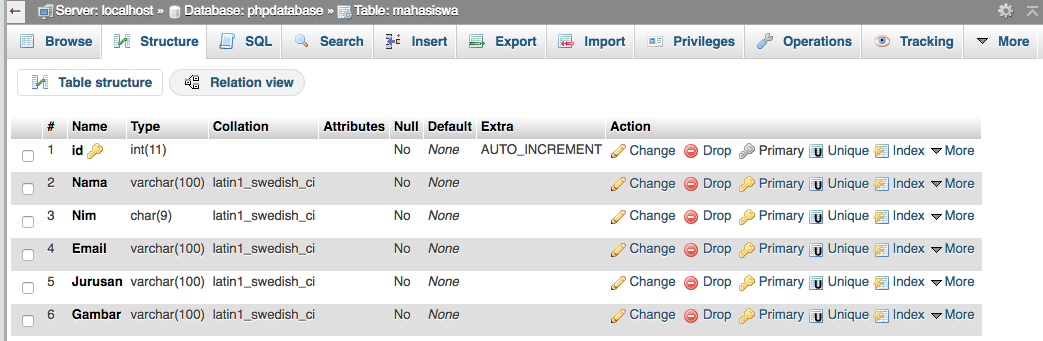
**Bab 10**

**PHP dan Database**

**(Membuat Koneksi dan Menampilkan data)**

Pada bab ini menjelaskan cara untuk menghubungkan antara PHP dan Database sehingga isi dari database dapat ditampilkan pada halaman browser. Untuk menghubungkan antara php dan mysql dapat menggunakan beberapa cara yaitu menggunakan extensi mysql, menggunakan extensi mysqli atau menggunakan PDO (PHP Data Object). untuk kali ini kita menggunakan extensi Mysqli ( Mysql Improve) untuk menghubungkan antara php dan mysql. Pertama kali kita membuat database dengan nama phpdatabase dengan 5 field seperti pada gambar 10.1



gambar 10.1 Desain database

setelah kita membuat database maka selanjutnya kita membuat halaman index.php untuk menampilkan isi dari database seperti pada gambar 10.3. Sebelum kita memulai code tambahkan data pada tabel mahasiswa seperti pada gambar 10.2. Untuk codenya seperti pada gambar 10.4 sebagai uji coba data yang akan ditampilkan merupakan data static ( belum mengambil dari database ).



Gambar 10.2 tambahkan data pada table mahasiswa

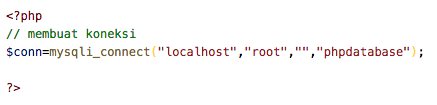


Gambar 10.3 hasil tampilan

<h1>

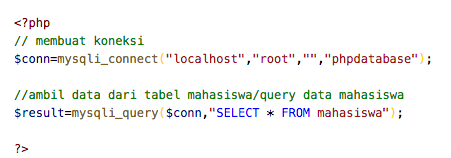
Gambar 10.4 code menggunakan data static

pada line 11 kita set bordernya 1 artinya tebal border 1 pixel, cellpadding digunakan untuk mengatur jarak dari border sisi dalam tabel dengan isi text tabel itu sendiri dan cellspacing digunakan untuk mengatur jarak antara garis tepi (border) bagian dalam dan luar. jika kita hasilnya ukuran dari picture terlalu besar maka dapat kita gunakan height dan width agar gambar tidak terlalu besar. Setelah kita membuat code untuk membuat tampilan langkah selanjutnya kita membuat koneksi ke database dan melakukan seleksi data menggunakan query seperti pada gambar 10.5



Gambar 10.5 koneksi ke database

mysqli\_connect adalah sebuah function untuk menghubungkan ke database. Function ini terdapat beberapa paramater yaitu **(“Nama Host”, “Username”, ”Password”, “Nama Database”)**. Sehingga untuk kasus kita parameternya adalah **(“localhost”, “root”,””,”phpdatabase”).** Setelah function dan paramaternya sudah, maka langkah selanjutnya adalah membuat variable baru untuk menampung isi dari function tersebut. Sehingga codenya menjadi seperti pada gambar 10.5. Setelah membuat koneksi ke database maka langkah selanjutnya mengambil data dari database menggunakan function **mysqli\_query** seperti pada gambar 10.6. ( koneksi , querynya mau apa)

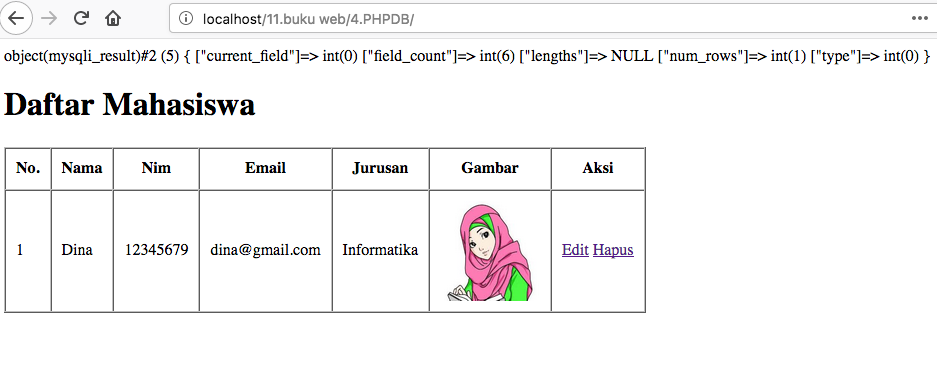


Gambar 10.6 ambil data dari tabel mahasiswa

pada function mysqli\_query terdapat 2 paramater yaitu **( koneksi, querynya )**, sehingga pada kasus ini **koneksinya** adalah **$conn** dan **querynya** adalah “**SELECT \* FROM mahasiswa**”. Langkah berikutnya adalah mengambil data dari object **$result**. Data yang ditampung pada variable **$result** berupa object hal ini dapat anda lihat dengan memberikan perintah **var\_dump** pada code gambar 10.7 dan panggil halaman index.php seperti pada gambar 10.8



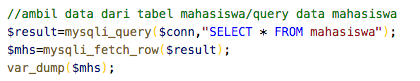
Gambar 10.7 var\_dum($result)



Gambar 10.8 Hasil dari var\_dump

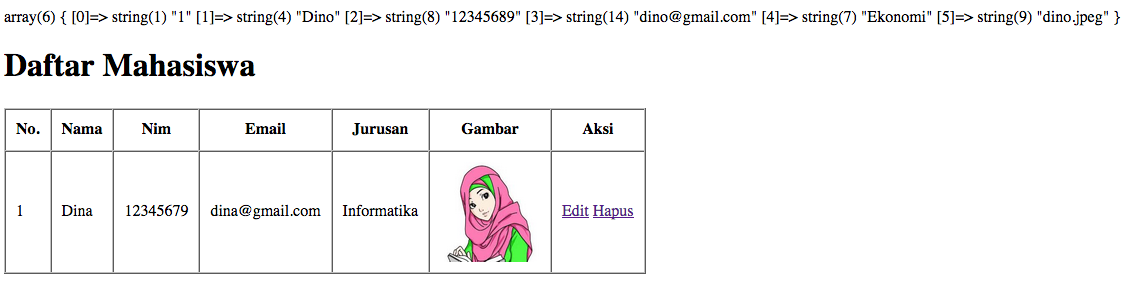
Jika kita lihat pada browser data masih berupa object sehingga kita perlu menambahkan fetch untuk menampilkan atau mengambil data mahasiswa. Ada 4 cara untuk mengambil data ( fetch ) yaitu

1. mysqli\_fetch\_row()
   1. function ini mengembalikan data array numerik ( nilai yang indexnya angka)
   2. sebagai contoh lakukan langkah-langkah berikut
2. tambahkan code berikut pada index.php



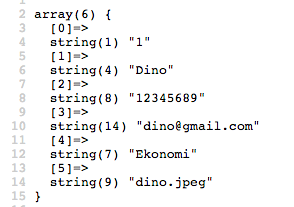
Gambar 10.9 mysqli\_fetch\_row

1. buka halaman index.php dan klik kanan brower pilih view page source



Gambar 10.10 view browser

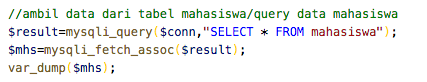
1. hasil dari view page source



Gambar 10.11 view page source

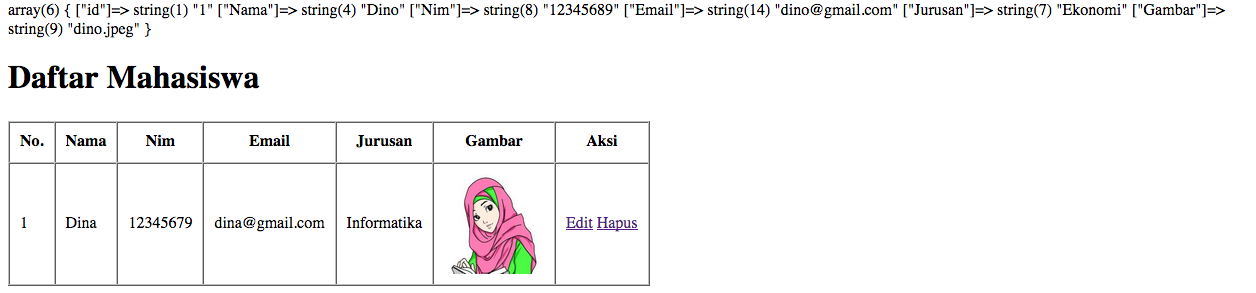
jika lihat pada view page source data dikembalikan dalam bentuk array dimana indexnya berupa angka sehingga disebut array numerik

1. mysqli\_fetch\_assoc()
   1. function ini mengembalikan data array associative
   2. sebagai contoh lakukan langkah-langkah berikut
2. tambahkan code berikut pada index.php ( **cat : code function mysqli\_fetch\_row dihapus** )



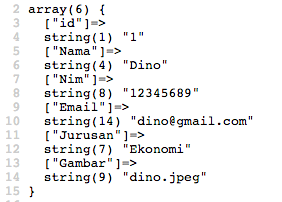
Gambar 10.12 mysqli\_fetch\_assoc

1. buka halaman index.php dan klik kanan brower pilih view page source



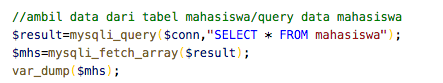
Gambar 10.13 view browser

1. hasil dari view page source



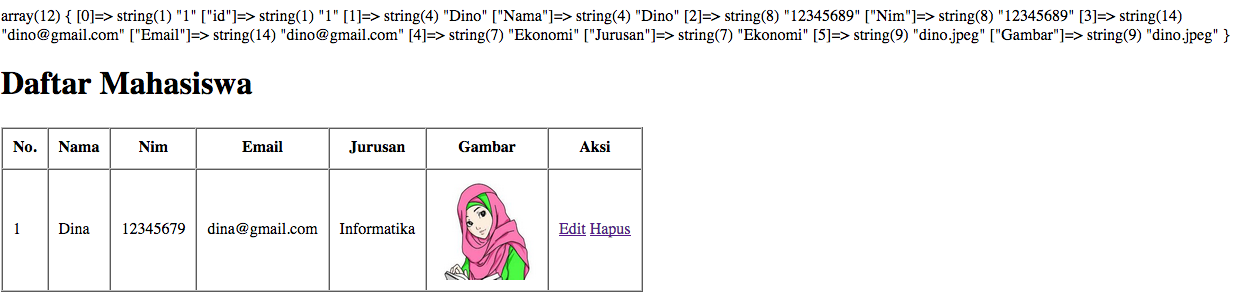
Gambar 10.14 View page source

1. mysqli\_fetch\_array()
   1. function ini mengambilkan data array numerik dan data associative
   2. sebagai contoh lakukan langkah-langkah berikut
2. Tambahkan code berikut pada index.php( **cat : code function mysqli\_fetch\_assoc dihapus**)



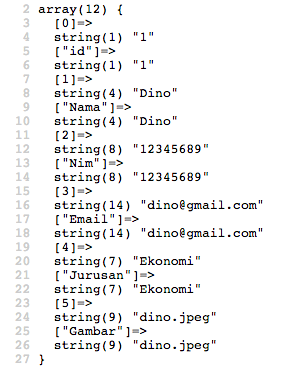
Gambar 10.15 mysqli\_fetch\_array

1. buka halaman index.php dan klik kanan brower pilih view page source



Gambar 10.16 view page browser

1. hasil dari view page source

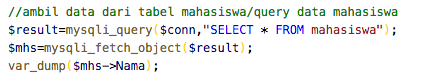


Gambar 10.17 view page source

Jika kita menggunakan function ini kita dapat melihat didalam arraynya terdapat index numeriknya dan index associative, sehingga jika kita menggunakan function ini maka secara tidak langsung ada 2 data array yang akan diload oleh browser.

