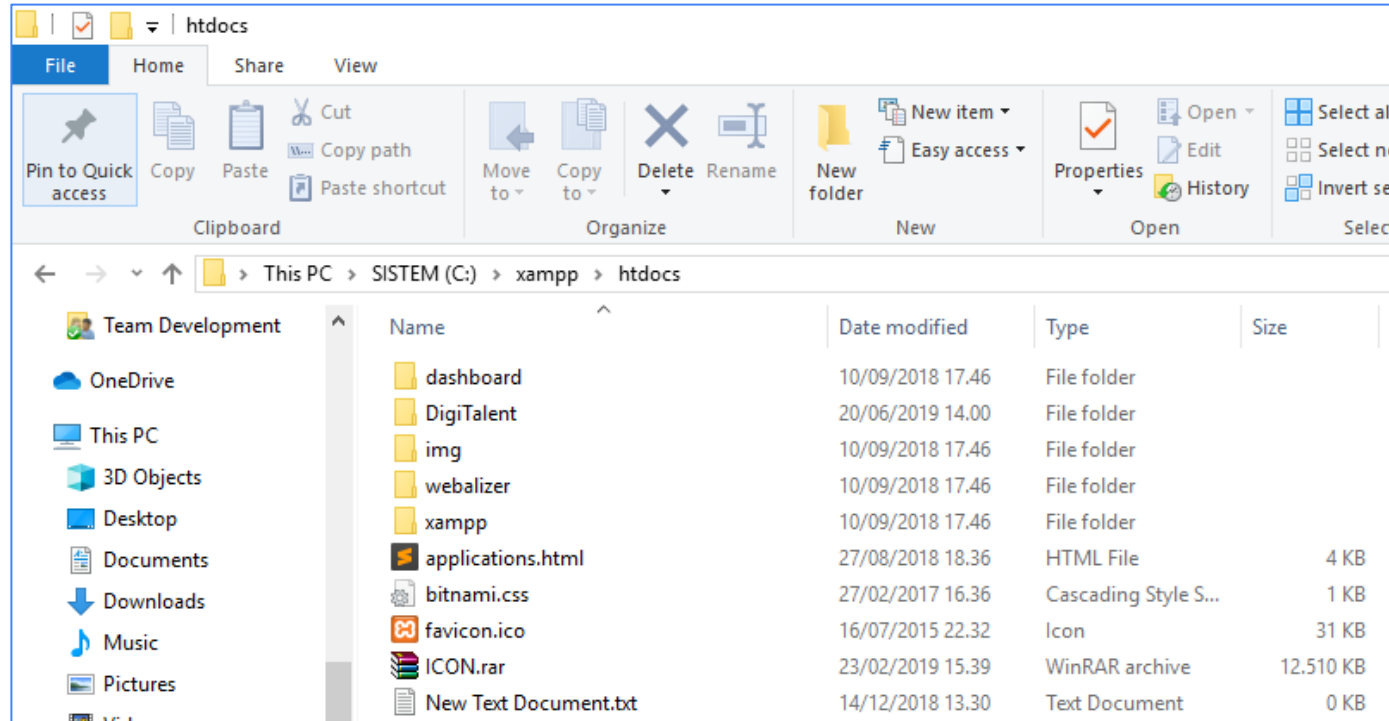
```
1  <!DOCTYPE html>
2  <html>
3      <body>
4
5      <?php
6          ECHO "Hello Indonesia!<br>";
7          echo "Hello Indonesia!<br>";
8          EcHo "Hello Indonesia!<br>";
9      ?>
10
11     </body>
12 </html>
```

PHP Case Sensitivity - 2

```
1  <!DOCTYPE html>
2  <html>
3      <body>
4
5      <?php
6          $warna = "merah";
7          echo "Rumahku " . $warna . "<br>";
8          echo "Darahku " . $WARNA . "<br>";
9          echo "Mobilku " . $waRnA . "<br>";
10     ?>
11
12     </body>
13 </html>
```

