|  |  |
| --- | --- |
| Materi | Nilai |
| PHP- CODEIGNITER–VID1-9-WEEK17 | **96** |

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

1. Saya mengerti baha CodeIgniter merupakan salah satu framework PHP yang terdiri dari class-class yang dibuat oleh ahli.
2. Saya bisa membuat folder dalam Vscode.
3. Saya bisa menjalankan dan menyalakan apache dan mysql di xampp kemudian membukanya di browser.
4. Saya suda menerti bahwa tujuan dibuatnya framework CodeIgniter yaitu supaya para pengembang web atau pembuat web atau aplikasi bisa membuat web dengan lebih cepat dan lebih baik.
5. Saya sudah bisa mendownload framework CodeIgniter, caranya yaitu dengan mengetikkan codeigniter.com di browser kemudian pencet enter, jika suah masuk webnya carilah pilihan download, di dalam pilihan download juga terdapat keterangan versi terbarunya.
6. Framework CodeIgniter sendiri terdiri dari tiga sistem, yaitu controller, model, view. Semua permintaan yan g diminta user akan diterima oleh controller kemudian ke database dengan menggunakan model lalu ditampilkan oleh view.
7. Sepertihalnya framework lain (bootstrap) codeigniter juga mempunyai panduan untuk penggunaanya.
8. Untuk mempelajari codeigniter, kita harus memastikan bahwa diri kita sudah belajar PHP terlebih dahulu, mulai dari dasar-dasarnya, kemudia OOP PHP, dan kemudian project dengan bahasa PHP.
9. Saat muncul error **“WHOOPS! We seem…”** ketika memuka folder public sehingga tak muncul halaman **“Welcome to CodeIgniter”** , Saya bisa mengatasinya dengan cara menghilangkan tanda pgar sebelum **CI\_ENVIRONTMENT** di file env, menghapus tanda titik koma sebelum **extension=intl** di dalam file php.ini, lalu mengganti **ini\_set(“display\_errors’,’0;’)** di file production.php menjadi **ini\_set(“display\_errors’,’1;’)** .
10. Saat mengeluarkan isi dari folder public keluar file ci4maka akan muncul error, cara mengatasinya adalah kita harus membuka file index.php yang dikeluarkan dari folder ci4, kemudian cari variabel path dan ganti isinya menjadi **‘ci4/app/config/paths.php’** .
11. Saya tahu bahwa controller dalam codeigniter adalah sesuatu yang menerima semua permintaan atau request dari user atau pengguna.
12. Saya tahu bahwa dalam memberi nama clas dalam suatu file itu harus sama persis dengan nama file dari class tersebut, untuk penamaannya juga harus diawali dengan huruf kapital, misalnya **‘Kategori.php’** .
13. Saya sudah berhasil membuat sebuah function contohnya

**public function select(){ echo "Untuk menampilkan data"; }** kemudian menampilkan isinya, yaitu dengan menuliskan nama function tersebut setelah nama file dari function tersebut misalnya **“localhost/menu/select”** .