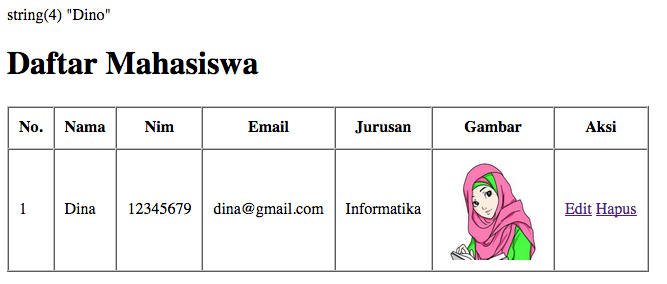
1. mysqli\_fetch\_object()
   1. function ini pengembalian datanya berupa object
   2. sebagai contoh lakukan langkah-langkah berikut
2. Tambahkan code berikut pada index.php( **cat : code function mysqli\_fetch\_array**

**dihapus**)



Gambar 10.18 mysqli\_fetch\_object

1. buka halaman index.php dan klik kanan brower pilih view page source



Gambar 10.19 view browser

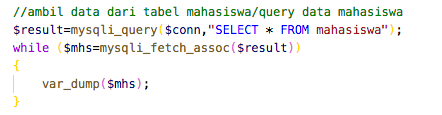
1. hasil view page source

../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05-

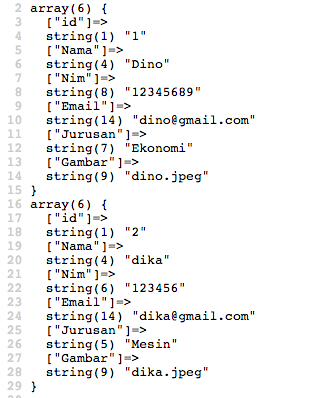
Gambar 10.20 view page source

Jika kita lihat data yang dikembalikan bukan berupa array tetapi berupa object.

Setelah kita mengetahui cara-cara mengambil data dengan beberapa function fetch maka langkah selanjutnya adalah kita menentukan fetch mana yang akan digunakan. Untuk kasus ini kita akan menggunakan cara associative (**mysqli\_fetch\_assoc** ). Sebelum kita lebih lanjut kita pelajari dulu konsep bagaimana mengambil data dan menampilkan data ke halaman web jika data yang akan ditampilkan itu lebih dari dari satu. Jika kita melakukan seperti pada gambar 10.12 diatas maka data yang akan ditampilkan hanya satu saja maka dari itu kita membutuhkan **perintah while** untuk melakukan proses lopping. ( Gambar 10.21 )



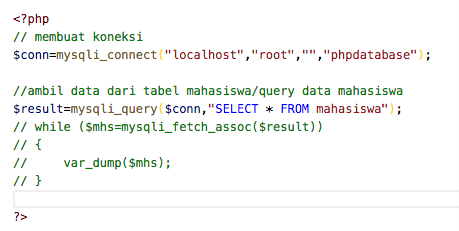
Gambar 10.21 perintah looping menggunakan while



Gambar 10.22 hasil dari view page source

dari gambar 10.22 kita dapat melihat hasil dari view page source berbeda dengan gambar 10.14 diatas hal ini disebabkan karena ada perintah looping didalamnya. Setelah kita memahami hal diatas maka langkah selanjutnya adalah bagaimana data array ( gambar 10.22 ) tersebut masuk kedalam tabel yang telah kita buat. Langkah-langkahnya sebagai berikut :

1. Comment while yang telah kita buat tadi sehingga tersisa koneksi dan ambil data



Gambar 10.23 comment while

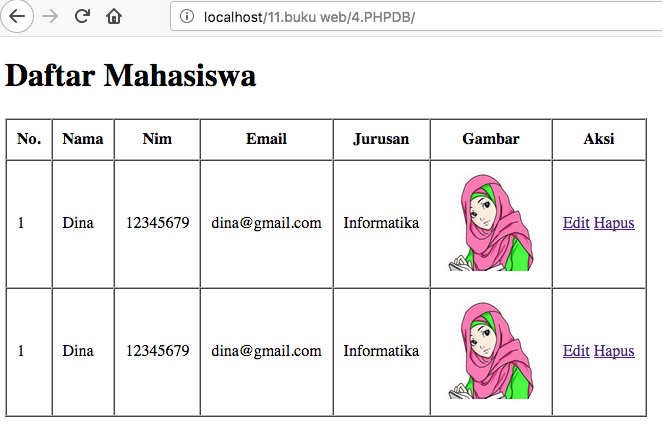
1. tambahkan sintak php dan looping pada tabel



Gambar 10.24 looping data

tambahkan code seperti pada gambar 10.24 line 38 dan 51. Line 38 adalah proses while dengan menggunakan function mysqli\_fetch\_assoc sedangkan pada line 51 adalah akhir proses dari while.

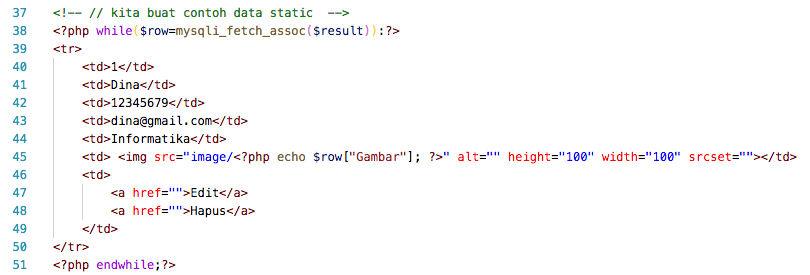
1. hasil sementara



Gambar 10.25 hasil sementara

Jika kita jalankan maka pada browser akan menampilkan data sejumlah data yang ada di database tetapi isinya belum berubah.

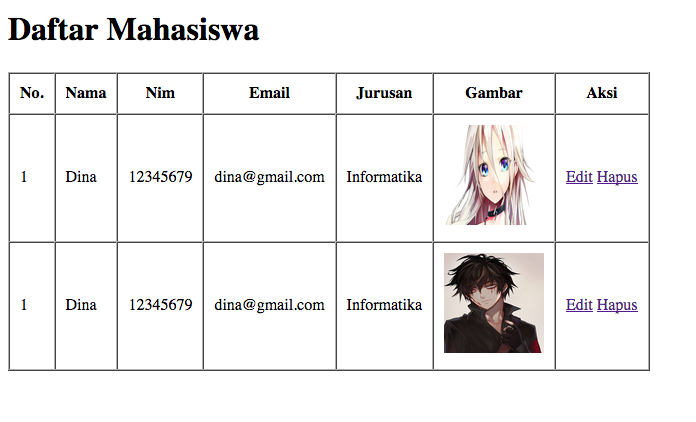
1. tambahkan code pada image agar image berubah sesuai dengan isi dari folder image



Gambar 10.26 code image

pada line 45 gambar 10.26 terdapat sintak **php ( <?php echo $row[“Gambar”] ?>)** ini digunakan menampilkan gambar ( **CAT : $row[“Gambar”] tulisan gambar besar kecilnya harus sama dengan nama FIELDNYA** )

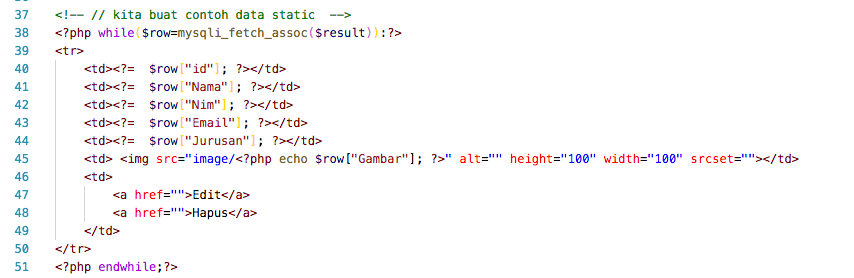
1. hasil sementara image



Gambar 10.27 Hasil sementara

Jika kita lihat gambar yang ditampilkan sudah sesuai dengan isi dari folder image.

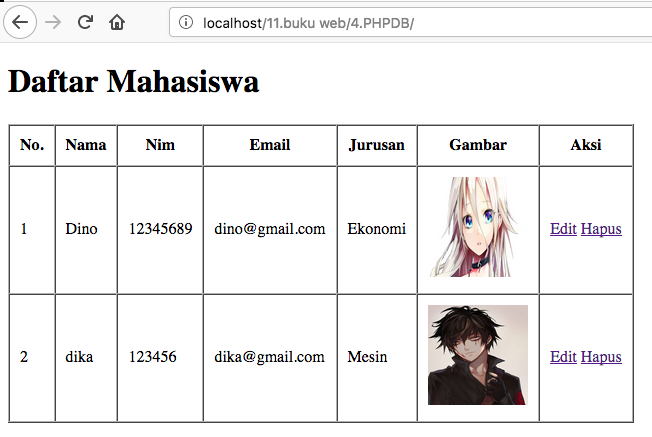
1. setelah kita berhasil menampilkan gambar maka selanjutnya kita tampilkan data yang lain



Gambar 10.28 code all

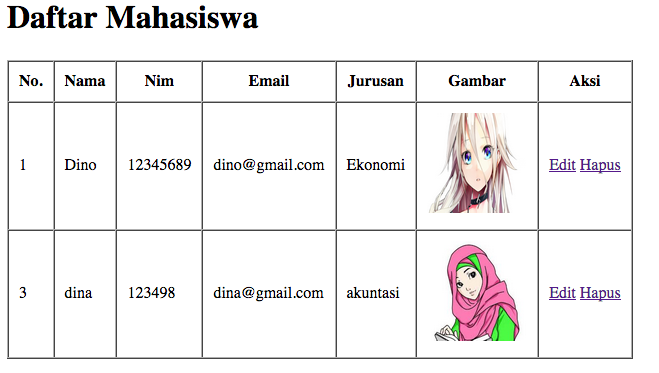
jika kita lihat pada line 40 sampai 44 terdapat code **php <?= …?>** ini sama dengan code **<?php echo …. ?>**

1. hasil sementara data yang ditampilkan



Gambar 10.29 Hasil sementara data yang ditampilkan

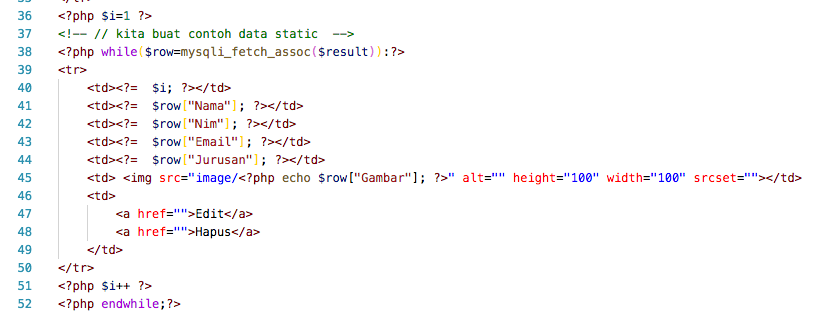
Jika kita lihat pada gambar 10.29 untuk kolom no terlihat urut tapi sebenarnya itu tidak ( tambahkan terlebih dahulu data pada database semisal 1 data jadi tota data ada 3 data). Hal ini bisa dibuktikan jika kita tambahkan satu data dan kita hapus satu data maka hasilnya akan seperti pada gambar 10.30



Gambar 10.30 penomoran yang tidak urut

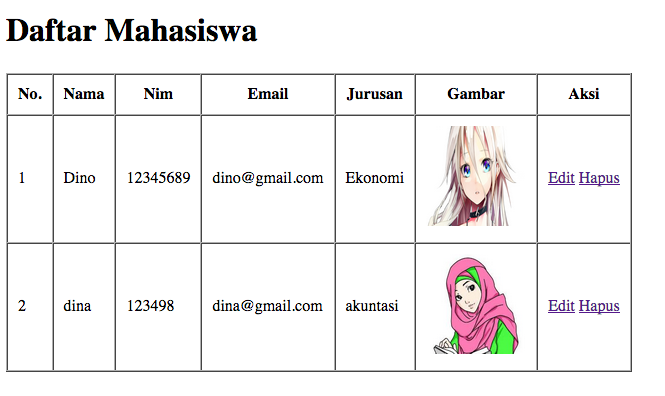
hal ini disebabkan karena data no yang ditampilkan adalah data yang ada field id pada tabel mahasiswa sehingga isi dari dari field id akan ditampilkan ke halaman index.php.

