

# **MODUL PRAKTIKUM PEMROGRAMAN WEB**

*Oleh:*

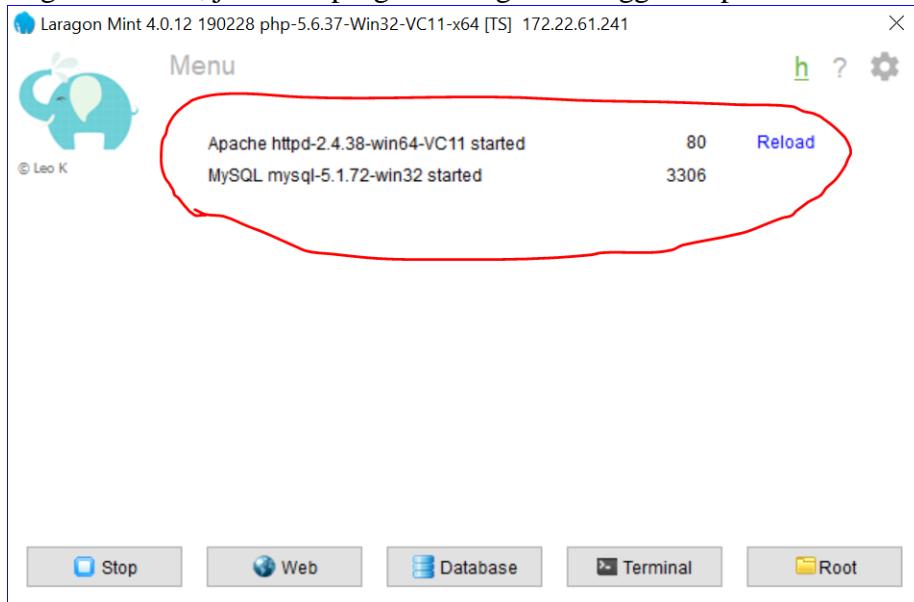
**RAHMAN, S. KOM., M.T.**

**2019**

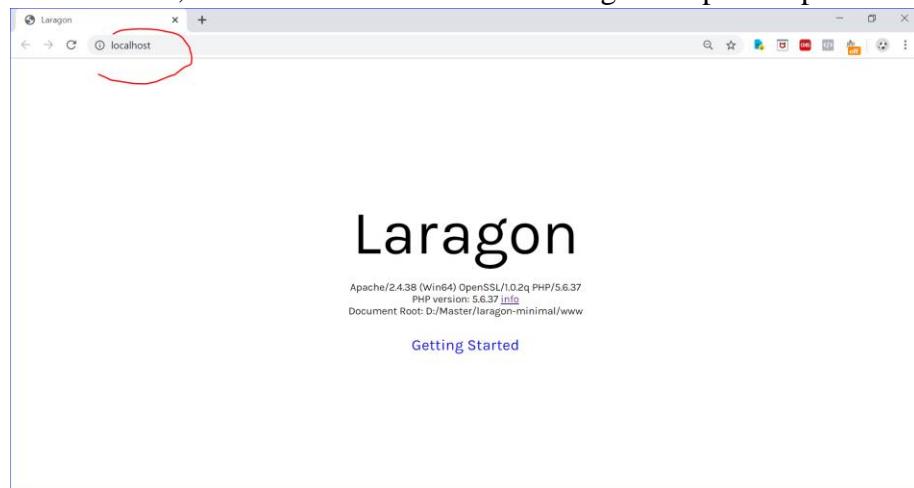
## A. MEMBUAT HALAMAN WEB DENGAN FILE PHP

Berikut langkah-langkahnya:

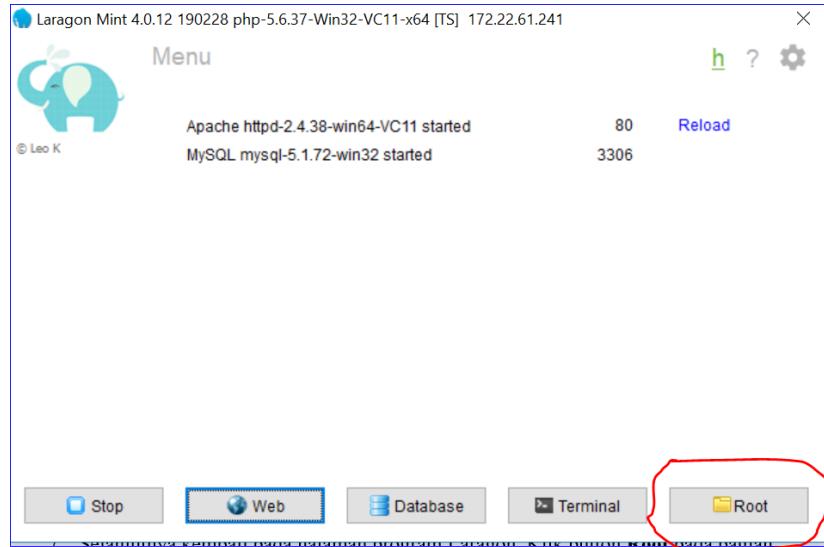
1. Pertama-tama instal Laragon. Pastikan versi laragon yang anda instal cocok untuk versi windows anda. Beberapa versi Laragon terbaru tidak bisa berjalan pada windows 7 dan windows 8. Sebelum mendownload laragon, anda harus membaca keterangan versinya dan sistem windows yang didukung.
2. Setelah laragon terinstal, jalankan program laragon sehingga tampil halaman berikut:



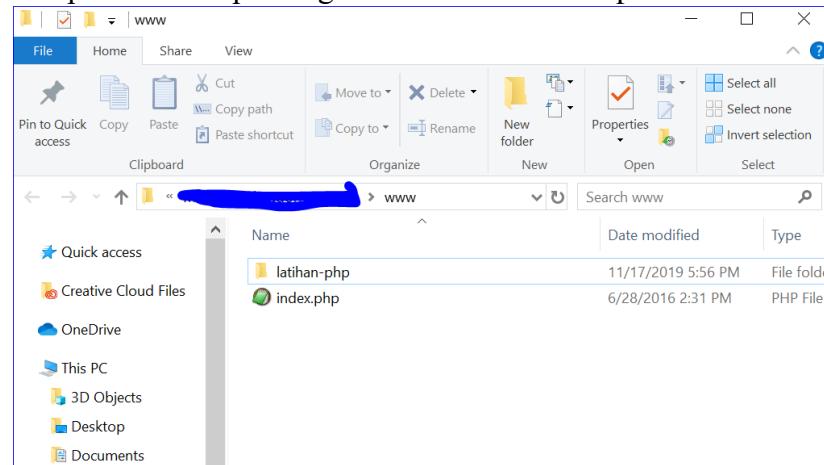
3. Pastikan **Apache** dan **MySQL** berjalan dengan baik seperti pada gambar.
4. Klik pada button **Web**, maka browser akan terbuka dengan tampilan seperti berikut:



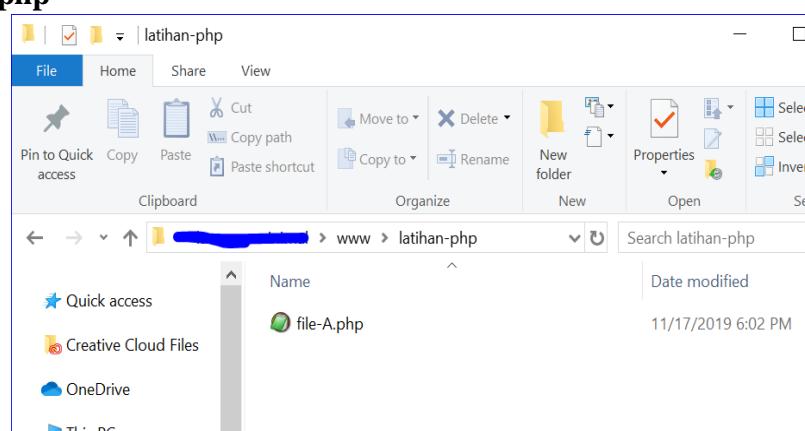
5. Pada url browser tertulis alamat lokal yaitu: **localhost**
6. Sampai disini berarti instalasi **web server** lokal anda menggunakan **Laragon** berjalan dengan baik.
7. Selanjutnya kembali pada halaman program Laragon. Klik button **Root** pada bagian sudut kanan bawah



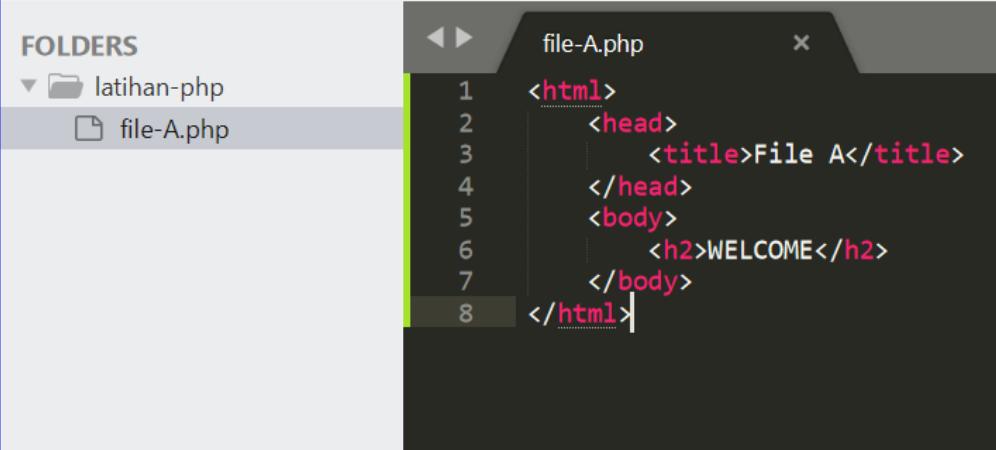
8. Direktori root laragon yaitu folder **www** akan terbuka.
9. Selanjutnya buat sebuah folder atau directory baru dalam folder **www** dengan nama "**latihan-php**" tanpa tanda kutip. Dengan demikian terlihat seperti ini:



10. Kita kan selanjutnya menggunakan folder **latihan-php** ini sebagai directory kerja kita untuk semua file php yang akan kita praktekan.
11. Selanjutnya masuk kedalam folder/directory **latihan-php** lalu buat sebuah file baru dengan nama **file-A.php**



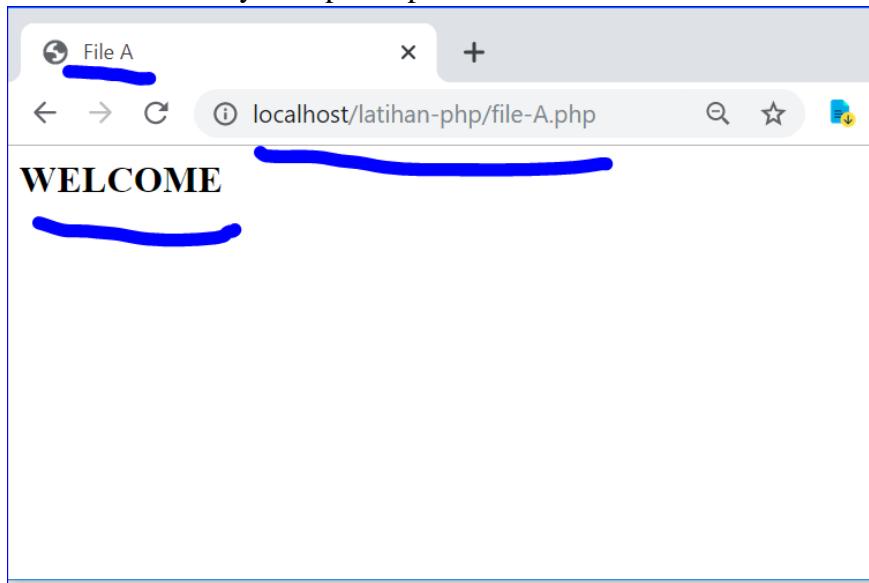
12. Buka file tersebut dan ketikan kode HTML berikut:



The screenshot shows a code editor window with a sidebar labeled 'FOLDERS'. Inside the sidebar, there is a folder named 'latihan-php' which contains a file named 'file-A.php'. The main area of the editor displays the following HTML code:

```
<html>
<head>
    <title>File A</title>
</head>
<body>
    <h2>WELCOME</h2>
</body>
</html>
```

13. Sekarang kita telah berhasil membuat file web dengan ekstensi php. File ini dapat diakses melalui browser dengan memasukan alamat server lokal: **localhost** dan nama folder/directory tempat file: **latihan-php** dan nama file: **file-A.php**. Dengan alamat lengkap akses file adalah: **localhost/latihan-php/file-A.php**
14. **localhost** adalah alamat server lokal yang merujuk pada folder **www** aplikasi laragon
15. **latihan-php** adalah folder/directory yang telah kita buat sebelumnya dalam folder root **www**.
16. **file-A.php** adalah nama file web php
17. Sehingga untuk mengakses halam tersebut maka pada browser harus dimasukan alamat: **localhost/latihan-php/file-A.php**
18. Maka pada browser seharusnya tampil output berikut:



19. Kita telah berhasil membuat file web php pertama dan berhasil pula menjalankannya.

## B. MEMBUAT HALAMAN PHP DENGAN PERINTAH VARIABEL DAN PERINTAH ECHO PHP

Perintah **echo** adalah perintah output sebagai luaran kode PHP. Pada dasarnya kode-kode PHP hanya diterjemahkan dan diproses oleh interpreter PHP. Hasil proses tersebut tidak akan menghasilkan apa-apa sebagai output visual yang bisa dilihat pada browser **kecuali** jika output itu ditampilkan melalui perintah **echo**.

Berikut ini adalah langkah-langkah menggunakan output dengan **echo**:

1. Masuk kedalam direktori latihan-php
2. Buat sebuah file dengan nama **file-B.php**
3. Buka file **file-B.php** kemudian ketikan kode berikut:

```
<html>
  <head>
    <title>File B</title>
  </head>
  <body>

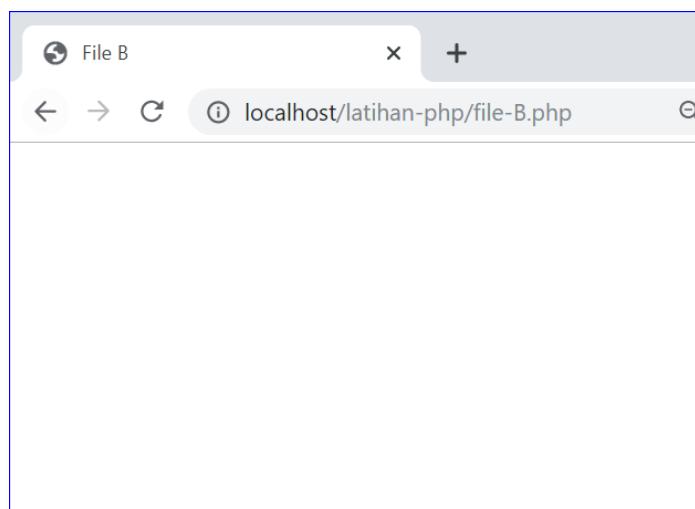
    <?php

      $output = "<h2>WELCOME</h2>";

    ?>

  </body>
</html>
```

4. Pada contoh **file-B.php** diatas, kita menggunakan kode php didalam tag **<?php** dan ditutup dengan **?>**
5. Variabel **output** (dideklarasikan dengan tanda \$), menampung data **String** yaitu **<h2>WELCOME</h2>**
6. Jika kita jalankan file **file-B.php** pada browser dengan mengklik: **localhost/latihan-php/file-B.php** maka tulisan WELCOME tidak akan tampil dilayar. Perhatikan output berikut:



7. Tulisan WELCOME dengan tampilan bold tidak tampil karena String `<h2>WELCOME</h2>` hanya disimpan dalam variabel `$output`
8. Untuk menampilkan tulisan WELOCOME, maka kita harus mengoutputkan String `<h2>WELCOME</h2>` dengan statemen `echo` pada variabel `$output`. Yaitu dengan menambahkan baris kode: `echo $output;`
9. Sehingga kode program menjadi seperti berikut ini:

```
<html>
  <head>
    <title>File B</title>
  </head>
  <body>

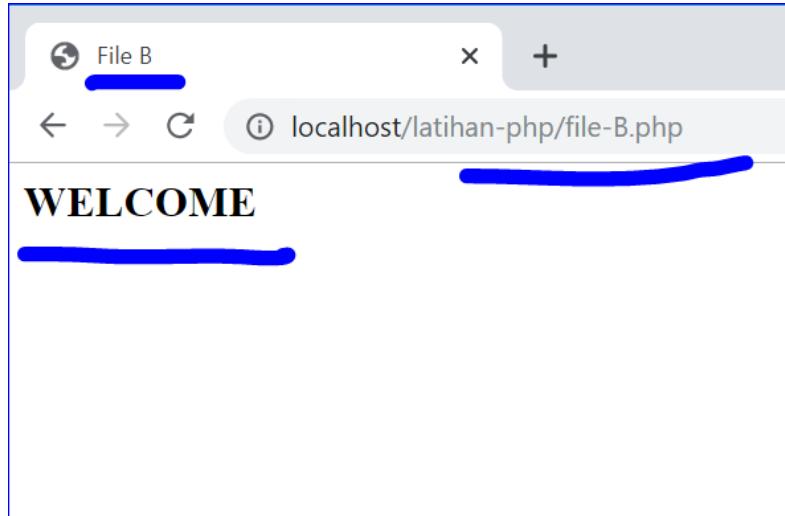
    <?php

      $output = "<h2>WELCOME</h2>";
      echo $output;

    ?>

  </body>
</html>
```

10. Sehingga saat kita mengakses kembali file **file-B.php** dengan alamat url browser: `localhost/latihan-php/file-B.php` maka tampil berikut ini:



11. Dengan demikian data yang telah diproses jika ingin ditampilkan pada pengguna melalui browser, maka data tersebut harus dioutputkan dalam program dengan menggunakan perintah `echo`.
12. Selanjutnya kita bisa juga memproses String dari 2 variabel dengan menggabung String seperti pada kode **file-B.php** lanjutan berikut ini:

```
<html>
  <head>
    <title>File B</title>
  </head>
  <body>

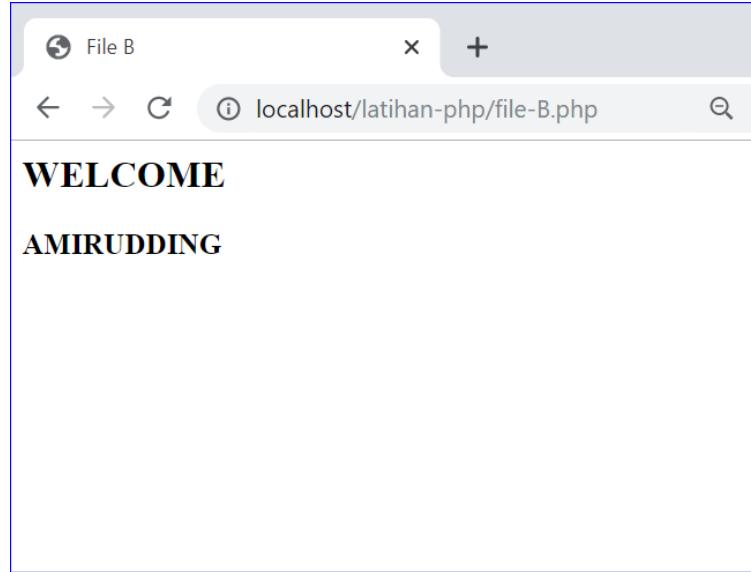
    <?php

      $output = "<h2>WELCOME</h2>";
      $user = "<h3><b>AMIRUDDING</b></h3>";
      echo $output . " " . $user;

    ?>

  </body>
</html>
```

13. Kode diatas menampilkan gabungan String yang ada pada variabel **\$output** dan variabel **\$user**. Gabungan String dilakukan menggunakan operasi titik.
14. Jika dijalankan, akan menghasilkan tampilan berikut:



15. Kita telah berhasil menggunakan kode php dalam halaman web, berhasil menggunakan variabel, melakukan operasi gabung String dengan tanda titik(.), serta menampilkan output dengan perintah **echo**.

### C. MENGGUNAKAN PERULANGAN DENGAN PHP

Perulangan dalam bahasa pemrograman secara umum menggunakan beberapa bentuk variasi diantaranya looping **for**, **while** dan **foreach**. Perulangan digunakan untuk memproses koleksi data atau untuk melakukan proses berulang. Contoh dalam latihan ini kita akan peraktekan adalah mencetak output yang sama secara berulang. Misalnya cetak kata WELCOME sebanyak 10 kali.

Langkah-langkahnya adalah:

1. Buka kembali folder atau directory **latihan-php** dan buat file baru
2. Beri nama file baru tersebut: **file-C-looping-for.php**
3. Ketikan kode program berikut:

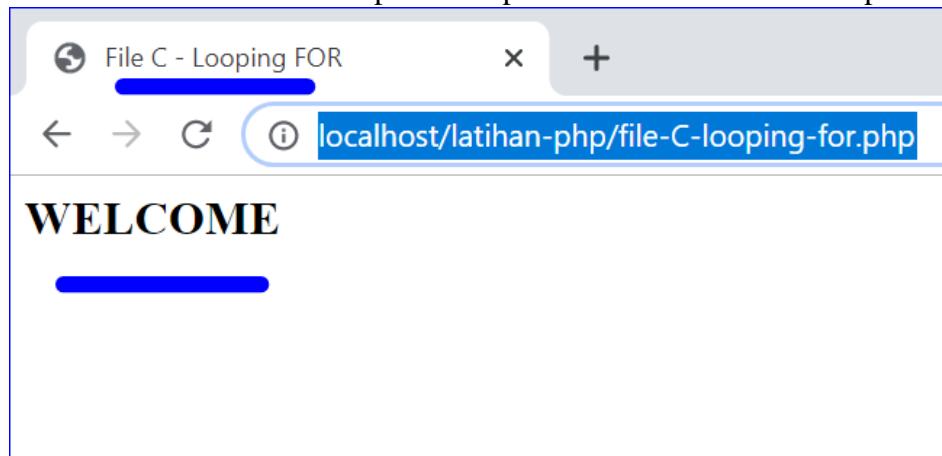
```
<html>
    <head>
        <title>File C - Looping FOR</title>
    </head>
    <body>

        <?php
            $output = "<h2>WELCOME</h2>";
            echo $output;

        ?>

    </body>
</html>
```

4. Jalankan halaman tersebut pada browser melalui url: **localhost/latihan-php/file-C-looping-for.php**
5. Maka halaman tersebut akan menampilkan output 1 tulisan WELCOME seperti berikut:



6. Selanjutnya kita akan memodifikasi kode program pada file **file-C-looping-for.php**
7. Kita akan menggunakan logika perulangan **for** untuk mencetak variabel \$output sebanyak 10 kali.

8. Ubah kode file **file-C-looping-for.php** seperti berikut ini:

```
<html>
  <head>
    <title>File C - Looping FOR</title>
  </head>
  <body>

    <?php

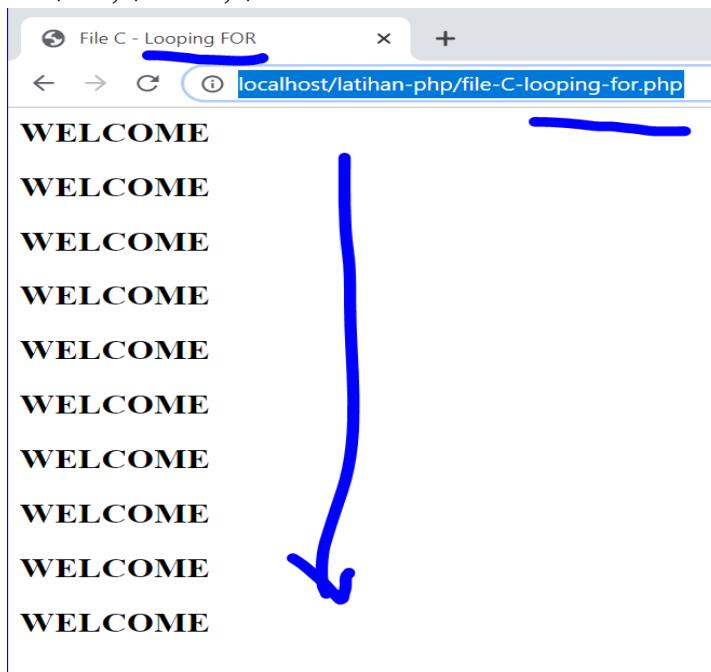
      $output = "<h2>WELCOME</h2>";

      for($i=1; $i<=10; $i++){
        echo $output;
      }

    ?>

  </body>
</html>
```

9. Pada kode revisi diatas, logika perulangan atau **looping** telah ditambahkan, dan perintah **echo \$output** ditempatkan didalam blok logika looping yaitu didalam blok kurung {}  
10. Jalankan kembali halaman tersebut dengan mengakses url: **localhost/latihan-php/file-C-looping-for.php** maka akan tampil output WELCOME sebanyak 10 kali sesuai logika perulangan variabel **\$i=1; \$i<=10; \$i++**



11. Kita telah berhasil membuat perulangan dengan menggunakan looping **FOR**

## D. PERULANGAN DENGAN OBJEK PHP

Adakalanya perulangan diperlukan bukan untuk memproses data yang sama seperti pada contoh sebelumnya. Dalam suatu program terkadang ada koleksi data dalam bentuk variabel **array** yang harus ditampilkan. Misalnya array dari 6 bilangan kelipatan 3 atau array koleksi nama-nama pulau di Indonesia dan seterusnya.

Data-data array contoh tersebut dapat pula ditampilkan menggunakan perintah **echo** yang dieksekusi dalam blok logika **looping**. Kita bisa menggunakan logika looping **FOR** seperti sebelumnya. Akan tetapi cara yang paling baik adalah dengan menggunakan looping **FOREACH** atau dengan menggunakan **WHILE** dan **DO..WHILE** dengan iterator.

Berikut ini adalah langkah-langkah menampilkan setiap data pada koleksi data dengan menggunakan 3 variasi logika looping.

1. Buka kembali folder atau directory **latihan-php** dan buat file baru
2. Beri nama file baru tersebut: **file-D-looping-foreach.php**
3. Kemudian ketik kode berikut ini:

```
<html>
  <head>
    <title>File D - Looping FOREACH</title>
  </head>
  <body>

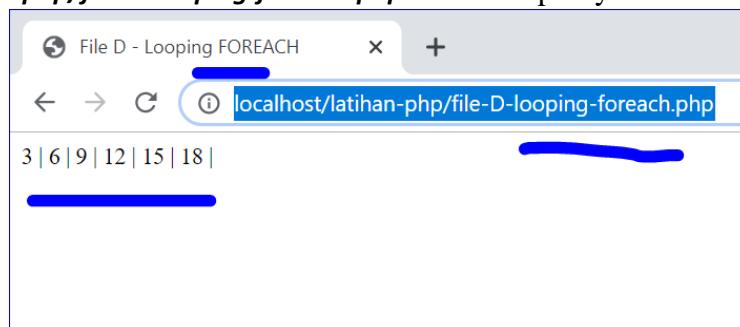
    <?php
      $kelipatan_3 = array(3, 6, 9, 12, 15, 18);

      foreach ($kelipatan_3 as $angka) {
        echo "$angka | ";
      }

    ?>

  </body>
</html>
```

4. Pada kode diatas terdapat variabel array yang berisi 6 angka yang merupakan kelipatan angka 3. Kemudian logika looping **foreach** digunakan untuk mencetak setiap elemen data array dalam variabel **\$kelipatan\_3**
5. Jalankan halaman file **file-D-looping-foreach.php** dengan mengakses url: **localhost/latihan-php/file-D-looping-foreach.php** maka outputnya:



6. Selanjutnya tambahkan variabel array lain yaitu nama-nama pulau. Edit file **file-D-looping-foreach.php** menjadi seperti berikut:

```
<html>
    <head>
        <title>File D - Looping FOREACH</title>
    </head>
    <body>

        <?php

            $kelipatan_3 = array(3, 6, 9, 12, 15, 18);

            foreach ($kelipatan_3 as $angka) {
                echo "$angka | ";
            }

            $pulau_pulau = ["Sulawesi", "Jawa", "Kalimantan", "Sumatera", "Bali", "Papua"];

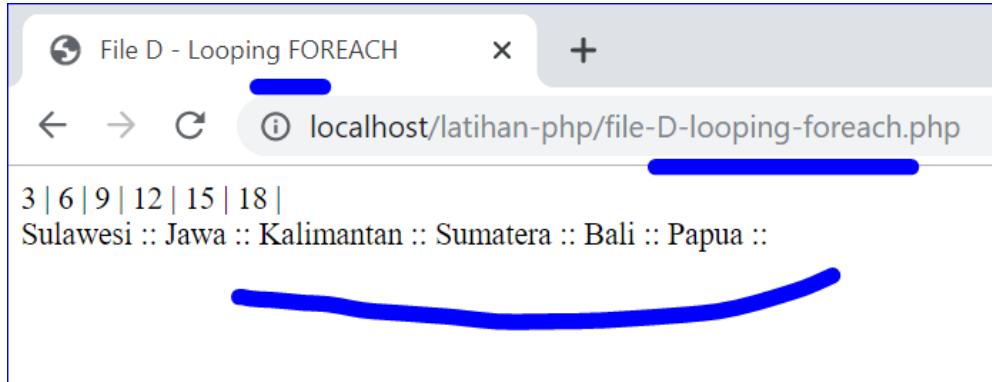
            echo "<br />";

            foreach ($pulau_pulau as $pulau) {
                echo "$pulau :: ";
            }

        ?>

    </body>
</html>
```

7. Sama seperti variabel array pertama, yang berbeda pada array kedua ini adalah cara pendeklarasian array dan tipe data elemen array yang bukan angka atau numerik tapi bertipe string.  
8. Jalankan kembali halaman file **file-D-looping-foreach.php** dengan mengakses url: **localhost/latihan-php/file-D-looping-foreach.php** maka outputnya akan jadi:



9. Kita telah berhasil mengolah koleksi data dengan menggunakan logika looping **FOREACH**. Selanjutnya kita akan memproses data yang sama dengan menggunakan logika looping **WHILE** dan **DO/WHILE**  
10. Buat file baru dengan nama **file-D-looping-while.php**  
11. Selanjutnya pada file ini kita akan mengimplementasikan logika looping **WHILE** pada data array angka kelipatan 3.  
12. Sedangkan logika looping **DO..WHILE** kita akan mengimplementasikan logika looping **DO—WHILE**.  
13. Baik logika **WHILE** maupun **DO—WHILE**, keduanya bisa mengakses elemen array data dengan menggunakan indeks. Akan tetapi pada contoh ini, kita akan menggunakan iterator untuk logika **WHILE** dan akses indeks untuk logika **DO/WHILE**

14. Buka file **file-D-looping-while.php** dan ketik kode berikut ini:

```
<html>
  <head>
    <title>File D - Looping WHILE</title>
  </head>
  <body>

    <?php

      $kelipatan_3 = array(3, 6, 9, 12, 15, 18);

      $arrObject1 = new ArrayObject($kelipatan_3);
      $arrayIterator1 = $arrObject1->getIterator();

      while( $arrayIterator1->valid() )
      {
        $angka = $arrayIterator1->current();
        echo "$angka | ";
        $arrayIterator1->next();
      }

      echo "<br />";

      $pulau_pulau = ["Sulawesi", "Jawa", "Kalimantan", "Sumatera", "Bali", "Papua"];

      $i = 0;

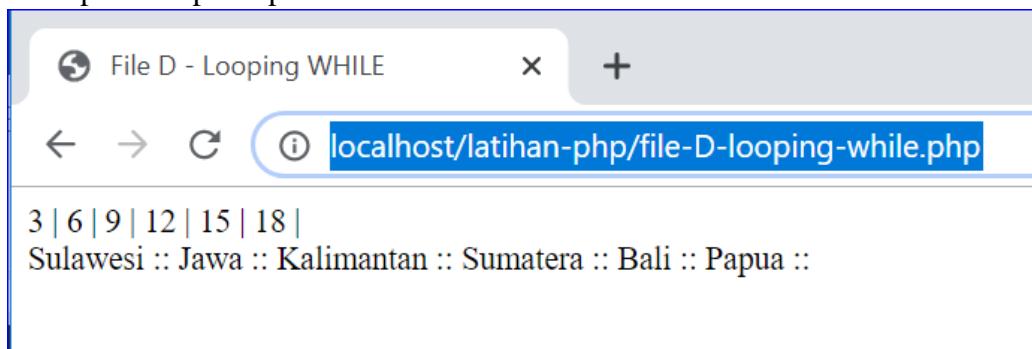
      do {
        echo "$pulau_pulau[$i] :: ";
        $i++;
      }
      while ( $i < count($pulau_pulau));

    ?>

  </body>
</html>
```

15. Jalankan file dengan mengakses url : **localhost/latihan-php/file-D-looping-while.php**

16. Maka tampilan output seperti berikut:



17. Jika berhasil, maka anda telah sukses mempraktekan logika perulangan pada koleksi data.

## E. MENGGUNAKAN LOGIKA SELEKSI

Logika seleksi juga sangat sering digunakan dalam memproses sebuah data. Seleksi memungkinkan pemrosesan berbeda terhadap data berdasarkan kriteria yang dipenuhi data tersebut.

