

MODUL PRAKTIKUM PEMROGRAMAN WEB
MINGGU – 3



PHP LANJUT & MYSQL

Program Studi Teknik Informatika
Jurusan Teknik Elektro, Informatika, dan Sistem Fisis
Institut Teknologi Sumatera
2020

Modul 6. PHP Lanjut

1. Perulangan Foreach

A. Array

Array merupakan tipe data yang sering digunakan dalam membuat program menggunakan PHP. Kemampuan array dalam menyimpan banyak data dalam satu variabel akan sangat berguna untuk menyederhanakan dan menghemat penggunaan variabel.

Untuk menampilkan dan memproses data pada array, kita dapat memanfaatkan perulangan.

B. Perulangan Foreach

Perulangan foreach merupakan perulangan khusus untuk pembacaan nilai array. Perulangan foreach memiliki pasangan *key* dan *value*. *Key* adalah posisi atau index dari array, dan *value* adalah isi dari array.

Ada 2 cara penulisan perulangan foreach, tanpa key dan dengan key

- Struktur penulisan foreach tanpa key

```
foreach ($nama_array as $value)
{
    statement (...$value...)
}
```

Keterangan :

\$nama_array => nama dari array yang telah didefinisikan

\$value => nama '*variabel perantara*' yang berisi data array pada perulangan tersebut. Anda bebas memberikan nama untuk variabel perantara ini, walaupun pada umumnya banyak programmer menggunakan **\$value**, atau **\$val** saja.

Contoh penggunaan perulangan foreach tanpa key

```
1 <?php
2 $nama = array("Bagus","Budi","Setiawan","Suva","Rahadiyan");
3
4 foreach ($nama as $val)
5 {
6     echo "$val";
7     echo "<br />";
8 }
9 ?>
```

Berikut adalah hasil yang didapatkan

← → ↻ ⓘ localhost/foreach/tanpakey.php

Bagus
Budi
Setiawan
Suva
Rahadiyan

- Struktur penulisan foreach dengan key

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

Perbedaan dengan format sebelumnya, disini PHP menyediakan variabel perantara kedua, yaitu variabel \$key. Variabel \$key ini menampung nilai key dari array.

Contoh penggunaan perulangan foreach dengan key

```
1 <?php
2 $nama = array(
3     1=>"Bagus",
4     6=>"Budi",
5     12=>"Setiawan",
6     45=>"Suva",
7     55=>"Rahadiyan");
8
9 foreach ($nama as $key =>$isi)
10 {
11     echo "Urutan ke-$key adalah $isi";
12     echo "<br />";
13 }
14 ?>
```

Berikut adalah hasil yang didapatkan

← → ↻ ⓘ localhost/foreach/dengankey.php

Urutan ke-1 adalah Bagus
Urutan ke-6 adalah Budi
Urutan ke-12 adalah Setiawan
Urutan ke-45 adalah Suva
Urutan ke-55 adalah Rahadiyan

2. Fungsi

Fungsi (atau Function) dalam bahasa pemrograman adalah kode program yang dirancang untuk menyelesaikan sebuah tugas tertentu, dan merupakan bagian dari program utama. Kita dapat membuat fungsi sendiri, atau menggunakan fungsi yang dibuat oleh programmer lain.

A. Struktur Fungsi

```
function functionName (parameter) {
    ...
    statement
    ...
}
```

Nama Fungsi Parameter

} Function Body

Struktur fungsi diatas dapat dijelaskan sebagai berikut :

- i. Nama fungsi merupakan deklarasi nama fungsi yang akan kita buat
Deklarasi fungsi memiliki beberapa aturan, antara lain :
 - Tidak boleh sama dengan fungsi yang sudah ada didalam PHP. Ini dikarenakan PHP tidak mendukung fasilitas *overloading* yaitu kondisi dimana fungsi dapat menggunakan nama yang sama namun memiliki keluaran yang berbeda dengan syarat parameter pada fungsi tersebut berbeda
 - Hanya terdiri dari huruf, angka dan garis bawah (underscore).
 - Tidak boleh diawali dengan angka.
- ii. Parameter adalah data / nilai masukan yang akan diolah oleh fungsi.
- iii. Function body adalah statements / perintah yang akan dikerjakan oleh fungsi.