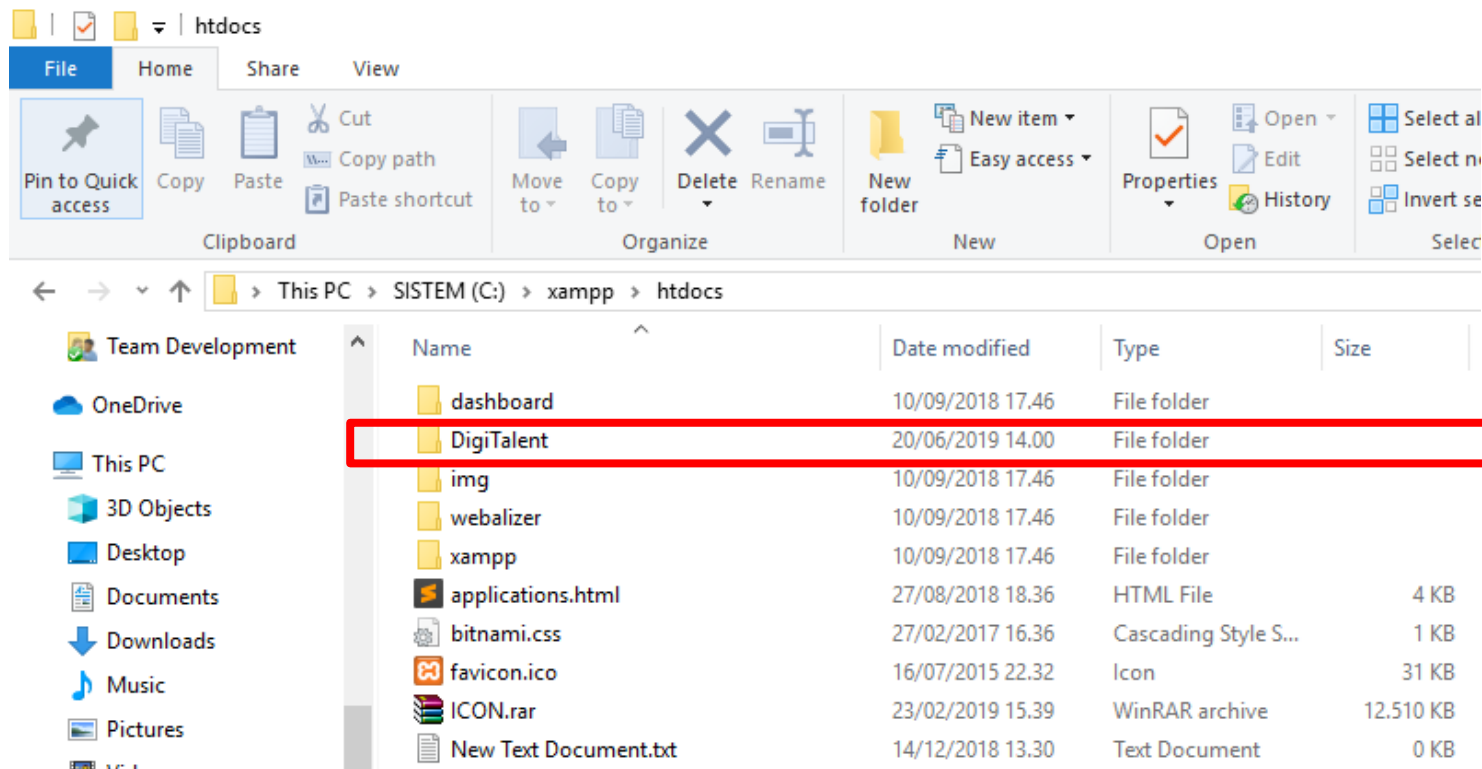
Cara Membuka File PHP

1. Cari folder instalasi **XAMPP**, masuk ke folder **htdocs**. Contoh C:\xampp\htdocs



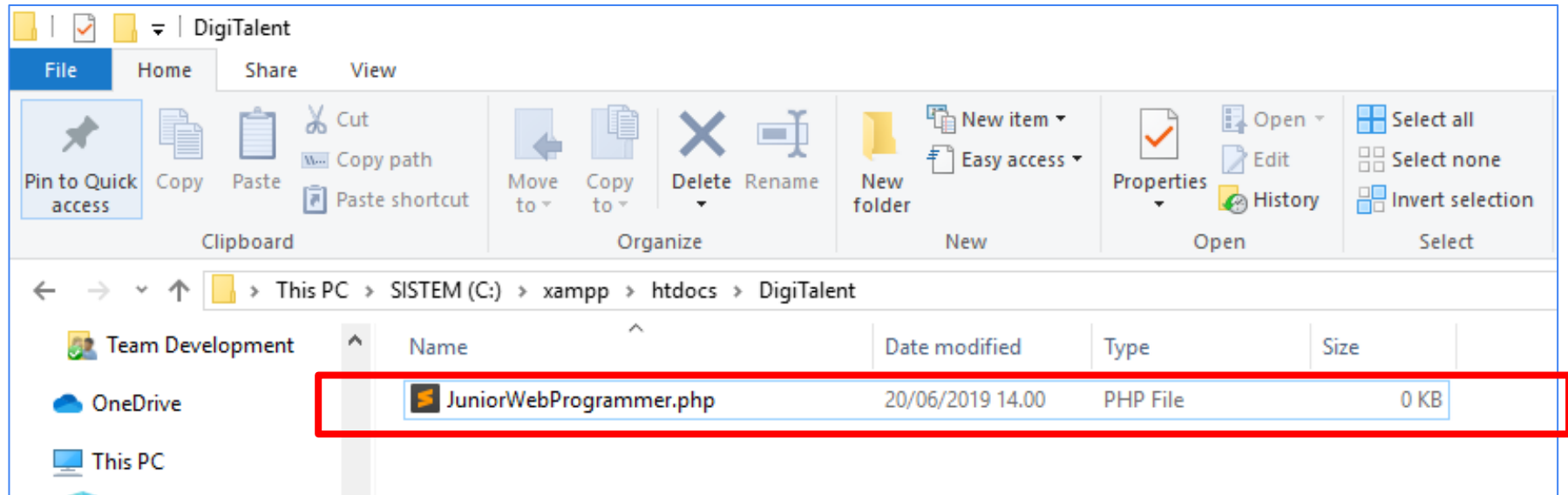
Cara Membuka File PHP

2. Buat folder baru dalam folder **htdocs**.



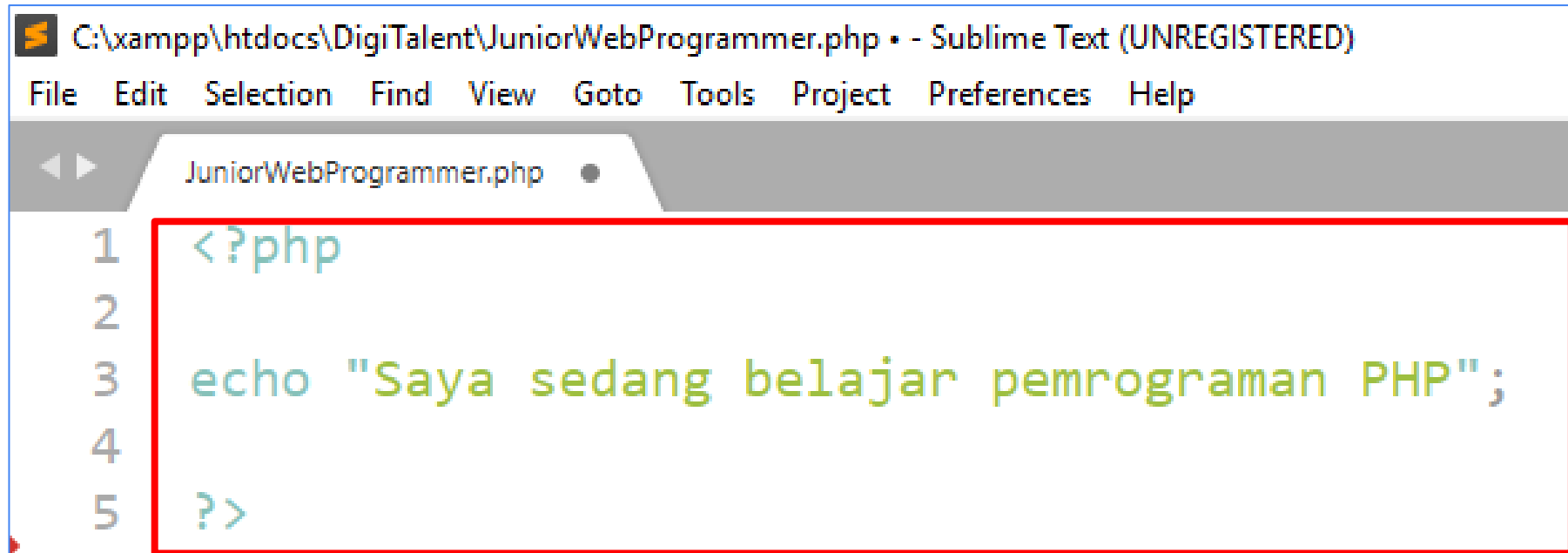
Cara Membuka File PHP

3. Buat file baru dengan cara klik kanan > pilih New > pilih Text Document. Ganti format file dengan **.php**



Cara Membuka File PHP

4. Buka file tersebut menggunakan Sublime Text. Buatlah program PHP.



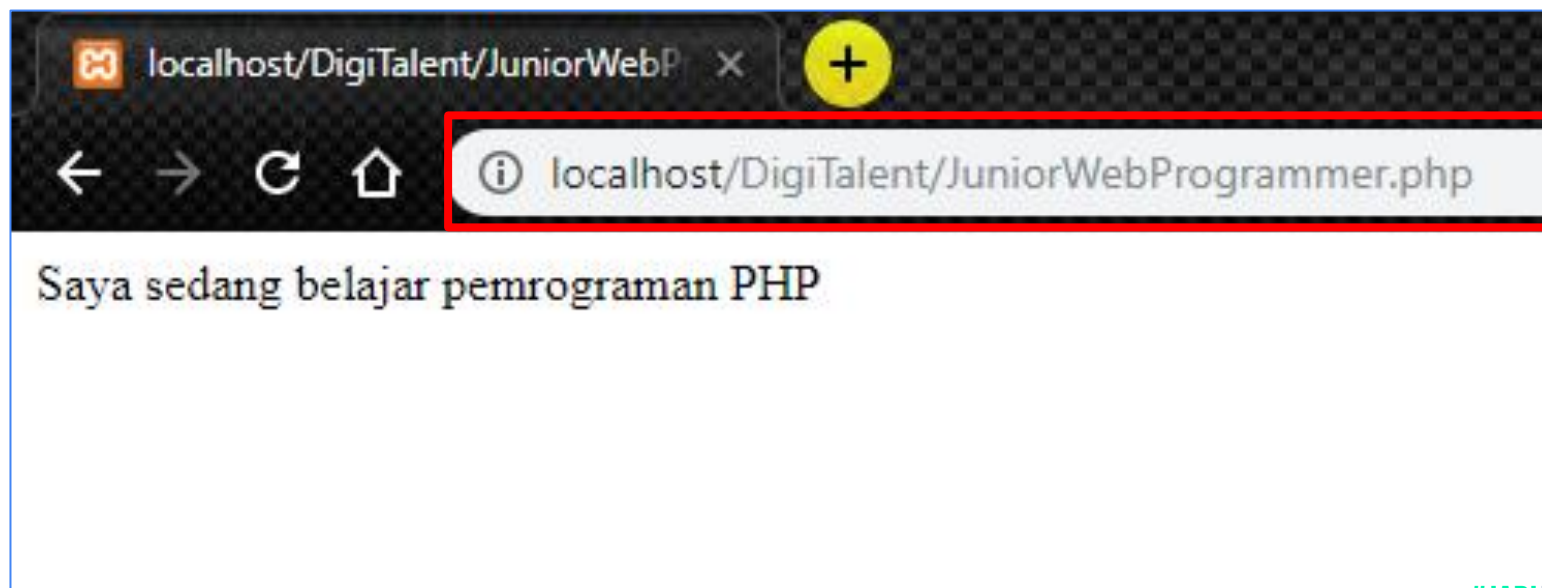
```
C:\xampp\htdocs\DigiTalent\JuniorWebProgrammer.php • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

JuniorWebProgrammer.php

1 <?php
2
3 echo "Saya sedang belajar pemrograman PHP";
4
5 ?>
```

