|  |  |
| --- | --- |
| Materi | Nilai |
| Pembelajaran PHP Video 31 - 40  ( Tugas Minggu ke – 13 ) | 95 |

1. Fungsi dari post pada php adalah untuk mengirim data yang menggunakan form dan isinya bersifat tidak umum.
2. Pada penggunaan method="post", informasi pada post tidak dapat dilihat oleh orang lain, karena method="post" itu bersifat rahasia.
3. Form berisi sebagai tempat untuk menampung dan mengisi data, yang biasanya di dalam form terdapat tag input ( <input> )
4. Input pada form berfungsi untuk tempat apabila data dimasukkan dan berbentuk kotak.
5. Di dalam input terdapat attributes type" ", digunakan untuk menentukan tipe dari jenis input yang akan diproses dan terdapat attributes name" ", digunakan untuk mengatur nama dari input.
6. Menambahkan input yang bertipe submit, digunakan sebagai tombol untuk mengirimkan data ke server.
7. required yang berada di dalam <input> menandakan bahwa <input> tersebut harus diisi.
8. Placeholder pada <input> berfungsi sebagai teks pemberitahuan di dalam kotak <input>.
9. Menambahkan beberap class dari bootstrap, yaitu form-group dan form-control.
10. Form-control digunakan untuk membuat panjang kotak form menjadi panjang browser / yang sudah diatur.
11. Untuk menggunakan form-control harus menggunakan div yang berisi form-group, lalu menambahkan form-control pada tag <input>.
12. Menyimpan nilai dari isi $\_POST menggunakan if.

* Cth : if (isset($\_POST['simpan'])) {

$menu = $\_POST['menu'];

}

Artinya, apabila $\_POST['menu'] di klik, maka akan disimpan di dalam $menu.

1. Insert data adalah memasukkan data dari PHP ke dalam database.
2. Penulisan perintah dari insert adalah menggunakan INSERT INTO nama\_tabel VALUES(' ','nama\_data\_yang\_ditambah')

* Cth : $sql = "INSERT INTO tblmenu VALUES ('','$menu')";

echo $sql;

Artinya, di dalam $sql terdapat perintah INSERT INTO untuk menambah data di dalam tbl menu dengan values / hasil yang sama dengan $menu (' '.'$menu'). pada pengaturan values, terdapat 2 pasang tanda petik yang 1 pasang berisi $menu dan 1 pasang lainnya kosong karena pada tblmenu terdapat 2 pilihan kolom data.

1. Menyambungkan data yang disimpan ke dalam database dengan menggunakan perintah function runSQL.

* Cth : $db->runSQL($sql);

Artinya, terdapat objek $db yang menjalankan perintah runSQL dan memasukkan data yang ditambahkan ke dalam database.

1. Header berfungsi untuk menampilkan halaman web awal yang ditautkan apabila selesai melakukan seseuatu, maka tampilan akan kembali ke halaman web awal.
2. Mengembalikan tampilan web awal setelah terjadi insert data menggunakan header.

* Cth : header("location:?f=menu&m=select");

Artinya, apabila selesai menambahkan data pada insert.php maka web akan kembali kepada tampilan pada select.php/ halaman ?f=menu&m=select.

1. Mengedit tampilan tabel dengan mengatur width / lebar pada class.

* Cth : <table class="table table-bordered w-75">

Artinya, tabel tersebut mempunyai class table-bordered yang berfungsi untuk membuat garis tepian pada tabel dan mempunyai width 75.

1. Untuk menambahkan angka pada form, maka menambahkan number agar dapat menambahkan / membaca angka pada form.

* Cth : <input type="text" name="harga" number required placeholder="isi harga!" class="form-control" >

Artinya, pada <input> tersebut terdapat baris yang mempunyai nama "harga" dengan menunjukkan bahwa isinya angka menggunakan "number", baris tersebut wajib diisi karena terdapat required, dan terdapat teks pemberitahuan di dalam kotak <input> sebelum kotak tersebut diisi karena menggunakan placeholder.

1. <input> type="file" digunakan ketika apabila di dalam form kita ingin menambahkan gambar / file lain, maka menggunakan type data file.

* Cth : <input type="file" name="gambar" >

Artinya, pada <input> tersebut mempunyai nama "gambar" dan <input> tersebut bertipe "file" karena kotak kolom input tersebut akan digunakan untuk mengupload gambar.

1. Apabila <input> menggunakan tipe file adalah, menghasilkan button bertuliskan pilih file dan apabila button tersebut di klik maka akan menyambungkan ke dalam file explorer dan kita dapat memilih file / gambar apa yang ingin kita tambahkan.
2. CTRL + SHIFT + L berfungsi untuk mengganti semua kata yang sama seperti kata yang di blok, jadi hanya mengubah satu kata dan kata lain yang otomatis ikut berubah. Mengganti semua kata kategori menjadi menu.
3. select di dalam form digunakan untuk membuat objek form menjadi list yang dapat dipilih.
4. option pada form digunakan untuk menampilkan list list yang dapat dipilih.
5. Menuliskan perintah fungsi SELECT di dalam satu variabel dengan function $db, tidak usah membuat variabel lain untuk menyimpan perintah fungsi.

* Cth : $row = $db->getALL("SELECT \* FROM tblkategori ORDER BY kategori ASC");

Artinya, pada $row terdapat objek $db yang di dalamnya berisi getALL untuk memilih data dari tblkategori yang data pada kategori diurutkan secara ASC / dari nilai kecil ke besar dengan menggunakan ORDER BY.

1. Memasukkan semua data kategori ke dalam option menggunakan foreach untuk menampilkan nilai perulangan dengan menggunakan array.

* Cth : <?php foreach($row as $r): ?>

<option value="<php echo $r['idkategori'] ?>"><?php echo $r['kategori'] ?> </option>

<?php endforeach ?>

Artinya, terdapat foreach untuk melakukan perulangan, dan sebagai key adalah $row sama dengan $r, di dalam foreach terdapat option dengan value $r['idkategori'] yang akan muncul nilai idkategori ketika $r['kategori'] di klik dan diatur menggunakan if.

