|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| PROJECT PHP OOP minggu ke 16 (video 51-55 pemrograman web php) | 95 |

**Saya Sudah Belajar dan Mengerti dan Saya BISA**

1. Membuatan Admin Page (Back End) pada bagian Menampilkan Order Pembelian. Jadi histori atau daftar orderan dari pelanggan akan masuk ke dalam navbar order pada back end atau yang hanya bisa dilihat oleh admin. Karena ketika login pelanggan bersamaan dengan login admin terjadi eror karena isi session ada 5 indeks array dan baru dideklarasikan untuk 2 indeks yaitu pelanggan dan idpelanggan maka di dalam file-file yang memiliki function berisi foreach session yang berisi if diberi tambahan setelah $key <> ‘pelanggan’ && (dan seterusnya sesuai nama key arraynya). Untuk mengetahui apa saja array sessionnya bisa dilihat melalui var\_dumb($\_SESSION); di dalam functionnya. Jadi dia akan menggunakan session untuk login admin(backend) dan login pelanggannya(frontend).

2. Setelah mengatur pada bagian login, selanjutnya di dalam navbar atau folder order yang ada di backend pada bagian file select.php bisa diisi dengan mengcopas file histori dari bagian frontend. Setelah mengcopas tentu ada beberapa bagian yang harus ditambah dan diubah, sebelumnya tambahkan kolom untuk pelanggan yang nanti barisnya berisi $r[‘pelanggan’] atau sesuai dengan nama pelanggan yang melakukan order, selanjutnya ubah kolom detail diubah dengan kolom bayar dengan td nya: <?php echo $status; ?>. karena $status belum diisi maka sebelum <tr> dibuat sebuah php yang berisi

<?php

if ($r['status'] == 0) {

                    $status = '<td><a href="?f=order&m=bayar&id=' . $r['idorder'] . '">Bayar</a></td>';

                } else {

                    $status = '<td>LUNAS</td>';

                }

?>

Jadi kolom ini akan menampilkan hyperlink bayar apabila belum membayar atau statusnya 0, dan apabila sudah melakukan pembayaran atau statusnya tidak sama dengan 0 maka baris pada kolom ini akan muncul tulisan LUNAS. Untuk daftar ordernya yang ditampilkan bukan sesuai dengan satu pelanggan tertentu saja tetapi untuk semua daftar order dari semua pelanggan akan ditampilkan di order backend ini. Untuk itu pada bagian $jumlahdata yang berisi db->rowCOUNT(“SELECT … ”) hilangkan WHERE email=’$email’ nya agar dia tidak menampilkan untuk pelanggan tertentu saja dan hapus juga $email yang ada diatas $jumlahdata. Selain di $jumlahdata pada bagian lainnya yang memiliki WHERE email dihilangkan saja, lalu pada $sql nya ubaah ORDER BY tglorder DESC menjadi ORDER BY status,idorder ASC agar dia menampilkan baris orderan sesuai status nya dengan diurutkan mulai dari 0 berarti diurutkan mulai dari yang atas sendiri yaitu status bayar ke status lunas dan diurutkan dari idordernya juga. Terakhir pada bagian page halaman ubah f=home menjadi f=order, m=order menjadi m=select agar dia menampilkan sesuai dengan halaman order bagian file select.php dengan urutan baris sesuai data orderannya.

3. Pembuatan Admin Page (Back End) pada bagian Pembayaran Order. Untuk mengatur bagian ini sebelumnya buat sebuah file di dalam folder order yang bernama bayar.php. di dalam file ini bisa diisi dengan mengcopas file update.php dari folder kategori. Setelah mengcopas ubah judul <h3> nya menjadi Pembayaran Order lalu nonaktifkan atau comment runSQL dan headernya agar ketika dalam proses masih mencoba tidak langsung terkirim ke tabel yang salah. Di dalam if isset $\_GET[‘id’] ganti isi $sql nya FROM tblorder bukan tblkategori lagi dan WHERE idorder bukan idkategori agar dia mengambil item dari tabelorder dengan menyesuaikan idordernya.

4. Setelah mengatur awal pada file bayar.php selanjutnya (masih di dalam file ini) pada bagian form nya seperti ini:

<div class="form-group">

    <form action="" method="POST">

        <div class="form-group w-50">

            <label for="">Total</label>

            <input type="number" name="total" required value="<?php echo $row['total'] ?>" class="form-control">

        </div>

        <div class="form-group w-50">

            <label for="">Bayar</label>

            <input type="number" name="bayar" required class="form-control">

        </div>

        <div>

            <input type="submit" name="simpan" value="Bayar" class="btn btn-primary">

        </div>

    </form>

</div>

Jadi di dalam form itu terdapat 2 form untuk input bertype angka sebagai input dari total orderan dan berapa jumlah yang dibayar pelanggan nya lalu 1 input bertype submit untuk mengambil tindakan dari 2 form yang sudah terisi yang nantinya akan di if isset $\_POST. Di dalam if isset $\_POST dibawah form ubah menjadi seperti ini:

if (isset($\_POST['simpan'])) {

    $bayar = $\_POST['bayar'];

    $kembali = $bayar - $row['total'];

    $sql = "UPDATE tblorder SET bayar= $bayar, kembali=$kembali, status=1 WHERE idorder=$id";

    if ($kembali < 0) {

        echo "<h3>Pembayaran Kurang</h3>";

    } else {

        $db->runSQL($sql);

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

    }

}

Jadi ketika menekan submit bernama simpan, dia akan mengupdate ke tabel order, untuk kolom bayar akan diisi sesuai dengan jumlah pembayaran yang diberikan pelanggan, untuk kolom kembali akan diisi sesuai dengan hasil pembayaran dikurangi total tagihannya (sebagai jumlah uang kembalian), untuk kolom status isi dengan 1 agar nanti setelah bayar akan berubah menjadi lunas, untuk baris yang dipilih sesuai dengan id pelanggan yang melakukan pembayaran. Apabila uang kembaliannya dibawah 0 atau negative misalkan tagihannya 10.000 dan bayar nya hanya 5.000 maka 5.000-10.000 = -5.000 dan negative 5000 berada di bawah 0 maka akan muncul tulisan Pembayaran Kurang dan apabila kembalian tidak dibawah 0 atau uang yang dibayar lebih dari total tagihan maka dia akan menjalankan sqlnya dengan mengupdate tabel lalu mengembalikan ke halaman order backend. Setelah mengatur bagian itu tadi selanjutnya kembali lagi ke file select.php dan ubah tabel ordernya dengan menambahkan kolom untuk Bayar dan Kembali, td bayar berisi $r[‘bayar’] dan td kembali berisi $r[‘kembali’]. Jadi apabila belum melakukan pembayaran kolom bayar dan kembali akan menampilkan 0 dan apabila sudah akan menampilkan jumlah sesuai dengan bayar dan kembaliannya.

5. Pembuatan Admin Page (Back End) pada bagian Detail Pembelian. Jadi pada bagian ini bertujuan untuk mengetahui detail pembelian dari orderannya, jadi nanti ambil datanya dari vorderdetail dalam phpmyadmin. Di dalam folder orderdetail file select.php akan diisi dengan mengcopas file select.php yang ada di folder order.

6. Setelah mengcopas file select.php dari folder order, ubah judulnya terlebih dahulu yaitu Detail Pembelian, lalu pada $jumlahdata yang berisi select from ubah vorder nya menjadi vorderdetail karena mengambil dari vorderdetail dan ubah idorder nya menjadi idorderdetail karena menggunakan id yang ada pada vorderdetail. Lalu pada bagian $sql yang berfungsi untuk mengambil data dari vorder ubah juga menjadi vorderdetail dan diurutkan atau ORDER BY nya berdasarkan idorderdetail dengan urutan DESC atau dari yang paling terakhir membeli akan ditampilkan dinomer 1. Pada bagian tabel nya ubah kolomnya menjadi No, Pelanggan, Tanggal, Menu, Harga, Jumlah, Total, Alamat. Untuk th pada tabel bisa diisi dengan tulisan berupa string sebagai judulnya, sedangkan pada td nya menggunakan missal $r[‘pelanggan’] apabila th nya untuk pelanggan, karena $r ini sudah diatur untuk mengambil data dari tblvorderdetail pada bagian kolom-kolom sesuai nama yang diletakkan di dalam kurung staplesnya. Pada bagian Total ini berbeda, karena total merupakan hasil perkalian antara jumlah dan harga maka td nya: <td><?php echo $r['jumlah'] \* $r['harga'] ?></td>. Agar tampilan tabel tidak hanya menampilkan 2 baris isi disetiap kolomnya maka di dalam variable banyak angka 2 diubah menjadi angka 4 agar menampilkan 4 baris disetiap kolomnya. Apabila ukuran tabelnya terlalu kecil, maka width nya bisa diubah dari w-50 menjadi w-70. Setelah itu dibawah tabel akan dibuatkan sebaris yang digunakan untuk menghitung total pembelian dari total-total baris diatasnya atau kata lainnya membuat Grand Total, letak baris ini berada dalam satu php foreach dengan tr td tetapi dibuatkan tr td sendiri untuk grand total agar selalu ada dibagian bawahnya dan di dalam td grand total ini ditambahi h3 agar tulisannya terlihat besar dari baris-baris lainnya. Agar panjang kolom grand total ini menyambungkan 6 kolom dan nantinya jumlah grand total akan tepat dibawah total, maka di dalam td grand total ini ditambahkan colspan=”6”. Masih di dalam tr dari grand total, selanjutnya ditambahkan td untuk bagian jumlah totalnya, codenya: <td><h4><?php echo $total ?></h4></td> yang berarti dia akan menampilkan isi dari $total. Karena $total ini belum dibuatkan sebelumnya, maka di dalam php yang paling atas sendiri diketikkan $total=0; agar dimulai dari awal lalu di bawah td dari no, pelanggan, alamat, dll. Dibuatkan php yang isinya seperti ini: $total = $total + ($r['jumlah'] \* $r['harga']); yang berarti jumlah total untuk baris pertama akan ditambahkan dengan 0 dan menjadi $total dan selanjutnya akan melooping total pada baris kedua lalu ditambahkan dengan $total yang sudah terisi tadi dan menjadi $total lagi dan begitu seterusnya hingga baris yang ada pada tabel tiap halaman berakhir. Setelah itu membuat sebuah form yang digunakan untuk mencari order berdasarkan tanggalnya, mula-mula judulnya dipindahkan keatas php, lalu dibawah judul ditambahkan div from yang dicopas dari insert.php yang ada dalam folder kategori, setelah itu div class yang berisi input type text dicopas ke bawahnya, jadi ada 2 input text setelah itu type text ini diubah menjadi date agar berfungsi untuk mengatur tanggal, untuk label pada input date yang pertama diberi nama Tanggal Awal, dan input date yang kedua diberi nama Tanggal Akhir lalu untuk name dari tanggal awal berupa tawal dan name dari tanggal akhir berupa takhir. Selanjutnya untuk placeholder dari kedua input date tadi dihapus saja. Setelah mengatur input date selanjutnya input submit diganti value nya saja yaitu Cari. Jika sudah mengatur form nya selanjutnya adalah membuat if isset $\_POST untuk formnya, letaknya ada dibawah $sql yang codenya seperti ini:

if (isset($\_POST['simpan'])) {

    $tawal = $\_POST['tawal'];

    $takhir = $\_POST['takhir'];

    $sql = "SELECT \* FROM vorderdetail WHERE tglorder BETWEEN '$tawal' AND '$takhir'";

    // echo $sql;

}

Jadi dia akan mengambil tindakan setelah menekan submit bernama simpan dengan mempost $tawal dan $takhir yang dipilih lalu $sql akan mencari data yang tanggalnya berada di antara tanggal awal dan tanggal akhir atau mulai tanggal awal ke tanggal akhir yang dipilih dari vorderdetail.

7. Pembuatan Admin Page (Back End) pada bagian Pembuatan Level Akses. Jadi admin bisa membuka bagian-bagian tertentu sesuai dengan level adminnya, misalkan koki hanya bisa membuka bagian order detail saja dan kasir bisa membuka bagian order dan order detail saja. Pertama di dalam file index.php yang ada di folder admin ubah judul Restoran menjadi Admin Page dan beri hyperlink yang alamat link nya menuju index.php atau tampilan awal index.php dan div class nya tambahkan mt-4 agar berjarak dengan bagian atas nya. setelah itu div class dibawahnya, didalamnya tambahkan div class untuk navbar yang tampil disebelah kiri user, div class ini bisa dicopas dari div classnya logout dan diletakkan dibawahnya div class user lalu diberi nama Level: lalu diberi php yang codenya: <?php echo $\_SESSION['level'] ?> Jadi dia akan menampilkan level sesuai dengan akun yang digunakan masuk.

8. Setelah mengubah itu tadi, selanjutnya adalah membagi menu apa saja yang akan tampil sesuai dengan levelnya. Apabila level admin akan tampil semua menu, apabila level kasir akan menampilkan menu order dan orderdetail saja, apabila level koko hanya akan menampilkan menu orderdetail saja. Untuk membagi-bagi halaman ini di dalam div class nya ul li atau menu-menu, diatas li menu ditambahkan sebuah php yang berisi $level = $\_SESSION[‘level’]; yang berarti $level akan diisi sesuai level akun yang digunakan untuk login. Lalu dibawah $level adalah switch($level) untuk membagi-bagi atau mengelompokkan menu apa saja yang akan ditampilkan sesuai dengan level yang digunakan, case pertama untuk admin (case’admin’:) yang isinya

echo '

<li class="nav-item"><a class="nav-link active" href="?f=kategori&m=select">Kategori</a></li>

 <li class="nav-item"><a class="nav-link active" href="?f=menu&m=select">Menu</a></li>

 <li class="nav-item"><a class="nav-link active" href="?f=pelanggan&m=select">Pelanggan</a></li>

 <li class="nav-item"><a class="nav-link active" href="?f=order&m=select">Order</a></li>

 <li class="nav-item"><a class="nav-link active" href="?f=orderdetail&m=select">Order Detail</a></li>

 <li class="nav-item"><a class="nav-link active" href="?f=user&m=select">User</a></li>

 ';

break;

Yang berarti apabila level nya admin dia akan menampilkan semua menu itu, li ini diambil dari li sebelumnya. Lalu case kedua untuk kasir (case ’kasir’:) yang berisi li order dan orderdetail, dan case ketiga untuk koki (case ‘koki’:) yang berisi li orderdetail saja. Setelah li dipindahkan ke case, maka li yang lama atau yang diluar case dihapus saja agar tidak ikut tampil ke halaman. Setelah 3 case untuk admin, kasir, dan koki sudah terisi selanjutnya bagian default diisi echo ‘Tidak Ada Menu’; dengan begitu apabila level tidak sesuai dengan 3 level yang ada maka dia akan menampilkan tulisan Tidak Ada Menu.

9. Pembuatan Admin Page (Back End) pada bagian Enkripsi Password. Bagian ini adalah bagian yang terakhir dari tutorial pembuatan aplikasi restoran. Jadi Enkripsi ini digunakan untuk menyamarkan password dari password awal menjadi sebuah code, jadi ketika melihat password di dalam database yang tampil adalah sebuah code bukan password aslinya. Untuk itu di dalam inset.php dalam folder user, pada bagian $password = ditambahkan hash(‘sha256’, $\_POST[‘simpan’]) menggunakan sha256 karena varchar yang digunakan di dalam passwordnya adalah 255 jadi digunakan sha256 itu tadi, karena bagian $konfirmasi harus sama dengan $password atau $password === $konfirmasi, maka $konfirmasi juga diubah menjadi hash(‘sha256’, $\_POST[‘simpan’]) juga. Setelah itu di dalam file login.php yang ada di dalam folder admin pada bagian $passwordya juga ditambahkan hash(‘sha256’, $\_POST[‘password’]) agar ketika login dia bisa masuk karena $password yang ada di dalam file login.php juga sudah ditujukan untuk password terenkripsi. Selanjutnya di dalam database tbluser bisa dihapus semua datanya keculai admin, lalu menginsert data lagi lewat halaman browser yang sudah disediakan di dalam menu user yang hanya bisa dibuka admin dengan membuat akun untuk koki dan kasir agar password nya terenkripsi dari password asli menjadi sebuah code-code di dalam database nya. Karena password admin ini belum terenkripsi passwordnya, maka akun admin di hapus dulu lalu di dalam vscode untuk case koki bisa ditambahkan menu user dulu sehingga dari akun koki bisa menambahkan data untuk user adminnya agar password admin ikut terenkripsi. Apabila sudah ada akun admin dan passwordnya terenkripsi maka di bagian case koki untuk menu usernya dihapus saja. Dengan begitu semua akun sudah terenkripsi passwordnya dan aplikasi restoran sudah jadi.

**Saya Belum Mengerti**

1.