Cara Membuka File PHP

5. Save file tersebut lalu buka kembali file tersebut menggunakan browser. Atau buka browser terlebih dahulu, lalu ketikkan URL **localhost/DigiTalent/JuniorWebProgrammer.php** Maka program PHP yang telah dibuat akan muncul.



Cara Membuka File PHP

- ❖ Buatlah sebuah Folder di **C:\xampp\htdocs\Latihan**, kita akan membuat sebuah file baru. Untuk pembuatan file ini anda dapat menggunakan notepad bawaan Windows maupun editor lainnya

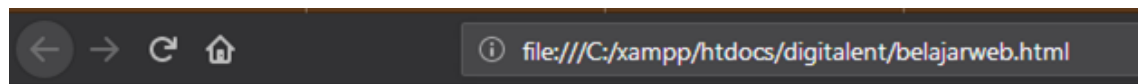
Isi pada file yang akan dibuat lalu save dengan format **belajarweb.html**

```
<!DOCTYPE html>
<html>
  <head>
<title> belajar web</title>
  <style>
</style>
</head>
<body>  <h1>Selamat Datang, jangan lupa bahagia  </h1>  <p>
Selamat belajar pemrograman web</p> </body>
</html>
```

Eksekusi File PHP

Jika Web Server Apache belum berjalan, silahkan menjalankannya terlebih dahulu, buka browser, dan ketikkan alamat berikut:

localhost/digitalent/belajarweb.html pada address bar lalu tekan **Enter**. Maka di dalam web browser akan tampil seperti berikut:



Selamat Datang, jangan lupa bahagia

Selamat belajar pemrograman web

Membuat Form PHP

Contoh : Membuat Form Validasi dengan PHP

Buatlah Sebuah File PHP sebagai berikut : Lakukan Penyimpanan dengan nama **koinfo.php**

```
<html>
    <head>
        <title>Membuat Form Validasi </title>
    </head>
    <body>
        <h1>Membuat Form Validasi Dengan PHP <br/> ---</h1>
        <?php
            if(isset($_GET['nama'])) {
                if($_GET['nama'] == "kosong") {
                    echo "<h4 style='color:red'>Nama Belum Di Masukkan
!</h4>";
                }
            }
        }
```

Penanganan Form

```
<h4>Masukkan Nama Anda :</h4>
<form action="cek.php" method="post">
  <table>
    <tr>
      <td>Nama</td>
      <td><input type="text" name="nama"></td>
      <td><input type="submit" value="Cek"></td>
    </tr>
  </table>
</form>
</body>

</html>
```

PHP Example

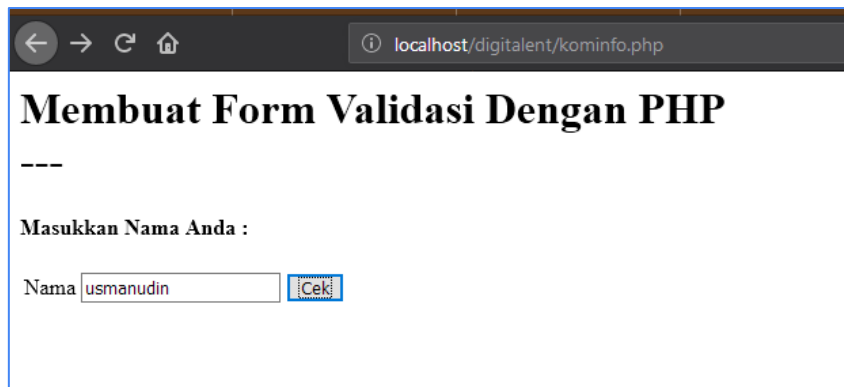
Kemudian buatlah sebuah file untuk mengecek Tulisan pada Komponen Edit, dan beri penamaan File **cek.php** dengan isian perintah sebagai berikut:

```
<?php
    $nama = $_POST['nama'];

    if($nama == ""){
        header("location:koinfo.php?nama=kosong");
    }else{
        echo "Nama anda adalah ". $nama;
    }
?>
```

PHP Example

Hasilnya ketika tidak ada inputan pada komponen edit



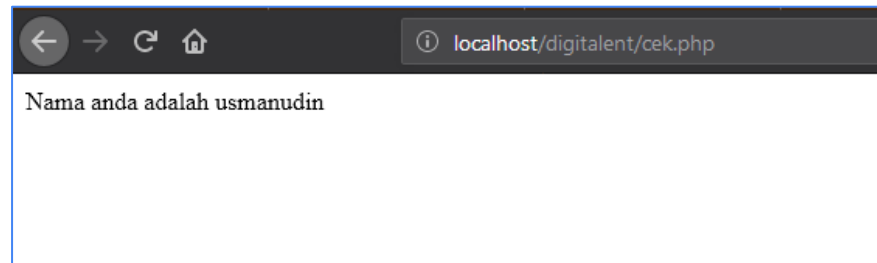
← → ↻ 🏠 localhost/digitalent/kominfo.php

Membuat Form Validasi Dengan PHP

Masukkan Nama Anda :

Nama

Hasil ketika nama diinputkan pada komponen edit



← → ↻ 🏠 localhost/digitalent/cek.php

Nama anda adalah usmanudin

Soal Latihan

❖ Lampiran Studi Kasus Project

Menggunakan tipe data dan *control* program

Variabel

- Suatu lokasi penyimpanan (di dalam memori) yang berisikan nilai atau informasi dan dapat diubah nilainya pada saat kode program sedang berjalan
- Memiliki aturan penamaan dan sifat (Sesuai dengan bahasa Pemrograman yang dipergunakan); misalkan di Php
 - Diawali dengan karakter dolar (\$) dan diikuti dengan nama pengenalan
 - Nama pengenalan dimulai dengan huruf atau garis bawah (_), tidak boleh diawali dengan angka
 - Tidak boleh mengandung spasi dan terdiri dari minimal satu karakter
 - Bersifat case sensitive (huruf besar dan kecil dibedakan)
 - Tidak diperlukan deklarasi type variable, tetapi type variable akan mengikuti type nilai yg diberikan
 - Setiap variabel terbentuk dalam tipe data variant (dapat menampung jenis data apapun)

Variabel

- Contoh
- Yang benar

```
<?php
    $i;
    $nama;
    $Umur;
    $_lokasi_memori;
    $ANGKA_MAKSIMUM;
?>
```

