|  |  |
| --- | --- |
| Materi | Nilai |
| PHP- CODEIGNITER minggu 20 | **80** |

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

1. Saya sudah mengerti bahwa dalam pembuatan view order kita harus memastikan terlebih dahulu bahwa idpelanggan dalam database sudah sesuai dan sudah sama.
2. Saya sudah bisa memanggil view kemudian menampilkannya di halaman browser, untuk memanggilnya kit abisa menggunakan code **echo view(‘namafolder/namafile’);** . contohnya jika saya ingin memanggil file select di dalam folder order **echo view(‘order/select’);** .
3. Saya sudah bisa membuat tabel dengan menggunakan tag table ( **<table>** ) dari html, kemudian menggabungkannya dengan **class=”table”** dari bootstrap supaya terlihat lebih rapi. Saya juga bisa menambah kolom pada tabel.
4. Dengan menggunakan perintah **ASC** pada perintah **“SELECT \* FROM vorder ORDER BY STATUS ASC”** saya bisa mengurutkan data mulai dari yang terkecil baru yang terbesar.
5. Untuk menghubungkan ke database terdapat dua cara yaitu dengan menggunakan model kemudian yang kedua yaitu dengan menggunakan perintah langsung dengan query builder2 , misalnya **$sql=”SELECT \* FROM…”;** . Keduanya memiliki fungsi yang sama tapi untukpenggunaannya kita bisa melakukan yang mana saja dan tentunya sesuai kebutuhan project yang kita buat.
6. Saya sudah mengerti bahwa perPage dalam program pagination merupakan suatu variabel yang berguna untuk menampilkan dan menentukan jumlah page dalam satu halaman.
7. Saya sudah bisa membuat penotalan data secara otomatis dengan menggunakan function bernama **count().** Contoh penggunaanya yaitu **count($row)** .
8. Saat kita membuat pagination dengan menggunakan function makelink, paging tersebut tidak serta merta langsung jadi atau berjalan, untuk menjalankannya kita bisa menggunakan if else di dalam controller contohnya

***if* (isset($\_GET['page'])) { $page = $\_GET['page']; $mulai = ($tampil \* $page) - $tampil; }** .

1. Yang harus diperhatikan dari pembuatan view order atau view order detail adalah idnya, misalnya id menu atau id kategori.
2. Saya sudah memahami konsep penggunaan dari tanda petik dalam PHP, yaitu jika sudah terdapat petik 2, maka di dalamnya seharusnya menggunakan tanda petik satu.
3. Saya sudah bisa menghilsngksn underline atau tanda garis bawah dengan menggunakan class dari bootstrap yaitu **text-decoration-none**, contohnya yaitu **class=”text-decoration-none”** .
4. Saya bisa menampilkan array dengan rapi dengan menggunakan tag pre dari html, contoh penggunaanya yaitu :

**echo "<pre>"; print\_r($row); echo "</pre>";**

1. Saya bisa memunculkan garis yang bisa berfungsi sebagai penyekat dengan menggunakan tag hr **(<hr>),** contohnya **echo "<hr>"; .**
2. Saya sudah mengerti bahwa segala hal yang dikirim ke dalam view harus selalu berbentuk array asosiatif atau array berpasangan.
3. Saya sudah bisa membuat array asosiatif atau array berpasangan yang digunakan untuk mengirim data ke dalam view , contohnya **$data = [ ‘judul’ => “Tambah Data”;** .
4. Untuk mengambil satu data pada array asosiatif kita perlu memperhatikan dua hal yaitu nomor index array, dan nama index array yang berada di dalam index array, misalnya jika saya ingin mengambil data nama pelanggan dari array berindex 0 saya bisa menggunakan kode **$order[0][‘pelanggan’]** .
5. Saya bisa membuat tulisan di dalam web menjadi tebal atau bold dengan menggunakan tag **<strong>…</strong>,** contohnya **<strong>Hello World!</strong>** . Maka hasilnya adalah **“Hello World”** .
6. Saya bisa membuat angka ribuan yang digunakanuntuk membuat harga menjadi lebih rapi dengan menggunakan function number\_format(), contohnya **number\_format($order[0]['total']) .**
7. Dengan menggunakan function **date("d-m-Y",strtotime($order[0]['tglorder'] ))** kita bisa mengubah format tanggal yang sebelumnya string menjadi berbentuk tanggal, jika menggunakan kode tersebut maka hasilnya **21-03-2021** yang sebelumnya **2021-03-21** .
8. Saat kita melihat ke dalam database dan disana tertulis bahwa ada dua data ber id menu sama namun yang tampil hanya satu saja kemungkinan terbesar penyebabnya adalah karena id kategorinya tidak ada dan kit aharus membuatnya lagi atau bisa juga kita ganti kidkategorinya.
9. Dengan menggunakan perintah **$sql = "UPDATE tblorder SET bayar=$bayar, kembali=$kembali, status=1";** saya sudah bisa membuat update data order.
10. Setelah melakukan proses pembayaran saya bisa langsung mengembalikan halaman tersebut ke halaman order dengan menggunakan kode,