1. Memanggil $r['idkategori'] dan menambahkannya sebagai value di dalam option yang berisi $r['kategori'] sebagai list, agar ketika salah satu list kategori diklik, maka akan muncul nomor id kategori nya.
2. Untuk menjalankan perintah insert data pada form menggunakan if.

* Cth : if (isset($\_POST['simpan'])) {

$menu = $\_POST['menu'];

$sql = "INSERT INTO tblmenu VALUES ('','$menu')";

echo $sql;

$db->runSQL($sql);

header("location:?f=menu&m=select");

Artinya, apabila $\_POST['simpan' di klik, maka $menu akan menyimpan $\_POST['menu'] terlebih dahulu, lalu untuk menjalankan perintah nya, maka $sql menyimpan perintah untuk menambahkan data INSERT INTO pada tblmenu dengan values sesuai dengan $menu. $sql akan ditampilkan menggunakan echo, dan disambungkan ke dalam database melalui objek $db dengan function runSQL($sql) yang akan menjalankan perintah $sql, setelah $sql dengan perintah insert dijalankan, maka akan kembali ke dalam tampilan web select.php / ?f=menu&m=select karena terdapat header yang disambungkan dengan link select.php.

1. Menambahkan beberapa variabel di dalam if sesuai dengan kotak <input>.

* Cth : $idkategori = $\_POST['idkategori'];

$menu = $\_POST['menu'];

$harga = $\_POST['harga'];

$gambar = $\_FILES['gambar']['name'];

$temp = $\_FILES['gambar']['tmp\_name'];

Artinya, pada $idkategori menyimpan data dari $\_POST['idkategori'], pada $menu menyimpan data dari $\_POST['menu'], pada $gambar menyimpan data dari $\_POST['gambar'], gambar adalah name $\_POST['gambar'] tetapi karena <input> bertipe file, maka diberi tambahan $\_FILES, dan $temp menyimpan data yang ada di $\_POST['gambar'] dengan name di dalam <input> dan tmp\_name adalah data dari browser untuk tempat menyimpan data.

1. Input pada upload bertipe file, maka di dalam variabel $gambar diberi tambahan $\_FILES untuk membaca file yang diupload. Untuk menampilkan salah satu dari beberapa value, maka bisa mendeskripsikan variabel baru sebagai penyimpanan tempat value yang akan dipilih.

* Cth : $gambar = $\_FILES['gambar']['name'];

Artinya, karena input bernilai file, jadi ditambahkan $\_FILES, $\_FILES akan disimpan di dalam $gambar, $\_FILES yang ada di <input> mempunyai name "gambar" dan value yang akan ditampilkan adalah "name".

1. Membuat variabel untuk menyimpan tempat temporari file nya.

* Cth : $temp=$\_FILES['gambar']['tmp\_name'];

Artinya, pada $temp terdapat $\_FILES yang ada di name di dalam <input> gambar dan tmp\_name yang terdapat pada browser.

1. Di dalam form yang terdapat <input> bertipe file, maka pada <form> harus diberi tambahan enctype="multipart/form-data" agar dapat membaca <input> bertipe file.

* Cth : <form action="" method="post" enctype="multipart/form-data">

Artinya, terdapat form dengan method post dan pada form tersebut terdapat <input> yang bertipe file karena terdapat enctype pada form.

1. Pada perintah INSERT INTO yang terdapat pada $sql menambahkan beberapa variabel yang menyimpan data pada form agar data nya dapat terbaca.

* Cth : $sql = "INSERT INTO tblmenu VALUES ('',$idkategori,'$menu','$gambar',$harga)";

echo $sql;

Artinya, pada $sql terdapat perintah INSERT INTO untuk menambah data pada tblmenu dengan VALUES yang membaca data pada $idkategori,$menu,$gambar, dan $harga. pada $menu,$gambar menggunakan tanda petik karena data yang ada pada database bertipe varchar, dan untuk $idkategori dan $harga tidak menggunakna tanda petik dua karena tipe data pada database adalah integer dan float. Lalu data $sql akan ditampilkan menggunakan echo.

1. Tampilan apabila selesai mengisi <input> pada form lalu mengklik simpan adalah dengan menampilkan data data yang telah dimasukkan.

* Cth : INSERT INTO tblmenu VALUES ('',30,'Minuman Kaleng','minuman-kaleng.jpg',5000)

Artinya, data pada $sql jika ditampilkan akan tertulis seperti ini, dengan membaca perintah menambahkan data pada tblmenu dengan values 30 ->$idkategori, 'Minuman Kaleng'->$menu, 'minuman-kaleng.jpg'->$gambar, dan 5000->$harga.

1. if(empty()) digunakan untuk mengatur apabila terdapat kondisi yang harus diisi, maka akan memuncullkan tulisan error / peringatan.
2. Melakukan pengaturan jika $gambar tidak diisi / kosong maka akan memunculkan tulisan GAMBAR KOSONG. MOHON MASUKKAN GAMBAR! dengan menggunakan if.

* Cth : if (empty($gambar)) {

echo "<h5> GAMBAR KOSONG. MOHON MASUKKAN GAMBAR! <h3>";

}else{

$sql = "INSERT INTO tblmenu VALUES ('',$idkategori,'$menu','$gambar',$harga)";

echo $sql;

}

Artinya, apabila $gambar kosong, maka akan memunculkan tulisan ukuran <h5> dengan isi GAMBAR KOSONG. MOHON MASUKKAN GAMBAR!, maka user diharuskan untuk mengisi $gambar, apabila gambar sudah diisi maka akan menjalankankan perintah INSERT INTO yang ada di dalam $sql.

1. Untuk menambahkan / mengupload file yang berada di dalam folder, maka harus menambahkan perintah move\_uploaded\_file.