- Yang salah

```
<?php
    $4ever; //variabel tidak boleh diawali
dengan angka
    $_salah satu; //varibel tidak boleh
mengandung spasi
    $nama*^; //variabel tidak boleh mengandung
karakter khusus: * dan ^
?>
```

Tipe Data

- Variable merupakan 'tempat' dari data
- Data yang diinput kedalam variable memiliki tipe tertentu (angka, desimal, text)
- Tipe Data di PHP : integer, floating, string, Boolean, array dan object

Tipe Data: Integer

- Tipe data berupa bilangan bulat positif maupun negatif
- Dapat dilakukan operasi matematis (tambah, kurang, kali, bagi)
- Contoh:

```
<?php
    $umur=21;
    $harga=15000;
    $rugi=-500000;

    echo $umur; //21
    echo "<br />";
    echo $harga; //15000
    echo "<br />";
    echo $rugi; //-500000
?>
```

```
<?php
    $a=14;
    $b=16;
    $c= $a + $b;
    echo $c; // 30

    $d=$a * $b;
    echo $d; // 224
?>
```

Tipe Data: Float

- Tipe data floating point/real number, berupa bilangan bilangan decimal (pecahan)
- Contoh angka float: 0.9 , 2.80 , 3.14 , 0.314E1
- Contoh:

```
<?php
    $angka_float1= 0.78;
    $angka_float2= 14.99;
    $angka_scientific1=0.314E1;

    echo $angka_float1; // 0.78
    echo "<br />";
    echo $angka_float2; //14.99
    echo "<br />";
    echo $angka_scientific1; //3.14
?>
```

```
<?php
    $a=10.66;
    $b=12.4;
    $c= $a + $b;
    echo $c; // 23.06

    $d=$a / $b;
    echo $d; // 0.85967741935484
?>
```

Tipe Data: String

- Tipe data berisi text, kalimat, atau kumpulan karakter
- Penulisannya diapit oleh single quoted/ petik satu (') atau double quoted/petik ganda (")
- Contoh string: 'a', 'saya sedang belajar PHP', 'emailku19@gmail.com'

```
<?php
$string1='Ini adalah string sederhana';
$string2='Ini adalah string yang bisa memiliki beberapa
baris';
$string4="Dia berkata: \"I'll be back\"";
$string5="Variabel akan otomatis ditampilkan:
echo $string1; echo "<br>";
echo $string2; echo "<br>";
echo $string4; echo "<br>";
echo $string5; echo "<br>";
?>
```

Tipe Data: Boolean

- Tipe data bernilai Benar (true) atau Salah (false)
- Dipergunakan dalam operasi logika
- Contoh:

```
<?php
    $benar=true;
    $salah=false;

    echo "benar = $benar, salah = $salah";
    // hasil output: benar = 1, salah =

    $x = FALSE; // false
    $x = ""; // string kosong dianggap false
    $x = " "; // string dengan karakter adalah true;
    $x = 0; // false
    $x = 1; // true

?>
```

SYNTAX PROGRAM

- Syntax PHP : Aturan Penulisan kode program
 - Script diapit tanda
 - `<?php dan ?>`
 - `<script language="php">` dan `</script>`
 - `<? dan ?>`
 - `<% dan %>`
 - Perintah harus diakhiri dengan tanda titik koma (;)

```
<html>
  <head>
    <title>Program PHP Pertama Saya</title>
  </head>
  <body>
    <?php echo "Hello World!"; ?>
  </body>
</html>
```

← php

1. Struktur Kendali/ Kontrol

- Kontrol Program: bagaimana urutan eksekusi perintah dalam program.
- Bentuk:
 - Percabangan (Branching)
 - Perulangan (Looping)
 - Lompatan/ Perpindahan (Jumping)

2. Struktur Kontrol Percabangan (Branching)

- Disebut juga sebagai stuktur kontrol keputusan
- Memungkinkan aplikasi untuk menguji suatu kondisi (isi sebuah variabel, hasil perhitungan ataupun ekspresi) dan mengambil tindakan sesuai dengan hasil pengujian tersebut
- Bentuk:
 - `if`
 - `if - else`
 - `if - elseif - else`
 - `switch`

Struktur Percabangan: if

- Format

```
if(ekspresi) {  
    statements  
    ...  
}
```

- ekspresi : kondisi yang harus dipenuhi agar daftar perintah (statements) antara tanda “{” dan “}” dapat dijalankan.
- Jika kondisi tersebut True maka statements dilakukan jika tidak maka eksekusi langsung ke perintah setelah tanda “}”.
- Contoh:

```
<?php  
    $usia = 30;  
    if($usia > 29)  
        echo "Maaf ... Anda tidak memenuhi Syarat";  
?>
```

Bagaimana jika variabel \$usia diganti dengan nilai 25?

Struktur Percabangan: if - else

- Format:

```
if (ekspresi_logika) {  
    statements1  
    ...  
} else {  
    statements2  
    ...  
}
```

- IF – ELSE membuat percabangan dengan 2 alternatif. Ketika kondisi IF tidak terpenuhi (ekspresi IF menghasilkan nilai FALSE) maka pernyataan pada bagian else yang akan dijalankan

- Contoh:

```
<?php
```

```
$nilai = 80;  
if($nilai >= 60){  
echo "Selamat anda lulus!";  
} else {  
echo "Coba lagi semester depan.";  
  
}
```

```
?>
```

Struktur Percabangan: if - elseif - else

- Jika lebih dari dua alternatif
- Format:

```
<?php
    if (ekspresiA) {
        statement1;
    } elseif (ekspresiB) {
        statement2;
    } else {
        statement3;
    }
?>
```

Contoh:

```
<?php
$nilai = 80;
if(($nilai >= 85) && ($nilai <= 100)){
    $index = "A";
} elseif($nilai >= 70) {
    $index = "B";
} elseif($nilai >= 50) {
    $index = "C";
} elseif($nilai >= 30){
    $index = "D";
} else {
    $index = "E";
}

echo "Nilai anda adalah : ".$nilai;
echo "<br />";

echo "Dengan index      : ".$index;
?>
```

Struktur Percabangan: switch

- Bentuk alternatif dari if-elseif-else.
- Ideal untuk membuat keputusan berdasarkan sejumlah kondisi dengan hanya membandingkan variabel tunggal dengan dengan beberapa kemungkinan nilai.

- Format:

```
switch($variable) {  
    case value_1:  
        statement 1;  
        break;  
    case value_2:  
        statement 2;  
        break;  
    case value_3:  
        statement 3;  
        break;  
    default:  
        statement 4;  
        break;  
}
```


Contoh:

```
<?php
$plat_nomor = "AB";
switch($plat_nomor){
    case "AB":
        echo "Yogyakarta";
        break;
    case "AD":
        echo "Surakarta";
        break;
    case "BE":
        echo "Lampung";
        break;
    case "B":
        echo "Jakarta";
        break;
    default:
        echo "Plat kendaraan tidak diketahui.";
        break;
}
?>
```

Struktur Kontrol Pengulangan (Iterasi)

- Digunakan untuk mengulang satu atau beberapa baris perintah selama kondisi tertentu terpenuhi
- Bentuk:
 - `for`
 - `while`
 - `do-while`
 - `foreach`

Struktur Kontrol Pengulangan (Iterasi)

- Terdapat empat komponen
 - *inisialisasi* : kondisi awal perulangan, inisialisasi (pemberian nilai awal) pada variabel counter
 - *condition* : kondisi yang harus dipenuhi agar perulangan dijalankan
 - *increment* : bagian untuk memproses variable counter agar bisa memenuhi kondisi akhir perulangan
 - *statement* : baris perintah yang akan diulang

For Loop

- Perulangannya (iterasi) telah ditentukan sejak awal pernyataan; sering disebut determinate loop

- **Format:**

```
for (inisialisasi; condition; increment)  
{  
    statement;  
}
```

- **Contoh:**

```
<?php  
for($i=1;$i<=10;$i++){  
    echo "Perulangan ke-".$i."<br />";  
}  
?>
```

While Loop

- Perulangan akan dilakukan terus menerus sampai dengan kondisi yang menjadi prasyarat bernilai false.