1. Untuk mengatasi permasalahan nomor maka diperlukan tambahan code seperti pada gambar 10.31



Gambar 10.31 code penomoran

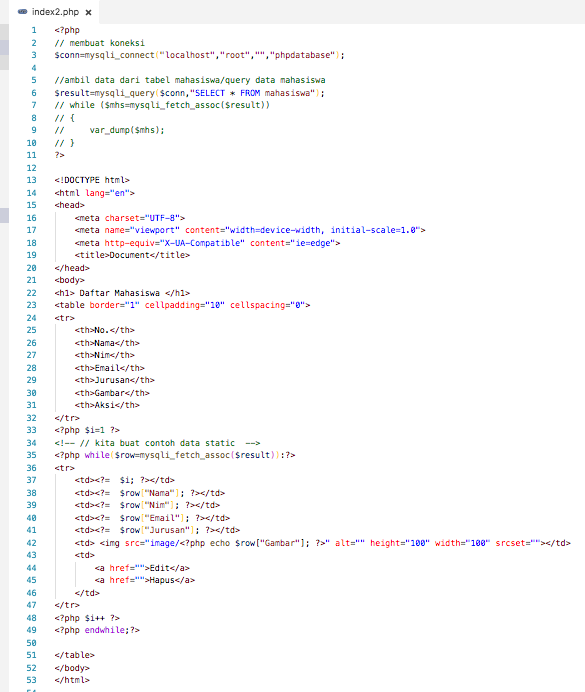
Pada line 36, 40 dan 51 terdapat code yang ditambah dan dirubah. Pada line 36 dimaksudkan kita membuat variable baru dengan nama $i dimana diberi data 1. Pada line 40 dimaksudkan untuk mencetak nilai $i dan yang terakhir pada line 51 merupakan proses penjumlahan $i sehingga hasilnya dapat kita lihat pada gambar 10.32



Gambar 10.32 Hasil index.php

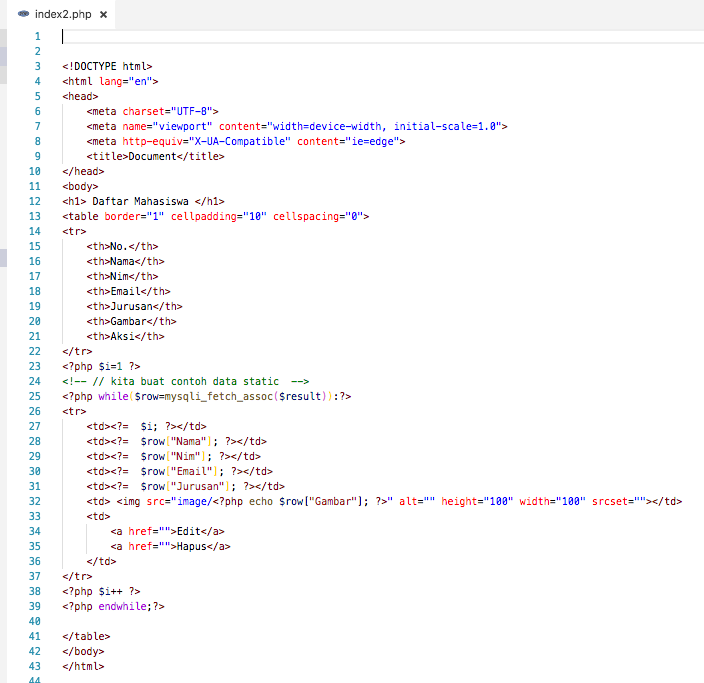
Selain menggunakan cara diatas untuk menampilkan data, terdapat juga cara yang lebih effesien daripada diatas. Untuk cara kedua ini kita membuat function sendiri untuk koneksi dan untuk melakukan querynya. Untuk lebih jelasnya anda ikuti langkah-langkah berikut :

1. Buat file baru dengan nama index2.php dan copy paste isi dari index.php yang telah kita buat tadi seperti pada gambar 10.33



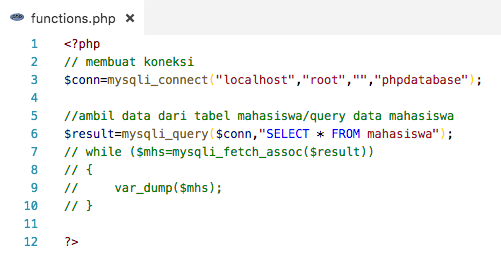
Gambar 10.33 index2.php

1. pada index2.php code php ( untuk koneksi dan query select all ) dihapus seperti pada gambar 10.34



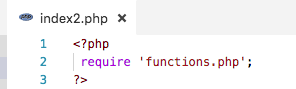
Gambar 10.34 index2.php

1. buat file baru dengan nama functions.php dan isikan code seperti pada gambar 10.35



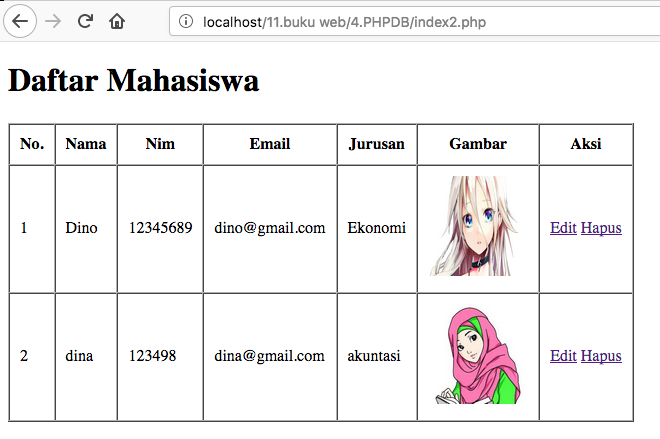
Gambar 10.35 functions.php

1. pada index2.php tambahkan code required seperti pada gambar 10.36. fungsi dari function require adalah untuk memanggil file php yang lain.



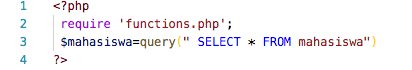
Gambar 10.36 require

1. coba buka halaman index2.php melalui browser jika benar maka tampilannya akan sama dengan index.php seperti pada gambar 10.37



Gambar 10.37 Hasil dari index2. php

1. pada **index2.php** tambahkan variable baru dengan nama mahasiswa, dimana variable baru ini digunakan untuk menampung string dengan beberapa parameter ( Gambar 10.38)



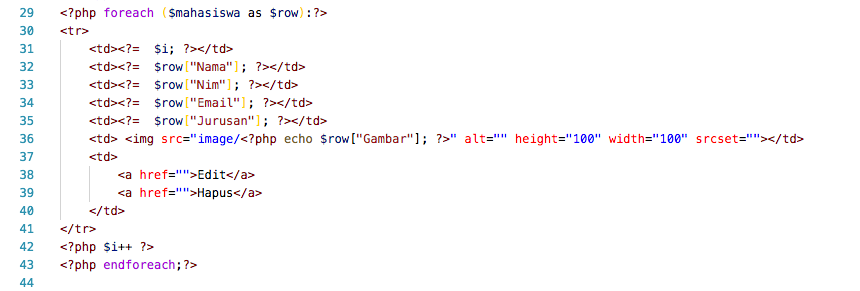
Gambar 10.38 variabel baru $mahasiswa

1. selanjutnya pada **functions.php** kita membuat function baru dengan nama query ( nama harus sesuai dengan string yang ditampung di **$mahasiswa** yang ada pada index2.php ) yang berisi proses looping (Gambar 10.39)



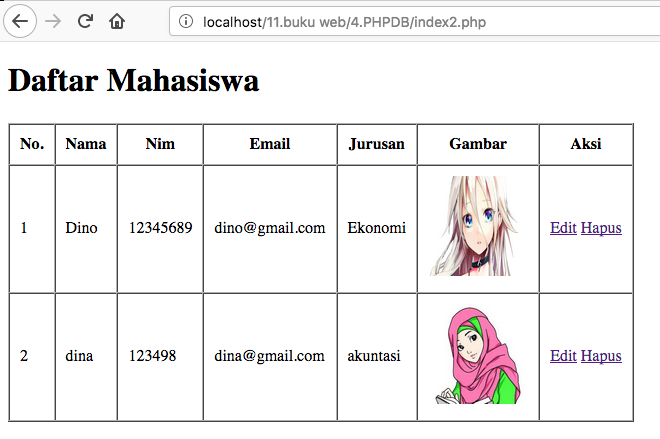
Gambar 10.39 function query

1. Pada code sebelumnya kita menggunakan looping while. Untuk kasus ini kita akan menggunakan looping foreach sehingga anda harus merubah code nya pada line 29 dan 43. ( Gambar 10.40 looping foreach)



Gambar 10.40 looping foreach

1. hasil akhirnya



Gambar 10.41 hasil

**BAB 11**

**INSERT DAN DELETE**

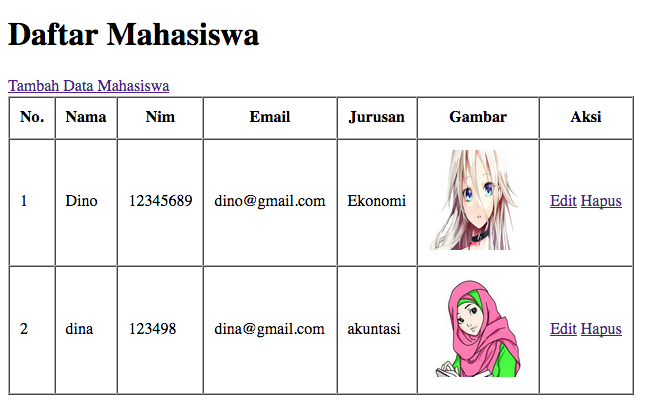
**11.1 Tambah Data**

Pada bab ini kita akan membahas tentang proses insert dan delete. Dimana bab ini merupakan kelanjutan dari bab sebelumnya sehingga kita memerlukan code-code pada bab sebelumnya. Untuk bab ini kita akan membutuhkan code **index2.php, function.php** dan **file image**. langkah pertama yang harus anda lakukan adalaha merubah **index2.php** menjadi **index.php**. Setelah kita sudah mempunyai index.php maka untuk selanjutnya ikuti langkah-langkah berikut :

1. pada Halaman **index.php** tambahkan link yang menghubungkan ke **tambah.php** ( Gambar 11.1 )



Gambar 11.1 tambah data



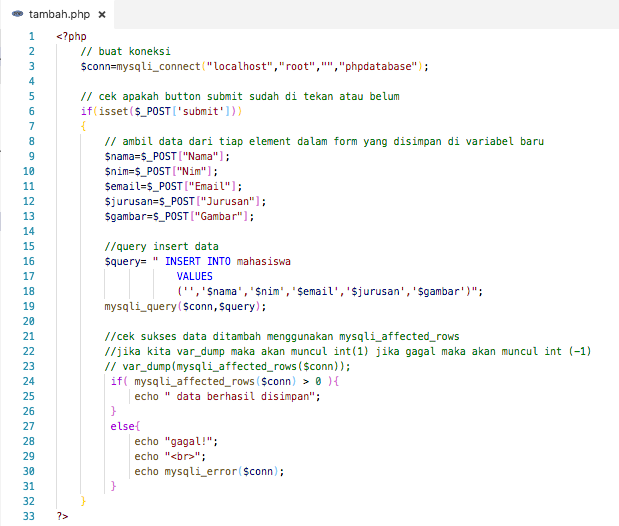
Gambar 11.2 Hasil

1. buat file baru dan beri nama **tambah.php**
2. pada file **tambah.php** ini kita akan membuat form untuk menambahkan data ( gambar 11.2)



Gambar 11.3 tambah.php

1. Setelah kita membuat formnya maka langkah berikutnya tambahkan kode php pada **tambah.php** untuk mengecek button submit sudah ditekan atau belum dan untuk proses query insert data (Gambar 11.4)



Gambar 11.4 code php pada file tambah.php

jika kita lihat pada gambar 11.4 terdapat 5 proses yang terjadi antara lain :

1. membuat koneksi
2. mengecek apakah button submit udah ditekan atau belum
3. ambil data dari tiap element pada form tambah data dan disimpan sementara pada variable baru
4. query insert data
5. mengecek apakah data berhasil atau tidak masuk ke database
6. setelah itu coba jalankan url nya dan tambahkan data

Setelah kita memahami alur atau jalannya proses penambahan data langkah selanjutnya adalah merapikan codingan kita. Jika kita lihat code yang telah kita buat code html dan code php dibuat menjadi satu halaman. Ini secara tidak langsung terlihat tidak rapi maka untuk selanjutnya kita akan membuat agar code php dan code html terpisah. Untuk langkah-langkah seperti berikut :

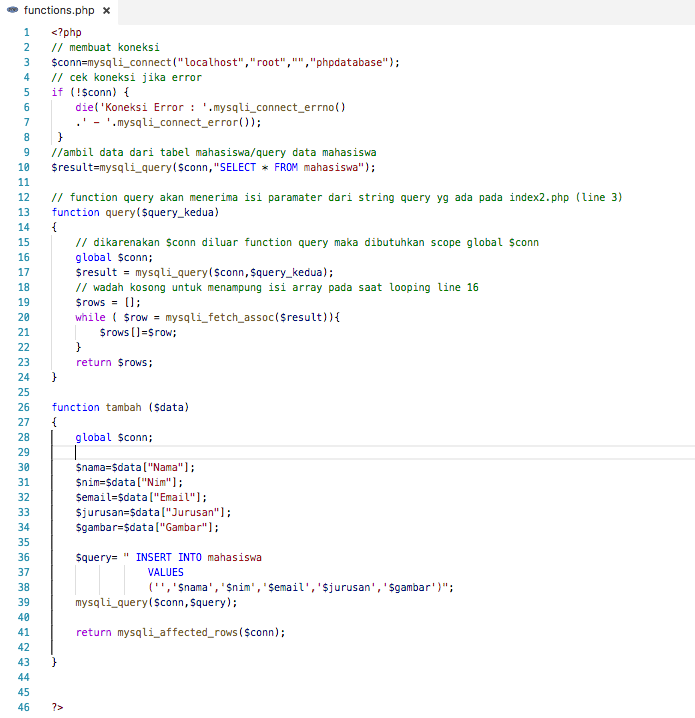
1. buat file baru dengan nama **tambah\_data.php** dimana isinya sama dengan **tambah.php**
2. **Hapus dan rubah beberapa code pada tambah\_data.php (Gambar 11.5)**