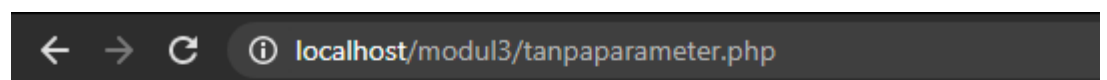
B. Implementasi Fungsi

Implementasi Fungsi tanpa parameter

```
1 <?php
2 function belajar_web(){
3     echo "<h2>Tips Pemrograman Web</h2>";
4     echo "Berlatih terus menerus HTML, CSS, PHP, dan MySQLi";
5 }
6 belajar_web();
7 ?>
```

- Fungsi belajar web tanpa menggunakan parameter
- Berisi 2 output
- Output pertama adalah <h2> dengan isi "Tips Pemrograman Web"
- Output kedua adalah "Berlatih terus menerus HTML, CSS, PHP, dan MySQLi"

Hasil yang didapatkan



Tips Pemrograman Web

Berlatih terus menerus HTML, CSS, PHP, dan MySQLi

Implementasi Fungsi dengan Parameter

```
1 <?php
2 function jumlahkan($x,$y)
3 {
4     $total=$x+$y;
5     echo "<h2>Penjumlahan dari :</h2>";
6     echo "$x + $y = $total";
7 }
8 jumlahkan(11,23);
9 ?>
```

- Fungsi jumlahkan memiliki parameter \$x dan \$y
- \$total menyimpan hasil penjumlahan \$x + \$y

Implementasi fungsi dengan return

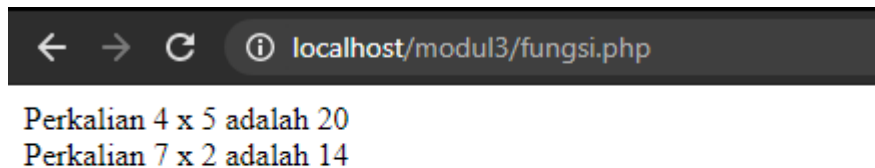
```

1 <?php
2 //pembuatan fungsi
3 function perkalian($angka1, $angka2)
4 {
5     $hasil= $angka1*$angka2;
6     return $hasil;
7 }
8
9 //pemanggilan fungsi
10 $hasil=perkalian(4,5);
11 echo "Perkalian 4 x 5 adalah $hasil";
12 echo "<br />";
13 echo "Perkalian 7 x 2 adalah ".perkalian(7,2);
14 ?>

```

- Pada kode program diatas, kita membuat fungsi perkalian dengan parameter \$angka1 dan \$angka2.
- Variable \$hasil untuk menyimpan hasil perkalian \$angka1 dan \$angka2
- Return untuk mengembalikan nilai \$hasil.

Hasil yang didapatkan dari baris kode diatas



Perkalian 4 x 5 adalah 20
Perkalian 7 x 2 adalah 14

Passing Parameter

A. By Value

Passing by value ialah teknik memasukkan parameter ke dalam sebuah fungsi dengan cara membuat salinan dari variabel asli, sehingga variabel asli tidak terpengaruh. Untuk lebih jelasnya, perhatikan contoh latihan berikut :



```

1 <?php
2 function increment($value){
3     $value++;
4 }
5 $x = 0;
6 increment($x);
7 echo $x;
8 ?>

```

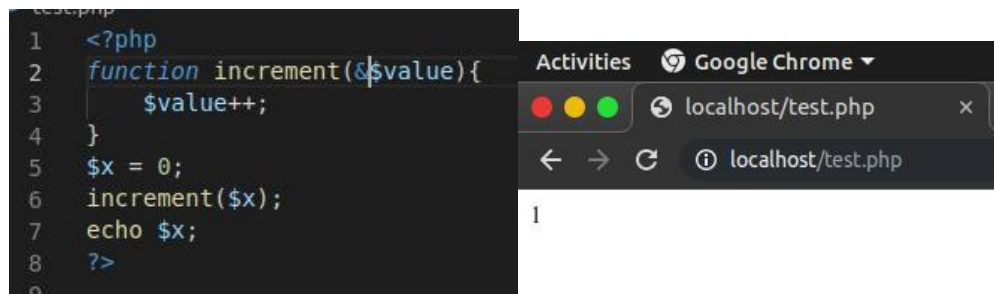
0

Awalnya variabel \$x memiliki nilai 0, ketika menjalankan fungsi increment, maka nilai dari variabel \$x akan disalin ke variabel \$value. Maka operasi