- **Format:**

```
inisialisasi;  
while(condition)  
{  
    statement;  
    ...  
    increment;  
}
```

- **Contoh:**

```
<?php  
    $i=1;  
    while($i<=10) {  
        echo "Perulangan ke-".$i."<br  
>";  
        $i++;  
    }  
?>
```

Do-while Loop

- Mirip dengan while loop namun pengecekan kondisi perulangan dilakukan pada akhir perulangan. Perulangan akan dilakukan terus menerus sampai dengan kondisi yang menjadi prasyarat bernilai false. Minimum dilakukan SATU kali.

- **Format:**

```
inisialisasi;  
do  
{  
    statement;  
    ...  
    increment;  
} while(condition)
```

- **Contoh:**

```
<?php  
    $i=1;  
    do{  
        echo "Perulangan ke-".$i."<br/>";  
        $i++;  
    } while($i<=10) ;  
?>
```

For-each-Loop

- Pengulangan khusus untuk array untuk mengakses elemen pada array tanpa mengidentifikasi index-nya.
- Contoh:

```
<?php
$nama = array("Adi", "Budi", "Carli", "Dedy");
foreach($nama as $val){
    echo "$val";
    echo "<br />";
}
?>
```

Menggunakan for:

```
<?php
$nama = array("Adi", "Budi", "Carli", "Dedy");
for($i=0; $i<4; $i++){
    echo "$nama[$i]";
    echo "<br />";
}
?>
```

- SAMA ARTINYA dengan =>

Struktur Lompatan (Jumping)

- Memungkinkan aplikasi untuk keluar dari loop atau exit dari program.
- Bentuk:
 - Break
 - Continue
 - Exit

Break

- Fungsinya untuk keluar dari sebuah perulangan.
- Contoh:

```
<?php
for($i=1;$i<=6;$i++) {
    if($i == 2) {
        break;
    }
    echo "Nilai i : ".$i."<br />";
}
?>
```

exit()

- Digunakan untuk keluar dari sebuah program. Walaupun terdapat statements lain dibawah baris kode exit yang belum dikerjakan untuk keluar dari sebuah perulangan.
- Contoh:

```
<?php
for($i=1;$i<=6;$i++) {
    if($i == 3) {
        exit();
    }
    echo "Nilai i : ".$i."<br />";
}
// statement dibawah ini tidak akan dijalankan
echo "Contoh Penggunaan Exit Pada Program";
?>
```

Membuat program sederhana

Baca Tulis dari Keyboard

- Untuk menerima input/masukan dari keyboar menggunakan <input> yang merupakan tag form html
- Atribut input:
 - Type (jenis input) : text
 - Name: nama elemen
 - value : nilai isian

- Contoh:

```
<html>
<head>
    <title>Input Text</title>
</head>
<body>
<form>
    Nama: <input type="text"
name="nama"/>
</form>
</body>
</html>
```

Membuat program menggunakan prosedur dan fungsi

Program dengan menggunakan prosedur/ fungsi

- Sebuah blok program yang merupakan sekumpulan statement yang bertujuan untuk menyelesaikan suatu tugas tertentu
- Tujuannya : untuk kepraktisan dan kemudahan membuat program
- Setelah dijadikan prosedur/fungsi, maka untuk menjalankan tugas yang sama, kita tinggal memanggil fungsi tersebut, tanpa perlu membuat kembali kode programnya

Format Prosedur/ Fungsi

```
function nama_fungsi ($parameter1, $parameter2)
{
    // kode program fungsi
    return $nilai_akhir // untuk fungsi yang mengembalikan nilai
}
```

- **function** adalah instruksi kepada PHP bahwa kita akan membuat prosedur/ fungsi
- **nama_fungsi** adalah nama dari fungsi yang akan ditulis
- **\$parameter1, \$parameter2** : variabel perantara yang akan menyimpan inputan yang diperlukan dalam pemrosesan fungsi (argumen).
 - Jumlah parameter sesuai yang dibutuhkan.
- **return** : perintah khusus untuk fungsi, dimana kata return menginstruksikan kepada PHP bahwa pemrosesan fungsi telah selesai. `return $nilai_akhir` berarti bahwa fungsi akan 'mengembalikan' `$nilai_akhir` sebagai hasil dari fungsi. Beda prosedur dengan fungsi adalah bahwa prosedur tidak mengembalikan nilai

Contoh prosedur

```
<?php
    function sayHello(){
        echo "Hello..  

```

Contoh fungsi

```
<?php
    function perkalian($bil_1, $bil_2){
        $hasil = $bil_1 * $bil_2;
        return $hasil;
    }
    echo "5 x 10 = ".perkalian(5, 10);
?>
```


Parameter Fungsi

Passing Variable By Value

Yaitu teknik memasukkan parameter ke dalam sebuah fungsi dengan cara membuat copy dari variabel asli, sehingga variabel asli tidak terpengaruh.

Passing Variable By Reference

Memungkinkan kita untuk melakukan manipulasi terhadap variabel yang menjadi parameter melalui sebuah fungsi.

Fungsi

Passing Variable By Value

```
<?php

function tambahSatu($value){
    $value = $value + 1;
}

$a = 5;
tambahSatu($a);
echo $a;

?>
```

Output:

5

Passing Variable By Reference

```
<?php

function tambahSatu(&$value){
    $value = $value + 1;
}

$a = 5;
tambahSatu($a);
echo $a;

?>
```

Output:

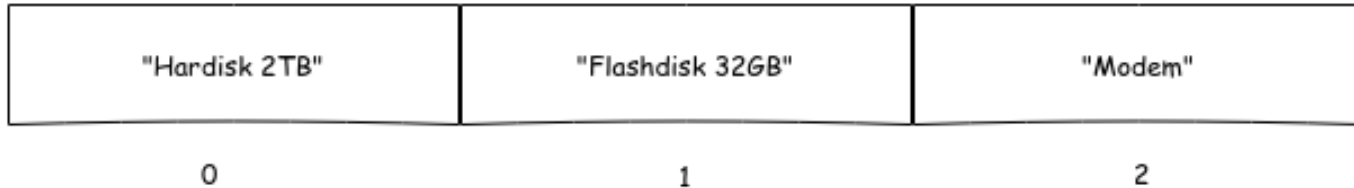
6

Passing Variable By Reference mengubah nilai variabel yang menjadi parameter

Membuat program menggunakan *array*

Apa Itu Array?

- Array adalah salah satu struktur data yang berisi sekumpulan data dan memiliki indeks. Indeks digunakan untuk mengakses nilai array.
- Indeks array selalu dimulai dari nol (0).