Gambar 11.5 tambah\_data.php

keterangan gambar 11.5 :

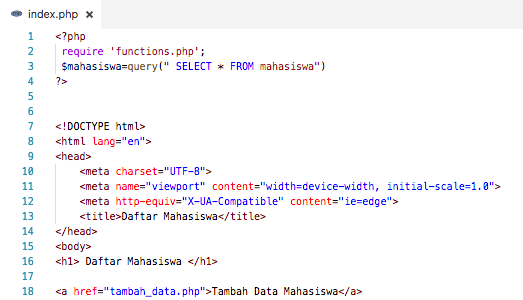
1. line 2 digunakan untuk menambahkan file **functions.php** hal ini digunakan agar function yang ada pada functions.php dapat dibaca pada file **tambah\_data.php**
2. line 4 digunakan untuk mengecek apakah button submit sudah ditekan atau belum
3. line 8 sampai line 25 adalah proses cek data apakah data berhasil disimpan.
4. line 8 terdapat **tambah** yang merupakan salah satu function yang terdapat pada functions.php. Jadi proses yang terjadi pada line 8 ini adalah **function tambah** dengan **parameter $\_POST** akan menampung sementara isi dari element form dengan method post (tambah\_data.php -> inputan dari user baik nama, nim dst) dan akan ditangkap oleh **$data** pada file **functions.php** untuk diolah lebih lanjut.
5. line 11 sampai line 14 adalah javascipt untuk menampilkan pesan dan mengembalikan user ke halaman index.php
6. line 24 digunakan untuk mencetak error jika terdapat kesalahan
7. **Tambahkan code pada functions.php yang telah kita buat pada bab 10 (Gambar 11.6)**



Gambar 11.6 function tambah

Keterangan gambar 11.6 :

1. line 26 function tambah dengan parameter **$data** dimana isi dari **$data** ini berasal dari masukan user dari parameter **$\_POST** ( tambah\_data.php -> method post ).
2. line 28 adalah digunakan untuk koneksi di database
3. line 30 sampai line 34 adalah variable baru untuk menampung sementara isi dari $data
4. line 36 sampai line 38 adalah query untuk tambah data
5. line 39 adalah function untuk menambahkan data ke database
6. line 41 adalah nilai kembalian dari function tambah
7. untuk mencobanya jangan lupa link pada index.php dirubah menjadi **tambah\_data.php** yang sebelumnya **tambah.php ( Gambar 11.7)**



Gambar 11.7 rubah line 18 menjadi tambah\_data.php

1. jalankan urlnya Gambar 11.8

|  |  |
| --- | --- |
| 1. halaman belum ditambah data   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- | 1. tambah data   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- |
| 1. data berhasil ditambah   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- | 1. halaman index.php   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- |

gambar 11.8 Hasil inputan

jika kita lihat data berhasil disimpan tetapi masih terdapat celah saat proses input data antara lain :

1. coba masukkan data sembarang dan salah satu text box kita kosongkan apa yang akan terjadi

|  |  |
| --- | --- |
| 1. awal halaman index.php   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- | 1. masukkan data sembarang   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- |
| 1. data berhasil disimpan   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- | 1. hasil dari inputan   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- |

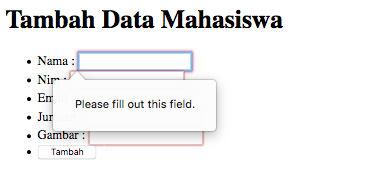
Gambar 11.9 inputan ada yang kosong

jika kita lihat pada gambar 11.9 pada saat user memasukkan data dan ada beberapa textbox yang sengaja dikosongkan maka sistem masih dapat menyimpan maka dari itu kita memerlukan sebuah notifikasi agar textbox tersebut tidak boleh kosong.Untuk pemberian notifikasi tersebut dengan cara menambahkan atribut di html 5 yaitu required pada setiap textbox pada tambah\_data.php (gambar 11.10)



Gambar 11.10 atribut required

setelah kita tambahkan atribut required pada html maka kita lakukan uji coba hasilnya seperti apa.( gambar 11.11)



Gambar 11.11 please fill out this field

kita bisa lihat hasilnya pada saat text box sengaja dikosongkan maka terdapat notifikasi bahwa textbox tersebut harus diisi atau wajib diisi.

1. setelah melakukan percobaan pertama diatas mungkin kita rasa sistem kita sudah berjalan lebih baik tetapi nyatanya belum. Hal ini bisa kita buktikan dengan cara coba masukkan kode html kedalam textbox dan lihat apa yang akan terjadi.( **Cat: jika terdapat text box yang tidak ada atribut required semisal hapus atribut required pada textbox nama dan lihat dampaknya**)

|  |  |
| --- | --- |
| 1. halaman awal html   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- | 1. masukkan file css pada salah satu text box   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- |
| 1. berhasil ditambahkan   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- | 1. tampilan index.php   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- |

Gambar 11.12 Halaman index.php di hack

Jika kita lihat pada saat user memasukkan data berupa file css maka data akan tetap berhasil disimpan akan tetapi pada saat kita tekan ok maka pada halaman index akan berubah menjadi halaman yang di hack. untuk mengatasi permasalahan ini maka kita memerlukan function **htmlspecialchars pada functions.php ( Gambar 11.13)**

|  |  |
| --- | --- |
| 1. tambahkan data   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- | 1. hasil index.php   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- |

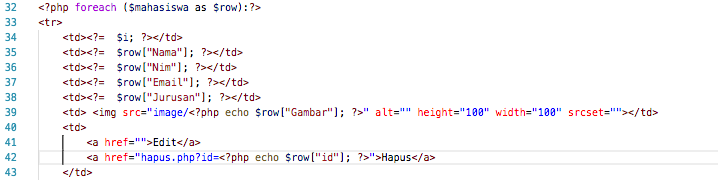
Gambar 11.13 perbaikan sistem

Jika kita lihat pada saat user menambahkan file css pada tambah\_data.php maka hasilnya yang akan ditampilkan pada index.php adalah isi dari file css tersebut. Sistem tidak menjalankan file css tersebut dikarenakan adanya function htmlspecialchars.

**11.2 Delete data**

Pada bagian ini kita akan membuat proses untuk menghapus data. Adapun langkah-langkahnya sebagai berikut :

1. buka halaman index.php dan tambahkan code berikut (Gambar 11.14)

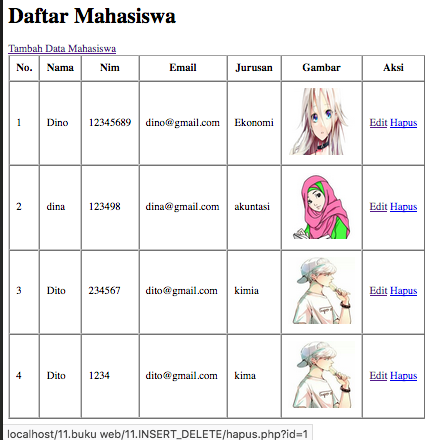


Gambar 11.14 index.php

jika kita lihat pada line 42 terdapat penambahan kode dimana link tersebut mengarah ke file hapus.php. Setelah hapus.php terdapat tanda tanya (?) dan diikuti id ini artinya adalah kita mengirimkan data id **( lihat step berikutnya )** dan id mana yang akan dikirimkan **( <?php echo $row[“id”]; ?>)**

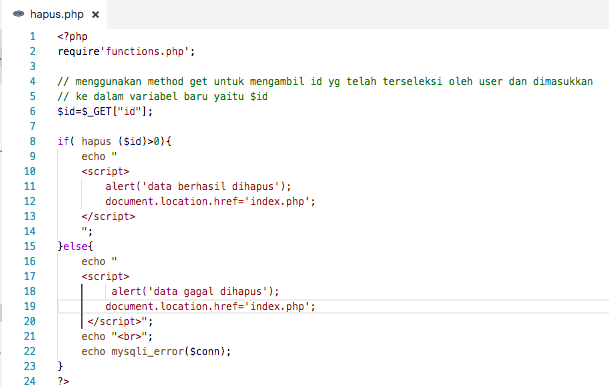
1. amati browser anda

jika kita lihat pada browser kita arahkan kursor kita ke hapus pada baris pertama. maka dibawah browser akan terlihat alamat link yang dituju dan terdapat no id yang akan dihapus (**localhost//11.buku web/11.INSERT\_DELETE/hapus.php?1)**



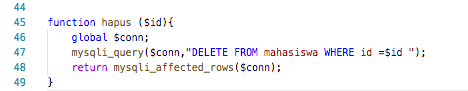
Gambar 11.15 link

1. buat file hapus.php (gambar 11.16)



Gambar 11.16 hapus.php

1. tambahkan function hapus pada file functions.php (Gambar 11.17)



Gambar 11.17 menambah function hapus pada functions.php

1. maka hasil sebagai berikut ( gambar 11.18)

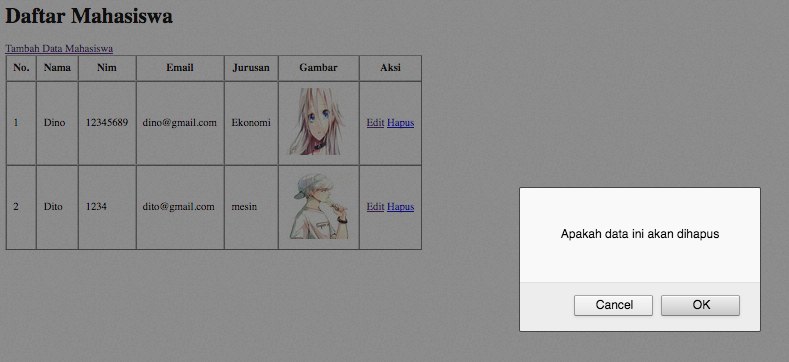
|  |  |
| --- | --- |
| 1. halaman awal index.php   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- | 1. berhasil hapus data   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- |
| 1. hasil akhir index.php   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- |  |

Gambar 11.18 Hasil akhir Hapus data

tambahkan javascript seperti pada gambar 11.19 agar menampilkan pesan seperti pada gambar 11.20

../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05-

Gambar 11.19 javascript



Gambar 11.20 Hasil dari javascript

**BAB 12**

**UPDATE DATA**

Pada bab ini kita akan melanjutkan proses CRUD yaitu update data. Update data adalah proses merubah data dimana data tersebut sudah ada di dalam database. Langkah-langkah untuk melakukan proses update data sebagai berikut :

1. Untuk merubah data maka kita memerlukan proses pengambilan data dari id. Oleh karena itu kita tambahkan code pada index.php berikut ( Gambar 12.1)

../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05-

Gambar 12.1 code edit data pada index.php

1. setelah itu buat dan buka edit.php
2. Dikarenakan isinya sama dengan tambah\_data.php maka kita copy paste ke file edit.php
3. setelah isi dari edit.php sama dengan isi dari tambah\_data.php maka lakukan beberapa perubahan kode seperti pada gambar 12.2 dan 12.3 ( rubah kata-kata tambah dengan edit)

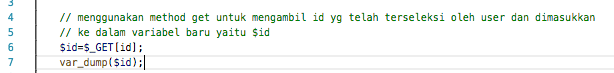


Gambar 12.2 edit.php (1)



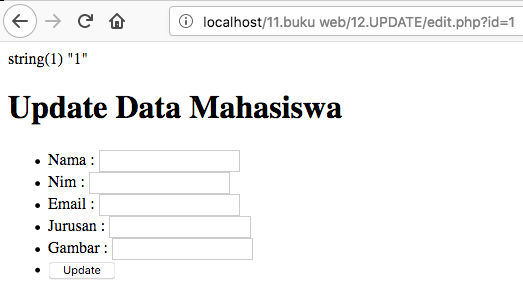
Gambar 12.3 edit.php (2)

1. langkah berikutnya kalian buat variable baru untuk menampung id pada edit.php ( gambar 12.4 )



Gambar 12.4 variabel baru

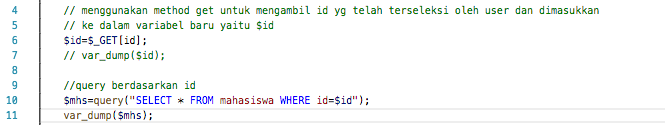
1. jika kita lihat atau kita var\_dump dan coba kalian amati (Gambar 12.5)



Gambar 12.5 hasil var dump

Jika kita lihat pada gambar 12.5 pada halaman edit.php terdapat tulisan string(1)”1” ini berarti terdapat string pada id=1.Tapi data masih belum dapat ditampilkan semuanya. untuk menampilkan isi data tersebut maka diperlukannya query.

1. Diperlukannya Query untuk menampilkan data pada index.php (gambar 12.6)



Gambar 12.6 query untuk seleksi data berdasarkan id

pada line 10 terdapat variable baru yaitu $mhs dimana fungsinya untuk menampung hasil sementara id. Query adalah function yang terdapat pada functios.php.