* Cth : move\_uploaded\_file($temp,'../upload/'.$gambar);

Artinya, file diupload dengan temporari yang disimpan di dalam $temp, lalu karena $gambar yang disimpan diletakkan di dalam foler upload yang letaknya beda folder dengan index.php yang ada di folder admin, maka untuk menambahkan harus dengan memberi tanda ../ untuk mencari folder, dan disambungkan dengan $gambar yang berisi nama dari gambar yang dipilih.

1. Mengatasi error karena objek $db tidak terbaca sebab perubahan pada vscode belum saya save / tekan ctrl +s.
2. Membuat file update.php di dalam menu untuk menyimpan data tentang update.php pada menu.
3. Menampilkan id dari data jika hyperlink update di klik menggunakan if.

* Cth : if (isset($\_GET['id'])) {

$id = $\_GET['id'];

echo $id;

}

Artinya, apabila $\_GET['id'] di klik, maka data dari $\_GET['id'] akan disimpan di dalam $id, sehingga nilai dari $id sama dengan nilai dari $\_GET['id'], lalu $id akan ditampilkan menggunakna echo.

1. Menambahkan $sql yang berisi perintah SELECT untuk menampilkan nilai $id dan menampilkan perintah SELECT di dalam if.

* Cth : if (isset($\_GET['id'])) {

$id = $\_GET['id'];

$sql = "SELECT \* FROM tblmenu WHERE idmenu = $id";

echo $sql;

}

Artinya, apabila $\_GET['id'] di klik, maka nilai dari $\_GET['id'] akan disimpan di dalam $id, terdapat $sql yang berisi perintah memilih data / SELECT dari tblmenu dan nilai yang berada di dalam idmenu sama dengan $id. Lalu menampilkan $sql menggunakan echo.

1. Memanggil / menampilkan nilai dari satu data menggunakan function getITEM pada dbcontrol.

* Cth : $item = $db->getITEM($sql);

Artinya, $item berisi objek $db yang memanggil function getITEM dan menjalankan getITEM pada $sql.

1. Pada function getITEM terdapat $result yang berisi perintah mysqli\_query, mysqli\_query adalah fungsi yang digunakan untuk mengirimkan perintah sql ke seerver mysql untuk melakukan aktivitas - aktivitas seperti :

* menambahkan data dengan perintah SQL INSERT.
* membaca data dengan perintah SQL SELECT.
* mengubah data dengan perintah SQL UPDATE
* menghapus data dengan perintah SQL DELETE.

1. Pada function getITEM juga terdapat $row yang berisi perintah mysqli\_fetch\_assoc, mysqli\_fetch\_assoc adalah fungsi php yang menghasilkan array assosiatif dimana indexnya sesuai dengan nama kolom yang diseleksi.
2. Menampilkan nilai dari $item yang berisi $db->getITEM($sql) menggunakan var\_dump.

* Cth : var\_dump($item);

Artinya, menampilkan nilai dari $item, maka jika ditampilkan, $item akan menampilkan nilai sesuai data yang dipilih, urutan tampilan nilai adalah, ["nama\_kategori"]=>tipedata() "hasil", akan dilakukan berulang sesuai jumlah kategori pada data.

1. Menginputkan value menu pada form sesuai dengan data yang dipilih.

* Cth : <input type="text" name="menu" required value="<?php echo $item['menu'] ?>" class="form-control" >

Artinya, terdapat <input> bertipe text yang mempunyai nama menu, <input> tersebut wajib untuk diisi, <input> mempunyai value yang mempunyai nilai yang sama dari data $item yang mengambil data dari ['menu'] pada tblmenu. Maka akan menampilkan menu yang dipilih.

1. Menginputkan value harga pada form sesuai dengan data yang dipilih.

* Cth : <input type="text" name="harga" number required value="<?php echo $item['harga'] ?>" class="form-control" >

Artinya, terdapat <input> bertipe text yang mempunyai nama harga. <input> tersebut wajib untuk diisi, <input> mempunyai value yang bernilai dama dari data $item yang mengambil nilai dari data ['harga'] pada tblmenu. Maka akan menampilkan harga dari menu yang dipilih.

1. Membuat 2 variabel yang berisikan idkategori untuk disambungkan ke dalam <option> dan gambar untuk disambungkan di dalam form gambar.

* Cth : $idkategori = $item['idkategori'];

$gambar = $item['gambar'];

Artinya, $idkategori mengambil nilai yang sama seperti $item['idkategori'], dan $gambar mengambil nilai yang sama seperti di dalam $item['gambar'].

1. Mengatur tampilan option menggunakan if.

* Cth : <?php foreach($row as $r): ?>

<option <?php if($idkategori == $r['idkategori']) echo "selected" ?> value="<?php echo $r['idkategori'] ?>"><?php echo $r['kategori'] ?> </option>

<?php endforeach ?>

Artinya, terdapat php foreach untuk melooping list yang terdapat pada option dengan $row sebagai $r, di dalam option terdapat if yang berisi apabila $idkategori nilai nya sama dengan $r['idkategori'] maka list yang muncul / selected adalah list yang sesuai dengan menu kategori yang dipilih, karena valuenya adalah $r['kategori'], maka akan menjalankan perintah if, option mempunyai hasil list yaitu $r['kategori'].

1. Menambahkan perintah untuk update data menggunakan UPDATE.

* Cth : $sql = "UPDATE tblmenu SET idkategori=$idkategori,menu='$menu',gambar='$gambar',harga=$harga";

echo $sql;

Artinya, $sql terdapat perintah update data dengan UPDATE pada tblmenu dengan pengaturan SET idkategori bernilai sama dengan $idkategori. menu bernilai sama dengan $menu, gambar bernilai sama dengan $gambar, dan harga bernilai sama dengan $harga. Lalu $sql akan ditampilkan menggunakan echo.

1. Membuat $gambar yang mengambil data dari $item['gambar'].

* Cth : $gambar = $item['gambar'];

Artinya, $gambar mempunyai nilai yang sama dengan $item['gambar'], ['gambar'] diambil dari tblmenu dengan kolom gambar.

1. Mengatur update gambar dengan menggunakan if yang berisi apabila $temp yang berisi temporari dari file gambar kosong, maka akan mengambil $temp dari gambar yang sama pada $item['gambar'], tetapi jika terjadi perubahan gambar, maka akan menjalankan if untuk menampilkan $temp dan mengupload gambar.

* Cth : if (!empty($temp)) {

$gambar = $\_FILES['gambar']['name'];

move\_uploaded\_file($temp,'../upload/'.$gambar);

}

Artinya, apabila $temp tidak kosong / notempty, maka akan menjalankan $gambar yang berisi $\_FILES['gambar']['name'] dan memindahkan gambar ke dalam $temp yang berisi temporari, di dalam folder upload dan dengan nama sesuai nama $gambar yang dipilih.

1. Menambahkan objek $db berisi perintah untuk menjalankan update pada update.php.

* Cth : $sql = "UPDATE tblmenu SET idkategori=$idkategori,menu='$menu',gambar='$gambar',harga=$harga WHERE idmenu = $id";

$db->runSQL($sql);

Artinya, pada $sql terdapat perintah untuk update data di dalam tblmenu dengan mengatur SET idkategori bernilai sama dengan $idkategori. menu bernilai sama dengan $menu, gambar bernilai sama dengan $gambar, dan harga bernilai sama dengan $harga yang diubah dan mengatur WHERE sesuai dengan idmenu = $id dari data yang dipilih. Lalu $sql akan dsambungkan dengan $db yang berisi function untuk menjalankan perintah (runSQL) pada $sql.

1. Setelah melakukan update, maka halaman akan kembali pada select.php diatur dengan menggunakan header.

* Cth : header("location:?f=menu&m=select");

Artinya, halaman akan dikembalikan pada tampilan dari select.php / ?f=menu&m=select setelah melakukan perintah update.

1. Memperbaiki error pada gambar, karena pada $sql belum diberi perintah WHERE, jadi jika WHERE belum diatur maka setelah gambar dipilih, maka menu akan berubah tampilannya menjadi sama semua, tetapi apabila jika WHERE sudah diatur, maka menu akan berubah / dipudate sesuai dengan idmenu yang dipilih.

* Cth : $sql = "UPDATE tblmenu SET idkategori=$idkategori,menu='$menu',gambar='$gambar',harga=$harga WHERE idmenu = $id";

Artinya, pada $sql terdapat perintah untuk update data di dalam tblmenu dengan mengatur SET idkategori bernilai sama dengan $idkategori. menu bernilai sama dengan $menu, gambar bernilai sama dengan $gambar, dan harga bernilai sama dengan $harga yang diubah dan mengatur WHERE sesuai dengan idmenu = $id dari data yang dipilih agar menu yang diupdate sesuai dengan idmenu.

1. Menambahkan perintah untuk menghapus data / DELETE pada menu.

* Cth : if (isset($\_GET['id'])) {

$id=$\_GET['id'];

$sql = "DELETE FROM tblmenu WHERE idmenu=$id";

$db->runSQL($sql);

header("location:?f=menu&m=select");

}

Artinya, jika $\_GET['id'] di klik, maka $\_GET['id'] akan disimpan nilainya di dalam $id. Menjalankan $sql yang berisi perintah untuk menghoaus data / DELETE dari tblmenu yang ada di idmenu dengan nilai yang sama seperti $id. Setelah itu akan menjalankan $db->runSQL untuk menyambungkan ke dalam database. Apabila selesai menghapus data, maka akan kembali ke halaman select.php / ?f=menu&m=select dengan menggunakan header.

1. Membuat 3 kolom tambahan di dalam tblpelanggan, yaitu email, password, dan aktif untuk keterangan data dari pelanggan, sehingga data untuk keterangan pelanggan ada 7, idpelanggan, pelanggan alamat, telp, email, password, dan aktif.
2. Mengcopy code dari select.php kategori dan paste di dalam select.php yang ada di dalam pelanggan, dan mengatur semua teks yang bertuliskan kategori diganti menjadi pelanggan dengan menggunakan ctrl + shift + l.
3. Mengatur tampilan tabel dan menambahkan beberapa kolom dengan isi pelanggan, alamat, telp, email, dan mengganti tabel Update menjadi status.
4. Mengatur tabel status, apabila tabel berstatus aktif, maka akan memunculkan tulisan aktif, dan jika tidak aktif maka akan memunculkan tulisan nonaktif, pengaturan dilakukan menggunakan if.

* Cth : <?php

if ($r['aktif']==1) {

$status = 'AKTIF';

}else{

$status = 'NONAKTIF';

}

?>

Artinya, apabila $r['aktif'] nilainya sama dengan 1, maka akan menjalankan $status yang berisi AKTIF, dan jika $r tidak sama dengan 1, maka akan menjalankan $status dan memunculkan tulisan NONAKTIF. ['aktif'] diambil dari database pada kolom tblpelanggan yang mempunyai nama ['aktif'].

1. Mengambil nilai dari tabel yang dipilih menggunakan perintah SELECT dan menampilkan nilainya menggunakan var\_dump.

* Cth : $row = $db->getITEM("SELECT \* FROM tblpelanggan WHERE idpelanggan = $id");

var\_dump($row);

Artinya, di dalam $row terdapat objek $db yang memanggil function getITEM dengan peirntah memilih data / SELECT data dari tblpelanggan terdapat dalam idpelanggan yang mempunyai nilai sama dengan $id, lalu nilai dari $row akan ditampilkan menggunakan var\_dump.

1. Mengatur dan memberi perintah UPDATE di dalam kolom aktif pada tblpelanggan.

* Cth : $sql = "UPDATE tblpelanggan SET aktif=$aktif WHERE idpelanggan=$id";

echo $sql;

Artinya, di dalam $sql terdapat perintah UPDATE data pada tblpelanggan dan mengatur kolom aktif nilainya sama dengan $aktif sesuai dengan idpelanggan yang nilainya sama dengan $id, lalu $sql nilai dari $sql akan ditampilkan menggunakan echo.

1. Pada kolom aktif di tblpelanggan, terdapat 2 pilihan, yaitu 1 dengan nilai aktif, dan 0 dengan nilai nonaktif, apabila ingin mengupdate nilai aktif menjadi nonaktif dan nilai nonaktif menjadi nilai aktif maka mengaturnya menggunakan if.

* Cth : if ($row['aktif']==0) {

$aktif=1;

}else {

$aktif = 0;

}

Artinya, apabila $row['aktif'] bernilai 0, maka akan mengubah nilai dari 0/ nonaktif menjadi 1 / aktif, dan jika $row['aktif'] bernilai 1, maka akan mengubah nilai dari 1/ aktif menjadi 0 / nonaktif.

1. Menambahkan perintah untuk DELETE / menghapus data.

* Cth : $sql = "DELETE FROM tblpelanggan WHERE idpelanggan=$id";

Artinya, di dalam $sql terdapat perintah untuk DELETE data dari tblpelanggan yang di idpelanggan yang mempunyai nilai sama dengan $id.

1. Menjalankan perintah DELETE menggunakan if.

* Cth : if (isset($\_GET['id'])) {

$id=$\_GET['id'];

$sql = "DELETE FROM tblpelanggan WHERE idpelanggan=$id";

$db->runSQL($sql);

header("location:?f=pelanggan&m=select");

}

Artinya, apabila $\_GET['id'] di klik, maka nilai dari $\_GET['id'] akan disimpan di dalam $id, lalu menjalankan perintah DELETE yang ada di dalam $sql, dan disambungkan pada database melalui $db yang menyambungkan function runSQL untuk menyambungkan $sql ke dalam database dan menjalankan $sql. Apabila selesai melakukan delete data, maka tampilan akan kembali ke dalam halaman select.php yang ada di folder pelanggan / ?f=pelanggan&m=select.

1. Menambahkan tbluser di dalam database dan menambahkan beberapa data di dalam tbluser, yaitu iduser, nama, email, password, level, dan aktif.
2. Mengcopy code dari select.php kategori dan paste di dalam select.php yang ada di dalam user, dan mengatur semua teks yang bertuliskan kategori diganti menjadi user dengan menggunakan ctrl + shift + l.
3. Mengatur tabel status, apabila tabel berstatus aktif, maka akan memunculkan tulisan aktif, dan jika tidak aktif maka akan memunculkan tulisan banned, pengaturan dilakukan menggunakan if.

* Cth : <?php

if ($r['aktif']==1) {

$status = 'AKTIF';

}else{

$status = 'BANNED';

}

?>

Artinya, apabila $r['aktif'] nilainya sama dengan 1, maka akan menjalankan $status yang berisi AKTIF, dan jika $r tidak sama dengan 1, maka akan menjalankan $status dan memunculkan tulisan BANNED. ['aktif'] diambil dari database pada kolom tblpelanggan yang mempunyai nama ['aktif'].

1. Mengambil nilai dari tabel yang dipilih menggunakan perintah SELECT dan menampilkan nilainya menggunakan var\_dump.

* Cth : $row = $db->getITEM("SELECT \* FROM tbluser WHERE iduser= $id");

var\_dump($row);

Artinya, di dalam $row terdapat objek $db yang memanggil function getITEM dengan peirntah memilih data / SELECT data dari tbluser terdapat dalam tbluser yang mempunyai nilai sama dengan $id, lalu nilai dari $row akan ditampilkan menggunakan var\_dump.

1. Mengatur dan memberi perintah UPDATE di dalam kolom aktif pada tbluser.

* Cth : $sql = "UPDATE tbluser SET aktif=$aktif WHERE iduser=$id";

echo $sql;

Artinya, di dalam $sql terdapat perintah UPDATE data pada tbluser dan mengatur kolom aktif nilainya sama dengan $aktif sesuai dengan iduser yang nilainya sama dengan $id, lalu $sql nilai dari $sql akan ditampilkan menggunakan echo.

1. Pada kolom aktif di tbluser, terdapat 2 pilihan, yaitu 1 dengan nilai aktif, dan 0 dengan nilai nonaktif, apabila ingin mengupdate nilai aktif menjadi nonaktif dan nilai nonaktif menjadi nilai aktif maka mengaturnya menggunakan if.

* Cth : if ($row['aktif']==0) {

$aktif=1;

}else {

$aktif = 0;

}

Artinya, apabila $row['aktif'] bernilai 0, maka akan mengubah nilai dari 0/ nonaktif menjadi 1 / aktif, dan jika $row['aktif'] bernilai 1, maka akan mengubah nilai dari 1/ aktif menjadi 0 / nonaktif.

1. Apabila selesai melakukan update data maka tampilan web akan kembali pada select.php / ?f=user&m=select.

* Cth : header("location:?f=user&m=select");

1. Menjalankan perintah DELETE menggunakan if.

* Cth : if (isset($\_GET['id'])) {

$id=$\_GET['id'];

$sql = "DELETE FROM tbluser WHERE iduser=$id";

$db->runSQL($sql);

header("location:?f=pelanggan&m=select");

}

Artinya, apabila $\_GET['id'] di klik, maka nilai dari $\_GET['id'] akan disimpan di dalam $id, lalu menjalankan perintah DELETE yang ada di dalam $sql, dan disambungkan pada database melalui $db yang menyambungkan function runSQL untuk menyambungkan $sql ke dalam database dan menjalankan $sql. Apabila selesai melakukan delete data, maka tampilan akan kembali ke dalam halaman select.php yang ada di folder pelanggan / ?f=pelanggan&m=select.

1. Membuat file bernama insert.php pada folder user untuk tempat menambah data.
2. Mengcopy code dari insert.php yang ada di dalam kategori dan dipaste di dalam insert.php yang ada di dalam user, dan menambahkan beberapa kolom tabel yaitu, nama user, email, password, konfirmasi pasword, dan menambahkan select option dengan list, admin, koki, dan kasir.
3. Menambahkan perintah INSERT INTO untuk menambahkan data.

* Cth : if (isset($\_POST['simpan'])) {

$user = $\_POST['user'];

$email = $\_POST['email'];

$password = $\_POST['password'];

$konfirmasi = $\_POST['konfirmasi'];

$level = $\_POST['level'];

$sql = "INSERT INTO tbluser VALUES ('','$user','$email','$password','$level',1)";

echo $sql;

}

Artinya, apabila $\_POST['simpan'] di klik, maka akan menjalankan, nilai dari $\_POST['user'] akan disimpan di dalam $user, nilai dari $email = $\_POST['email'] akan disimpan di dalam $email, nilai dari $\_POST['password'] akan disimpan di dalam $password, nilai dari $\_POST['konfirmasi'] akan disimpan di dalam $konfirmasi, dan nilai dari $level = $\_POST['level'] akan disimpan di dalam $level. Setelah itu akan menjalankan $sql yang terdapat perintah INSERT INTO tbluser dengan values $user,$email,$password,$level, dan 1, karena kolom aktif bernilai 1. Dan $sql akan ditampilkan menggunakan echo.

1. Mengatur isi dari $password harus sama dengan $konfirmasi menggunakan if.

* Cth : if ($password === $konfirmasi) {

$sql = "INSERT INTO tbluser VALUES ('','$user','$email','$password','$level',1)";

echo $sql;

}else {

echo "<h4>PASSWORD TIDAK SAMA</h4>";

}

Artinya, jika $password bernilai sama dengan $konfirmasi, maka akan menjalankan perintah INSERT INTO tbluser dengan values $user,$email,$password,$level, dan 1, karena kolom aktif bernilai 1. Dan $sql akan ditampilkan menggunakan echo. Tetapi jika $password != / tidak sama dengan $konfirmasi, maka akan menampilkan tulisan PASSWORD TIDAK SAMA.

1. Session pada PHP digunakan untuk menyimpan variabel dan variabel tersebut tidak akan hilang selama browser tetap menyala dan selama session tidak dihapus.
2. Sebelum digunakan, session harus diaktifkan terlebih dahulu, yaitu menggunakan session\_start yang berfungsi untuk memulai session dan menyimpannya di dalam browser.
3. Menambahkan pengaturan apabila sebelum masuk ke dalam website harus menggunakan login form terlebih dahulu, pengaturan dilakukan menggunakan if.

* Cth : if (!isset($\_SESSION['user'])) {

header("location:http://localhost/phpsmk/restoran/admin/login.php");

}

Artinya, apabila $\_SESSION['user'] belum di klik, maka tampilan halaman akan diarahkan di dalam tampilan login.php di dalam admin.

1. Menyambungkan link bootstrap ke dalam file login.php.
2. Mengganti label, type, dan name di dalam form menjadi email dalam login.php, dan mengganti value dari <input> submit menjadi login.
3. Menambahkan form yang mempunyai label PASSWORD dan mengatur type, name menjadi paaword, serta menambahkan placeholder yang berisi "isi password!".
4. Membuat 2 variabel baru, yaitu $email untuk menyimpan data dari $\_POST['email'] dan $password untuk menyimpan data dari $\_POST['password'].
5. Membuat if yang digunakan untuk mengatur apabila $\_POST['login'] di klik, maka akan mengambil nilai dari data yang dimasukkan

* Cth : if (isset($\_POST['login'])) {

$email = $\_POST['email'];

$password = $\_POST['password'];

echo $email.' -- '.$password;

}

Artinya, apabila $\_POST['login'] di klik, maka data dari $\_POST['email'] akan disimpan di dalam $email, dan data dari $\_POST['password'] akan disimpan di dalam $password. Nilai dari $email dan $password akan ditampilkan menggunakan echo.

1. Menambahkan perintah untuk memilih data / SELECT di dalam $sql.

* Cth : $sql = "SELECT \* FROM tbluser WHERE email = '$email' AND $password";

echo $sql;

Artinya, di dalam $sql terdapat perintah untuk memilih data / SELECT data dari tbluser pada kolom email yang nilainya sama dengan $email dan $password. Maka data $sql akan ditampilkan menggunakan echo.

1. Menyambungkan login.php dengan db.control menggunakan require\_once, menyambungkan objek $db dan menambahkan session\_start();

* Cth : <?php

session\_start();

require\_once "../dbcontrol.php";

$db = new DB;

?>

Artinya, di dalam <php> terdapat session\_start dan memanggil semua isi dari file dbcontrol.php serta menyambungkan objek $db.

1. Membuat $count yang memanggil function rowcount dari $db.

* Cth : $sql = "SELECT \* FROM tbluser WHERE email = '$email' AND '$password'";

$count = $db->rowCOUNT($sql);

Artinya, pada $sql terdapat perintah untuk memilih data / SELECT data dari tbluser pada kolom email yang nilainya sama dengan $email dan $password. Lalu $count terdapat perintah untuk memanggil function dari $db yaitu rowCOUNT, sehingga apabila dijalankan maka setelah memasukkan email dan password maka akan diberikan hasil bahwa email yang dimasukkan jika berada di dalam database email tersebut aktif atau tidak, dan jika tidak ada di dalam database maka akan muncul 0 / tidak aktif.

1. Pembuatan function rowCOUNT yang digunakan untuk memanggil hasil data.
2. Mengatur tampilan apabila $email dan $password salah, maka akan muncul pesan bahwa email / password error, tetapi jika benar maka akan menjalankan $count.

* Cth : if ($count == 0) {

echo "<h4>EMAIL ATAU PASSWORD SALAH </h4>";

}else {

echo $count;

}

Artinya, apabila $count bernilai 0, maka akan menampilkan pesan EMAIL ATAU PASSWORD SALAH, tetapi apabila $count tidak bernilai 0 / bernilai 1, maka akan menampilkan / menjalankan $count yang berisi perintah menampilkan data aktif / tidak pada tbluser.

1. Melakukan pengaturan apabila $count tidak bernilai 0 maka akan menjalankan perintah memilih data / SELECT.

* Cth : $sql = "SELECT \* FROM tbluser WHERE email = '$email' AND '$password'";

$row = $db->getITEM($sql);

var\_dump($row);

Artinya, di dalam $sql terdapat perintah untuk memilih data / SELECT data pada tbluser di dalam kolom email yang nilainya sama dengan $email dan $password. Lalu $row akan menjalankan perintah objek $db yaitu geiITEM untuk menampilkan item $sql. Nilai dari $row akan ditampilkan menggunakna var\_dump.

1. Membuat session yang berisi nilai yang sama seperti $row.

* Cth : $\_SESSION['user'] = $row['email'];

$\_SESSION['level'] = $row['level'];

Artinya, $\_SESSION['user'], terdapat session dengan nama user yang mengambil nilai / data yang sama pada $row['emal'], dan $\_SESSION['level'], terdapat session dengan nama level yang mengambil nilai / data yang sama pada $row['level'].

1. Menambahkan hyperlink pada logout.

* Cth : <div class="float-right mt-4"><a href="?log=logout">Logout</a></div>

Artinya, di dalam logout terdapat variabel ?log yang berisi ?log=logout, jadi ketika link di klik, maka url akan menambahkan ?log=logout.

1. session\_destroy digunakan untuk menghapus semua session yang disimpan di session daat ini, sehingga $\_SESSION akan kosong karena data dari $\_SESSION saat ini sudah dihapus oleh session\_destroy.
2. Mengosongkan session sehingga apabila logout di klik maka akan diarahkan untuk login kembali.

* Cth : if (isset($\_GET['log'])) {

session\_destroy();

header("location:index.php");

}

Artinya, apabila $\_GET['log'] di klik, maka semua session yang ada saat ini akan dihapus sehingga apabila mengklik $\_GET['log'] maka akan diarahkan ke dalam index.php yang sebelum masuk ke akan diperiksa apakah $\_SESSION['user'] sudah diisi atau belom, jika belum diisi maka dalam index.php harus dilakukan login terlebih dahulu.

1. Update user digunakan untuk proses update dari user yang login.
2. Menambahkan session baru bernama iduser yang nilainya sama dengan $row['iduser'].

* Cth : $\_SESSION['iduser'] = $row['iduser'];

Artinya, terdapat session baru yang bernama iduser dan mempunyai nilai yang sama dengan $row['iduser'].

1. Meletakkan isi dari $\_SESSION['user'] di sebelah User.

* Cth : <div class="float-right mt-4 mr-4">User : <?php echo $\_SESSION['user'] ?></div>

Artinya, terdapat div User yang tempatnya di pojok kanan dengan margin-top 4 dan margin-right 4, jika ditampilkan, pada sebelah User akan ditampilkan juga $\_SESSION['user'].

1. Menambahkan file updateuser.php dan menambahkan hyperlink ke dalam $\_SESSION['user'], yang apabila $\_SESSION['user'] di klik, maka di dalam url akan memunculkan link ?f=user&m=updateuser&id=(yang sama seperti data yang dipilih)

* Cth : <div class="float-right mt-4 mr-4">User : <a href="?f=user&m=updateuser&id=<?php echo $\_SESSION['iduser']?>"> <?php echo $\_SESSION['user'] ?></a></div>

Artinya, terdapat div User yang tempatnya di pojok kanan dengan margin-top 4 dan margin-right 4, jika ditampilkan, pada sebelah User akan ditampilkan juga $\_SESSION['user'] yang di dalamnya terdapat link untuk menhubungkan ke dalam file updateuser.php, dan pada url akan tertuliskan nama file juga id sesuai dari data yang dipilih.

1. Mengcopy semua code dari insert.php yang ada di user dan mengcopas nya ke dalam updateuser.php.
2. Mengganti placeholder yang ada di tabel menjadi value, yang setiap value isinya akan dipanggil sesuai dengan user, email, dan password, pemanggilan menggunakan $row.

* Cth : <input type="text" name="user" required value="<?php echo $row['user']?>" class="form-control" >

Pada input tersebut mempunyai name user dan mempunyai value $row['user'], jadi pada kolom tersebut akan ditampillkan isi dari $row['user'].

1. Mengatur list pada option agar otomatis mengikuti level dari user ketika dimasukkan, pengaturan dilakukan di dalam <option> lalu diberi if.

* Cth : <option value="admin"<?php if($row['level'] === "admin") echo "selected" ?> >admin</option>

Artinya, apabila <option> yang mempunyai $row['level'] mempunyai nilai yang sama dengan admin, maka list tersebut yang akan diselected / otomatis ditampilkan.

1. Menambahkan perintah untuk update data / UPDATE di dalam $sql.

* Cth : if ($password === $konfirmasi) {

$sql = "UPDATE tbluser SET user='$user',email='$email',password='$password',level='$level' WHERE iduser = $id";

$db->runSQL($sql);

header("location:?f=user&m=select");

}else {

echo "<h4>PASSWORD TIDAK SAMA</h4>";

}

}

Artinya, apabila $pqssword nilainya sama dengan $konfirmasi, maka akan menjalankan $sql yang di dalamnya terdapat periintah untuk UPDATE data pada tbluser dengan mengatur user yang nilainya sama dengan $user, email yang nilainya sama dengan $email, password yang nilainya sama dengan $password, level yang nilainya sama dengan $level, yang pengaturannya sesuai dengan iduser = $id. Lalu akan menjalankan objek $db yang mempunyai function menjalankan sql, sehingga perubahan akan terhubung ke dalam database, dan aapabila selesai mengupdate data maka halaman akan kembali ke select.php dalam user , tetapi jika password tidak sama, maka akan menampilkan tulisan PASSWORD TIDAK SAMA.

1. Menambahkan pengaturan menggunakan if jika tabel kosong, maka tidak terjadi error.

* Cth : <?php if(!empty($row)){ ?>

<?php foreach($row as $r): ?>

<tr>

<td><?php echo $no++ ?></td>

<td><?php echo $r['kategori'] ?></td>

<td><a href="?f=kategori&m=delete&id=<?php echo $r['idkategori'] ?>">Delete</a></td>

<td><a href="?f=kategori&m=update&id=<?php echo $r['idkategori'] ?>">Update</a></td>

</tr>

<?php endforeach ?>

<?php } ?>

Artinya, apabila $row tidak kosong, maka akan menjalankan dan menampilkan kolom tabel yang ada di php foreach.

1. Membuat file index.php di folder restoran dan mencopy code yang ada di dalam index, lalu mengganti dan menghapus beberapa hyperlink.
2. Menambahkan beberapa menu pada navbar, yaitu login, pelanggan, dan daftar.
3. Menambahkan $sql pada index.php restoran yang berisi perintah untuk SELECT data.

* Cth : $sql = "SELECT \* FROM tblkategori ORDER BY kategori ASC";

$row = $db->getALL($sql);

var\_dump($row);

Artinya, pada $sql terdapat perintah untuk memilih data / SELECT data dari tblkategori yang diurutkan dari nilai kecil ke besar sesuai data yang ada pada kategori. $row akan menyambungkan objek $db yang berisi function untuk getALL data pada $sql, dan untuk mengetahui nilai dari $row maka dipanggil menggunakan var\_dump.

1. Mengatur agar tampilan dari <li> tidak error jika kosong menggunakan if dan mengatur agar data yang ditampilkan sesuai dengan kategori pada tblkategori, menggunakan foreach.

* Cth : <?php if (!empty($row)) {?>

<ul class="nav flex-column">

<?php foreach($row as $r): ?>

<li class="nav-item"><a class="nav-link" href="#"><?php echo $r['kategori'] ?></a></li>

<?php endforeach ?>

</ul>

<?php } ?>

Artinya, apabila $row tidak kosong, maka akan menjalankankan <ul> berisi <li> yang di dalamnya akan melakukan pengulangan / foreach pada $row sebagai $r, dan di dalam <li> akan $r akan dipanggil sesuai dengan yang ada pada kategori.

1. Membuat folder home yang berisi file produk.php untuk menampilkan isi dari produk dan mengcopy semua isi yang ada di dalam insert.php menu dan mempaste di dalam file produk.php.
2. Mengatasi error gambar tidak muncul pada produk .php, dan untuk mengatasinya dengan memperbaiki path dari pemanggilan file.
3. Menyambungkan isi dari produk.php ke dalam index.php restoran dengan if else.

* Cth : if (isset($\_GET['f']) && (isset($\_GET['m']))) {

$f = $\_GET['f'];

$m = $\_GET['m'];

$file = $f.'/'.$m.'.php';

require\_once "$file";

}else{

require\_once "home/produk.php";

}

Artinya, apabila $\_GET['f'] dan $\_GET['m'] di klik maka nilai dari $\_GET['f'] akan disimpan di dalam $f, dan nilai dari $\_GET['m'] akan disimpan di dalam $m , dan akan menjalankan $file yang berisi $f./.$m atau nama dari file, dan semua isi dari $file akan dipanggil menggunakan require\_once, apabila $\_GET['f'] dan $\_GET['m'] tidak di klik maka akan menjalankan else yaitu memanggil semua yang ada di dalam produk.php.

1. Mengatur tampilan menu menjadi seperti card dengan mengcopy code dari bootstrap, mengedit tampilan dari gambar menjadi width:200px, dan menambahkan css yaitu width car 15rem, float:left, dan memberi margin:10px antar card
2. Style="clear:both" berfungsi untuk tidak mengizinkan elemen menggunakan float, float-left atau float-right.
3. Menambahkan id pada URL dengan menambahkan variabel.

* Cth : <a class="nav-link" href="?f=home&m=produk&id=<?php echo $r['idkategori']?>">

Artinya, terdapat link dengan mengatur f=home m=prdouk dan id sesuai dengan $r['idkategori'].

1. Menampilkan menu sesuai dengan kategori dengan menambahkan $where

* Cth : if (isset($\_GET['id'])) {

$id = $\_GET['id'];

$where = "WHERE idkategori = $id";

echo $where;

}else {

$where = "";

}

Artinya, apabila $\_GET['id'] di klik, maka nilai dari $\_GET['id'] akan disimpan di dalam $id, dan isi dari $where adalah perintah menunjukkan idkategori nilainya sama dengan $id, lalu akan ditampilkan menggunakan echo. Apabila $\_GET['id'] belum di klik, maka akan menjalankan else yang berisi $where nilainya kosong.

1. Menambahkan $where pada perintah yang disambungkan dalam database.

* Cth : $sql = "SELECT \* FROM tblmenu $where ORDER BY menu ASC LIMIT $mulai,$banyak";

Artinya, pada $sql terdapat perintah untuk memilih data / SELECT data dari tblmenu dan $where diurutkan dari nilai kecil ke besar dengan limit sesuai dengan $mulai, dan $ banyak.

1. Menambahkan link untuk mengembalikan halaman ke tampilan awal setelah melihat kategori.

* Cth : <a href="index.php">Restoran Vocational-School</a>

Artinya, apabila restoran vocational-school di klik maka akan mengarahkan ke halaman index.php

**Saya Belum Mengerti**