Membuat Array di PHP

Array di PHP dapat kita buat dengan fungsi array() dan tanda kurung kotak [].

```
1  <?php
2
3  // membuat array kosong
4  $buah = array();
5  $hobi = [];
6
7  // membuat array sekaligus mengisinya
8  $minuman = array("Kopi", "Teh", "Jus Jeruk");
9  $makanan = ["Nasi Goreng", "Soto", "Bubur"];
10
11 // membuat array dengan mengisi indeks tertentu
12 $anggota[1] = "Komang";
13 $anggota[2] = "Medi";
14 $anggota[0] = "Usman";
15 ?>
```

Membuat Array di PHP

Array dapat di isi dengan tipe data apa saja. Bahkan dicampur dengan data lain juga boleh.

```
1  <?php
2
3  $item = ["Bunga", 123, 39.12, true];
4
5  ?>
```

Menampilkan isi Array

Untuk menampilkan isi array, kita bisa mengaksesnya melalui indeks.

```
1  <?php
2  // membuat array
3  $barang = ["Buku Tulis", "Penghapus", "Spidol"];
4
5  // menampilkan isi array
6  echo $barang[0]."<br>";
7  echo $barang[1]."<br>";
8  echo $barang[2]."<br>";
9  ?>
```

Menampilkan isi Array

Contoh:

```
1 <?php
2 // membuat array
3 $barang = ["Buku Tulis", "Penghapus", "Spidol"];
4
5 // menampilkan isi array dengan perulangan for
6 for($i=0; $i < count($barang); $i++){
7     echo $barang[$i]."<br>";
8 }
9 ?>
```

```
1 <?php
2 // membuat array
3 $barang = ["Buku Tulis", "Penghapus", "Spidol"];
4
5 // menampilkan isi array dengan perulangan foreach
6 foreach($barang as $isi){
7     echo $isi."<br>";
8 }
9
10 echo "<hr>";
11
12 // menampilkan isi array dengan perulangan while
13 $i = 0;
14 while($i < count($barang)){
15     echo $barang[$i]."<br>";
16     $i++;
17 }
18 ?>
```

- Menggunakan loop untuk menampilkan sejumlah isi array
- Fungsi count() untuk menghitung banyaknya isi array.
 - Pada contoh di kiri, isi array sebanyak 3, maka perulangan akan dilakukan sebanyak 3x.

Menghapus isi Array

- fungsi `unset()` untuk menghapus isi array. Fungsi ini juga dapat digunakan untuk menghapus variabel.

```
1  <?php
2  // membuat array
3  $hewan = [
4      "Burung",
5      "Kucing",
6      "Ikan"
7  ];
8  // menghapus kucing
9  unset($hewan[1]);
10
11 echo $hewan[0]. "<br>";
12 echo $hewan[1]. "<br>";
13 echo $hewan[2]. "<br>";
14 echo "<hr>";
15 echo "<pre>";
16 print_r($hewan);
17 echo "</pre>";
18 ?>
```

fungsi `print_r()` untuk menampilkan array secara mentah (raw).
Biasanya untuk debugging.

Menambahkan isi Array

Dua cara untuk menambah isi array:

1. Mengisi langsung ke nomer indeks yang ingin ditambahkan
2. Mengisi langsung ke indeks terakhir

```
1 <?php
2 // membuat array
3 $hobi = [
4     "Membaca",
5     "Menulis",
6     "Ngeblog"
7 ];
8 // menambahkan isi pada idenks ke-3
9 $hobi[3] = "Coding";
10
11 // menambahkan isi pada indeks terakhir
12 $hobi[] = "Olahraga";
13
14 // cetak array dengan perulangan
15 foreach($hobi as $hobiku){
16     echo $hobiku."<br>";
17 }
18 ?>
```

Menambahkan isi Array

- Penambahan pada indeks yang sudah memiliki isi, maka isinya akan diganti/ ditindih dengan yang baru.

```
1  <?php
2  // membuat array
3  $user = [
4      "Komang",
5      "Medi",
6      "Usman"
7  ];
8
9  // mengisi array pada indek ke-1 ("Evans")
10 $user[1] = "Evans";
11
12 // mencetak isi array
13 echo "<pre>";
14 print_r($user);
15 echo "</pre>";
16 ?>
```

Array Asosiatif

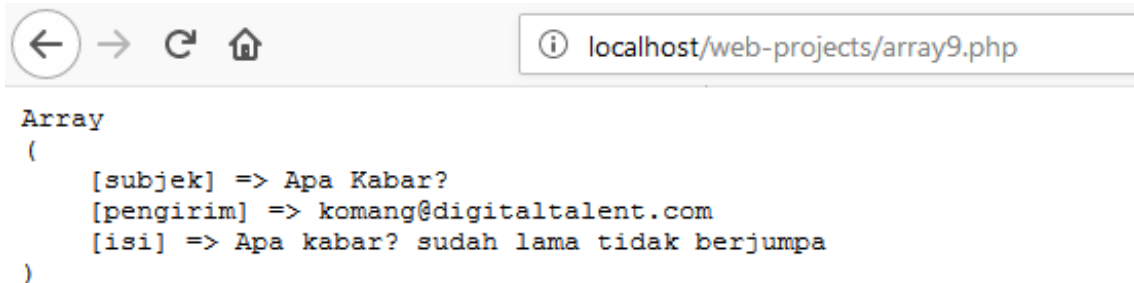
- Array yang indeksnya tidak menggunakan nomer atau angka. tapi berbentuk kata kunci.

```
1 <?php
2 // membuat array asosiatif
3 $artikel = [
4     "judul" => "Belajar Pemrograman PHP",
5     "penulis" => "Digital Talent",
6     "view" => 128
7 ];
8
9 // mencetak isi array assosiatif
10 echo "<h2>".$artikel["judul"]."</h2>";
11 echo "<p>oleh: ".$artikel["penulis"]."</p>";
12 echo "<p>View: ".$artikel["view"]."</p>";
13 ?>
```

Array Asosiatif

- Pada array asosiatif, kita menggunakan tanda => untuk mengasosiasikan sebuah kata kunci dengan isi array.
- Selain menggunakan tanda =>, kita juga bisa membuat array asosiatif seperti **ini**:

```
1 <?php
2 $email["subjek"] = "Apa Kabar?";
3 $email["pengirim"] = "komang@digitaltalent.com";
4 $email["isi"] = "Apa kabar? sudah lama tidak berjumpa";
5
6 echo "<pre>";
7 print_r($email);
8 echo "</pre>";
9 ?>
```



Array Multi Dimensi

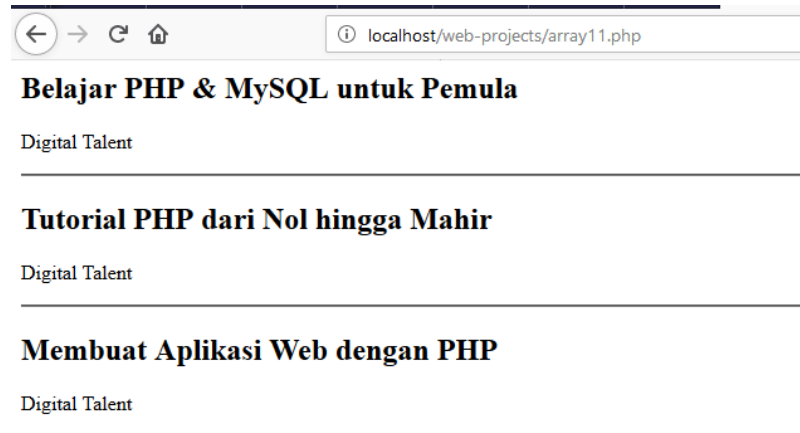
Array yang memiliki dimensi lebih dari satu, untuk membuat matrik, graph, dan struktur data rumit lainnya.

```
1  <?php
2  // ini adalah array dua dimensi
3  $matrik = [
4      [2,3,4],
5      [7,5,0],
6      [4,3,8],
7  ];
8
9  // cara mengakses isinya
10 echo $matrik[1][0]; //-> output: 7
11 ?>
```

Array Multi Dimensi

Contoh lain:

```
1 <?php
2 // membuat array 2 dimensi yang berisi array asosiatif
3 $artikel = [
4     [
5         "judul" => "Belajar PHP & MySQL untuk Pemula",
6         "penulis" => "Digital Talent"
7     ],
8     [
9         "judul" => "Tutorial PHP dari Nol hingga Mahir",
10        "penulis" => "Digital Talent"
11    ],
12    [
13        "judul" => "Membuat Aplikasi Web dengan PHP",
14        "penulis" => "Digital Talent"
15    ]
16 ];
17 // menampilkan array
18 foreach($artikel as $post){
19     echo "<h2>".$post["judul"]."</h2>";
20     echo "<p>".$post["penulis"]."<p>";
21     echo "<hr>";
22 }
```



Membuat program untuk akses *file*

Operasi File pada PHP

PHP memiliki beberapa fungsi untuk:

- √ Membuka file
- √ Menulis data ke dalam file
- √ Membaca data dari file
- √ Menutup file

Membuka File

Sebelum melakukan operasi file, terlebih dahulu kita buka file yang akan dioperasi.