yang dilakukan oleh fungsi increment adalah operasi yang dilakukan terhadap variabel \$value yang merupakan salinan dari variabel \$x. Sehingga nilai variabel \$x tidak berubah.

B. By Reference

Passing parameter by reference memungkinkan kita untuk melakukan manipulasi terhadap variabel yang menjadi parameter melalui sebuah fungsi. Untuk lebih jelasnya, perhatikan contoh latihan berikut yang merupakan modifikasi dari skrip program sebelumnya :

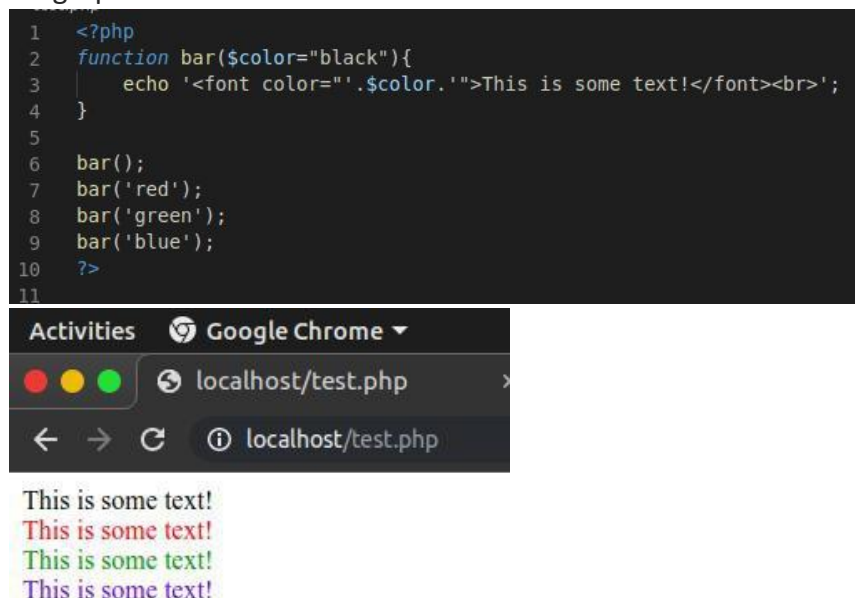


```
1 <?php
2 function increment(&$value){
3     $value++;
4 }
5 $x = 0;
6 increment($x);
7 echo $x;
8 ?>
```

The browser window shows the output of the script at localhost/test.php, which is the number 1.

C. Default Parameter

Default parameter adalah nilai otomatis yang digunakan saat pemanggilan fungsi parameter tidak berisi nilai



```
1 <?php
2 function bar($color="black"){
3     echo '<font color="'. $color. '>This is some text!</font><br>';
4 }
5
6 bar();
7 bar('red');
8 bar('green');
9 bar('blue');
10 ?>
```

The browser window shows the output of the script at localhost/test.php, which displays four lines of text with different colors: black, red, green, and blue.

Secara default, teks tersebut akan berwarna hitam jika fungsi bar dipanggil tanpa ada parameter. Namun jika pemanggilan fungsi disertai dengan parameter \$color, maka parameter default akan diabaikan dan menggunakan parameter yang dimasukkan.

3. Include dan Require

Include dan **Require** adalah dua fungsi atau dua perintah dalam PHP yang bertugas untuk memuat suatu file PHP, di dalam file PHP yang lain.