Sebagai contoh skenario yang akan kita terapkan pada koleksi data kelipatan 3 pada program sebelumnya. Kali ini data tidak hanya sekedar dicetak tapi akan diperiksa jika bilangan tersebut ganjil atau genap. Kondisi ini bisa dicapai dengan logika seleksi.

Berikut langkah-langkah implementasi logika seleksi **IF-ELSE**

1. Buka kembali folder atau directory **latihan-php** dan buat file baru
2. Beri nama file baru tersebut: **file-E-seleksi-ifelse.php**
3. Kemudian ketik kode berikut:

```
<html>
    <head>
        <title>File E - SELEKSI</title>
    </head>
    <body>

        <?php

            $kelipatan_3 = array(3, 6, 9, 12, 15, 18);

            $arrObject1 = new ArrayObject($kelipatan_3);
            $arrayIterator1 = $arrObject1->getIterator();

            while( $arrayIterator1->valid() )
            {
                $angka = $arrayIterator1->current();
                if($angka%2==0){
                    echo "$angka | GENAP | <br />";
                }else{
                    echo "$angka | GANJIL | <br />";
                }

                $arrayIterator1->next();
            }

        ?>

    </body>
</html>
```

4. Kode diatas mencetak array data dengan menggunakan logika looping **WHILE**. Tapi sebelum dicetak dengan perintah **echo**, maka setiap elemen bilangan diperiksa jika elemen tersebut habis di modulo 2 ( $\$angka \% 2 == 0$ ), jika habis dibagi 2 maka bilangan genap, sebaliknya adalah bilangan ganjil.

5. Jalankan file dengan mengakses url: ***localhost/latihan-php/file-E-seleksi-ifelse.php***
6. Tampilan file akan seperti berikut:

The screenshot shows a browser window with the following details:

- Title bar: File E - SELEKSI
- Address bar: localhost/latihan-php/file-E-seleksi-ifelse.php
- Content area:
  - 3 | GANJIL |
  - 6 | GENAP |
  - 9 | GANJIL |
  - 12 | GENAP |
  - 15 | GANJIL |
  - 18 | GENAP |

7. Jika berhasil maka kita telah sukses menguji logika seleksi atau percabangan.

## F. MENGGUNAKAN FUNGSI

Pada contoh sebelumnya, looping **FOREACH** digunakan untuk menampilkan 2 buah variabel array. Sehingga kode **FOREACH** harus ditulis 2 kali karena ada 2 variabel array. Hal ini tentu tidak efisien menulis berkali-kali kode yang sama untuk setiap koleksi data yang diproses.

Keberadaan fungsi dapat digunakan untuk mengorganisasi kode sehingga kode dapat dipakai berulang-ulang (reusability) untuk melakukan pemrosesan data yang berbeda.

Namun fungsi tidak semata bertujuan untuk penggunaan kode secara berulang. Fungsi juga dapat digunakan untuk memetakan kode secara lebih baik dengan memisahkan kode-kode dengan fungsi berbeda.

Pada kali ini, kita akan memodifikasi kode program php bagian D yaitu logika **FOREACH** dengan menambahkan fungsi tampilkan data.

Berikut langkah-langkah menggunakan fungsi:

1. Buka kembali folder atau directory **latihan-php** dan buat file baru
2. Beri nama file baru tersebut: **file-F-fungsi.php**
3. Tuliskan kode program berikut:

```
<html>
    <head>
        <title>File F - Looping FOREACH FUNGSI</title>
    </head>
    <body>

        <?php

            $kelipatan_3 = array(3, 6, 9, 12, 15, 18);

            foreach ($kelipatan_3 as $angka) {
                echo "$angka | ";
            }

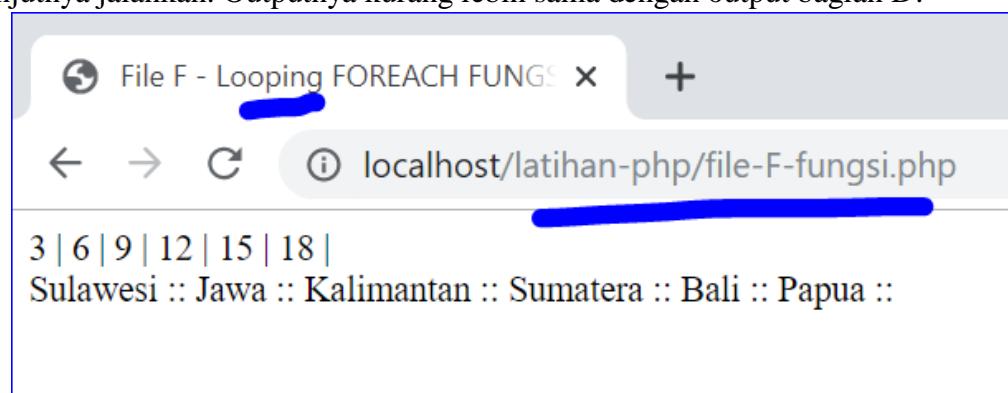
            $pulau_pulau = ["Sulawesi", "Jawa", "Kalimantan", "Sumatera", "Bali", "Papua"];

            echo "<br />";

            foreach ($pulau_pulau as $pulau) {
                echo "$pulau :: ";
            }

        ?>
    </body>
</html>
```

4. Selanjutnya jalankan. Outputnya kurang lebih sama dengan output bagian D.



5. Kode di atas tidak menggunakan fungsi.

6. Selanjutnya edit kode diatas dengan menambahkan fungsi yang akan mengeksekusi logika **FOREACH**, seperti berikut ini:

```
<html>
  <head>
    <title>File F - Looping FOREACH FUNGSI</title>
  </head>
  <body>

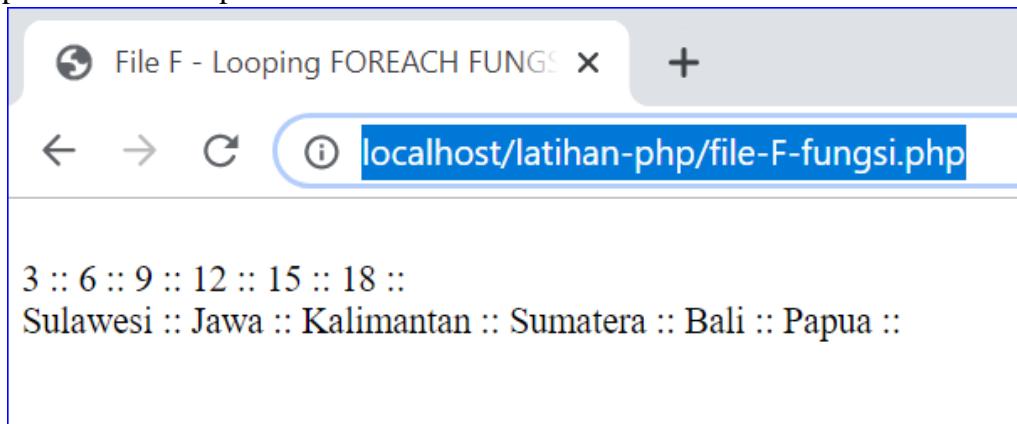
    <?php

      function cetakData($koleksi){
        echo "<br />";
        foreach ($koleksi as $data) {
          echo "$data :: ";
        }
      }

      $kelipatan_3 = array(3, 6, 9, 12, 15, 18);
      $pulau_pulau = ["Sulawesi", "Jawa", "Kalimantan", "Sumatera", "Bali", "Papua"];
      cetakData($kelipatan_3);
      cetakData($pulau_pulau);

    ?>
  </body>
</html>
```

7. Kode diatas menggunakan **fungsiCetaData** yang menerima argumen array data. Selanjutnya pada fungsi cetakData logika looping digunakan untuk menampilkan elemen data.  
8. Jalankan file dengan mengakses url: **localhost/latihan-php/file-F-fungsi.php**  
9. Tampilan file akan seperti berikut:



10. Halaman menghasilkan output yang sama menggunakan fungsi cetakData.

## G. MENGGUNAKAN FILE PHP EKSTERNAL DALAM KODE PHP

Untuk mengorganisasi kode program dengan baik, selain menggunakan fungsi, kita juga bisa memisahkan fungsi-fungsi tertentu didalam file eksternal terpisah. File -file ini nanti akan di-**include** ke dalam file yang akan menggunakan kode tersebut.

Kita akan membuat contoh fungsi pada program sebelumnya bagian F yaitu fungsi **cetakData** akan kita tempatkan dalam file eksternal kemudian kita include.

Berikut langkah-langkahnya:

1. Buka kembali folder atau directory **latihan-php** dan buat file baru
2. Beri nama file baru tersebut: **fungsi-eksternal.php**
3. Selanjutnya ketik kode php berikut ini:

```
<?php

function cetakData($koleksi){
    echo "<br />";

    foreach ($koleksi as $data) {
        echo "$data :: ";
    }
}

?>
```

4. Perhatikan bahwa pada file yang baru saja kita buat, hanya ada kode php murni didalamnya, yaitu kode fungsi **cetakData**.
5. Lalu buat lagi file lain dan beri nama file baru tersebut: **file-G-fungsi-eksternal.php**
6. Ketik kode berikut ini:

```
<html>
    <head>
        <title>File G - Include File Eksternal</title>
    </head>
    <body>

        <?php

            include("fungsi-eksternal.php");

            $kelipatan_3 = array(3, 6, 9, 12, 15, 18);

            $pulau_pulau = ["Sulawesi", "Jawa", "Kalimantan", "Sumatera", "Bali", "Papua"];

            cetakData($kelipatan_3);
            cetakData($pulau_pulau);

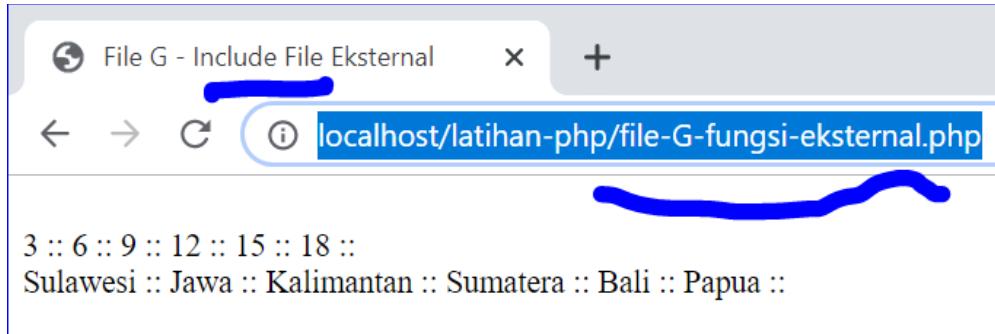
        ?>

    </body>
</html>
```

7. Pada kode diatas kita meng-**include** file eksternal yaitu **fungsi-eksternal.php** yang telah kita buat sebelumnya. Karena itu program di atas bisa menggunakan fungsi **cetakData**

meskipun tidak didefinisikan dalam file tersebut. Ini karena fungsi **cetakData** sudah didefinisikan dalam file eksternal **fungsi-eksternal.php**

8. Jalankan file dengan mengakses url: **localhost/latihan-php/file-G-fungsi-eksternal.php**
9. Tampilan file akan seperti berikut:



10. Jika tampilan seperti diatas maka anda berhasil membuat kode dengan fungsi dari file eksternal.

## H. MENGGUNAKAN STRUKTUR KELAS

Beberapa kasus dalam pemrosesan data, data yang dikelola adalah data dalam bentuk struktur data. Contoh data seperti ini adalah data mahasiswa. Data mahasiswa adalah data yang harus dibentuk oleh struktur data, sebab atribut data seperti NIM, nama adalah atribut data yang tidak terpisahkan sebagai satu kesatuan data.

Salah satu cara penanganan struktur data adalah dengan menggunakan **class** php. struktur **class** dapat dengan mudah dibentuk menyesuaikan struktur entity yang direpresentasikan **class** tersebut.

Kali ini kita akan memproses koleksi data dari objek struktur data **class**. berikut langkah-langkahnya:

1. Buka kembali folder atau directory **latihan-php**
2. Buat file struktur kelas Mahasiswa, beri nama: **Mahasiswa.php**
3. Tuliskan struktur data mahasiswa berikut:

```
<?php

    class Mahasiswa{
        public $nim;
        public $nama;

        public function __construct($nim, $nama){
            $this->nim = $nim;
            $this->nama = $nama;
        }
    }

?>
```

4. Buat file baru dan beri nama: **file-H-struktur-kelas.php**
5. Kemudian ketik kode berikut:

```
<html>
    <head>
        <title>File H - Struktur Kelas</title>
    </head>
    <body>

        <?php

            include("Mahasiswa.php");

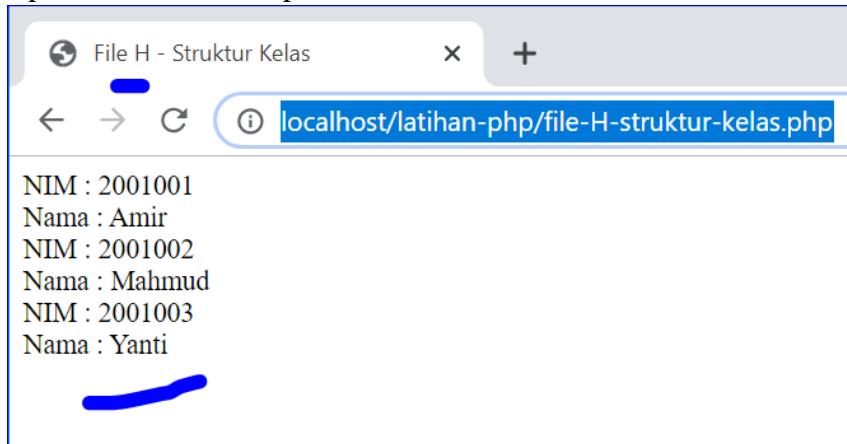
            $data [] = new Mahasiswa("2001001", "Amir");
            $data [] = new Mahasiswa("2001002", "Mahmud");
            $data [] = new Mahasiswa("2001003", "Yanti");

            foreach($data as $mahasiswa){
                echo "NIM : $mahasiswa->nim <br />";
                echo "Nama : $mahasiswa->nama <br />";
            }

        ?>

    </body>
</html>
```

6. Pada contoh diatas struktur kelas **Mahasiswa** digunakan sebagai struktur data untuk menampung data mahasiswa.
7. File program **file-H-struktur-kelas.php** memanfaatkan struktur **Mahasiswa** dengan meng-**include** struktur kelas Mahasiswa.
8. Jalankan file dengan mengakses url: **localhost/latihan-php/file-H-struktur-kelas.php**
9. Tampilan output browser akan seperti ini:



10. dad

## I. LINK HALAMAN

Halaman aplikasi web adalah sejumlah halaman-halaman terpisah yang terhubung menjadi satu. Pengguna bisa berpindah dari satu halaman ke halaman lain melalui link/url halaman tujuan yang ada pada satu halaman. Kali ini kita akan mempraktekan menghubungkan halaman web dengan url link. Kita akan menggunakan 3 halaman web.