Gunakan perintah : fopen();

```
1  <?php
2  $file = fopen("welcome.txt", "r");
3  ?>
```

Asumsi

Menutup File

Setelah selesai menggunakan file maka file harus ditutup kembali.
Gunakan perintah: `fclose()`;

```
1  <?php
2  //buka file
3  $file = fopen("welcome.txt", "r");
4  //tutup file
5  fclose($file);
6  ?>
```

Operasi Membaca File

- Menggunakan perintah : `fgets(String filename, int length);`
Untuk `int length`, itu hanya opsi saja, menandakan berapa panjang karakter yang kita baca dalam file itu.

```
1  <?php
2  //buka file
3  $file = fopen("welcome.txt","r");
4  //baca file
5  echo fgets($file);
6  //tutup file
7  fclose($file);
8  ?>
```

Membaca file dengan tag HTML

Berikut contohnya (usahakan file anda terdapat tag HTML-nya), sebagai contoh pada file welcome2.txt terdapat tag HTML.

```
1  <?php
2  //buka file
3  $file = fopen("welcome2.txt","r");
4  //baca file dengan fgetss
5  echo fgetss($file);
6  //tutup file
7  fclose($file);
8  ?>
```

Membaca file Tingkat Lanjut

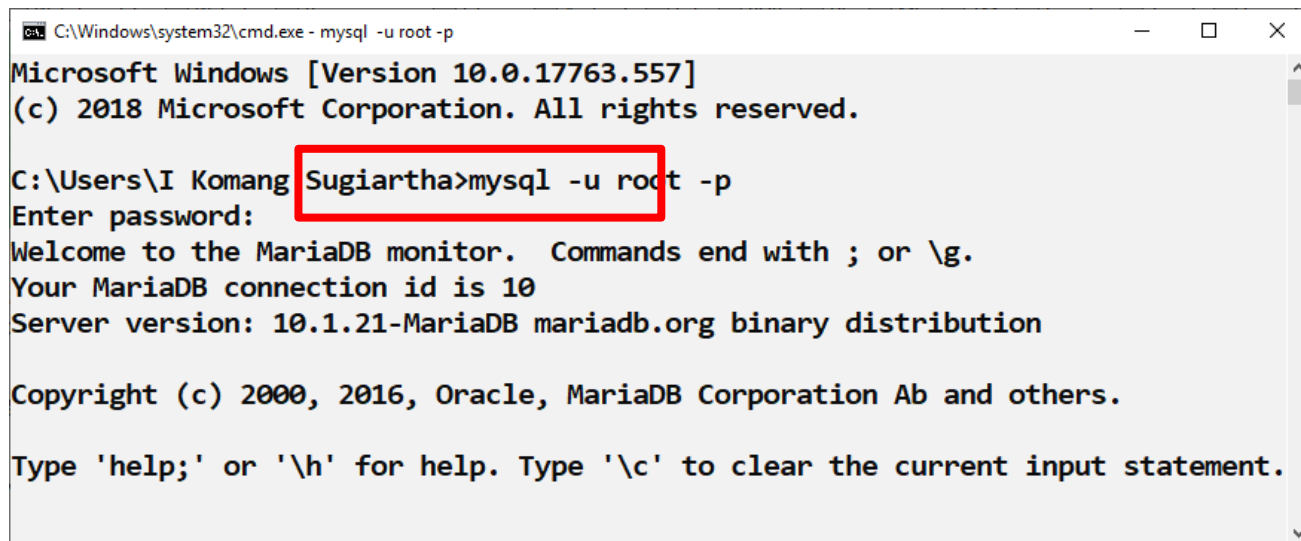
```
1  <?php
2  //membuka file
3  $data = "welcome.txt";
4  $bukafile = fopen($data, "r");
5  if (!$bukafile)
6  { print ("File $data gagal dibuka ! ");
7    exit;
8  }
9  while (!feof($bukafile))
10 { //membaca file
11   $data = fgets($bukafile,50);
12   //menampilkan data
13   print (" $data <br>");
14 }
15 //menutup file
16 fclose($bukafile);
17 ?>
```

Akses Database pada PHP

- ❖ Membuat Koneksi dengan database MySQL
- ❖ Menjalankan query MySQL dari PHP
- ❖ Menampilkan Data
- ❖ Menambahkan Data
- ❖ Mengupdate Data
- ❖ Menghapus Data