Secara umum kedua perintah tersebut memiliki tugas yang sama. Hanya saja, dari segi error, keduanya memiliki perbedaan:

1. Jika file yang di-*load* ternyata tidak ada, perintah **include** akan tetap mengeksekusi baris kode program selanjutnya.
2. Sedangkan jika file yang di-*load* dengan fungsi **require** ternyata tidak ada, maka ia akan berhenti di mana fungsi **require** tersebut dipanggil, dan tidak melanjutkan pengeksekusian kode program berikutnya.

Contoh implementasi Include

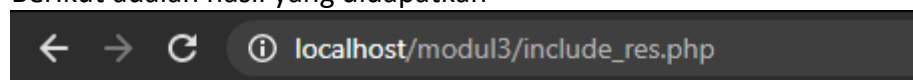
- Buat file `include_func.php` yang diisi dengan sebuah fungsi, disini kita membuat fungsi untuk melakukan penjumlahan

```
1 <?php
2 function jumlahkan($x,$y){
3     $total=$x+$y;
4     echo "<h2>Penjumlahan dari :</h2>";
5     echo "$x + $y = $total";
6 }
7 ?>
```

- Buat file `include_res.php` untuk menginclude fungsi jumlahkan dari file `include_func.php`

```
1 <?php
2 include("include_func.php");
3 jumlahkan(11,23);
4 ?>
```

- Berikut adalah hasil yang didapatkan



Penjumlahan dari :

11 + 23 = 34

Contoh Implementasi Require

- Buat file `require_func.php` yang diisi dengan sebuah fungsi, disini kita membuat fungsi untuk melakukan penjumlahan

```

1  <?php
2  function jumlahkan($x,$y){
3      $total=$x+$y;
4      echo "<h2>Penjumlahan dari :</h2>";
5      echo "$x + $y = $total";
6  }
7  ?>

```

- Buat file include_res.php untuk menginclude fungsi jumlahkan dari file include_func.php

```

1  <?php
2      require("include_func.php");
3      jumlahkan(11,23);
4  ?>

```

- Hasil yang didapatkan

← → ↻ ⓘ localhost/modul3/require_res.php

Penjumlahan dari :

11 + 23 = 34

Include once dan Required once

Dua fungsi tersebut hanya akan me-load file satu kali saja. Artinya jika file tersebut sudah di-load sebelumnya, dia tidak akan di-load lagi di file yang sama.

Implementasi Include once

```

1  <?php
2      include_once("include_func.php");
3      jumlahkan(11,23);
4  ?>

```

Implementasi Require once

```

1  <?php
2      require_once("require_func.php");
3      jumlahkan(11,23);
4  ?>

```

4. POST dan GET

Ada 2 Method dalam pemrograman PHP yang dapat kita gunakan pada form, yakni method POST dan GET.

- Method POST akan mengirimkan data atau nilai langsung ke action untuk ditampilkan, tanpa menampilkan pada URL.



Implementasi Method POST

```
// Tampung data dengan var post  
$data = $_POST['nama'];  
// Cetak  
echo $data;
```

- Method GET akan menampilkan data/nilai pada URL, kemudian akan ditampung oleh action.



Implementasi method GET

```
localhost/form.php?nama=reza  
// Tampung data dengan var get  
$data = $_GET['nama'];  
// Cetak  
echo $data;
```

Perbedaan Method POST dan GET

PERBEDAAN	POST	GET
History	Parameter/nilai tidak tersimpan di browser	Parameter/nilai tersimpan di browser dan bisa diakses ulang karena nilai menjadi kesatuan URL
Bookmark	Parameter/nilai tidak dapat dibookmark	Parameter bisa dibookmark
Tombol Kembali/Kirim Ulang	Biasanya browser memberitahukan/menanyakan apakah data akan dikirim ulang	Dengan method get, halaman sebelumnya langsung ditampilkan tanpa mengeksekusi kembali karena tersimpan dalam cache browser
Encoding type (enctype attribute)	Mendukung pengiriman data biner dengan teknik multipart/form-data atau application/x-www-form-urlencoded	hanya mendukung application/x-www-form-urlencoded
Parameters	Dapat mengirimkan parameter url sekaligus mengirimkan data tersembunyi ke server	Hanya bisa mengirimkan parameter melalui URL
Kecurangan data	Lebih sulit karena tertutup	Lebih terbuka karena data terlihat
Keamanan	Dengan method POST lebih aman karena data langsung dikirim ke server	Dengan jelas bisa ketahui data apa saja yang dikirim
Penggunaan	POST digunakan untuk mengirim data yang lebih rahasia seperti password	Digunakan lebih untuk keperluan mengambil data
Besar data yang dikirim	Standard 8 Mb, tergantung konfigurasi server	7607 karakter

5. Tugas

1. Buatlah fungsi untuk menghitung faktorial dari suatu bilangan!
2. Buatlah fungsi untuk menghitung harga bet berdasarkan panjang nama dan menampilkannya dengan warna font yang diinginkan. Jika parameter warna kosong, maka gunakan warna merah.

Panjang nama 1 - 10 : Rp. 300/karakter

Panjang nama 11-20 : Rp. 500/karakter

Panjang nama >20 : Rp. 700/karakter

Ouput dari fungsi adalah nama dan harga total dengan warna font yang diinginkan.

Modul 7. MySQLi

3. Pengertian

- **Mysqli** merupakan kependekan dari **MySQL Improved Extension**. Seperti yang terlihat dari namanya, extension ini merupakan versi perbaikan dan penambahan dari extension **mysql** sebelumnya yang umum digunakan. Extension **mysqli** dibuat untuk mendukung fitur-fitur terbaru dari MySQL Server versi 4.1 keatas.
- Terhitung dari PHP versi 5.5, extension **mysql** tidak lagi disarankan penggunaannya dan berstatus *deprecated*, yang berarti mungkin akan dihapus pada PHP versi berikutnya. Kita disarankan untuk mulai beralih menggunakan **mysqli** atau **PDO (PHP Data Object)**.

4. Langkah-langkah koneksi database MySQLi

a) Membuka koneksi ke server MySQLi

Mysqli_connect()

Digunakan untuk melakukan uji dan koneksi kepada server database MySQL.

```
$koneksi = mysqli_connect("localhost","root","", "mahasiswa");
```

- \$koneksi adalah variable penampung hasil status koneksi
- Localhost adalah nama host yang digunakan
- Root adalah username yang digunakan untuk mengakses database
- " " berisi password untuk masuk ke dalam database
- Mahasiswa adalah nama database yang digunakan

b) Memilih database yang akan digunakan di server

Mysqli_select_db()

Digunakan untuk melakukan koneksi kepada database yang dalam server yang berhasil dikoneksi dengan perintah `mysqli_connect()`.

```
mysqli_select_db($koneksi, "mahasiswa");
```

c) Mengambil query dari sebuah database

Mysqli_query()

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

```
$hasil = mysqli_query($koneksi, "select * from informatika");
```

d) Mengambil record dari table pada database

- **Mysqli_fetch_array()**

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

```
$row=mysqli_fetch_array
```

- **Mysqli_fetch_assoc()**

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

```
$row=mysqli_fetch_assoc
```

- **Mysqli_fetch_row()**

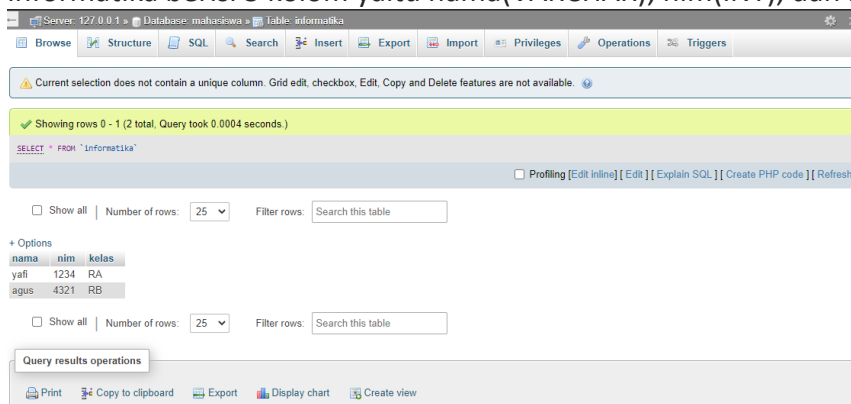
Fungsi ini hampir sama dengan fungsi `mysql_fetch_array()`, hanya saja array yang dihasilkan hanya array numeris.

```
$row=mysqli_fetch_row
```