1. hasil var\_dump



Gambar 12.7 hasil var\_dump

pada gambar 12.7 terdapat isi dari id=1 berbeda dengan gambar 12.5. Dimana di gambar 12.5 hanya terdapat keterangan string(1) “1” sedangkan pada gambar 12.7 isi dari id=1 ditampilkan. permasalahan berikutnya bagaimana memasukkan data tersebut ke masing-masing textbox

1. untuk memasukkan data pada gambar 12.5 ke masing-masing text box maka kita harus dapat mengambil dan menampilkan salah satu data. sebagai contoh kita ingin menampilkan data nama saja maka tambahkan kode berikut ( gambar 12.8)

../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05-

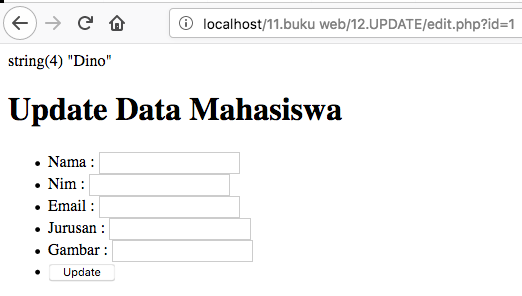
gambar 12.8 var dump

jika kita lihat line 11 kita melakukan var\_dump untuk menampilkan data nama saja. Pertanyaannya apa maksud dari [0] pada line 11. Jika kita var\_dump($mhs) dan kita view page source maka array yang ditampilkan adalah array numerik ditandai index nol (gambar 12.9 (2)). Maka untuk mengambil datanya kodenya seperti pada gambar 12.8

|  |  |
| --- | --- |
| 1. var\_dump($mhs)   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- | 1. hasil view page source   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- |

Gambar 12.9 hasil view page source

1. hasil var\_dump



Gambar 12.10 hasil var\_dump

1. setelah kita mengerti cara mengambil salah satu data pada array numerik seperti pada gambar 12.9 maka selanjutnya adalah memodifikasi variable $mhs pada edit.php( gambar 12.11)

../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05-

Gambar 12.11 variabel baru

maksud dari line 11 adalah kita memanggil function query dan dimasukkan ke dalam array rows nya yang kita ambil adalah index ke nolnya atau intinya sama dengan gambar 12.8

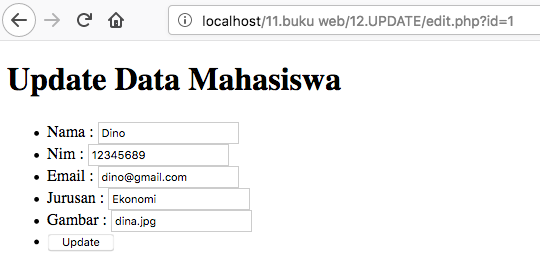
1. setelah kita tambahkan value pada code html pada edit.php seperti pada gambar 12.11



Gambar 12.12 code value

jika kita lihat pada line 57, 61, 65, 69, dan 73 terdapat code php yang digunakan untuk mengambil data masing-masing textbox berdasarkan fieldnya sebagai contoh pada line 57 codenya **value=”<?= $mhs[Nama] ?>”** ini mempunyai arti value atau nilai dari textbox dengan **name = “Nama”** adalah **isi dari tabel mahasiswa field Nama pada id 1 (id 1 karena yang saya seleksi tadi adalah data pertama dengan id 1).**

1. coba jalankan dan lihat hasilnya ( Gambar 12.13)



Gambar 12.13 edit.php

1. setelah data berhasil ditampilkan ke tiap-tiap textbox maka langkah selanjutnya adalah menambahkan **function edit** pada **functions.php** codenya seperti pada gambar 12.14

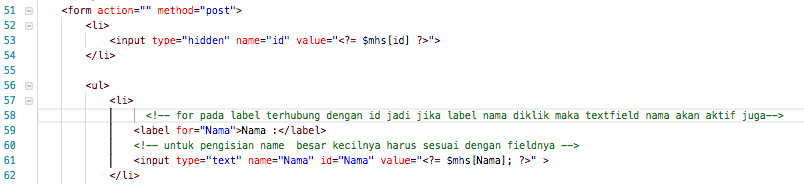


Gambar 12.14 function edit

isi dari function edit sebenarnya hampir sama dengan function tambah tetapi ada beberapa perbedaannya antara lain :

* + - * kita menambahkan variable baru dengan nama id pada line 54. variable ini berfungsi untuk mengidentifikasi data pada id mana yang akan kita rubah ( berhubungan dengan query update line 67)
      * querynya pasti berbeda pada function tambah menggunakan query insert into sedang pada function edit menggunakan query update

1. Setelah kita membuat function edit pada functions.php maka langkah berikutnya adalah menambahkan code pada edit.php ( gambar 12.5)



Gambar 12.5 edit.php

pada line 53 kita membuat sebuah sebuah inputan tapi dengan **type=”hidden”** hal ini digunakan untuk mendapatkan data id yang akan kita update dan data id tersebut akan dikirimkan ke function edit yang nati digunakan sebagai identifikasi data yang akan dirubah.

1. hasilnya

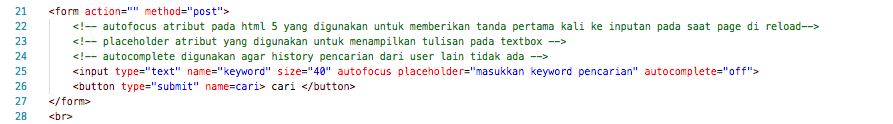
|  |  |
| --- | --- |
| 1. data awal   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- | 1. rubah data   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- |
| 1. data berhasil dirubah   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- | 1. hasil akhir   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- |

**BAB 13**

**SEARCHING**

pada bab ini kita akan melakukan proses searching atau pencarian. Untuk membuat proses searching maka ikuti langkah-langkah berikut :

1. Buka halaman index.php dan tambahkan beberapa code ( Gambar 13.1)

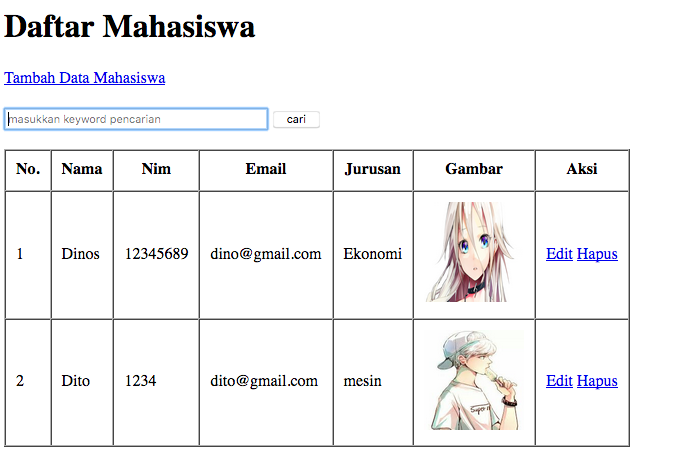


Gambar 13.1 index.php

Pada gambar 13.1 pada line 25 terdapat beberapa atribut yang akan kita tambahkan antara lain :

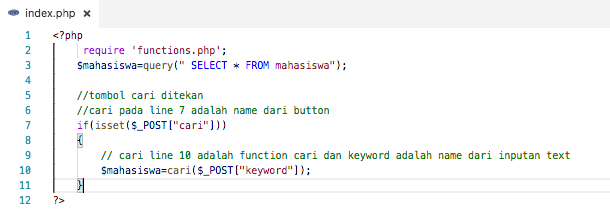
* 1. autocomplete digunakan agar pada saat user membuka halaman index.php kursor akan secara otomatis mengarah ke textbox pencarian.
  2. placeholder digunakan untuk memberikan keterangan pada textbox.
  3. autocomplete digunakan untuk menghilangkan history pada textbox.

1. Hasil dari index.php



Gambar 13.2 hasil dari index.php

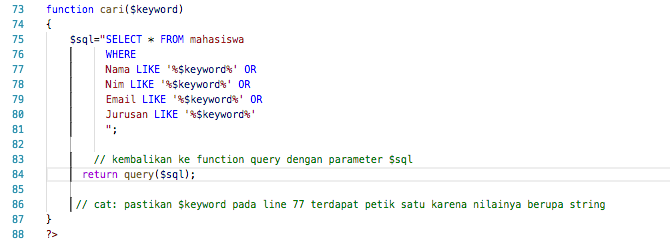
1. setelah kita membuat tampilan searchingnya maka kita lanjutkan dengan memberikan logic nya pada halaman index.php ( Gambar 13.3 )



Gambar 13.3 code pada index.php

Keterangan pada gambar 13.3

1. line 2 **require ‘functions.php’** maksud dari code ini adalah index.php menambahkan atau memanggil isi dari functios.php
2. line 3 **$mahasiswa=query( “SELECT \* FROM mahasiswa”)** ini artinya adalah terdapat variable baru dengan nama $mahasiswa dimana $mahasiswa ini melakukan proses **function query** yang terdapat pada **functions.php** dengan parameter ( **SELECT \* FROM mahasiswa**)
3. line 7 digunakan untuk mengecek apakah button dengan nama cari sudah ditekan atau belum oleh user
4. line 10 **$mahasiswa=cari($\_POST[“keyword”]);** artinya adalah bahwa $mahasiswa akan melakukan proses pada **function cari** dimana parameternya adalah masukan dari user ( **keyword adalah name dari inputan text (Gambar 13.1 line 25)**))
5. tambahkan function cari pada file functions.php



Gambar 13.4 function cari di file functions.php

Keterangan Gambar 13.6

1. line 75 sampai line line 81 adalah perintah sql untuk seleksi data berdasarkan field tertentu (Nama, Nim, Email, Jurusan) yang ditampung sementara pada variable $sql. Setelah ditampung pada variable $sql proses selanjutnya adalah dikembalikan nilai dari **$sql** tersebut ke **function query** yang terdapat pada **functions.php**
2. line 84 pengembalian nilai $sql yang akan di proses pada function query
3. jalankan urlnya

|  |  |
| --- | --- |
| 1. cari berdasarkan nama   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- | 1. hasil pencarian   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- |

Gambar 13.5 hasil

1. untuk pencarian berdasarkan email, nim, jurusan silakan anda coba

**Bab 14**

**FILE UPLOAD**

Pada bab ini kita akan membuat proses upload. Untuk lanjut lebih dalam maka kita harus mengerti terlebih dahulu file handling. File handling adalah suatu cara untuk menangani file dengan menggunakan sintak php. Ada beberapa hal yang harus kita ketahui dari file handling ini, antara lain

1. **<input type=”file”……..>**

ini adalah sebuah **input** pada php dengan **atribut** **typenya** adalah **file**. Jika kita lihat code sebelum-sebelumnya **type** dari **input** adalah **text**, untuk kali ini atau untuk proses upload ini kita membutuhkan **type** **file** untuk mengolah file kita.

1. **enctype ( encoding type) pada form**

ini merupakan salah satu atribut pada form untuk menentukan encoding apa yang akan kita gunakan nanti.

1. **$\_FILES**

ini adalah salah satu variable superglobal. Superglobal ini digunakan khusus untuk menangani files nantinya

1. **move\_uploaded\_file**

ini adalah sebuah function pada php yang digunakan untuk memindahkan file ke dalam computer ke server

Pada database kita dapat memasukkan data berupa gambar dengan menggunakan **type BLOB (Binary Large OBject)**. Tetapi jika kita menggunakan cara ini maka ukuran database akan menjadi lebih besar dan membuat database akan menjadi berat dalam proses pengolahan data. **Jadi nanti kita akan melakukan cara lain untuk proses upload, yaitu dengan cara kita hanya menyimpan nama gambar dari file gambar yang akan kita upload, dimana gambar tersebut disimpan di sebuah direktori.** Setelah kita mengetahui tentang file handling dan cara untuk melakukan proses penyimpanan pada database maka langkah selanjutnya adalah kita mulai dengan proses coding. Untuk langkah-langkah proses codingnya sebagai berikut:

1. Kita akan memodifikasi **tambah.php ( Gambar 14.1 )**. Hal ini kita lakukan karena pada proses-proses sebelumnya kita, pada proses tambah data pada field gambar kita hanya memasukkan nama dari gambar tersebut dan dimana gambar sudah ada di folder image.



Gambar 14.1 tambah\_data.php

Keterangan Gambar 14.1

* 1. pada line 67 kita rubah **type** yang sebelumnya adalah **text** kita rubah menjadi **file.** Maksud dari kita merubah nilai dari atribut type ini adalah karena kita akan mengolah file maka nilai dari type bukan lagi text tetapi file.
  2. pada line 45 pada **form** kita tambahkan atribut **enctype=”multipart/form-data**”. Maksud dari kita tambahkan atribut ini adalah untuk melakukan proses encoding file.( untuk lebih jelasnya ada pada langkah berikutnya).