Membuat Database

Terlebih dahulu login ke database dengan perintah:



```
C:\Windows\system32\cmd.exe - mysql -u root -p
Microsoft Windows [Version 10.0.17763.557]
(c) 2018 Microsoft Corporation. All rights reserved.

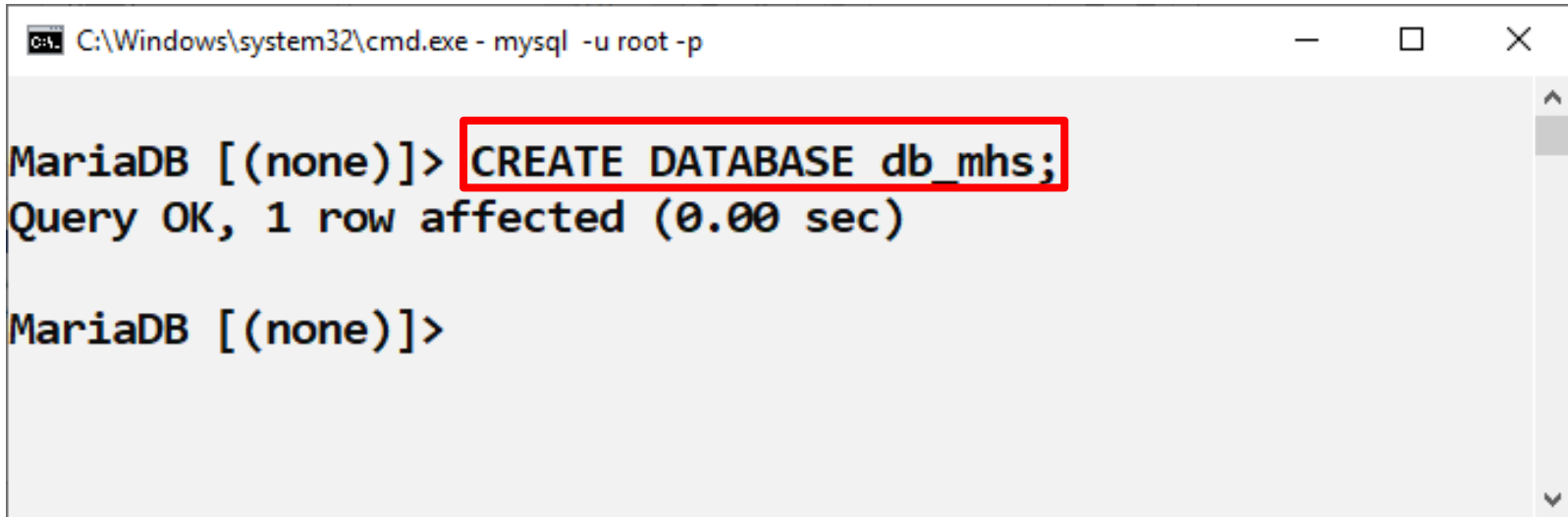
C:\Users\I Komang Sugiarta>mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 10
Server version: 10.1.21-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

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


Membuat Database

Buat database dengan nama db_mhs;



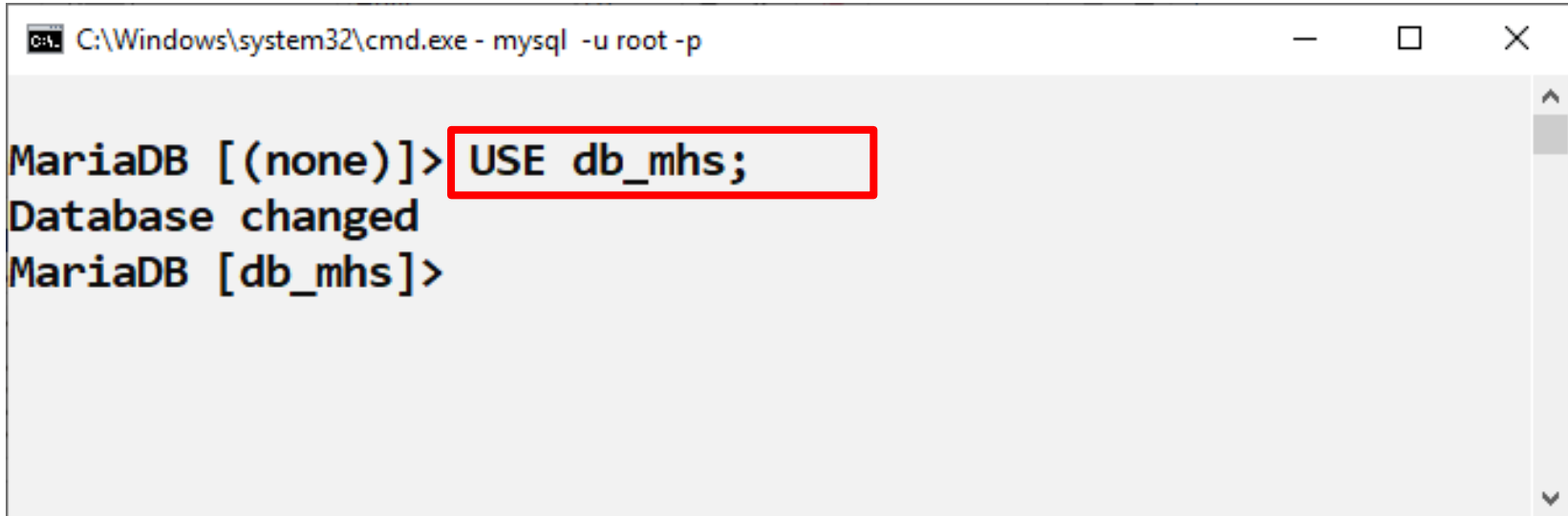
```
C:\Windows\system32\cmd.exe - mysql -u root -p

MariaDB [(none)]> CREATE DATABASE db_mhs;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]>
```

Membuat Database

Gunakan database yang sudah dibuat.

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Windows\system32\cmd.exe' followed by 'mysql -u root -p'. The command prompt shows the MySQL command-line interface. The prompt is 'MariaDB [(none)]>'. The user has entered the command 'USE db_mhs;', which is highlighted with a red rectangular box. The output shows 'Database changed' on the next line, followed by the new prompt 'MariaDB [db_mhs]>'.

```
C:\Windows\system32\cmd.exe - mysql -u root -p

MariaDB [(none)]> USE db_mhs;
Database changed
MariaDB [db_mhs]>
```

Membuat Database

Buat Table dengan perintah

```
C:\Windows\system32\cmd.exe - mysql -u root -p
MariaDB [db_mhs]> CREATE TABLE mahasiswa (
  ->   nim varchar(13) NOT NULL,
  ->   id_mhs int(11) NOT NULL PRIMARY KEY AUTO_INCREMENT,
  ->   nama varchar(40) NOT NULL,
  ->   jenis_kelamin enum('P','L') NOT NULL,
  ->   jurusan varchar(30) NOT NULL,
  ->   alamat text NOT NULL);
Query OK, 0 rows affected (0.03 sec)

MariaDB [db_mhs]>
```

Membuat Database

Struktur Table mahasiswa

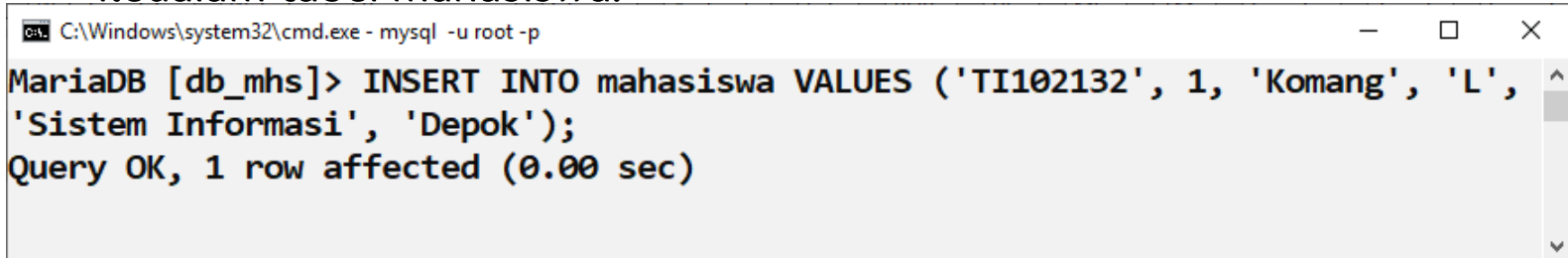
```
C:\Windows\system32\cmd.exe - mysql -u root -p
MariaDB [db_mhs]> DESC mahasiswa;
```

Field	Type	Null	Key	Default	Extra
nim	varchar(13)	NO		NULL	
id_mhs	int(11)	NO	PRI	NULL	auto_increment
nama	varchar(40)	NO		NULL	
jenis_kelamin	enum('P','L')	NO		NULL	
jurusan	varchar(30)	NO		NULL	
alamat	text	NO		NULL	

```
6 rows in set (0.02 sec)
```

Insert Data

- Untuk menampilkan data tentunya kita butuh data yang akan ditampilkan, oleh karena itu kita membutuhkan beberapa sample data untuk di tampilkan nantinya.
- Silahkan jalankan SQL Query berikut ini untuk memasukan beberapa record kedalam tabel mahasiswa.

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Windows\system32\cmd.exe - mysql -u root -p'. The command prompt shows the MySQL prompt 'MariaDB [db_mhs]>' followed by the SQL query 'INSERT INTO mahasiswa VALUES ('TI102132', 1, 'Komang', 'L', 'Sistem Informasi', 'Depok');'. Below the query, the output 'Query OK, 1 row affected (0.00 sec)' is displayed. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
C:\Windows\system32\cmd.exe - mysql -u root -p
MariaDB [db_mhs]> INSERT INTO mahasiswa VALUES ('TI102132', 1, 'Komang', 'L',
'Sistem Informasi', 'Depok');
Query OK, 1 row affected (0.00 sec)
```