2. Latihan

a) Latihan koneksi PHP ke database MySQLi

- Buat database mahasiswa dan buat table informatika di dalamnya. Table informatika berisi 3 kolom yaitu nama(VARCHAR), nim(INT), dan kelas(VARCHAR)



- Lakukan koneksi database dengan baris kode pada gambar berikut

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Koneksi Database MySQLi</title>
5 </head>
6 <body>
7     <h1>Demo Koneksi Database</h1>
8 <?php
9
10 $koneksi = mysqli_connect("localhost","root","","mahasiswa");
11
12 // Check connection
13 if (mysqli_connect_errno()){
14     echo "Koneksi database gagal : " . mysqli_connect_error();
15 }else{
16     echo "sukses";
17 }
18
19 ?>
20 </body>
21 </html>
22
```

- b) Latihan uji coba perbedaan `mysqli_fetch_array()`, `mysqli_fetch_assoc()`, dan `mysqli_fetch_row()`

Salinlah kode dibawah ini, dan lihat dan analisis perbedaan hasilnya

- `Mysqli_fetch_array`

```
1  <?php
2  $koneksi=mysqli_connect ("localhost","root","")
3  or die ("koneksi gagal");
4  mysqli_select_db($koneksi, "mahasiswa");
5  $hasil = mysqli_query($koneksi, "select * from informatika");
6  while ($row=mysqli_fetch_array($hasil)) {
7      print_r($row);
8  }
9  ?>
```

- `Mysqli_fetch_row`

```
1  <?php
2  $koneksi=mysqli_connect ("localhost","root","")
3  or die ("koneksi gagal");
4  mysqli_select_db($koneksi, "mahasiswa");
5  $hasil = mysqli_query($koneksi, "select * from informatika");
6  while ($row=mysqli_fetch_row($hasil)) {
7      print_r($row);
8  }
9  ?>
```

- `Mysqli_fetch_assoc`

```
1  <?php
2  $koneksi=mysqli_connect ("localhost","root","")
3  or die ("koneksi gagal");
4  mysqli_select_db($koneksi, "mahasiswa");
5  $hasil = mysqli_query($koneksi, "select * from informatika");
6  while ($row=mysqli_fetch_assoc($hasil)) {
7      print_r($row);
8  }
9  ?>
```

- c) Membuat Buku Tamu

- Buat table bukutamu

```
Create table bukutamu (
nama char(20) not null,
email char(20),
komentar char (40)
);
```

- Buat form untuk buku tamu, beri nama bukutamu.html

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Buku Tamu</title>
5 </head>
6 <body>
7     <h1>Buku Tamu untuk Database MySQLi</h1>
8     <form action="prosesTambah.php" method="post">
9         Nama      : <input type="text" name="nama" size="35" maxlength="50"> <br>
10        Email       : <input type="text" name="email" size="35" maxlength="50"> <br>
11        Komentar    : <textarea name="komentar" rows="5" cols="30"></textarea> <br>
12        <input type="submit" value="Simpan">
13        <input type="reset" value="Reset">
14    </form>
15
16 </body>
17 </html>

```

localhost/modul3/bukutamu.html

Buku Tamu untuk Database MySQLi

Nama :

Email :

Komentar :

- Buat file prosesTambah.php untuk memproses data dari bukutamu.html

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Simpan Buku Tamu</title>
5 </head>
6 <body>
7     <h1>Simpan Buku Tamu MySQLi</h1>
8     <?php
9         $nama = $_POST["nama"];
10        $email = $_POST["email"];
11        $komentar = $_POST["komentar"];
12        $conn=mysqli_connect ("localhost","root","")
13        or die ("koneksi gagal");
14        mysqli_select_db($conn, "tamu"); echo "Nama      : $nama <br>"; echo "Email       : $email <br>";
15        echo "Komentar : $komentar <br>";
16        $sqlstr="insert into bukutamu (nama,email,komentar) values ('$nama','$email','$komentar')";
17        $hasil = mysqli_query($conn, $sqlstr);
18        echo "Simpan bukutamu berhasil dilakukan";
19    ?>
20
21 </body>
22 </html>

```

localhost/modul3/prosesTambah.php

Simpan Buku Tamu MySQLi

Nama : telaga
 Email : telagamuhamad@gmail.com
 Komentar : tess
 Simpan bukutamu berhasil dilakukan

- Buat file view.php untuk melihat isi buku tamu

```

1  <?php
2  $conn = mysqli_connect("localhost","root","");
3  mysqli_select_db($conn, "tamu");
4  $hasil = mysqli_query($conn, "select * from bukutamu");
5  $jumlah = mysqli_num_rows($hasil);
6  echo "<center>Daftar Pengunjung</center>";
7  echo "Jumlah pengunjung : $jumlah";
8  $a=1;
9  while($baris = mysqli_fetch_array($hasil))
10 {
11     echo "<br>";
12     echo $a;
13     echo "<br>"; echo "Nama : "; echo $baris[0]; echo "<br>";
14     echo "Email : ";
15     echo $baris[1];
16     echo "<br>";
17     echo "Komentar : ";
18     echo $baris[2];
19     $a++;
20 }
21 ?>

```

localhost/modul3/view.php

Daftar Pengunjung

Jumlah pengunjung : 3

1
Nama : telaga
Email : telagamuhamad@gmail.
Komentar : tess

2
Nama : telaga
Email : telagamuhamad@gmail.
Komentar : tess

3
Nama : Suva Rahadiyan
Email : admin@admin.com
Komentar : tes2

- Buat file search.html untuk melakukan searching pada buku tamu

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Cari Database</title>
5  </head>
6  <body>
7      <h1>Searching Buku Tamu untuk Database MySQLi</h1>
8      <form action="hasilsearch.php" method="post">
9          <select name="kolom">
10             <option value="nama">nama</option>
11             <option value="email">email</option>
12          </select>
13          Masukkan kata yang anda cari
14          <input type="text" type="text" name="cari">
15          <input type="submit" value="cari" >
16      </form>
17
18  </body>
19  </html>

```

localhost/modul3/search.html

Searching Buku Tamu untuk Database MySQLi

Masukkan kata yang anda cari

- Buat file hasilsearch.php untuk menampilkan data

```

1 <?php
2     $kolom=$_POST['kolom'];
3     $cari=$_POST['cari'];
4     $conn=mysqli_connect("localhost","root","");
5     mysqli_select_db($conn, "tamu");
6     $hasil=mysqli_query($conn, "select * from bukutamu where $kolom like '%$cari%'");
7     $jumlah=mysqli_num_rows($hasil);
8     echo "<br>";
9     echo "Ditemukan: $jumlah"; echo "<br>";
10    while($baris=mysqli_fetch_array($hasil)){
11        echo "Nama : "; echo $baris[0]; echo "<br>";
12        echo "Email : "; echo $baris[1]; echo "<br>";
13        echo "Komentar : ";
14        echo $baris[2];
15    }
16    ?>

```

localhost/modul3/hasilsearch.php

Ditemukan: 1
 Nama : Suva Rahadiyan
 Email : admin@admin.com
 Komentar :tes2

3. Tugas

1. Buat tabel mahasiswa yang berisi 4 field : NRP, Nama, Alamat, ID_Jur
2. Buat table jurusan yang berisi 2 field : ID_Jur, Nama
3. Kedua table diatas saling berelasi
4. Buat tampilan web yang berisi :
 - Proses untuk tambah data
 - Proses untuk search data, dengan masukan berupa Nama. Data yang ditampilkan adalah NRP, Nama, Foto dan nama jurusan.
 - Proses untuk delete data, dengan masukan berupa NRP.

Contoh tampilan input :

http://localhost/awal.php

DATA MAHASISWA PENS

TAMBAH DATA

NRP :

Nama :

Foto :

Jurusan :

- Telekomunikasi
- Elka
- IT
- Elin

SEARCH DATA

Nama :