1. Cara kerja dari pemanggilan select adalah kita harus memanggil nama dari classnya (sama saja dengan nama file) di url kemudian ketik slash ( / ) dan panggil nama function lalu kita bisa memanggil parameter contohnya **( $id=null)** .
2. Dalam controller, perintah yang kita masukkan di url merupakan requestnya, dan yang tampil di halaman utama merupakan respond atau jawabannya.
3. Saya sudah berhasil memanggil controller yaitu dengan mengetikkan perintah di dalam urlnya.
4. Saya sudah mengerti bahwa terdapat beberapa berbedaan antara membuat aplikasi web dengan php native dan framework, jika menggunakan php native maka akan membutuhkan lebih banyak file, sedangan jika menggunakanframework file-file tersebut dapat diwakili dengan menggunakan function.
5. Supaya penyebab masalah error dapat diketahui kita harus mencari **CI\_ENVIRONTMENT = production** di dalam file env, namun sebelum itu nama file env harus diganti dengan ditambhi titik di depannya sehingga menjadi **.env** , jika sudah maka CI tadi dapat di copas dan productionnya dapat diubah menjadi development.
6. Saya sudah mengerti dan mengetahui bahwa setiap controller itu pasti membutuhkan Basecontroller.
7. Saya bisa mengatasi error **“404 error ! Not Found”**saat membuka function di dalam file kategori dengan cara menambahkan kode **use App\Controllers\Basecontroller**. Dan diatas kode tersebut kita dapat menambahkan \Admin setelah \Controllers, Setelah itu tinggal kita refresh.
8. Tujuan dari penulisan kode **use** seperti diatas adalah supaya class dari suatu file tersebut dapat menggunakan class dari Basecontroller.
9. Saya tahu bahwa controler itu diletakkan dalam folder-folder untuk memudahkan kita jika terjadi penambahan sesuatu misal penambahan menu dari aplikasi yang kita buat sehingga lebih tertata dan lebih mudah dalam perawatannya.
10. Saya sudah mengerti bahwa routes merupakan sebuah permintaan atau request dalam url yang bisa kita atur atau custom sehingga bisa kita atur sesuai keinginan.
11. Saya bisa mengatur permintaan atau request dengan mengatur file Routes.php, misal jika saya mengganti defaultcontrollernya menjado login dan mengganti variabel routes ($routes) menjadi **‘Login::index’** maka jika kita menjalankan folder ci4 di browser akan langsung menuju halaman dari Login.php.
12. Untuk memanggil fungsi berparameter di dalam file Routes.php kita bisa menggunakan kode **$routes->get(‘*nama file function*/(:any)’,*’folder function\nama file function::namafunction*/$parameter); .**
13. Saya sudah mengerti bahwa saat kita mengisi null dalam variabel id di dalam function itu agar jika salah variabel atau function tersebut akan tetap berjalan.
14. Saya sudah mengerti bahwa group routes merupakan salah satu fitur dari codeigniter yang memungkinkan kita untuk mengelompokkan suatu routes. Group routes ini menurut saya sangat bermanfaat jika ada lebih dari satu routes yang memiliki nama atau atribut yang sama sehingga tidak terjadi duplikasi.
15. Saya sudah bisa embuat group routes contohnya **$routes->group('admin', function($routes){ …………..}); .**
16. Untuk memanggil Route berparameter di dalam group route caranya hampir sama seperti memanggil route berparameter seperti biasanya namun sebelum itu ditambahi dengan ‘add’, misalnya **$routes->add('kategori/update/(:any)', '\Kategori::formUpdate/$1'); .**
17. Saya sudah bisa memanggil group routes, yaitu dengan mengetikkan nama group routes setelah nama restoran kemudian diikuti dengan nama routenya, misalnya **http://localhost:8080/titip-utk-htdocs/php-smk/restoran-ci4/admin(nama group routes) /kategori(nama routes) /update (detail routes)** .
18. Saya sudah bisa menggunakan fitur view, contohnya **echo view("kategori/select");** .
19. Kita bisa menambahkan (:any) yang erguna untuk memasukkan data sembarang yang kemudian bisa tampil di halaman utama.
20. Saya sudah mengerti bahwa view merupakan salah satu fitur dari framework codeigniter yang berguna untuk menangani urusan tampilan atau user interface, untuk melakukannya kita bisa membuat file di dalam folder view dari codeigniter kemudian mengisinya, bisa untuk header ataupun footer.
21. Saya sudah mengerti bahwa saat kita menuliskan direktori file did dalam routes ataupun view, nama file tak perlu diberi ekstensi ( .php) tapi langsung ditulis namanya saja.
22. Saya sudah tahu bahwa base\_url yang terletak paa href merupakan suatu function yang sudah diisi alamt url tertentu supaya format dari hrfnya itu betul.
23. Saya sudah memahami dan mengetahui bahwa semua pengiriman data di dalam codeigniter itu harus dibuat dala bentuk array asosiatif atau array berpasangan.
24. Saya bisa menampilka array dengan bentuk yang lebih tertata dan rapi, yaitu dengan menggunakan kombinasi atau gabungan dari tag pre dan print\_r. contohnya **echo ‘<pre>’; print\_r($data) echo ‘</pre>’;** .
25. Untuk mengirim data ke view kita hanya perlu menambahkan nama dari array asosiatif tersebut seteah direktori, contohnya **echo view(“kategori/select”, $data)** .
26. Untuk menampilkan array tersebut, kita bisa menuju file yang sudah ditaruh variabel array tesebut, kemudian ketik php open dan close, lalu isi dengn key dari array, contohnya **<h1><?php echo $judul; ?></h1>** .
27. Saya sudah bisa mengirimkan isi dari seluruh data ke view dengan menggunakan foreach dengan menggunakan kode sebagai berikut :

**<?php foreach ($kategori as $key => $value): ?>**

**<h2><?= $value ?></h2>**

**<?php endforeach; ?> .**

1. Saya sudah bis membuka menu panduan dari codeignite, caranya yaitu ketikkan codeigniter.com di browser, kemudian pilih read manual.
2. Supaya isi file dari controller tidak terlalu penuh kita dapat menggunakan fitur view layout.
3. Saya sudah bisa meembuat view layout dengan menggunakan method atau metode renderSection(), contohnya <**?= $this->renderSection(‘content’) ?>** .
4. Saya sudah mengerti bahwa saat kita ingin menggunakan view layout dan ingin memasukkan tampilan yang kita inginkan ke dalam view layout kita harus menggunakan extend(), section(), dan endSection(). Contohnya

**<?= $this->extend('default') ?>**

**<?= $this->section('content') ?>**

**<h1>Tampilan yang ingin dimasukkan ke dalam view layouth1>**

**<?= $this->endSection() ?>**

1. Saya sudah memahami dan mengetahui bahwa dengan menggunakan view layout kita dapat membuat aplikasi atau web dengan jumlah baris yang lebih sedikit dan lebih sederhana dengan fungsi yang sama dengan menggunakan php native.
2. Untuk menghubungkan project di codeigniter yang kita buat dengan database caranya cukup mudah, yaitu kita hanya perluu mencari file env kemudian cari menu database default kemudian aktifkan dan sesuaikan, seperti nama database, nama user, dll.
3. Perintah **use** dalam kode ( **use App\Controllers\Basecontroller;** ) memiliki fungsi dan peran yang sama dengan perintah require\_once() yaitu kita dapat menggunakan isi dari file yang sudah kita tentukan di dalam use.

**Saya kurang paham**

1. -