1. Untuk lebih jelas fungsi dari enctype maka kita lakukan var\_dump pada tambah\_data.php



Gambar 14.2 var\_dump

1. Setelah kita tambahkan data pada tambah\_data.php dan klik view\_page\_source

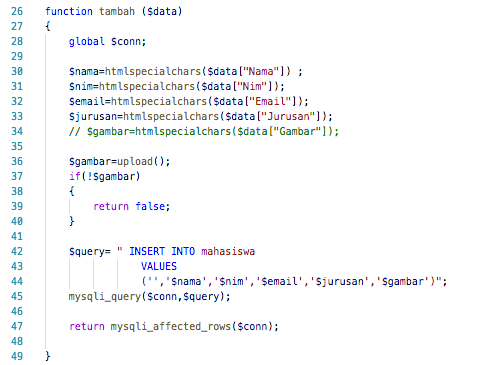


Gambar 14.3 hasil view page source

Jika kita lihat hasil dari gambar 14.3 maka kita dapat lihat bahwa ada 2 array associative disana. line 3 sampai line 14 itu merupakan array pertama yang merupakan hasil var\_dump($\_POST) sedangkan line 15 sampai line 29 merupakan hasil dari var\_dump($\_FILES). Jika kita lihat hasil dari dump files merupakan array 2 dimensi ( [ “Gambar”][“name”] dst) sedangkan pada hasil dump post merupakan array 1 dimensi [“Nama”]. Keterangan Gambar 14.3 pada line 18 sampai 27 adalah sebagai berikut:

* 1. line 18 menunjukkan bahwa nama filenya adalah dina.jpg
  2. line 20 menunjukkan type dari file tersebut adalah image/jpeg
  3. line 22 menunjukkan dimana file tersebut disimpan sementara
  4. line 24 menunjukkan jika terdapat error
  5. line 26 menunjukkan ukuran dari file tersebut

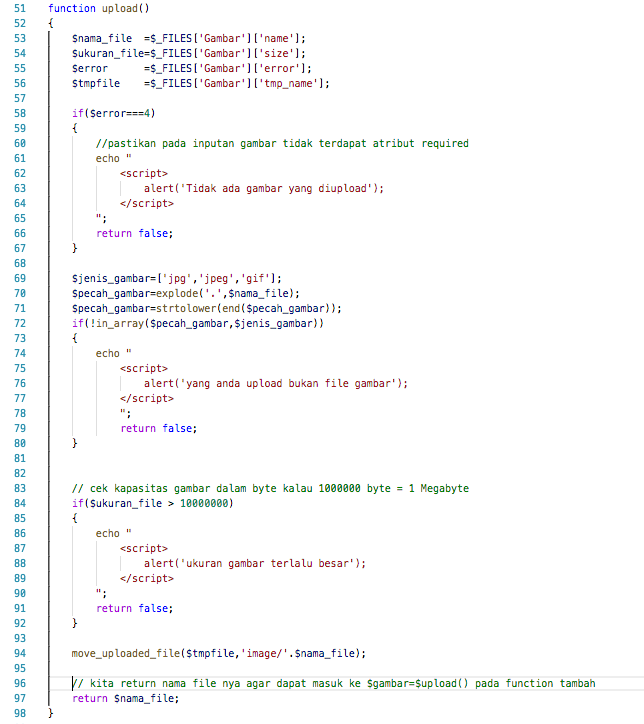
1. Memodifikasi function tambah pada functions.php ( Gambar 14.4)



Gambar 14.4 functions.php

Keterangan gambar 14.4

1. pada line 36 $gambar adalah variable yang digunakan untuk menampung hasil dari function upload
2. line 37 sampai line 40 adalah sebuah kondisi jika tidak ditemukan $gambar (!$gambar) maka akan dikembalikan ke nilai false
3. pada line 34 comment saja karena sudah tidak digunakan lagi
4. buat function upload pada functions.php ( Gambar 14.5)



Gambar 14.5 function update

keterangan gambar 14.5

1. line 53 sampai 56 adalah variable baru yang gunanya adalah menampung isi data array ( **data hasil array pada gambar 14.3** ). Tulisan **[‘Gambar]** adalah nama dari inputan gambar ( **gambar 14.1 line 67**), sedangkan **[‘name’],[‘size’]** dstnya adalah nama dari array **( gambar 14.3 line 18 dst ).**
2. line 58 sampai 67 adalah sebuah kondisi untuk mengecek error. jika terjadi error kategori 4 ( tidak ada data masukan ) (<http://php.net/manual/en/features.file-upload.errors.php)> maka terdapat pesan error gambar belum diupload
3. line 70 pada line ini kita tentukan extensi gambar apa saja yang boleh di upload oleh user.
4. line 71 kita akan memecah string menjadi array dengan menggunakan fungsi explode

( **explode (delimiter,string)** ). Di dalam function explode terdapat parameter dengan nama delimiter yang nanti digunakan untuk memecah string ke array ( kita set string apa yang digunakan sebagai pemisah, sebagai contoh titik maka **delimiter** akan memecah string ke array jika terdapat string titik ). sebagai contoh **andi.jpeg** maka **delimiter** akan memecahnya **menjadi [‘andi’,’jpeg’].**

1. line 72 artinya Setelah kita pecah menjadi array maka selanjutnya adalah mengambil string paling akhir menggunakan **end** sebagai contoh
   1. andi.jpeg setelah dipecah menjadi [‘andi’,’jpeg’] maka akan diambil string terakhir yaitu ‘jpeg
   2. andi.boy.jpeg setelah dipecah menjadi [‘andi’,’boy’,’jpeg’] maka string yang akan diambil adalah ‘jpeg’

setelah kita ambil string yang paling belakang yaitu extensinya maka langkah berikutnya adalah mengkonversi huruf besar menjadi huruf kecil menggunakan **strtolower,** agar kita tidak perlu lagi menulis 2 kali pada line 70 ( step 3)

1. line 73 kita akan mengecek apakah ada string didalam array menggunakan function **in\_array( $needle, $haystack ) -> in\_array( $pecah\_gambar, $jenis\_gambar) maksudnya isi dari $pecah\_gambar semisal jpeg dicek apakah ada pada array pada $jenis\_gambar jika tidak ada maka akan muncul pesan error.**
2. line 84 sampai line 92 ini digunakan untuk mengecek ukuran file jika lebih dari 1 Mb maka akan muncul pesan error
3. line 94 jika data sudah dicek mulai dari ada tidaknya file yang diupload, jenis filenya dan ukuran filenya maka langkah berikutnya adalah memindahkan dengan fungsi

**move\_uploaded\_file (‘tempat penyimpanan sementara’,’ tempat penyimpanan sesunggunya’ ) -> move\_uploaded\_file($tmpfile,’img/’. $nama\_file) ingat tanda titik digunakan untuk menggabungkan string. ( CATATAN: folder img sesuaikan dengan nama folder kalian dan sesuaikan dengan path folder image anda)**

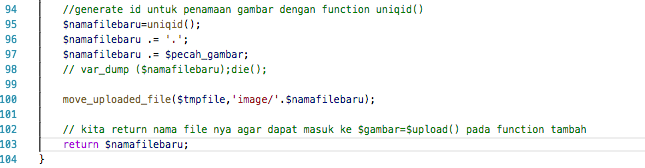
1. line 97 **return $nama\_file** langkah terakhir adalah kita kembalikan nilai dari **$nama\_file**. Kenapa harus dikembalikan ke **$nama\_file** -> coba kita lihat kembali pada **function** **tambah** terdapat code **$gambar=upload() jadi kita kembalikan nilai dari $nama\_file agar dapat disimpan sementara pada $gambar pada function tambah**
2. jika berhasil maka image yang diupload user akan berpindah ke folder image
3. Jika kita jalankan masih ada beberapa kekurangan seperti
   1. jika kita tambahkan data dengan nama gambar sama dengan yang sudah ada maka akan terjadi penumpukan ( Gambar 14.6 )

|  |  |
| --- | --- |
| 1. tampilan awal   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- | 1. tambah data   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- |
| 1. perhatikan gambar dika ( saya punya 2 gambar dika dengan nama dan extensi yang sama   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- | 1. hasil   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- |

Gambar 14.6 penumpukan gambar

Jika kita lihat pada gambar 14.6(4) terlihat gambar dika yang semula tertumpuk oleh gambar dika yang terbaru.

* 1. Untuk mengatasi hal berikut maka diperlukan generate id pada function update ( gambar 14.7) dan ada beberapa code yang harus dirubah



Gambar 14.7 uniqid

keterangan pada gambar 14.7

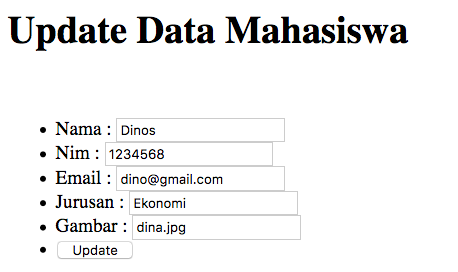
1. pada line 95 merupakan variable baru untuk menampung **uniqid** ( untuk menggenerate id).
2. line 96 gabungkan uniqid dengan titik **(contohnya:1312341.)**
3. line 97 gabungkan uniqid titik dan extensi gambarnya **(contohnya: 1312341.jpg)**
4. line 100 pindah file **$tempfile** ke **$namafilebaru** ( yang sebelumnya **$nama\_file**)
5. line 103 kita kembalikan nilai dari **$namafilebaru** yang sebelumnya **$nama\_file**
6. **pada line 96 dan 97 jika ingin mengetahuinya tinggal anda var\_dump**

* 1. hasilnya jika kita lihat di database maka akan seperti pada gambar 14.8



gambar 14.8 isi dari database

1. jika kita melakukan proses edit maka gambar tidak akan muncul dihalaman update sehingga kita tidak bisa melakukan update pada gambar ( Gambar 14.9)



Gambar 14.9 halaman update mahasiswa

1. untuk mengatasi hal diatas maka kita harus melakukan perubahan code pada halaman edit.php ( Gambar 14.10)



Gambar 14.10 halaman edit.php

Keterangan Gambar 14.10

1. line 52 tambahkan atribut **enctype**
2. line 55 digunakan untuk mengirimkan gambar yang lama. Hal ini dilakukan jika user tidak melakukan perubahan data **( beri name=”GambarLama” )**
3. line 80 kita tambahkan **img source** agar gambar dapat tampil dihalaman update
4. setelah kita melakukan modifikasi pada **edit**.**php** maka langkah berikutnya adalah memodifikasi **function edit** pada **functions.php** ( Gambar 14.11)



Gambar 14.11 function edit

keterangan gambar 14.11

1. tambahkan variable baru dengan nama $GambarLama seperti pada line 120
2. line 122 sampai line 129 digunakan untuk mengecek apakah user menekan tombol browse dengan cara kita cek error nya apakah bernilai 4 artinya tidak ada gambar yang akan diupload jika iya maka **$gambar=$GambarLama(line 125) jika tidak maka $gambar=upload() artinya lakukan proses function upload**
3. selesai silakan jalankan

**BAB 15**

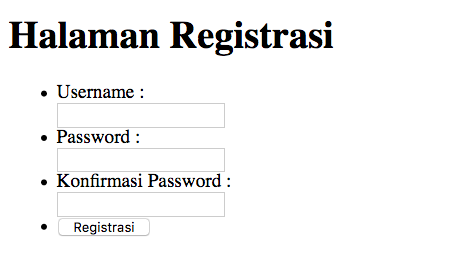
**REGISTRASI**

Pada bab ini kita akan membuat fitur registrasi atau fitur sign up. Untuk membuat fitur registrasi ini kita membutuhkan **tabel user.** Sebelum kita membuat tabel user kita membuat terlebih dulu form registrasi. Untuk memudahkan kita dalam membuat fitur registrasi ini maka ikutilah langkah-langkah membuat fitur registrasi sebagai berikut :

1. buat file baru dengan nama registrasi.php dimana isinya adalah form registasi ( Gambar 15.1)

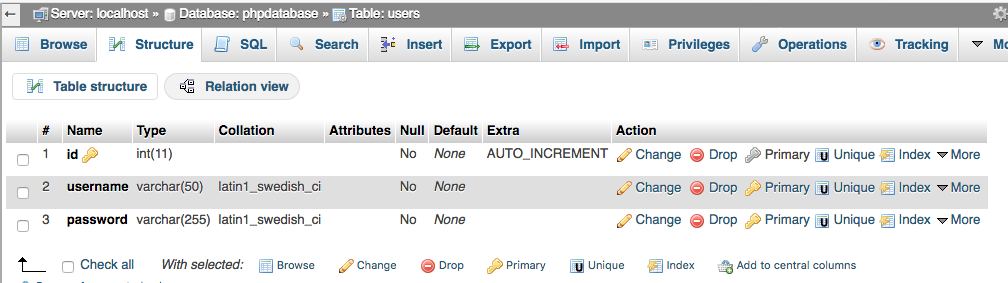


Gambar 15.1 code registrasi.php



Gambar 15.2 Hasil tampilan registrasi.php

1. setelah kita membuat form registrasi langkah berikutnya adalah membuat **tabel** dengan nama **user (** Gambar 15.3**)**