***return* *redirect*()->*to*(*base\_url*("/admin/order"));** .

1. Jika kita lupa menambahkan **WHERE** dalam perintah **$sql = "UPDATE tblorder SET bayar=$bayar, kembali=$kembali, status=1"; .** Maka bisa-bisa seluruh data dalam database akan ikut ter ubah, jika itu terjadi maka akan membuat ita kerepotan sendiri, Jadi kita bisa memperbaikinya menjadi **$sql = "UPDATE tblorder SET bayar=$bayar, kembali=$kembali, status=1 WHERE idorder=$idorder"; .**
2. Saya sudah bisa menampilkan data rincian order, dengan cara membuat controllernya terlebih dahulu, kemudian membuat models, dan terakhir membuat viewnya untuk bagian inteface.
3. Saya sudah mengeti bahwa models dalam codeigniter itu bisa berperan sebagai pemanggil tabel dari database, dan controller itu berperan sebagai pengendali yang mengatur aksi atau jalannya website yang kita buat.
4. Saya sudah bisa membuat form bertipe date yang berfungsi untukmemasukkan data yang berbentuk tanggal, jadi secara otomatis ketika di klik form tersebut akan berubah menjadi seperti kalender, Caranya yaitu kita bisa mengisikan **type=””** menjadi date, jadi **type=’date”** .
5. Saya bisa membuat form yang saya buat menjadi pindah atau mengapung ke kiri dengan menggunakan class bernama float-end, contoh penggunaanya yaitu **class=”float-end”** .
6. Saya sudah bisa menampilkan pencarian tanggal dengan menggunakan if kemudian dikombinasikan dengan oerintah query builder, seperti**$sql = "SELECT \* FROM vorderdetail WHERE tglorder BETWEEN '$awal' AND '$sampai' ";**
7. Saya sudah bisa membuka halaman phpmyadmin di dalam browser, dengan cara menuliskan **localhost/phpmyadmin** kemudian klik enter dan tunggu hingga terbuka.
8. Saya sudah bisa menjalankan file berekstensi PHP di browser dengan mengetikkan **localhost/namafile**.
9. Saya sudah bisa mengatifkan apache dan mysql di dalam XAMPP
10. Dengan menggunakan **protected $allowedFields = ['kategori', 'keterangan'];** . saya bisa menentukan kolom mana saja yang dapat di edit atau diubah.
11. Saya sudah bisa membuat validation atau validasi data.
12. Saya sudah mengerti bahwa validation atau validasi data itu dibagi menjadi 2 hal, yaitu rules untuk mengatur aturan dan message yang berisi respon atau pesan terhadap rules yang dibuat.
13. Dengan menggunakan rule berikut **'user'     => 'alpha\_numeric\_space|min\_length[3]|is\_unique[tbluser.user]',** saya bisa membuat data yang dimasukkan harus dalam bentuk angka atau huruf, minimal 3 karakter, dan tidak boleh ada data yang sama.
14. Saya sudah bisa membuat enkripsi password dengan menggunakan kode

**password\_hash($\_POST['password'], PASSWORD\_DEFAULT) .**

1. Dengan menggunakan enkripsi, password yang sudah dimasukkan tidak akan dapat dilihat secara langsung, meskipun dilihat melalui database, dengan begini data dari user akan semakin aman.
2. Setelah melakukan proses insert ataupun update data, saya bisa mengembalikan halaman insert atau update tersebut ke halaman utama dengan menggunakan function redirect(), contoh penggunaanya adalah

**return redirect()->to(base\_url("/admin/user")); .**

1. Saya sudah bisa membuat posisi dari suatu objek menjadi di tengan dengan class mx dari bootstrap, contoh penggunaanya yaitu, **class-“mx-auto”** .
2. Semua pengiriman data dalam login itu harus menggunakan request method dari codeigniter supaya lebih aman, kita tidak bisa langsung membuat **$\_POST** seperti pada menu-menu yang lain tapi kita harus menggunakan

**($this->request->*getMethod*() == 'post'),** hal ini juga demi keamanan aplikasi yang kita buat.

1. Codeigniter membuat function yang bernama filter, filter ini digunakan untuk menjalankan atau untuk mengecek controller mana yang akan di filter..
2. Saya sudah mengerti bahwa di dalam filters itu hanya boleh diisi oleh dua function saja, yaitu before dan after.
3. Setelah kita mebuat filter, filter itu harus di daftarkan di dalam file Filters.php yang ada di dalam folder config. Untuk codenya bisa kita tullis

**'nama file filters yang sudah kita buat' => \App\Filters\ApiAuth::class,** .

1. Yang harus diperhatikan dalam pembuatan filter yaitu yang pertama kit aharus membuat file bernama bebas di dalam folder filterkemudian di daftarkan di dalam file filter.php kemudian kita bisa atur filter tersebut di dalam routes.
2. Saya sudah bisa membua menu atau pilihan logou dengan menggunakan function session destroy, contohnya ***session*()->*destroy*(); .**
3. Saya sudah mengeri bahwa di dalam routes iu erbagi menjadi 2 hal aau dua blok, yaiu blok perama sebagai pemanggil dan blog kedua sebagai yang dipanggil. Contoh bloknya yaitu **$routes->*get*('/login', 'Admin\Login::index'); .**
4. Saya sudah mengerti bahwa yang memisahkan anara 2 blok yaitu tanda koma.
5. Untuk memasangkan url pada filter kita harus melakukannya secara manual satu per satu di dalam file routes.php , karena tidak langsung otomatis semua url yang kita buat sudah terpasang filter. Jika tidak di setting secraa manual, maka tidak akan bisa terfiter.
6. Saya sudah mengerti bahwa if di dalam php itu juga bisa dipisahkan atau dibagi menjadi dua bagian, satu sebagian sebagai if buka yang biasanya ditandai dengan titik dua, contohnya (**<?php *if*($level === "Admin"): ?>**) dan satu lagi sebagai if tutup contohnya (**<?php *endif;* ?>**).
7. Saya sudah bisa mengatur hak akses pengguna dengan menggunakan if. Dengan begini setiap level akan memiliki hak akses yang berbeda-beda.
8. PHP sudah menyediakan function atau fungsi bernama password\_verify(), function ini jika digunakan akan memverifikasi 2 inputan yaitu yang dikirim oleh form dan yang dilkan oleh pencarian di database, jadi dicocokkan antara yang dikirim oleh form dan yang ada di dalam daabase, contoh penggunaanya yaiu
9. Seelah mempelajari tentang framework PHP codeigniter dan PHP native atau PHP yang biasa saya dapa menyimpulkan bahwa kia bisa membuat aplikasi atau web dengan lebih cepat karena perintah-perintah dari bahasa PHP yang panjang sudah dibungkus ke dalam function-function yang ringkas oleh codeigniter sehingga kita hanya perlu membaca user guidenya untuk mencari tshu cara penggunaanya.

**Saya kurang paham**

1. -