Berikut langkah-langkahnya:

1. Buka kembali folder atau directory **latihan-php**
2. Buat file baru dan beri nama: **file-I1-link.php**
3. Ketik kode berikut:

```
<html>
  <head>
    <title>File I - Link Halaman</title>
  </head>
  <body>

    <h2>Link Halaman</h2>
    <a href="file-I2-link.php">Link ke: </a>file-I2-link.php
    <br />
    <a href="file-I3-link.php">Link ke: </a>file-I3-link.php
  </body>
</html>
```

4. Buat file baru dan beri nama: **file-I2-link.php**
5. Ketik kode berikut:

```
<html>
  <head>
    <title>File I - Link Halaman</title>
  </head>
  <body>

    <h2>Halaman: file-I2-link.php</h2>
    <a href="file-I1-link.php">Link kembali </a>file-I1-link.php
  </body>
</html>
```

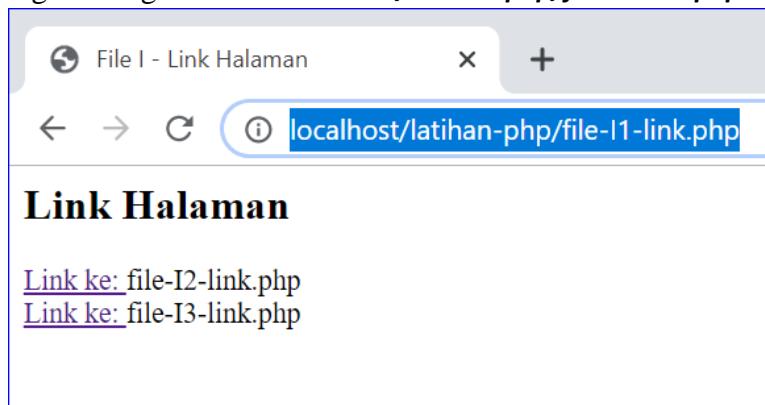
6. Buat file baru dan beri nama: **file-I3-link.php**
7. Ketik kode berikut:

```
<html>
  <head>
    <title>File I - Link Halaman</title>
  </head>
  <body>

    <h2>Halaman: file-I3-link.php</h2>
    <a href="file-I1-link.php">Link kembali </a>file-I1-link.php
  </body>
</html>
```

8. Ketiga file tersebut adalah halaman web terpisah yang telah dihubungkan dengan link.
9. File halaman **file-I1-link.php** memiliki 2 link menu masing-masing menuju ke **file-I2-link.php** dan menuju ke **file-I3-link.php**

10. Sedangkan pada halaman **file-I2-link.php** dan **file-I3-link.php** terdapat link menuju halaman **file-I1-link.php**
11. Jalankan file dengan mengakses url: **localhost/latihan-php/file-I1-link.php**



12. Klik link yang anda untuk berpindah halaman

## J. MENGIRIM DATA DARI SATU HALAMAN KE HALAMAN LAIN

Selain menghubungkan halaman web, setiap halaman dapat pula mengirim data ke halaman yang dituju. Data yang dikirim dapat dilampirkan sebagai parameter **url**. Parameter ini dimulai dengan tanda ?

Misalnya kita ingin mengirim parameter **nim** dan **nama** melalui sebuah **url** yang menuju file **file-J2-link.php** dengan nim 2001005 dan nama Agus Prasetyo, maka kita menformat url seperti : href="file-J2-link.php?nim=2001005&nama=Agus Prasetyo"

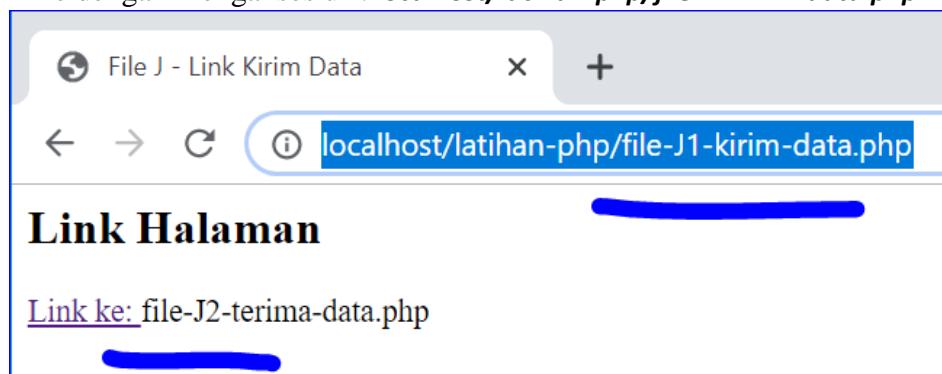
Berikut langkah-langkahnya:

1. Buka kembali folder atau directory **latihan-php**
2. Buat file baru dan beri nama: **file-J1-kirim-data.php**
3. Ketik kode berikut:

```
<html>
  <head>
    <title>File J - Link Kirim Data</title>
  </head>
  <body>

    <h2>Link Halaman</h2>
    <?php
      $nim = "2001005";
      $nama = "Agus Prasetyo";
      echo '<a href="file-J2-terima-data.php?nim=' . $nim . '&nama=' . $nama . '">
        Link ke: </a>file-J2-terima-data.php';
    ?>
  </body>
</html>
```

4. Pada kode diatas, pertama-tama data yang akan dikirim yaitu **nim** dan **nama** dideklarasikan dengan variabel \$nim dan \$nama dengan setiap nilai masing-masing.
5. Selanjutnya data ditambahkan pada href link tujuan, data **nim** ditambahkan pada akhir url setelah tanda ? dengan cara penulisan **?nim=\$nim**
6. Lalu data kedua yaitu nama ditambahkan dengan tanda pemisah &
7. Jadi untuk data pertama setelah url didahului dengan tanda ? selanjutnya data berikut untuk data kedua, ketiga atau seterusnya dipisah dengan tanda &
8. Jalankan file dengan mengakses url: **localhost/latihan-php/file-J1-kirim-data.php**



9. Klik link yang ada pada halaman tersebut, maka akan diarahkan ke halaman tujuan. Namun browser tidak akan menemukan file yang dimaksud karena kita belum membuat file **file-J2-terima-data.php** yang akan menerima data yang dikirim.
10. Selanjutnya kita akan membuat halaman file **file-J2-terima-data.php** yang menjadi tujuan link kirim data.

11. Buat file baru dan beri nama: **file-J2-terima-data.php**

12. Ketik kode berikut:

```
<?php
    echo "<html>";
    echo "  <head>";
    echo "    <title>File J - Halaman Terima Data</title>";
    echo "  </head>";
    echo "  <body>";
    echo "    <h2>INFO KIRIMAN DATA dari: file-J1-kirim-data.php</h2>";

    if(isset($_GET["nim"])){
        $nim = $_GET["nim"];
        echo "NIM : $nim";
    }else{
        echo "Data NIM tidak ditemukan!";
    }

    echo "<br />";

    if(isset($_GET["nama"])){
        $nama = $_GET["nama"];
        echo "Nama : $nama";
    }else{
        echo "Data Nama tidak ditemukan!";
    }

    echo "  </body>";
    echo "</html>";

?>
```

13. Pada kode diatas, bagian perintah untuk membaca data yang dikirim adalah **\$\_GET["nim"]** yang berarti data nim yang dikirim diterima pada sisi penerima dalam variabel global **\$\_GET**

14. Variabel **\$\_GET** merupakan array yang berisi data-data yang dikirim. Ini berarti jika kita mengirim data di url dengan kode **?data1=kalimantan&data2=sulawesi** maka data-data tersebut dapat diakses melalui variabel global **\$\_GET["data1"]** akan menghasilkan **kalimantan** dan **\$\_GET["data2"]** akan menghasilkan **sulawesi**

15. Namun jika kode mencoba mengakses data yang tidak terdapat dalam variabel global **\$\_GET** maka program akan error. Misalnya mencoba mengakses **\$\_GET["data100"]** dimana data100 ini tidak pernah dikirim.

16. Oleh karena itu penting menyisipkan logika seleksi dengan cek kondisi **isset()**

17. Fungsi **isset()** akan memeriksa jika data dalam argumennya bernilai

18. Fungsi **isset(\$\_GET["nim"])** akan bernilai **true** jika data **nim** ada.

19. Jalankan kembali program dan klik link menuju halaman dengan mengirim data melalui url

20. Maka halaman tujuan link akan tampak seperti berikut:



21. Perhatikan bagian **url** browser pada tampilan halaman di atas. url yang ada telah ditambahkan dengan informasi data yang telah dikirim yaitu:

<http://localhost/latihan-php/file-J2-terima-data.php?nim=2001005&nama=Agus%20Prasetyo>

22. Dengan demikian kita telah berhasil mengirim data dari satu halaman ke halaman lain melalui url.

## K. MEMBUAT FORM INPUT DAN MENGIRIM DATA DARI FORM INPUT KE HALAMAN ACTION

Aplikasi web membutuhkan interaksi pengguna misalnya menambahkan data, mengedit data atau mengetik pencarian data. Semua itu dapat dilakukan melalui antar muka **form** input. **Form** input berguna menerima data input pengguna dan menyimpannya kedalam variabel data. Kita akan mempraktekan menginput data dan setelah itu mengirim data tersebut ke halaman lain. Tapi sebelumnya kita akan membuat form input.

Berikut langkah-langkahnya:

1. Buka kembali folder atau directory **latihan-php**
2. Buat file baru dan beri nama: **file-K1-form-input.php**
3. Ketik kode berikut:

```
<html>
  <head>
    <title>File K - Form Input</title>
  </head>
  <body>

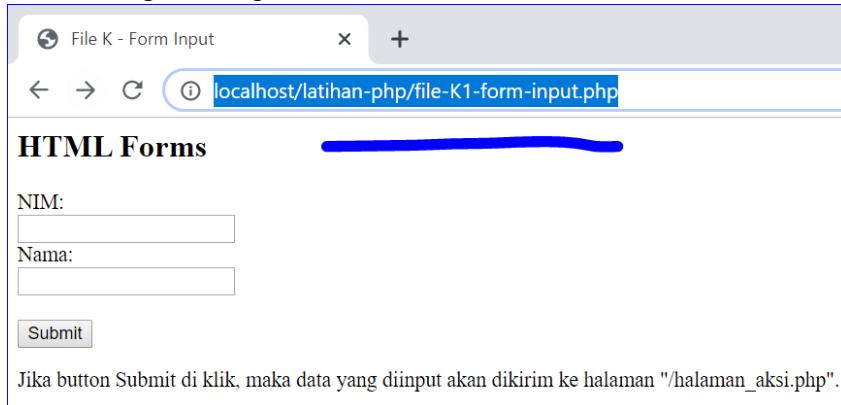
    <h2>HTML Forms</h2>

    <form action="halaman_aksi.php" method="post">
      NIM:<br>
      <input type="text" name="nim" >
      <br>
      Nama:<br>
      <input type="text" name="nama">
      <br><br>
      <input type="submit" value="Submit">
    </form>

    <p>Jika button Submit di klik, maka data yang
       |diinput akan dikirim ke halaman "/halaman_aksi.php".</p>
  </body>
</html>
```

4. Perhatikan pada kode **form** atribut **action** dan atribut **method**
5. Atribut **action** adalah tujuan halaman web dimana data yang diinputkan pada form akan dikirim. Pada contoh di atas data input pada form akan dikirim pada **halaman\_aksi.php** untuk diproses selanjutnya.
6. Atribut **method** digunakan untuk menentukan cara pengiriman data dari halaman **form** input ke **halaman\_aksi.php**
7. **Method** standar ada 2 yaitu **post** dan **get**
8. Selanjutnya setiap data input pada tag **input** ditandai dengan atribut **name**.
9. Atribut **name** itu adalah nama variabel data yang akan dikirim. Misalnya **name="nim"** artinya data input nanti dapat diakses dengan indeks **nim** pada variabel global. Jika menggunakan method **post** maka diakses di variabel global **\$\_POST["nim"]** sebaliknya Jika menggunakan method **get** maka diakses di variabel global **\$\_GET["nim"]**

10. Coba jalankan file dengan mengakses url: **localhost/latihan-php/file-K1-form-input.php**



File K - Form Input

localhost/latihan-php/file-K1-form-input.php

### HTML Forms

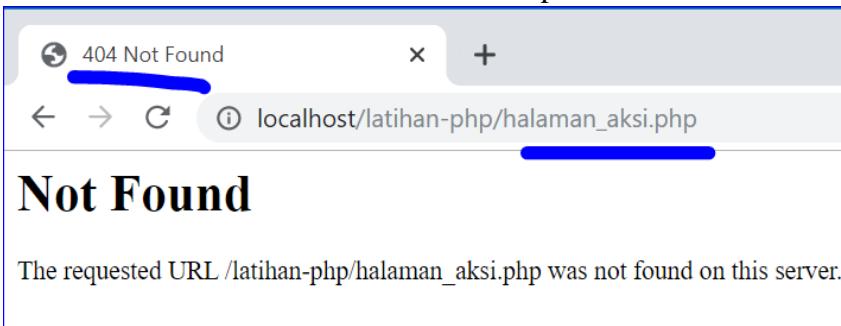
NIM:

Nama:

Submit

Jika button Submit di klik, maka data yang diinput akan dikirim ke halaman "/halaman\_aksi.php".

11. Jika anda mencoba klik button submit maka akan tampil error not found.



12. Error ini karena kita belum membuat file **halaman\_aksi.php**

13. Buat file baru dan beri nama: **halaman\_aksi.php**

14. Ketik kode berikut:

```
<?php
    echo "<h2>Halaman Aksi</h2>";
    if(isset($_POST["nim"])){
        $nim = $_POST["nim"];
        echo "NIM : $nim";
    }else{
        echo "Data NIM tidak ditemukan!";
    }

    echo "<br />";

    if(isset($_POST["nama"])){
        $nama = $_POST["nama"];
        echo "Nama : $nama";
    }else{
        echo "Data Nama tidak ditemukan!";
    }
?>
```

15. Pada kode diatas **key-index** yaitu **nim** dan **nama** dari variabel global **\$\_POST** disesuaikan dengan atribut TAG **input** yaitu **name** pada form input
16. Coba jalankan kembali file dengan mengakses url: **localhost/latihan-php/file-K1-form-input.php**
17. Kemudian input data NIM dan Nama seperti berikut:

File K - Form Input

localhost/latihan-php/file-K1-form-input.php

## HTML Forms

NIM:  
200106578

Nama:  
Suparmin

Submit

Jika button Submit di klik, maka data yang diinput akan dikirim ke halaman "/halaman\_aksi.php".

18. Klik submit sehingga halaman aksi akan tampil dan memproses data input sesuai data yang dimasukan seperti tampilan berikut:

localhost/latihan-php/halaman\_a

localhost/latihan-php/halaman\_aksi.php

## Halaman Aksi

NIM : 200106578

Nama : Suparmin

19. Dengan demikian kita telah berhasil mengirim data dari halaman input kehalaman lain sebagai halaman aksi yang akan memproses data input.

## L. MEMBEDAKAN METHODE POST DAN GET PADA FORM

Pada form sebelumnya kita menggunakan **method="post"** pada atribut form. Kali ini kita akan menggunakan **method="get"** sebagai pengganti post.

Berikut langkah-langkahnya:

1. Buka kembali folder atau directory **latihan-php**
2. Buat file baru dan beri nama: **file-L1-form-input.php**
3. Ketik kode berikut:

```
<html>
  <head>
    <title>File L - Form Input</title>
  </head>
  <body>

    <h2>HTML Forms</h2>

    <form action="halaman_aksi_get.php" method="get">
      NIM:<br>
      <input type="text" name="nim" >
      <br>
      Nama:<br>
      <input type="text" name="nama">
      <br><br>
      <input type="submit" value="Submit">
    </form>

    <p>Jika button Submit di klik, maka data yang
       diinput akan dikirim ke halaman "/halaman_aksi_get.php".</p>
  </body>
</html>
```

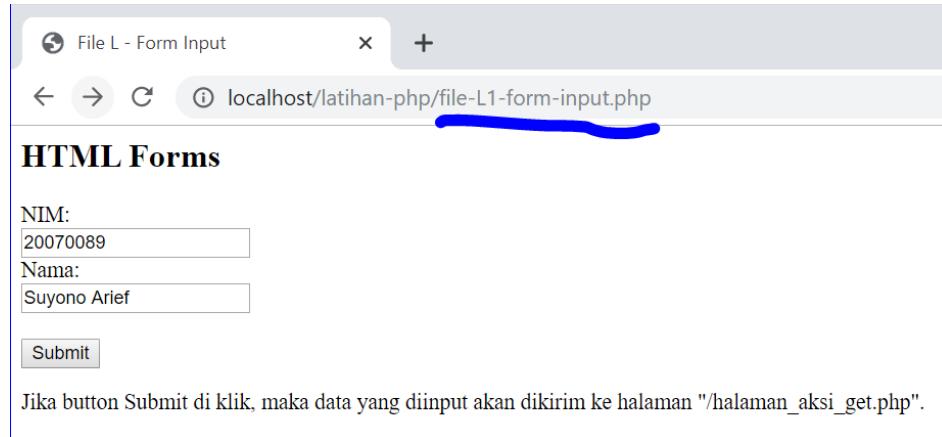
4. Kode diatas sama persis dengan file **file-K1-form-input.php** sebelumnya. Kecuali perbedaan pada **method="get"** dan **halaman\_aksi\_get.php**. halaman aksi sengaja dibedakan agar tidak perlu mengedit file **halaman\_aksi.php** sebelumnya
5. Selanjutnya buat file baru dan beri nama: **halaman\_aksi\_get.php**
6. Ketik kode berikut:

```
<?php
  echo "<h2>Halaman Aksi</h2>";
  if(isset($_GET["nim"])){
    $nim = $_GET["nim"];
    echo "NIM : $nim";
  }else{
    echo "Data NIM tidak ditemukan!";
  }

  echo "<br />";

  if(isset($_GET["nama"])){
    $nama = $_GET["nama"];
    echo "Nama : $nama";
  }else{
    echo "Data Nama tidak ditemukan!";
  }
?>
```

7. Pada kode diatas variabel global yang digunakan adalah `$_GET` berbeda dengan sebelumnya yang menggunakan `$_POST`
8. Coba jalankan file dengan mengakses url: `localhost/latihan-php/file-L1-form-input.php`
9. Kemudian input data NIM dan Nama seperti berikut:



File L - Form Input

localhost/latihan-php/file-L1-form-input.php

### HTML Forms

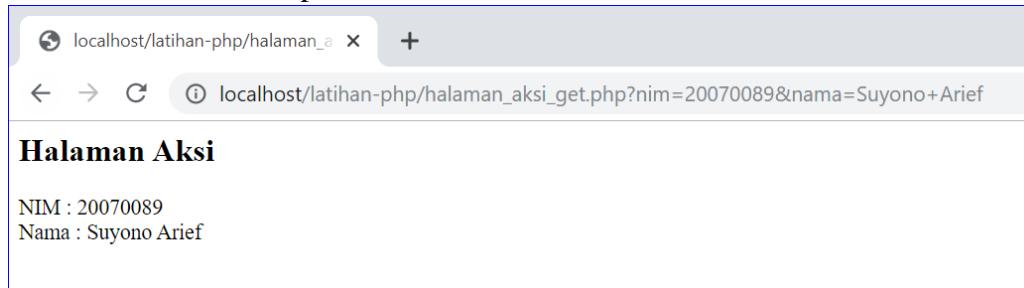
NIM:  
20070089

Nama:  
Suyono Arief

Submit

Jika button Submit di klik, maka data yang diinput akan dikirim ke halaman "/halaman\_aksi\_get.php".

10. Klik submit maka akan tampil halaman aksi



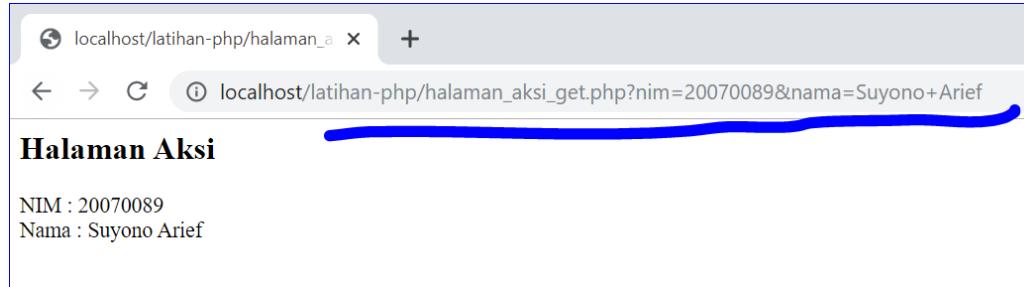
localhost/latihan-php/halaman\_a

localhost/latihan-php/halaman\_aksi\_get.php?nim=20070089&nama=Suyono+Arief

### Halaman Aksi

NIM : 20070089  
Nama : Suyono Arief

11. Perhatikan url yang terdapat pada halaman\_aksi\_get.php di url terdapat parameter data yang dikirim seperti pada gambar:



localhost/latihan-php/halaman\_a

localhost/latihan-php/halaman\_aksi\_get.php?nim=20070089&nama=Suyono+Arief

### Halaman Aksi

NIM : 20070089  
Nama : Suyono Arief

12. Jika halaman tampil maka anda telah berhasil mengirim data menggunakan methode GET pada form input.

## M. MEMBUAT DATABASE MYSQL

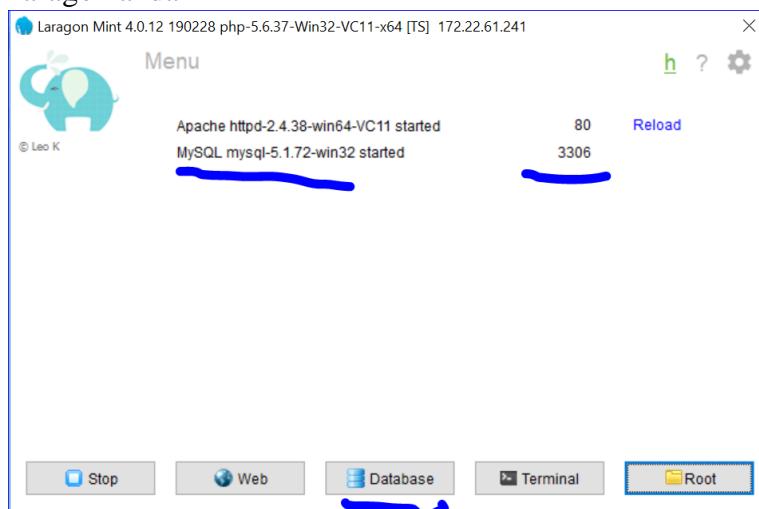
Selanjutnya kita akan membuat database untuk aplikasi web yang akan kita buat. Database yang akan kita buat menggunakan server database lokal MySQL yang telah terinstal pada Laragon. Pastikan program MySQL pada Laragon berjalan.

Untuk mengakses database MySQL, kita membutuhkan tool IDE. Ada beberapa IDE yang bisa kita pakai seperti PHPMyadmin DBeaver atau HeidiSQL. Kita akan menggunakan HeidiSQL dalam praktik ini.

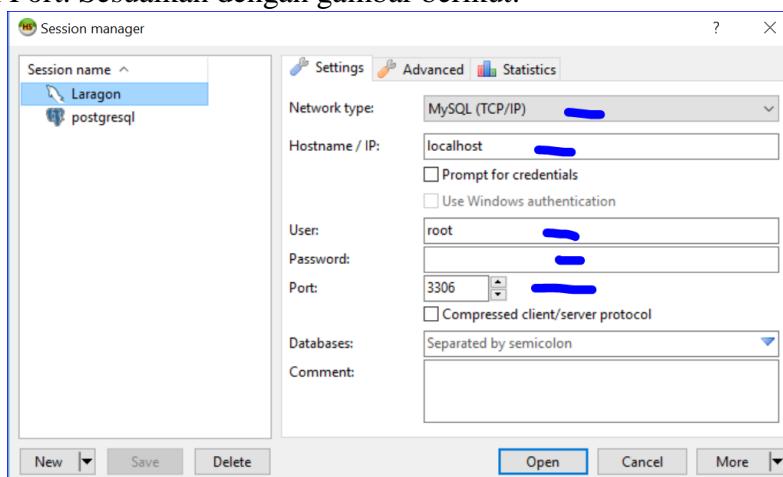
Pertama-tama download dan install HeidiSQL jika anda belum menginstallnya. Pastikan HeidiSQL terinstal dan menjadi program default untuk membuka akses server MySQL.

Kita akan membuat database untuk menyimpan data mahasiswa dalam MySQL. Berikut lanjkah-langkahnya:

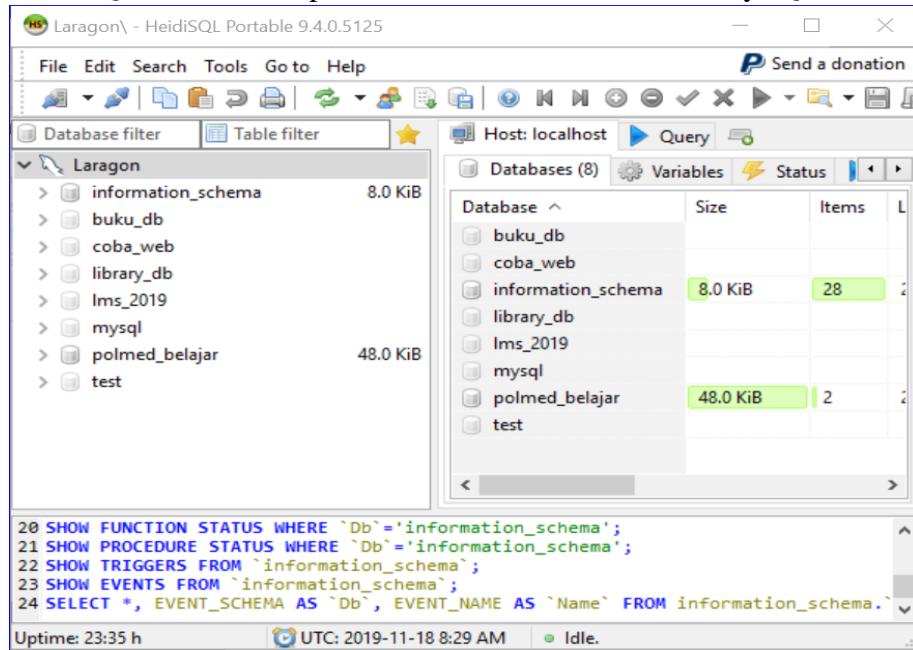
1. Buka program Laragon anda



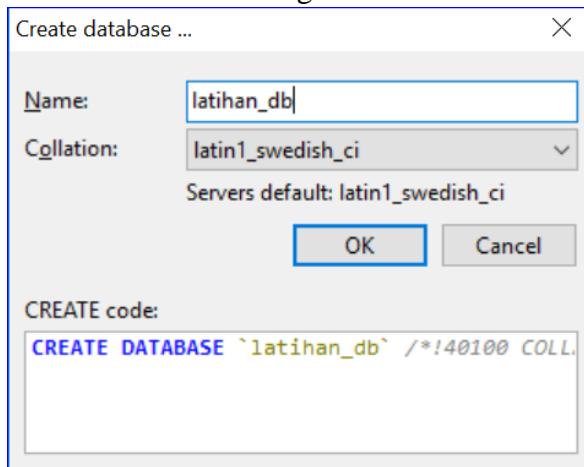
2. Pastikan MySQL berjalan dan nomor port **3306** tampil
3. Klik pada button **Database**
4. Jika konfigurasi benar, maka program **HeidiSQL** akan terbuka. Jika HeidiSQL tidak terbuka anda bisa menjalankan manual program HeidiSQL
5. Perhatikan konfigurasi program HeidiSQL pada bagian NetworkType, Hostname, User, Password dan Port. Sesuaikan dengan gambar berikut:



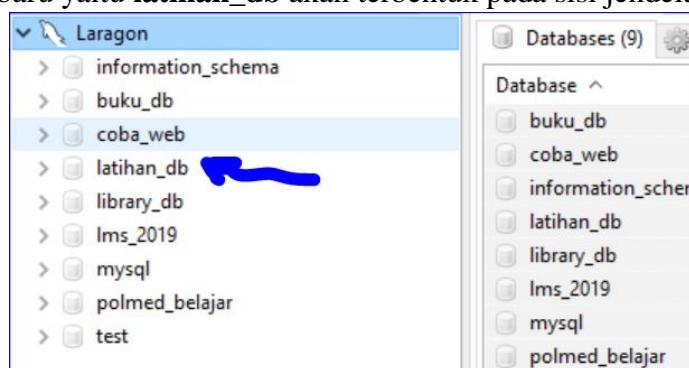
6. Klik button **Open**
7. Program HeidiSQL akan menampilkan halaman akses ke server MySQL



8. Kemudian **Klik kanan** pada **Laragon** -> **Create New -> Database**
9. Jendela create database akan muncul. Isi dengan nama database: **latihan\_db**



10. Kemudian Klik OK pada jendela create database
11. Sebuah database baru yaitu **latihan\_db** akan terbentuk pada sisi jendela kiri:



12. Selanjutnya kita akan membuat tabel di dalam database **latihan\_db**
13. Klik kanan pada database **latihan\_db** pilih **Create new -> table**
14. Halaman create table akan ditampilkan, isi dengan data untuk table mahasiswa seperti berikut ini:

The screenshot shows the MySQL Workbench interface for creating a new table. The top bar indicates 'Host: localhost', 'Database: latihan\_db', and 'Table: [Untitled]'. The 'Basic' tab is selected. The 'Name' field is highlighted with a blue underline and contains the text 'tmahasiswa'. The 'Comment' field is empty. Below the table name, the 'Columns' section is visible, showing two columns: 'nim' and 'nama'. The 'nim' column is defined as VARCHAR with a length of 20, and the 'nama' column is defined as VARCHAR with a length of 50.

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zer...
1	nim	VARCHAR	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
2	nama	VARCHAR	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

15. Pada **Name** kita isi dengan **tmahasiswa** yang merupakan nama tabel. Selanjutnya untuk field column kita isi dengan **nim** yang merupakan **Name** column, **Datatype Varchar** dengan **Length 20**
16. Kemudian column kita isi dengan **nama** yang merupakan **Name** column kedua, **Datatype Varchar** dengan **Length 50**
17. Selanjutnya klik button **save**
18. Tabel **tmahasiswa** akan terbentuk didalam database **latihan\_db** seperti pada gambar

The screenshot shows the MySQL Workbench interface with the 'Tables' list on the left and the table configuration on the right. The 'Tables' list shows several databases: Laragon, information\_schema, buku\_db, coba\_web, latihan\_db, library\_db, Ims\_2019, mysql, polmed\_belajar, and test. The 'tmahasiswa' table under the 'latihan\_db' database is selected and highlighted with a blue background. On the right, the table configuration window is open, showing the same details as the previous screenshot: Name: tmahasiswa, Columns: nim (VARCHAR, Length 20) and nama (VARCHAR, Length 50).

19. Kemudian Klik pilih **tmahasiswa** lalu pilih klik **Tab Data**

The screenshot shows the HeidiSQL interface. On the left, the database structure is visible with 'tmahasiswa' selected under 'latihan\_db'. On the right, the 'Data' tab is active for the 'tmahasiswa' table. The table has two columns: 'nim' and 'nama'. There are currently no data rows displayed.

20. Terlihat belum ada data dalam tabel. Kita akan menambah 2 data sebagai data awal tabel.

21. Untuk menambah data, **Klik** tanda bulatan hijau dengan gambar + pada menu bar di atas tab. Kemudian isikan data seperti gambar berikut:

The screenshot shows the HeidiSQL interface after data has been added. The 'Data' tab for the 'tmahasiswa' table now displays two rows of data. The first row has 'nim' value '20010099' and 'nama' value 'Arief Budiman'. The second row has 'nim' value '20010098' and 'nama' value 'Hendra Saputra'. A blue arrow points to the green '+' icon in the toolbar above the data grid.

22. Sampai di sini anda telah berhasil membuat database dan tabel serta menambahkan data. Selanjutnya kita akan mempraktekan bagaimana menghubungkan halaman web php dengan database dan menampilkan data yang ada dalam tabel.

## N. KONEKSI KE DATABASE DENGAN PHP

Halaman web dapat dihubungkan ke database dengan menggunakan kode PHP. ada beberapa cara yang bisa digunakan seperti MYSQLi dan PDO. Pada latihan ini kita akan menggunakan PDO untuk menghubungkan PHP dengan MySQL server.

Dalam menghubungkan kode ke database, beberapa variabel penting koneksi harus dipahami dengan baik seperti: alamat server atau servername, nomor port, username, password dan nama database yang terdapat dalam server MySQL. Server MySQL secara default akan beralamat pada **localhost** jika diinstal lokal menggunakan Laragon, nomor **port** default yaitu 3306, sedangkan username adalah **root** dengan password kosong. Variabel nama database tergantung dari nama database pengguna yang telah dibuatnya.

Berikut langkah-langkah koneksi halaman web dengan server MySQL:

1. Buka kembali folder atau directory **latihan-php**
2. Buat file baru dan beri nama: **file-N-koneksi-mysql.php**
3. Ketik kode berikut:

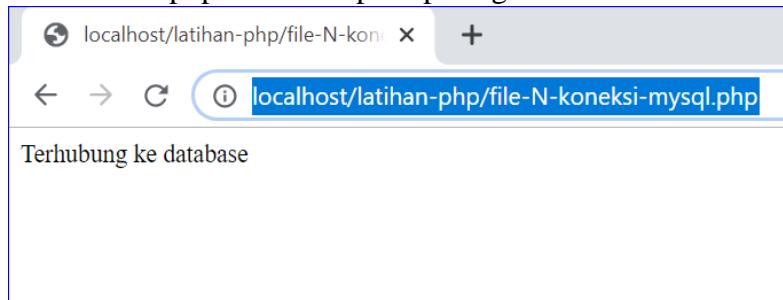
```
<?php

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "latihan_db";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    echo "Terhubung ke database";
}
catch(PDOException $e)
{
    echo "Gagal terhubung: " . $e->getMessage();
}

?>
```

4. Pada kode diatas, variabel **\$servername**, **\$username**, **\$password** dan **\$dbname** dideklarasikan sebagai variabel dengan inisialisasi nilai. Selanjutnya variabel ini digunakan oleh object PDO untuk menghubungi server
5. Coba jalankan file dengan mengakses url: **localhost/latihan-php/file-N-koneksi-mysql.php**
6. Halaman tersebut akan menampilkan informasi “Terhubung ke database” jika server bisa diakses dengan kode PDO php di atas. Seperti pada gambar:



7. Demikian cara koneksi dengan PDO ke server MySQL

## O. MEMBACA DATA DARI DATABASE

Selanjutnya kita akan membaca data dalam tabel **tmahasiswa** dalam database **latihan\_db** yang telah kita buat sebelumnya. Data ini akan ditampilkan tanpa format HTML utnuk memudahkan pemahamannya.

Berikut langkah-langkahnya:

1. Buka kembali folder atau directory **latihan-php**
2. Buat file baru dan beri nama: **file-O-baca-tabel.php**
3. Ketik kode berikut:

```
<?php

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "latihan_db";

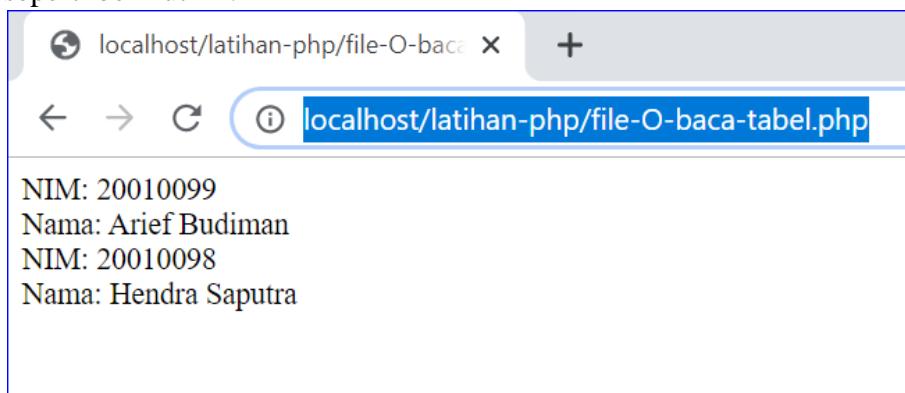
$query = 'SELECT * FROM tmahasiswa';

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $data = $conn->query($query);
    while ($row = $data->fetch())
    {
        echo "NIM: " . $row['nim'] . "<br />";
        echo "Nama: " . $row['nama'] . "<br />";
    }
} catch(PDOException $e)
{
    echo "Gagal terhubung: " . $e->getMessage();
}

?>
```

4. Kode diatas mirip dengan kode koneksi PHP ke server MySQL. Perbedaannya pada kode diatas didefinisikan variabel **\$query** yang merupakan query MySQL untuk membaca semua data pada tabel **tmahasiswa**
5. Selanjutnya terdapat statemen: **\$conn->query(\$query);**; kode ini bertujuan mengeksekusi perintah query dan mengembalikan hasilnya sebagai koleksi data.
6. Selanjutnya kode perulangan **while** berfungsi untuk menampilkan data pada browser.
7. Coba jalankan file dengan mengakses url: **localhost/latihan-php/file-O-baca-tabel.php**
8. Jika berhasil terhubung maka halaman tersebut akan menampilkan data yang dibaca dari database seperti berikut ini:



9. Pada contoh diatas, data yang dibaca langsung di **ECHO** kan sebagai output data. Jika kita ingin proses lebih lanjut, misalnya mengirim data yang telah dibaca tadi ke halaman lain, maka ada baiknya data yang telah dibaca disimpan dalam bentuk struktur data yang selanjutnya disimpan sebagai koleksi array. Untuk tujuan seperti itu, kita pertama-tama harus membuat struktur data **class Mahasiswa** seperti yang telah kita praktekan sebelumnya. Bedanya kali ini kita mengambil data dari database MySQL. Kita akan meninjau praktek koleksi data dengan struktur data pada modul selanjutnya.

## P. MENAMBAH DATA

Menambah data ke dalam tabel database pada dasarnya mirip dengan membaca data. Yang harus kita pahami adalah perintah **query** menambah data ke dalam tabel. Hanya saja, menambah data tidak bisa dilakukan dengan query yang fixed atau statis. Query yang diperlukan adalah query dinamis atau query yang dapat menerima variabel data input.

Sebagai contoh query berikut bersifat statis:

```
INSERT INTO tmahasiswa (nim, nama)
VALUES ('20010098', 'Hendra Saputra');
```

Query diatas jika dieksekusi hanya akan menambah data yang tetap yaitu mahasiswa dengan nim=20010098 dan nama=Hendra Saputra. Oleh karena itu kita membutuhkan query yang dinamis.

Query tersebut dapat dijadikan dinamis seperti berikut:

```
INSERT INTO tmahasiswa (nim, nama)
VALUES (:nim, :nama);
```

Sekarang querynya telah menjadi dinamis dengan menjadikan input nim dan nama sebagai variabel yang dapat dimodifikasi dari luar dengan kode PHP.

Berikut langkah-langkah menambah data dengan PDO:

1. Buka kembali folder atau directory **latihan-php**
2. Buat file baru dan beri nama: **file-P-insert-data.php**
3. Ketik kode berikut:

```
<?php

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "latihan_db";

$query = 'INSERT INTO tmahasiswa (nim, nama) VALUES (:nim, :nama)';
$nim = "2001238876";
$nama = "Hakim Usri";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $stmt = $conn->prepare($query);
    $stmt->bindParam(':nim', $nim);
    $stmt->bindParam(':nama', $nama);
    $stmt->execute();
    echo "Data berhasil ditambahkan!";
}

catch(PDOException $e)
{
    echo "Gagal terhubung: " . $e->getMessage();
}

?>
```

- Pada kode diatas, kita menggunakan perintah **bindParam** untuk memberikan nilai variabel **:nim** dan **:nama** yang terdapat pada query sehingga query nanti bisa digunakan jika ingin menambahkan data berbeda.
- Coba jalankan file dengan mengakses url: **localhost/latihan-php/file-P-insert-data.php**
- Jika berhasil maka data baru akan ditambahkan ke dalam tabel **tmahasiswa** yaitu data mahasiswa nim : 2001238876 dengan nama Hakim Usri
- Periksa kondisi data dalam tabel **tmahasiswa** dengan menggunakan heidisSQL:

Laragon\\latihan\_db\\tmahasiswa - HeidiSQL Portable 9.4.0.5125

File Edit Search Tools Go to Help

Database filter Table filter

Laragon

- information\_schema
- buku\_db
- coba\_web
- latihan\_db
- tmahasiswa** 1.0 KiB
- library\_db
- Ims\_2019
- mysql
- polmed\_belajar
- test

Host: localhost Database: latihan\_db.tmahasiswa: 3 rows tot

nim	nama
20010099	Arief Budiman
20010098	Hendra Saputra
2001238876	Hakim Usri

- Atau bisa juga memeriksanya dengan mengakses halaman baca tabel sebelumnya pada alamat url: **localhost/latihan-php/file-O-baca-tabel.php**
- Jika halaman tersebut ditampilkan maka akan terlihat data telah bertambah seperti gambar berikut ini:

localhost/latihan-php/file-O-baca-tabel.php

NIM: 20010099  
Nama: Arief Budiman  
NIM: 20010098  
Nama: Hendra Saputra  
NIM: 2001238876  
Nama: Hakim Usri

- Kita telah berhasil menulis kode implementasi logika menambah data dalam tabel. Di sini juga bisa dipahami data yang hendak ditambahkan bisa saja berasal dari halaman lain yang dikirim melalui **form** input.

## Q. MEMPERBAHARUI DATA

Update atau memperbaharui data sama persis dengan logika menambah data baru. Yang berbeda adalah query yang digunakan. Berikut ini query yang bisa digunakan memperbaharui data:

```
UPDATE tmahasiswa SET nama=:nama WHERE nim=:nim;
```

Berikut langkah-langkahnya:

1. Buka kembali folder atau directory **latihan-php**
2. Buat file baru dan beri nama: **file-Q-update-data.php**
3. Ketik kode berikut:

```
<?php

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "latihan_db";

$query = 'UPDATE tmahasiswa SET nama=:nama WHERE nim=:nim;';
$nim = "2001238876";
$nama = "Hakim Usri";
_____
```

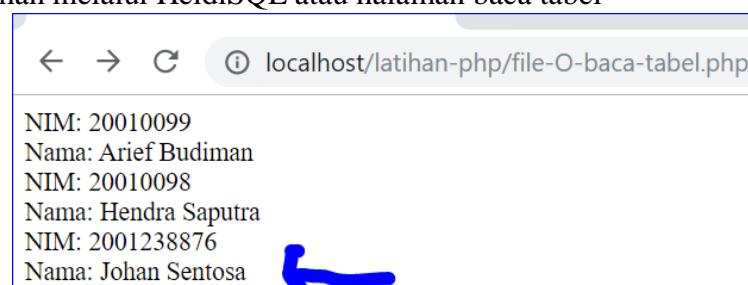
```
try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $stmt = $conn->prepare($query);
    $stmt->bindParam(':nim', $nim);
    $stmt->bindParam(':nama', $nama);
    $nama = "Johan Sentosa"; ←
    $stmt->execute();
    echo "Data berhasil diperbaharui!";

}
catch(PDOException $e)
{
    echo "Gagal terhubung: " . $e->getMessage();
}

?>
```

4. Pada kode di atas nama **Hakim Usri** yang telah ada dalam database dganti menjadi **Johan Sentosa** dengan query update
5. Coba jalankan file dengan mengakses url: **localhost/latihan-php/file-Q-update-data.php**
6. Jika berhasil maka data nama **Hakim Usri** akan diubah menjadi **Johan Sentosa**
7. Periksa perubahan melalui HeidiSQL atau halaman baca tabel



8. Perintah update berhasil. Dengan demikian kita telah berhasil mengimplementasikan logika pembaharuan data.

## R. MENGHAPUS DATA

Operasi menhapus data sama saja dengan operasi insert dan update data. Perbedaannya hanya sekali lagi pada query. Berikut query dinamis untuk menghapus data berdasarkan kata kunci primary key tabel yaitu kolom nim:

```
DELETE FROM tmahasiswa WHERE nim=:nim;
```

Berikut langkah-langkah hapus data:

1. Buka kembali folder atau directory **latihan-php**
2. Buat file baru dan beri nama: **file-R-delete-data.php**
3. Ketik kode berikut:

```
<?php

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "latihan_db";

$query = 'DELETE FROM tmahasiswa WHERE nim=:nim;';
$nim = "2001238876";
$nama = "Hakim Usri";

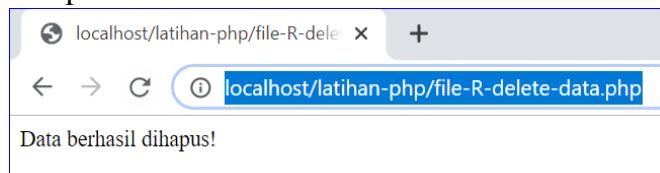
try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $stmt = $conn->prepare($query);
    $stmt->bindParam(':nim', $nim);
    $stmt->execute();
    echo "Data berhasil dihapus!";
}

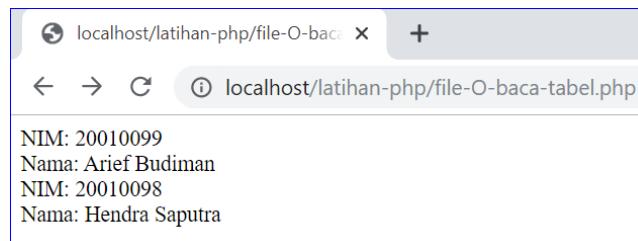
catch(PDOException $e)
{
    echo "Gagal terhubung: " . $e->getMessage();
}

?>
```

4. Jika file dijalankan pada url: **localhost/latihan-php/file-R-delete-data.php** maka data nim 2001238876 akan terhapus dari tabel



5. Data yang tersisa dalam tabel:



## S. UPLOAD FILE DENGAN PHP DAN FORM INPUT

Upload file seperti berkas pdf atau file gambar terkadang diperlukan jika aplikasi kita butuh menyajikan file gambar atau berkas pdf. Tidak ada perbedaan upload gambar atau berkas text. Hanya saja secara default server web membatasi ukuran file maksimal yang boleh diupload jika melebihi ketentuan maka akan terjadi error.

Berikut ini adalah langkah-langkah upload file:

1. Buka kembali folder atau directory **latihan-php**
2. Buat file baru dan beri nama: **file-S1-form-upload.php**
3. Ketik kode berikut:

```
<form action="aksi_upload_file.php" method="post" enctype="multipart/form-data">
    Pilih gambar:
    <input type="file" name="berkas" id="berkas">
    <input type="submit" value="Upload Gambar" name="submit">
</form>
```

4. Selanjutnya buat folder baru dengan nama **gambar**
5. Lalu buatlah file aksi form upload gambar yaitu: **aksi\_upload\_file.php**
6. Ketik kode berikut untuk proses file upload:

```
<?php
    // Asumsi file gagal diupload
    if ($_FILES['berkas']['error'] != UPLOAD_ERR_OK) {
        die("Upload gagal dengan error code: " . $_FILES['file']['error']);
    }

    $info = getimagesize($_FILES['berkas']['tmp_name']);
    if ($info === FALSE) {
        die("Tidak bisa membaca jenis file");
    }

    if (($info[2] != IMAGETYPE_GIF) && ($info[2] != IMAGETYPE_JPEG)
        && ($info[2] != IMAGETYPE_PNG)) {
        die("Bukan file gambar gif/jpeg/png");
    }

    // ambil data file
    $namaFile = $_FILES['berkas']['name'];
    $namaSementara = $_FILES['berkas']['tmp_name'];

    // Check file size
    if ($_FILES["berkas"]["size"] > 500000) {
        die( "File terlalu besar." );
    }

    // tentukan folder penyimpanan
    $dirUpload = "gambar/";

    // pindahkan file ke folder gambar
    $terupload = move_uploaded_file($namaSementara, $dirUpload.$namaFile);

    if ($terupload) {
        echo "Upload berhasil!<br/>";
        echo "Link: <a href='".$dirUpload.$namaFile."'>".$namaFile."</a>";
    } else {
        echo "Upload Gagal!";
    }
?>
```

7. Jalankan form upload dengan akses url: *localhost/latihan-php/file-S1-form-upload.php*

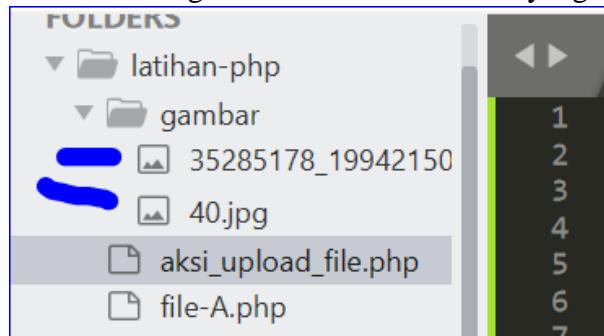
A screenshot of a web browser window. The address bar shows the URL *localhost/latihan-php/file-S1-form-upload.php*. Below the address bar is a form with a label "Pilih gambar:" followed by a "Choose File" button and a message "No file chosen". To the right of the button is a "Upload Gambar" button.

8. Pilih file gambar kemudian tekan upload gambar

9. Jika upload berhasil maka halaman akan menampilkan informasi berhasil

A screenshot of a web browser window. The address bar shows the URL *localhost/latihan-php/aksi\_upload\_file.php*. The page content displays the message "Upload berhasil!" and a link "Link: [40.jpg](#)".

10. Anda juga dapat memeriksa folder gambar untuk melihat file yang telah berhasil diupload



11. Jika berjalan lancar berarti anda telah berhasil mengimplementasikan logika upload file gambar.

## T. STUDI KASUS MEMBUAT APLIKASI WEB PHP DAN MYSQL.

Dalam membuat aplikasi web perlu dipahami ada 3 bagian utama web yang perlu kita rancang dan desain dengan baik sebelum menulis kode program. Tiga bagian itu adalah Database, Interface/tampilan dan proses data dari database ke tampilan pengguna.

Karena terdiri dari 3 bagian, maka pembuat web aplikasi bisa mulai bekerja dari sisi berbeda. Pekerjaan bisa dimulai dengan membuat struktur database terlebih dahulu. Namun kemungkinan ada yang memilih membuat tampilan lebih dahulu, sedangkan programmer lain melihat lebih penting mengokohkan proses perangkat lunak dibanding database dan tampilan. Ini tergantung dari kebiasaan programmer.

Pada studi kasus ini, kita akan membuat aplikasi koleksi buku. Berikut langkah-langkahnya:

### 1. Buat database

- a. Buatlah database yang kita perlukan dengan menggunakan HeidiSQL. Silahkan rujuk praktikum sebelumnya.
- b. Buat database dengan nama **db\_latihan\_kasus**. Klik kanan pada koneksi laragon lalu pilih Create database
- c. Buat tabel dengan nama **tbuku**. Klik kanan pada database, pilih Create table
- d. Tambahkan field tabel seperti berikut (pilih table **tbuku** lalu tambah field):

Field Name	Tipe Data	Length	Ket
isbn	VARCHAR	25	Primary Key
judul	VARCHAR	100	
deskripsi	VARCHAR	300	

- e. Kemudian tambahkan data awal ke dalam tabel tbuku seperti berikut:

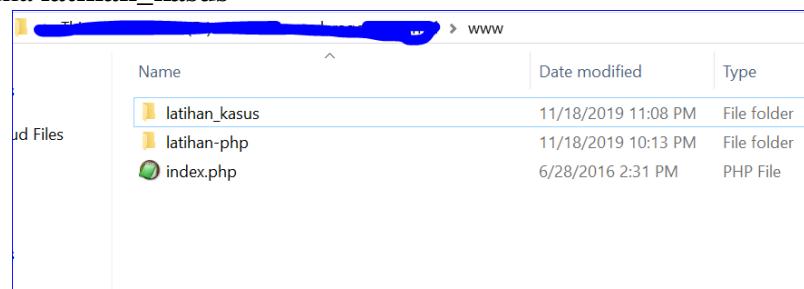
isbn	judul	deskripsi
200-02-1998-1-0012	Matematika Diskrit II	Buku standar tentang matematika diskrit tingkat lanjut
200-03-2000-6-0040	Pemrograman Web Python dan PHP	Buku belajar mandiri pemrograman web oleh Rahman

### 2. Buat kode program CRUD: Membuat halaman tambah data CREATE

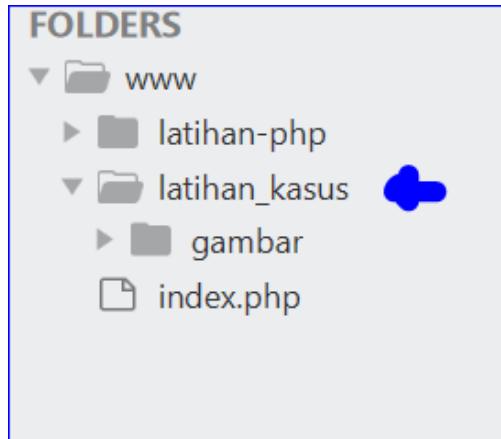
CRUD adalah singkatan dari Create Read Update Delete yang merupakan operasi menambah atau insert data, membaca satu atau semua data, mengubah atau memperbarui data dan menghapus satu atau beberapa data.

Kita akan mengimplementasikan operasi CRUD pertama kali adalah menambah data atau fungsi **CREATE** Berikut ini langkah-langkahnya:

- a. Pertama-tama pada **root** directory atau folder **www** Laragon, buat sebuah folder baru dengan nama **latihan\_kasus**



- b. Selanjutnya dalam directory **latihan\_kasus** buatlah folder atau directory baru dengan nama **gambar**



- c. Kita akan menggunakan folder **latihan\_kasus** untuk file-file berikutnya yang akan kita buat  
d. Masuk kedalam folder **latihan\_kasus**  
e. Buat file baru untuk konfigurasi koneksi ke database dengan nama **config\_db.php**

```

<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "db_latihan_kasus";

?>

```

- f. Untuk menambah data baru kita membutuhkan 2 halaman yaitu halaman input data yang berisi **form input** dan halaman **aksi form**.  
g. Pertama-tama buat form input dan beri nama **create\_buku.php**  
h. Ketik kode berikut:

```

1 <html>
2   <head>
3     <title>Create - Form Input</title>
4   </head>
5   <body>
6
7     <h2>HTML Forms</h2>
8
9     <form action="aksi_create_buku.php" method="post" enctype="multipart/form-data">
10    ISBN:<br>
11    <input type="text" name="isbn" >
12    <br>
13    Cover:<br>
14    <input type="file" name="gambar" >
15    <br>
16    Judul:<br>
17    <input type="text" name="judul" >
18    <br>
19    Deskripsi:<br>
20    <textarea rows="4" cols="50" name="deskripsi"></textarea>
21    <br><br>
22    <input type="submit" value="Simpan">
23  </form>
24 </body>
25 </html>

```

- i. Kode diatas akan menampilkan form input data.
- j. Coba jalankan form tersebut dengan mengakses url: [http://localhost/latihan\\_kasus/create\\_buku.php](http://localhost/latihan_kasus/create_buku.php) maka tampilan seperti berikut:

The screenshot shows a web page with the title "HTML Forms". It contains four input fields: "ISBN:" (text input), "Cover:" (file input with placeholder "Choose File No file chosen"), "Judul:" (text input), and "Deskripsi:" (text area). Below the form is a "Simpan" button.

- k. Selanjutnya dalam folder **latihan\_kasus** buat halaman aksi create buku yang akan dipanggil ketika button form **Simpan** diklik. Beri nama file **aksi\_create\_buku.php** file ini merupakan file action dari form create diatas.
- l. Ketik kode program berikut: (kode bersambungan dihalaman lain. Perhatikan nomor baris kode saat anda mengetik kode)

```

1 <?php
2 //error_reporting(0);
3
4 echo "<h2>Halaman Aksi CREATE Buku</h2>";
5 // deklarasikan variabel data
6 $isbn; $judul; $deskripsi;
7
8 // periksa input ISBN
9 if(isset($_POST["isbn"])){
10   $isbn = $_POST["isbn"];
11 }else{
12   die("Gagal proses ISBN kosong!<br />");
13 }
14 // periksa input JUDUL
15 if(isset($_POST["judul"])){
16   $judul = $_POST["judul"];
17 }else{
18   die("Gagal proses JUDUL kosong!<br />");
19 }
20 // periksa input DESKRIPSI
21 if(isset($_POST["deskripsi"])){
22   $deskripsi = $_POST["deskripsi"];
23 }else{
24   die("Gagal proses DESKRIPSI kosong!<br />");
25 }
26
27 // periksa input ISBN tidak boleh kosong
28 if($isbn == ""){
29   die("ISBN tidak boleh kosong!<br />");
30 }
31
32 // periksa input ISBN tidak boleh kosong
33 if($judul == ""){
34   die("JUDUL tidak boleh kosong!<br />");
35 }

```

```

36 // periksa input ISBN tidak boleh kosong
37 if($deskripsi == ""){
38     die("DESKRIPSI tidak boleh kosong!<br />");
39 }
40 /*
41     Selanjutnya proses menyimpan dimulai
42     proses simpan dilakukan dgn menggunakan
43     fungsi insert_buku yang mengembalikan true jika berhasil
44 */
45
46 if(insert_buku($isbn, $judul, $deskripsi, $pesan_error)){
47     // jika berhasil simpan kita juga akan menyimpan gambarnya
48     // proses simpan gambar juga dilakukan dalam fungsi
49     echo "Data berhasil ditambahkan!<br />";
50     if(simpan_gambar($info, $link)){
51         echo "Gambar berhasil disimpan: $link";
52         echo "<br /> <br /> <br />";
53         echo "Klik untuk kembali ke halaman ";
54         echo "<a href='read_buku.php'>read_buku.php</a>";
55     }else{
56         echo "Gagal simpan gambar: $info";
57     }
58 }else{
59     // jika gagal simpan
60     die($pesan_error . "<br />");
61 }
62
63
64
65 // fungsi untuk menyimpan gambar
66 function simpan_gambar(&$info_error, &$link_gambar){
67
68     // periksa jika file ada
69     if($_FILES['gambar'][['name']] == "") {
70         $info_error = "Tidak ada file yang diupload";
71         return false;
72     }
73
74     // Asumsi file gagal diupload
75     if ($_FILES['gambar'][['error']] != UPLOAD_ERR_OK) {
76         $info_error = "Upload gagal dengan error code: " . $_FILES['gambar'][['error']];
77         return false;
78     }
79
80
81     $info = getimagesize($_FILES['gambar'][['tmp_name']]);
82     if ($info === FALSE) {
83         $info_error = "Tidak bisa membaca jenis file";
84         return false;
85     }
86
87     if (($info[2] != IMAGETYPE_GIF) && ($info[2] != IMAGETYPE_JPEG)
88         && ($info[2] != IMAGETYPE_PNG)) {
89         $info_error = "Bukan file gambar gif/jpeg/png";
90         return false;
91     }
92
93     // ambil data file
94     $namaFile = $_POST['isbn'];
95     $namaSementara = $_FILES['gambar'][['tmp_name']];
96
97     // Check file size
98     if ($_FILES["gambar"]["size"] > 500000) {
99         $info_error = "File terlalu besar";
100        return false;
101    }
102
103    // tentukan folder penyimpanan
104    $dirUpload = "gambar/";
105
106    // pindahkan file ke folder gambar
107    $terupload = move_uploaded_file($namaSementara, $dirUpload.$namaFile);
108
109    if ($terupload) {
110        $link_gambar = "Link: <a href='".$dirUpload.$namaFile."'>".$namaFile."</a>";
111        return true;
112    } else {
113        $info_error = "Gagal move_uploaded_file";
114        return false;
115    }
116}
117

```

```

118     // fungsi untuk menambah data buku
119     function insert_buku($isbn, $judul, $deskripsi, &$info_error){
120         include("config_db.php");
121         $query = 'INSERT INTO tbuku (isbn, judul, deskripsi)
122             VALUES (:isbn, :judul, :deskripsi)';
123
124         try {
125             $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
126             // set the PDO error mode to exception
127             $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
128
129             $stmt = $conn->prepare($query);
130             $stmt->bindParam(':isbn', $isbn);
131             $stmt->bindParam(':judul', $judul);
132             $stmt->bindParam(':deskripsi', $deskripsi);
133             $stmt->execute();
134             return true;
135         }
136         catch(PDOException $e)
137         {
138             // jika terjadi error maka kita set error info
139             $info_error = "Gagal tambah data: " . $e->getMessage();
140             return false;
141         }
142     }
143
144
145 ?>
```

m. Selanjutnya jalankan halaman create dan silahkan tambahkan data.

### 3. Buat kode program CRUD: Membuat halaman tampilan data READ

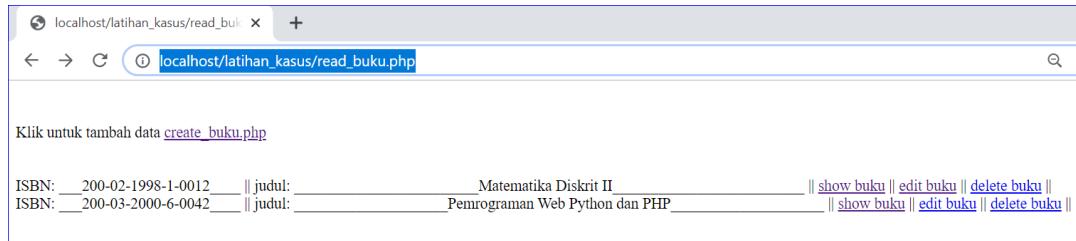
Halaman ini adalah halaman web yang akan menampilkan semua data yang ada didalam tabel **tbuku** dengan disertai link **show**, **edit** dan **delete**. Berikut langkah-langkah untuk membuat halaman tampilan data:

- Masuk kedalam folder **latihan\_kasus**
- Buat file baru dengan nama **read\_buku.php**
- Ketikan kode program berikut:

```

1  <?php
2      include("config_db.php");
3      $query = 'SELECT * FROM tbuku';
4      try {
5          echo "<br /> <br />";
6          echo "Klik untuk tambah data ";
7          echo "<a href='create_buku.php'>create_buku.php</a>";
8          echo "<br /> <br /> <br />";
9          $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
10         // set the PDO error mode to exception
11         $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
12
13         $data = $conn->query($query);
14         while ($row = $data->fetch()){
15             echo str_pad("ISBN:", 8, ' ', STR_PAD_RIGHT);
16             echo str_pad($row['isbn'], 25, '_', STR_PAD_BOTH);
17             echo " || ";
18
19             echo str_pad("judul:", 8, ' ', STR_PAD_RIGHT);
20             echo str_pad($row['judul'], 70, '_', STR_PAD_BOTH);
21             echo " || ";
22
23             echo "<a href='show_buku.php?isbn=" . $row["isbn"] . "'>show buku</a>" . " || ";
24             echo "<a href='update_buku.php?isbn=" . $row["isbn"] . "'>edit buku</a>" . " || ";
25             $confirm = "onclick=\"return confirm('Apakah anda yakin hapus?');\"";
26             echo "<a href='delete_buku.php?isbn=" . $row["isbn"] . "' $confirm >delete buku</a>" . " || ";
27             echo "<br />";
28         }
29     }
30     catch(PDOException $e)
31     {
32         echo "Gagal terhubung: " . $e->getMessage();
33     }
34 ?>
```

- Selanjutnya akses melalui url: [http://localhost/latihan\\_kasus/read\\_buku.php](http://localhost/latihan_kasus/read_buku.php)
- Halaman tampilan data yang dibaca dari tabel tbuku akan ditampilkan:



- f. Pada sudut kiri atas terdapat link untuk mengakses halaman **create\_buku.php**. sedangkan pada setiap baris data terdapat link untuk show data, edit dan hapus. Tapi jika anda mengklik link tersebut maka browser akan menampilkan error Not Found karena kita belum membuat halaman-halaman tersebut.
  - g. Selanjutnya kita akan membuat halaman **show\_buku.php**
4. Buat kode program CRUD: **Membuat halaman show data SHOW**
- Halaman show data adalah halaman kategori READ pada istilah CRUD. Pada bagian ini data yang dibaca adalah data tunggal yang diidentifikasi dengan menggunakan primary key tabel, dalam hal ini adalah kolom **isbn**. Kolom **isbn** akan selalu dikirim oleh suatu halaman jika memanggil halaman lain.
- Berikut langkah-langkah membuat halaman SHOW:
- Masuk kedalam folder **latihan\_kasus**
  - Buat file baru dengan nama **show\_buku.php**
  - Ketikan kode program berikut:

```

1  <?php
2  // baca parameter isbn yg dikirim di url
3  $isbn = "";
4  if(isset($_GET["isbn"])){
5      $isbn = $_GET["isbn"];
6  }
7  // Jika isbn kosong maka keluar
8  if($isbn==""){
9      die("Kode ISBN kosong");
10 }
11 include("config_db.php");
12 $query = 'SELECT * FROM tbuku WHERE isbn=:isbn LIMIT 1';
13 try {
14     $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
15     // set the PDO error mode to exception
16     $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
17     $stmt = $conn->prepare($query);
18     $stmt->bindParam(':isbn', $isbn);
19     $stmt->execute();
20     $row = $stmt->fetch();
21     echo "Cover: <br />";
22     echo "<img src='gambar/" . $row['isbn'] . "' alt='". $row['isbn'] . "' width='150' height='150'" . "<br />";
23     echo "ISBN: " . $row['isbn'] . "<br />";
24     echo "Judul: " . $row['judul'] . "<br />";
25     echo "Deskripsi: " . $row['deskripsi'] . "<br />";
26     echo "<br /> <br /> <br />";
27     echo "Klik untuk kembali ke halaman ";
28     echo "<a href='read_buku.php'>read_buku.php</a>";
29 }
30 catch(PDOException $e)
31 {
32     echo "Gagal baca data: " . $e->getMessage();
33 }
34 ?>

```

- d. Selanjutnya silahkan mengklik link **show buku** pada salah satu data di halaman tampilan data **read\_buku.php**
- e. Data akan ditampilkan pada halaman show seperti berikut ini:

Cover:

ISBN: 200-02-1998-1-0012  
 Judul: Matematika Diskrit II  
 Deskripsi: Buku standar tentang matematika diskrit tingkat lanjut

Klik untuk kembali ke halaman [read\\_buku.php](#)

- f. Halaman show menampilkan data buku dan cover buku.
5. Buat kode program CRUD: **Membuat halaman edit data EDIT/UPDATE**  
 Pada dasarnya halaman Edit/Update hampir sama dengan halaman Create/Insert. Kita membutuhkan 2 halaman untuk implementasi edit/update. Halaman pertama adalah halaman form input update dan halaman kedua adalah halaman aksi form update.  
 Hal penting lain yang perlu diperhatikan adalah kita harus menyisipkan primary key yaitu kode isbn pada halaman form update. Primary key kolom isbn akan menjadi identitas data pada tabel tbuku yang dapat dieksekusi melalui query kondisional dengan kode isbn.  
 Berikut langkah membuat halaman update/edit:
- a. Masuk kedalam folder **latihan\_kasus**
  - b. Buat file baru dengan nama **update\_buku.php**
  - c. Ketikan kode program berikut: (kode bersambung dihalaman berikut)

```

1  <?php
2
3      // baca parameter isbn yg dikirim di url
4      $isbn = "";
5      if(isset($_GET["isbn"])){
6          $isbn = $_GET["isbn"];
7      }
8
9      // Jika isbn kosong maka keluar
10     if($isbn==""){
11         die("Kode ISBN kosong");
12     }
13
14     include("config_db.php");
15
16     $query = 'SELECT * FROM tbuku WHERE isbn=:isbn LIMIT 1';
17     $data;
18     try {
19         $conn = new PDO("mysql:host=$servername;dbname=$databasename", $username, $password);
20         // set the PDO error mode to exception
21         $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
22
23         $stmt = $conn->prepare($query);
24         $stmt->bindParam(':isbn', $isbn);
25
26         $stmt->execute();
27         $data = $stmt->fetch();

```

```

28     }
29     catch(PDOException $e)
30     {
31         echo "Gagal baca data: " . $e->getMessage();
32     }
33 }
34
35 ?>
36 <?php ?>
37 <html>
38     <head>
39         <title>Form Edit</title>
40     </head>
41     <body>
42         <h2>HTML Forms</h2>
43         <form action="aksi_update_buku.php" method="post" enctype="multipart/form-data">
44             <input type="text" name="isbn_old" value=<?php echo $data['isbn']; ?>" hidden >
45             ISBN:<br>
46             <input type="text" name="isbn" value=<?php echo $data['isbn']; ?>" >
47             <br>
48             Cover:<br>
49             <input type="file" name="gambar" >
50             <br>
51             Judul:<br>
52             <input type="text" name="judul" value=<?php echo $data['judul']; ?>" >
53             <br>
54             Deskripsi:<br>
55             <textarea rows="4" cols="50" name="deskripsi"><?php echo $data['deskripsi']; ?></textarea>
56             <br><br>
57             <input type="submit" value="Update">
58         </form>
59     </body>
60 </html>

```

- d. Pada kode form kita menambahkan suatu input yang tersembunyi atau **hidden** yaitu tidak ditampilkan pada browser yaitu input dengan nama **isbn\_old**.
- e. Selanjutnya buat file baru untuk aksi form update, beri nama **aksi\_update\_buku.php**
- f. Ketikan kode berikut: (kode bersambung dihalaman lain)

```

1  <?php
2      //error_reporting(0);
3
4      echo "<h2>Halaman Aksi UPDATE Buku</h2>";
5      // deklarasikan variabel data
6      $isbn_old; $isbn; $judul; $deskripsi;
7
8      // periksa kode isbn lama sebagai acuan edit
9      if(isset($_POST["isbn_old"])){
10          $isbn_old = $_POST["isbn_old"];
11      }else{
12          die("Gagal proses referensi ISBN acuan!<br />");
13      }
14
15      // periksa input ISBN
16      if(isset($_POST["isbn"])){
17          $isbn = $_POST["isbn"];
18      }else{
19          die("Gagal proses ISBN kosong!<br />");
20      }
21      // periksa input JUDUL
22      if(isset($_POST["judul"])){
23          $judul = $_POST["judul"];
24      }else{
25          die("Gagal proses JUDUL kosong!<br />");
26      }
27      // periksa input DESKRIPSI
28      if(isset($_POST["deskripsi"])){
29          $deskripsi = $_POST["deskripsi"];
30      }else{
31          die("Gagal proses DESKRIPSI kosong!<br />");
32      }
33
34      // periksa input ISBN tidak boleh kosong
35      if($isbn == ""){
36          die("ISBN tidak boleh kosong!<br />");
37      }

```

```

39     // periksa input ISBN tidak boleh kosong
40     if($judul == ""){
41         die("JUDUL tidak boleh kosong!<br />");
42     }
43
44     // periksa input ISBN tidak boleh kosong
45     if($deskripsi == ""){
46         die("DESKRIPSI tidak boleh kosong!<br />");
47     }
48
49     /*
50      Selanjutnya proses menyimpan dimulai
51      proses simpan dilakukan dgn menggunakan
52      fungsi insert_buku yang mengembalikan true jika berhasil
53   */
54
55     if(update_buku($isbn_old, $isbn, $judul, $deskripsi, $pesan_error)){
56         // jika berhasil simpan kita juga akan menyimpan gambarnya
57         // proses simpan gambar juga dilakukan dalam fungsi
58         echo "Data berhasil diperbaharui!<br />";
59         if(simpan_gambar($info, $link)){
60             echo "Gambar berhasil disimpan: $link";
61             echo "<br /> <br />";
62             echo "Klik untuk kembali ke halaman ";
63             echo "<a href='read_buku.php'>read_buku.php</a>";
64         }else{
65             echo "Gagal simpan gambar: $info";
66         }
67     }else{
68         // jika gagal simpan
69         die($pesan_error . "<br />");
70     }
71
72     // fungsi untuk menyimpan gambar
73     function simpan_gambar(&$info_error, &$link_gambar){
74
75         // periksa jika file ada
76         if($_FILES['gambar']['name'] == "") {
77             $info_error = "Tidak ada file yang diupload";
78             return false;
79         }
80
81         // Asumsi file gagal diupload
82         if ($_FILES['gambar']['error'] != UPLOAD_ERR_OK) {
83             $info_error = "Upload gagal dengan error code: " . $_FILES['gambar']['error'];
84             return false;
85         }
86
87
88         $info = getimagesize($_FILES['gambar']['tmp_name']);
89         if ($info === FALSE) {
90             $info_error = "Tidak bisa membaca jenis file";
91             return false;
92         }
93
94         if (($info[2] != IMAGETYPE_GIF) && ($info[2] != IMAGETYPE_JPEG)
95             && ($info[2] != IMAGETYPE_PNG)) {
96             $info_error = "Bukan file gambar gif/jpeg/png";
97             return false;
98         }
99
100        // ambil data file
101        $namaFile = $_POST['isbn'];
102        $namaSementara = $_FILES['gambar']['tmp_name'];
103
104        // Check file size
105        if ($_FILES["gambar"]["size"] > 500000) {
106            $info_error = "File terlalu besar";
107            return false;
108        }

```

```

109     // tentukan folder penyimpanan
110     $dirUpload = "gambar/";
111
112     // pindahkan file ke folder gambar
113     $terupload = move_uploaded_file($namaSementara, $dirUpload.$namaFile);
114
115     if ($terupload) {
116         $link_gambar = "Link: <a href='".$dirUpload.$namaFile."'>".$namaFile."</a>";
117         return true;
118     } else {
119         $info_error = "Gagal move_uploaded_file";
120         return false;
121     }
122 }
123
124
125     // fungsi untuk menambah data buku
126 function update_buku($isbn_old, $isbn, $judul, $deskripsi, &$info_error){
127     include("config_db.php");
128     $query = 'UPDATE tbuku SET isbn = :isbn, judul = :judul, deskripsi = :deskripsi
129     WHERE isbn = :isbn_old;';
130
131     try {
132         $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
133         // set the PDO error mode to exception
134         $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
135
136         $stmt = $conn->prepare($query);
137         $stmt->bindParam(':isbn', $isbn);
138         $stmt->bindParam(':isbn_old', $isbn_old);
139         $stmt->bindParam(':judul', $judul);
140         $stmt->bindParam(':deskripsi', $deskripsi);
141         $stmt->execute();
142         return true;
143     }
144     catch(PDOException $e)
145     {
146         // jika terjadi error maka kita set error info
147         $info_error = "Gagal tambah data: " . $e->getMessage();
148         return false;
149     }
150 }
151
152
153 ?>
```

- g. Selanjutnya silahkan mengklik link **edit buku** pada salah satu data di halaman tampilan data **read\_buku.php**
- h. Data akan ditampilkan pada halaman edit seperti berikut ini:

Form Edit

localhost/latihan\_kasus/update\_buku.php?isbn=200-03-2000-6-0042

## HTML Forms

ISBN:  
200-03-2000-6-0042

Cover:  
 Choose File | No file chosen

Judul:  
Pemrograman Web Python

Deskripsi:  
Buku belajar mandiri pemrograman web oleh Rahman

- i. Lakukan beberapa perubahan, pilih gambar untuk Cover dan klik button **Update** maka akan tampil halaman info berhasil pembaharuan data

localhost/latihan\_kasus/aksi\_update\_buku.php

## Halaman Aksi UPDATE Buku

Data berhasil diperbaharui!  
Gambar berhasil disimpan: Link: [200-03-2000-6-0042](#)

Klik untuk kembali ke halaman [read\\_buku.php](#)

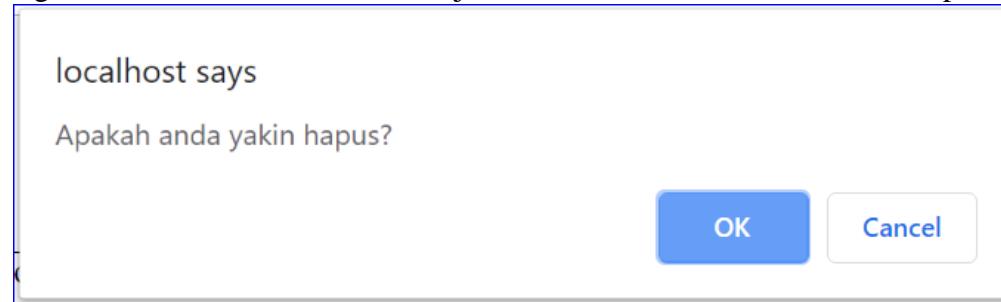
- j. Data edit dan file gambar cover telah berhasil disimpan
- k. Selanjutnya adalah membuat halaman aksi menghapus data
6. Buat kode program CRUD: **Membuat halaman hapus data DELETE**  
Halaman hapus akan mengeksekusi query penghapusan data. Karena itu dibutuhkan skenario pesan dialog konfirmasi penghapusan data. Ini mekanisme proteksi sehingga penghapusan data tidak terjadi secara tidak sengaja.  
Penghapusan data juga membutuhkan kunci primary key yaitu kode isbn baris data sehingga data yang terhapus adalah data yang sesuai dengan kode isbn yang diinginkan. Kode ini dikirim melalui url metode **get**. Berikut langkah-langkah halaman hapus data:
- Masuk kedalam folder **latihan\_kasus**
  - Buat file baru dengan nama **delete\_buku.php**
  - Ketikan kode program berikut:

```

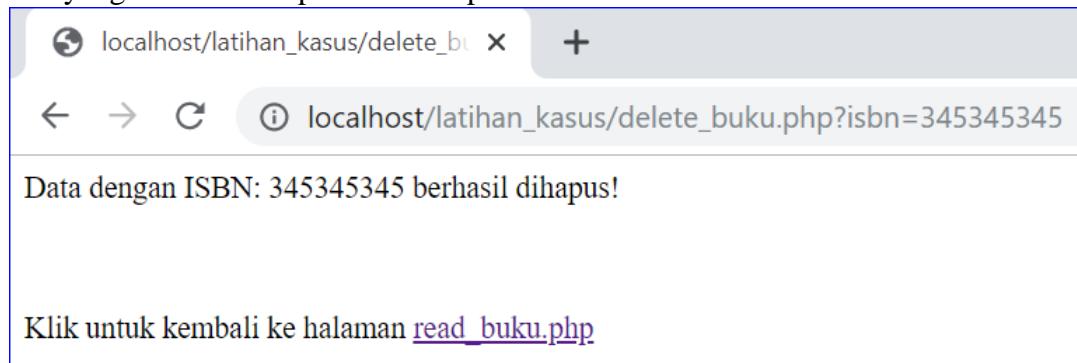
1  <?php
2
3      // baca parameter isbn yg dikirim di url
4      $isbn = "";
5      if(isset($_GET["isbn"])){
6          $isbn = $_GET["isbn"];
7      }
8
9      // Jika isbn kosong maka keluar
10     if($isbn==""){
11         die("Kode ISBN kosong");
12     }
13
14     include("config_db.php");
15
16     $query = 'DELETE FROM tbuku WHERE isbn=:isbn LIMIT 1';
17
18     try {
19         $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
20         // set the PDO error mode to exception
21         $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
22
23         $stmt = $conn->prepare($query);
24         $stmt->bindParam(':isbn', $isbn);
25         $stmt->execute();
26         echo "Data dengan ISBN: ".$isbn." berhasil dihapus! <br />";
27
28         echo "<br /> <br /> <br />";
29         echo "Klik untuk kembali ke halaman ";
30         echo "<a href='read_buku.php'>read_buku.php</a>";
31
32     } catch(PDOException $e) {
33     {
34         echo "Gagal baca data: " . $e->getMessage();
35     }
36
37
38 ?>

```

- d. Selanjutnya silahkan mengklik link **delete buku** pada salah satu data di halaman tampilan data **read\_buku.php**
- e. Dialog konfirmasi akan dimunculkan, jika memilih OK maka data akan dihapus



- f. Data yang berhasil dihapus akan tampil halaman berikut:



- g. Selanjutnya kita akan membuat halaman indek sebagai halaman awal aplikasi web

7. Buat halaman utama: **Membuat halaman index/homepage**

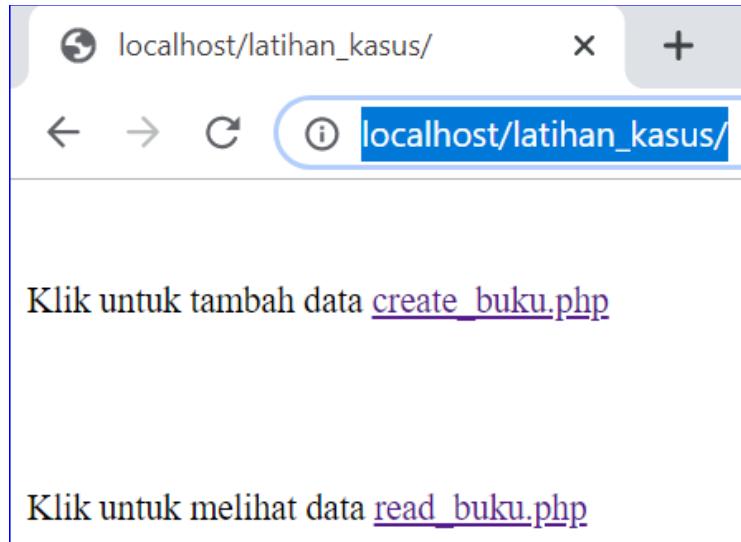
Berikut langkah-langkahnya:

- a. Masuk kedalam folder **latihan\_kasus**
- b. Buat file baru dengan nama **index.php**
- c. Ketikan kode program berikut:

```
1  <?php  
2  
3      echo "<br /> <br />";  
4      echo "Klik untuk tambah data ";  
5      echo "<a href='create_buku.php'>create_buku.php</a>";  
6      echo "<br /> <br /> <br />";  
7  
8      echo "<br /> <br />";  
9      echo "Klik untuk melihat data ";  
10     echo "<a href='read_buku.php'>read_buku.php</a>";  
11     echo "<br /> <br /> <br />";  
12  
13  
14 ?>
```

- d. Jalankan dengan mengakses url: [http://localhost/latihan\\_kasus/](http://localhost/latihan_kasus/)

- e. Tampilan seperti berikut:



- f. Halaman ini menyajikan link menuju tampilan data dan create data.