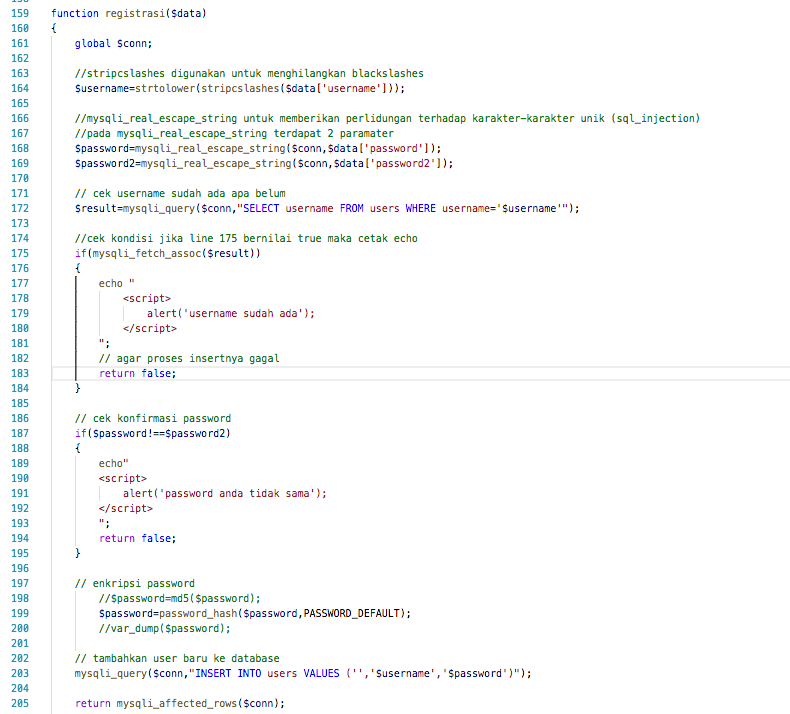
Gambar 15.3 tabel users

1. kita tambahkan code php pada halaman register.php (Gambar 15.4)



Gambar 15.4 register.php

1. setelah kita tambahkan code php pada registrasi.php maka langkah berikutnya adalah kita membuat function registrasi pada functions.php



Gambar 15.5 function registrasi

Keterangan gambar 15.5

1. line 164 terdapat variable baru $**username**. Pada variable username ini terdapat **function** **stripsclashes** digunakan untuk menghilangkan **blackslashes**
2. line 168 dan 169 terdapat variable **password** dan **password2**. password2 nanti digunakan sebagai pembanding dari password ( buat mengecek inputan user berupa password apakah sama dengan password2). Diline ini terdapat function mysqli\_real\_escape\_string ( parameter1, parameter2 ) yang fungsinya adalah untuk memberikan perlindungan terhadap karakter-karakter unik( menghindari serangan sql\_injection).
3. line 172 adalah **query** untuk mengecek apakah isi dari **username** yang diisikan oleh user ada pada database (**WHERE username=’$username’**) dan hasilnya dari querynya disimpan pada variable baru **$result**
4. line 175 sampai 184 adalah untuk mengecek kondisi dari result. Jika nilai dari result bernilai true ( line 175) maka akan mencetak **alert(“username sudah ada”).**

dan pada akhir line 183 terdapat **return false; ( ini digunakan agar proses berhenti tidak dilanjutkan). Jika ingin tahu kalian comment returnnya (line183) dan buat inputan username yang sudah ada pada database dan lihat hasilnya pada databasenya->data akan tetap masuk walaupun ada warningnya**

1. line 187 sampai line 195 digunakan untuk mengecek apakah **password** yang diinputkan oleh user tidak sama dengan **password2** ( konfirmasi password ).
2. line 199 untuk menggenkripsi password. Untuk enkripsi password ada 2 cara yaitu
   1. Menggunakan **md5**
      * jika kita menggunakan metode **md5** maka hasil enkripsi dapat mudah diketahui oleh orang lain yaitu dengan cara copy paste kode enkripsi dan search di google.( Gambar 15.6 (4))

|  |
| --- |
| 1. ubah code sebagai berikut   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- |
| 1. masukkan data   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- |
| 1. searching di google nilai enkripsinya   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- |
| 1. Hasil   ../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-05- |

Gambar 15.6 menggunakan md5

* 1. Menggunakan Hash
     + dengan menggunakan **metode** **hash** dirasa lebih aman. maka dari itu kita akan menggunakan **metode** **hash**

1. line 203 query yang digunakan untuk menambahkan ke **tabel** **user**
2. line 205 merupakan nilai kembalian ke parameter **$conn(koneksi)**

**BAB 16**

**LOGIN**

pada bab ini kita akan membuat proses login. Untuk lebih jelas kita ikuti langkah-langkah berikut:

1. Buat file baru dengan nama **login**.**php** dan buatlah **form** pada **login**.**php (**Gambar 16.1**)**

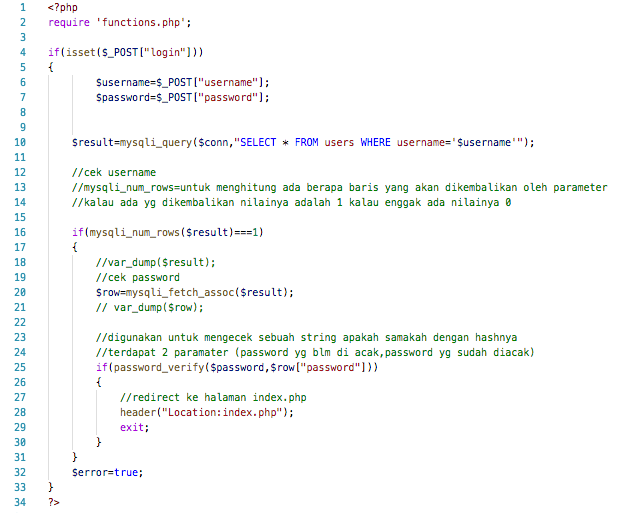


Gambar 16.1 form pada file login.php

Keterangan Gambar 16.1

* 1. buatlah form dengan method post
  2. Line 49 sampai 53 adalah untuk menampilkan pesan error dari variabel **$error** yang dikirimkan dari **code php line 32 Gambar 16.2**
  3. **pada button diberi name=”login”**

1. tambahkan kode php pada login.php (Gambar 16.2)



Gambar 16.2 code php pada file login.php

Keterangan Gambar 16.2

* 1. line 2 memanggil function yang ada pada functions.php
  2. line 4 mengecek dan mengambil data yang dimasukkan oleh user
  3. line 6 dan line 7 membuat variabel baru untuk menyimpan sementara isi dari inputan user berupa username dan password
  4. ­­ line 10 membuat variabel baru dengan nama $**result** yang digunakan menyimpan sementara hasil **query**. Dimana **query** nya adalah untuk menyeleksi semua data pada tabel user dimana isi dari **username(database**) sama **dengan $username(masukan user)**
  5. line 16 kita menggunakan **function mysqli\_num\_rows** dimana function ini digunakan untuk menghitung baris yang terdapat pada **parameter ($result)** yang nanti menghasilkan nilai dan nilai tersebut akan dikembalikan untuk dicek, dalam hal ini parameternya adalah **$result. Jika nilai kembalian bernilai 1 maka akan dilanjutkan ke proses berikutnya. Jika bernilai 0 maka akan menuju ke line 32 $error=true;**
  6. line 20 adalah variabel baru untuk menampung array dari result yang bernilai 1 pada line 16 **( var\_dump(row) -> jika ingin tahu isinya)**
  7. line 25 adalah terdapat function **password\_verify().** Fungsi dari function ini adalah digunakan untuk mengecek sebuah string yang dimasukkan user **($password) ->parameter 1** apakah sama dengan password yang sudah diacak (**$row[‘password’]**) **-> Parameter 2.**
  8. line 28 jika password sama maka akan redirect ke halaman index.php
  9. line 32 variabel error dengan nilai true. Jika bernilai true maka akan diproses pada line 49

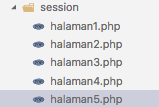
1. silakan dijalankan urlnya

**BAB 17**

**SESSION**

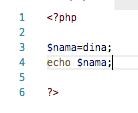
Pada bab ini kita akan membahas tentang session. Jika kita lihat pada hasil bab sebelumnya user dapat secara langsung masuk ke halaman index.php tapi melewati login.php ( dalam arti user mengetik url index.php atau tambah.php dapat secara langsung dan akan tampil halaman yang dituju ). Untuk itu kita memerlukan session untuk mengatasi hal ini. Dengan adanya session user tidak akan bisa langsung mengakses halaman tertentu sebelum login dahulu. Session adalah cara untuk menyimpan pada variabel yang disimpan di server agar dapat digunakan di beberapa halaman dan termasuk halaman dia sendiri. Untuk lebih mengerti maksudnya maka ikuti langkah-langkah berikut:

1. buat 5 file dan beri nama halaman1.php dst ( Gambar 17.1)



Gambar 17.1 halaman.php

1. pada halaman1.php berilah code sebagai berikut ( Gambar 17.2)



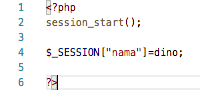
Gambar 17.2 halaman1.php

1. Pada halaman2.php berilah code sebagai berikut ( Gambar 17.3)



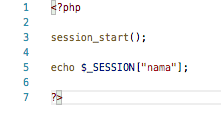
Gambar 17.3 halaman2.php

1. jalan kan dan lihat hasilnya
   1. jika kita buka halaman1.php maka akan muncul **dina**
   2. tapi pada kondisi saya ingin mengambil isi dari variabel nama pada halaman1.php yang kita panggil melalalui halaman2.php maka tidak akan berhasil atau bisa jadi terdapat pesan error. Hal ini dikarenakan pada halaman2.php tidak mengenali $nama. **Hal ini bisa kita lakukan dengan menggunakan session.**
2. pada halaman3.php berilah code sebagai berikut ( Gambar 17.4 )



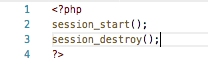
Gambar 17.4 halaman3.php

1. pada halaman4.php berilah code sebagai berikut: ( Gambar 17.5)



Gambar 17.5 Halaman4.php

1. Jika kita ingin menampilkan isi dari nama pada halaman yang lain maka kita menggunakan session. langkah-langkahnya adalah sebagai berikut :
   1. buka terlebih dahulu halaman3.php
   2. setelah kita buka halaman3.php maka selanjutnya kita buka halaman4.php maka pada halaman4.php akan muncul nama dino. Kenapa harus membuka halaman3.php terlebih dahulu??? -> karena untuk membuat session terlebih dahulu setelah session terbentuk baru setelah itu dicetak pada halaman4.php
2. pada halaman5.php berilah code sebagai berikut: ( Gambar 17.6)

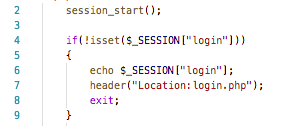


Gambar 17.6 halaman5.php

1. **jika kita tidak mendestroy session maka akan berakibat jika** **halaman4**.**php** **kita tutup setelah itu kita buka browser kembali dan membuka alamat halaman4.php maka nama dino akan muncul. Sehingga kita perlu destroy session dengan cara setelah kita buka halaman4.php maka selanjut kita buka halaman5.php untuk mendestroy session.**

setelah kita memahami proses dari session maka selajutnya kita terapkan pada sistem yang sudah kita buat sebelumnya. Untuk lebih jelasnya ikuti langkah-langkah berikut:

1. Buka halaman index.php dan tambahkan code sebagai berikut ( Gambar 17.7)



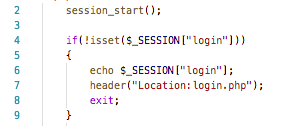
Gambar 17.7 code di index.php

code diatas digunakan jika user melakukan akses langsung ke halaman index.php melalui url maka secara otomatis user akan dikembalikan ke halaman login.php

Keterangan gambar 17.7

* 1. line 2 adalah untuk menjalankan session
  2. line 4 sampai line 9 digunakan untuk mengecek apakah ada session login jika tidak ada maka user akan dikembalikan ke halaman login.

1. setelah kita membuat code pada langkah pertama copykan code ( Gambar 17.8) atau code pada langkah pertama tadi ke halaman tambah\_data.php, edit.php dan hapus.php agar user tidak dapat langsung mengakses halaman tersebut melalui URL



Gambar 17.8 code pada tiap-tiap halaman

1. buka halaman login.php dan tambahkan kode berikut ( Gambar 17.9)



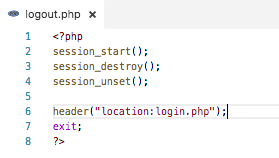
Gambar 17.9 halaman login

pada halaman login.php kita akan tambahkan beberapa code antara lain pada line 2, line 4 sampai line 8 dan line 26.

Keterangan gambar 17.9

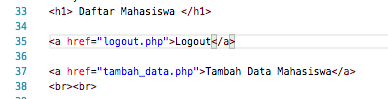
* 1. line 2 adalah untuk menjalankan session
  2. line 4 sampai line 8 digunakan jika user sudah masuk ke halaman login.php dan menuju halaman index.php dan setelah itu user mau menuju kembali ke halaman login maka sistem akan mengembalikan ke halaman index.php
  3. line 26 digunakan untuk mengset session

1. buat file dengan nama logout.php untuk mendestroy session ( Gambar 17.10)



Gambar 17.10 logout.php

1. tambahkan code berikut ( gambar 17.11) pada index.php agar terdapat button atau link logout



Gambar 17.11 halaman index.php

**BAB 18**

**COOKIES**

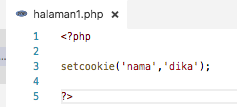
bab ini kita akan membuat cookies. Berbeda dengan session dimana informasi disimpan di server sedangkan cookies informasi disimpan di klien. untuk lebih jelasnya konsep dari cookie maka ikuti langkah-langkah berikut :

1. buat folder baru dengan nama cookie dan tambahkan 2 file yaitu halaman1.php dan halaman2 .php ( Gambar 18.1)

../../../../../Users/dimaswahyu/Desktop/Screen%20Shot%202018-06-

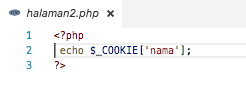
Gambar 18.1 folder cookie

1. code pada halaman1.php ( Gambar 18.2)



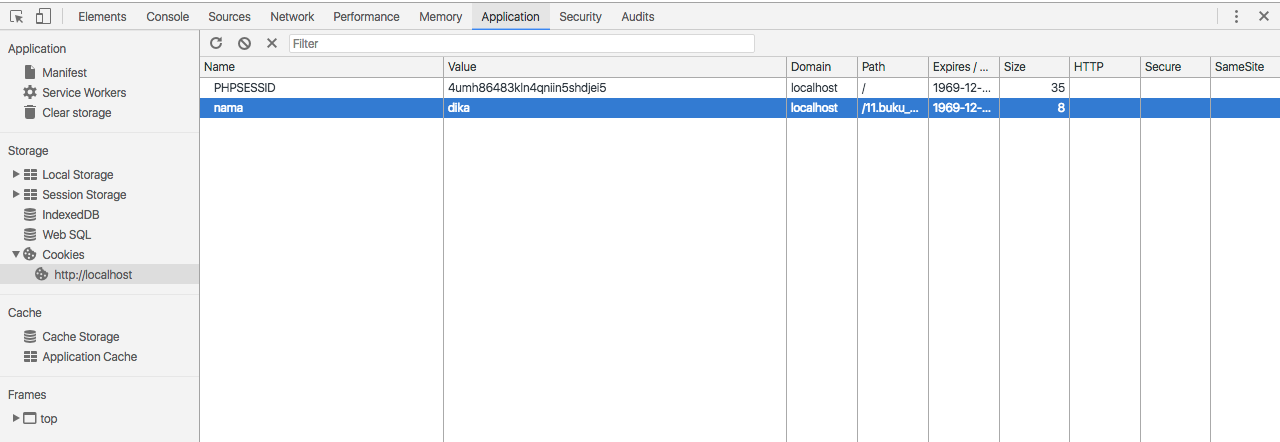
Gambar 18.2 halaman1.php

1. code pada halaman2.php ( Gambar 18.3)



Gambar 18.3 halaman2.php

1. untuk melihat hasilnya buka dulu halaman1.php setelah itu halaman2.php. Setelah itu buka inspect element cari tab application maka akan ada keterangan cookie (Gambar 18.4)



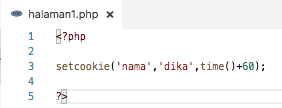
gambar 18.4 inspect element

Keterangan Gambar 18.4

1. PHPSESSID adalah id dari session yang pernah kita buat ( abaikan saja)
2. setelah PHPSESSID terdapat cookie dengan value dika ini dapat juga kita rubah menggunakan extensi/plugin edit this cookie di chrome atau dapat juga kita hapus.

jika kita tutup browser dan buka kembali halaman2.php maka secara otomatis cookie akan terhapus **( tutup browser bukan tab pada browser).** Kita dapat mengatur waktu agar cookie tidak secara otomatis terhapus jika kita tutup browsernya dengan cara :

1. buka halaman1.php tambahkan code berikut :



Gambar 18.5 setcookie

1. untuk mencobanya kita buka dulu halaman1.php setelah itu halaman2.php dan tutup browsernya. setelah kita tutup browsernya langsung buka halaman2.php. Bisa kita lihat cookie masih ada, hal ini disebabkan karena kita telah set penghapusan cookie 1 menit ( gambar 18.5)

setelah kita mengetahui kerja dari cookie maka selanjutnya adalah menerapkan cookie ini didalam sistem. Untuk langkah-langkahnya sebagai berikut :

1. buka halaman login.php dan tambahkan beberapa code ( Gambar 18.6 dan gambar 18.7)

  
Gambar 18.6 login.php

keterangan Gambar 18.6

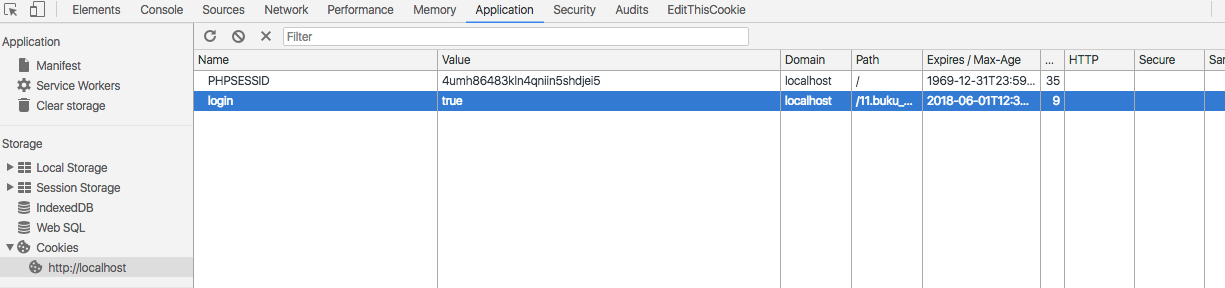
* 1. tambahkan code pada line 29 sampai line 32. Code ini digunakan untuk mengset cookie**. login adalah nama dari button submit, true adalah nilai Boolean, dan time()+60 adalah kita set cookie akan hilang setelah 1 menit**



Gambar 18.7 login.php

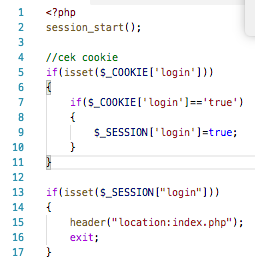
Keterangan Gambar 18.7

1. tambahkan code pada line 72 sampai line 75. Disini kita menambah code inputan berupa checkbox dan label remember me
2. untuk mengecek cookie apakah jalan buka halaman login.php dan centang remember me. setelah itu buka dengan menggunakan inspect element -> application maka akan terlihat cookienya ( Gambar 18.8)



gambar 18.8 inspect element

1. tambahkan code berikut ( Gambar 18.9) pada login.php



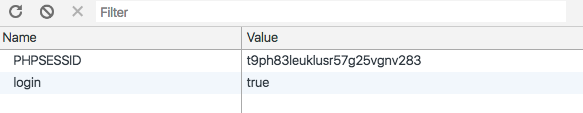
Gambar 18.9 login.php

jika terdapat kasus user masuk ke halaman index.php ( **login terlebih dahulu dan cek remember me**) setelah user masuk browser ditutup dan dibuka kembali maka tampilan **user** akan **index**.**php**. Agar lebih jelas ikuti langkah berikut:

* 1. buka halaman login.php dan cek remember me
  2. jika username dan password benar maka akan masuk ke halaman index.php
  3. setelah kita masuk ke halaman index.php tutup browser anda
  4. setelah browser kita tutup.Buka kembali browser anda dan ketik urlnya maka akan muncul halaman index.php ( **Syarat tidak lebih dari 1 menit. Karena sudah ke set time()+60).**

hal ini terjadi karena informasi cookie masih tersimpan dalam browser. Sehingga sebelum satu menit user akan ditampilkan halaman terakhir yang dia buka. Berbeda jika user **tidak mengcek remember me** pada saat user berhasil membuka halaman index.php dan user menutup browser**(exit browser)** setelah itu membuka kembali maka user tidak akan ditampilkan halaman terakhir yang dibuka user. Agar lebih jelas ikuti langkah berikut:

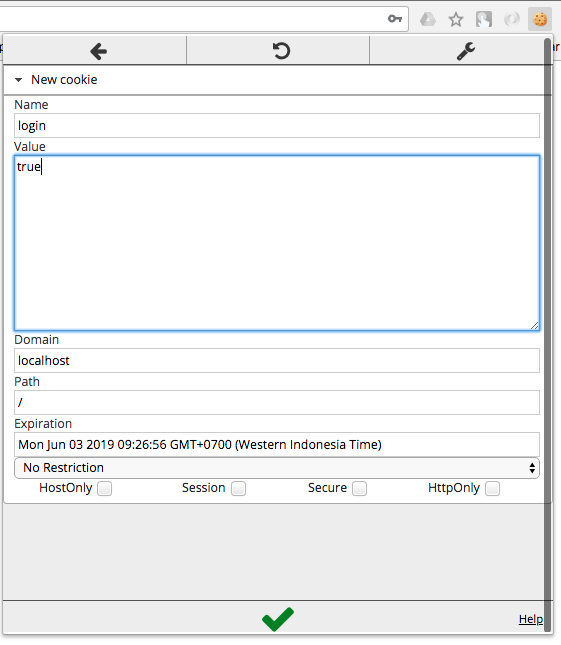
1. buka halaman login.php dan remember me tidak dicek
2. jika username dan password benar maka akan masuk ke halaman index.php
3. setelah kita masuk ke halaman index.php tutup browser anda **(exit browser)**
4. setelah browser kita tutup.Buka kembali browser anda dan ketik urlnya maka akan muncul halaman login.php
5. Ada kelemahan dari sistem ini. Kelemahannya adalah karena cookie mudah ditembak yaitu **namenya login dan valuenya true** ( Gambar 18.10).



Gambar 18.10 inspect element

maka kita dengan mudah menambahkan cookie dengan menggunakan plugin dari chrome yang **bernama edit this cookie.** Untuk lebih jelas kita coba langkah berikut:

1. buka halaman login.php
2. buka aplikasi edit this cookie dan tambahkan name login valuenya true ( Gambar 18.11)



Gambar 18.11 edit this cookie

1. refresh halaman login.php dan lihat apa yang akan terjadi. User akan langsung masuk ke halaman index.php.

jadi berdasarkan kejadian diatas maka kita harus berusaha untuk menutupi kelemahan cookie tersebut. Untuk menutupi kelemahan cookie tersebut dengan cara :

1. kita tambahkan proses enkripsi menggunakan hash tipe sha256. Rubah code yang sebelumnya seperti pada gambar 18.6 line 31 kita rubah menjadi gambar 18.12



Gambar 18.12 rubah code pada login.php

keterangan gambar 18.12

* + - * line 50 digunakan untuk mengset cookie dengan **nama id , pada row id , dan set timenya +60**
      * line 51 digunakan untuk mengset cookie dengan menggunakan enkripsi dengan **parameter namanya key, tipe enkripsinya hash (parameter hashnya adalah tipe hash sha256 , pada row username ), dan parameter cookie yang terakhirnya adalah set timenya +60**
      * **untuk link hash** [**http://php.net/manual/en/function.hash.php**](http://php.net/manual/en/function.hash.php)

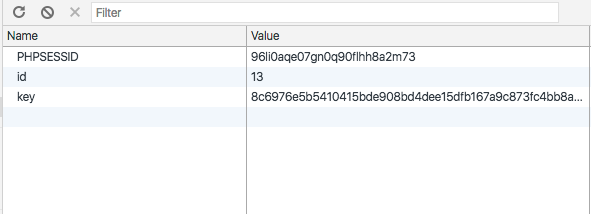
1. rubah dan tambahkan code berikut pada login.php yang sebelumnya codenya seperti pada gambar 18.9 dirubah menjadi gambar 18.13



Gambar 18.13 rubah code pada login.php

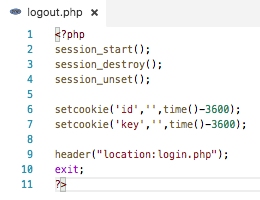
Keterangan gambar 18.12

* + - * line 6 sampai line 20 digunakan untuk mengecek cookies. Setelah cookie diset dan dienkripsi maka langkah berikutnya adalah mengcek cookie id dan cookie username
      * line 8 dan 9 adalah variabel baru untuk menampung sementara cookie id dan username
      * line 12 variabel baru untuk menampung isi dari query
      * line 13 variabel baru untuk merubah hasil query menjadi array assoc yang isinya username
      * line 16 mengcek apakah **key (username yang sudah diacak)** sama dengan **enkripsi $row[‘username’] menggunakan hash dengan tipe sha256** apakah sama jika sama maka session login akan true
  1. hasilnya gambar 18.14



Gambar 18.14 hasil enkripsi

* 1. setelah kita buat cookie menggunakan enkripsi langkah terakhir adalah menghapus cookie saat logout. Buka halaman logout dan tambahkan code berikut ( Gambar 18.15) line 6 dan 7



Gambar 18.15 logout.php

1. Dengan kita menambahkan enkripsi pada cookie kemungkinan user melakukan proses hacking semakin kecil.

**BAB 19**

**PAGINATION**

pada bab ini kita akan menambahkan fitur pagination. Pagination adalah fitur yang digunakan untuk membatasi tampilan data agar tidak terlalu panjang dan tampilan jadi lebih rapi. Untuk memahami lebih dalam maka ikuti langkah-langkah berikut: