**LAPORAN**

**BENGKEL PEMOGRAMAN FRAMEWORK**

**“P2 :Konfigurasi Dasar Codeigniter, Konsep MVC, Pemanggilan library, Pengenalan Function”**

****

**NAMA : Ikhsan Damasraya**

**NIM : 2255301076**

**KELAS : 2 TI D**

**DOSEN : Mutia Sari Zulvi, S.S.T., M.M.S.I**

**ILB : Ika Suhasmi**

**PROGRAM STUDI TEKNIK INFORMATIKA**

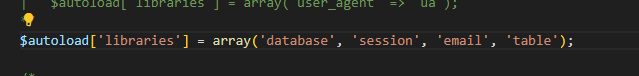
**POLITEKNIK CALTEX RIAU**

**TA 2023 / 2024**

1. **Pengenalan Dasar**
2. **Konfigurasi Dasar**

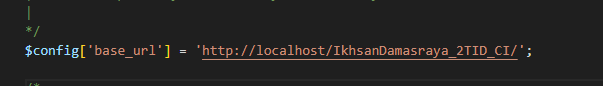
Membuka file **autoload.php** yang berada pada folder **application/config**

Lalu menambahkan codingan dibawah ini pada masing-masing **$autoload**

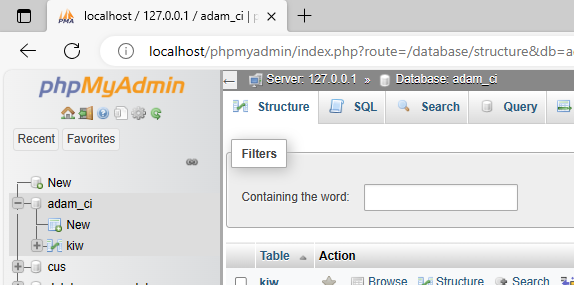




Buka file **config.php** yang ada difolder **application/config,** lalu tambahkan codingan dari folder yang dimiliki, berguna untuk menggunakan domain dengan lebih mudah

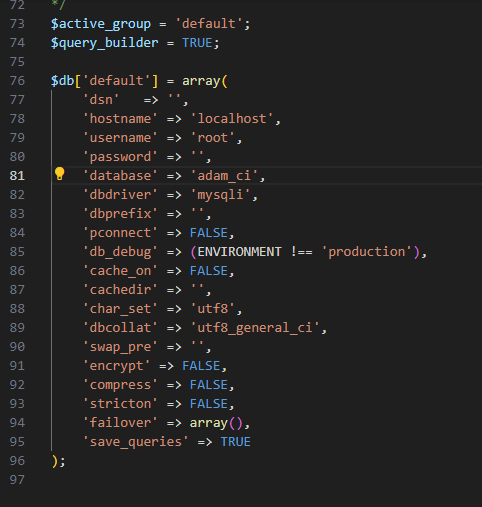


Selanjutnya buat database baru dengan format **nama\_ci**



Setelah itu buka file **database.php** pada folder **application/config** dan

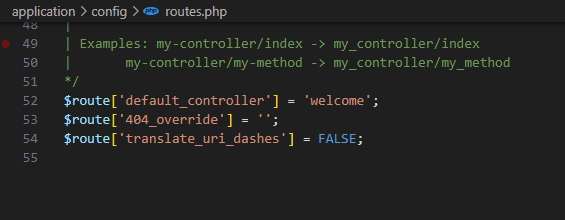
Tambahkan username dan database sesuai dengan database yang dibuat diatas tadi.



File **routes.php** ini berada pada folder **application/config**, route-route ini

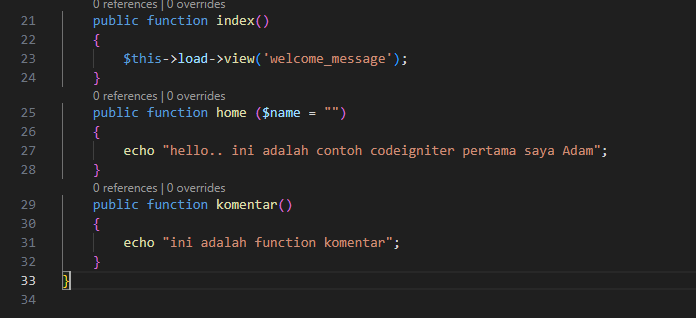
mempunyai peran masing-masing yaitu

| $route['default\_controller'] = 'welcome'; | Mengatur controller mana yang akan digunakan sebagai controller default pada website |
| --- | --- |
| $route['404\_override'] = ''; | Berguna untuk menampilkan halaman yang tidak ditemukan |
| $route['translate\_uri\_dashes'] = FALSE; | Otomatis mereplace hal yang false |

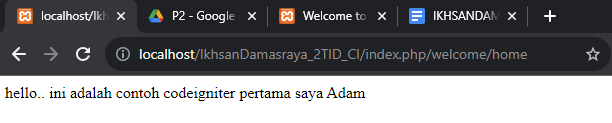


1. **Struktur URL**

* localhost/ : Sebuah induk/host dari website/web server yang dibuat
* adam\_ci : nama folder projek di web server atau htdocs
* Index.php : file yang merupakan wajib untuk menjalankan CI namun file ini bisa ditiadakan menggunakan .htaccess
* Welcome : Nama class pada gambar dibawah
* Index : Nama function pada gambar dibawah
* Membuka file **welcome.php** yang berada pada **application/controller**
* Controller berguna sebagai penengah/perentara antara Model, View, Core dan berbagai resources yang dibutuhkan untuk membuat sebuah web



**Output:**



**Tugas Dadakan**

**- home :** Ini adalah salah satu fungsi yang ada dalam controller. Biasanya,

fungsi ini akan merespons permintaan ketika URL tanpa segmen tambahan

(http://localhost/nama\_proyek/controller) diakses.

**- komentar():** Ini adalah fungsi lain dalam controller. Fungsi ini mungkin merespons

permintaan yang terkait dengan menampilkan komentar atau melakukan operasi

terkait komentar dalam aplikasi.

**- $nama=””;** Ini adalah deklarasi variabel $nama dengan nilai awal string kosong.

Variabel ini mungkin digunakan dalam fungsi-fungsi tersebut untuk menyimpan

atau memanipulasi data.

**-Jika $nama=””** dihilangkan, apakah terjadi error? Jika deklarasi variabel $nama

dihilangkan dari kode, maka jika fungsi-fungsi tersebut mencoba mengakses

atau memanipulasi variabel $nama, akan terjadi kesalahan (error) karena

variabel tersebut tidak ada. Ini akan tergantung pada bagaimana variabel

tersebut digunakan dalam fungsi-fungsi tersebut.

1. **Konsep MVC**

MVC adalah pola arsitektur yang membagi aplikasi menjadi tiga komponen utama:

1. Model: Menangani logika bisnis dan interaksi dengan database.

2. Controller: Mengkoordinasikan aktivitas antara Model dan View.

3. View: Bertugas untuk merepresentasikan data kepada pengguna.

Keuntungan penggunaan pola MVC meliputi komponen yang berfungsi secara independen, fleksibilitas dalam mengubah komponen individual, dan meningkatkan produktivitas tim pengembangan.

**Contoh penerapan Controlelr dan Viewer**

- Dalam CodeIgniter, controller adalah komponen yang mengatur logika aplikasi dan

merespons permintaan dari pengguna.

- Contoh kode controller "Blog.php" yang merespons permintaan dengan memuat

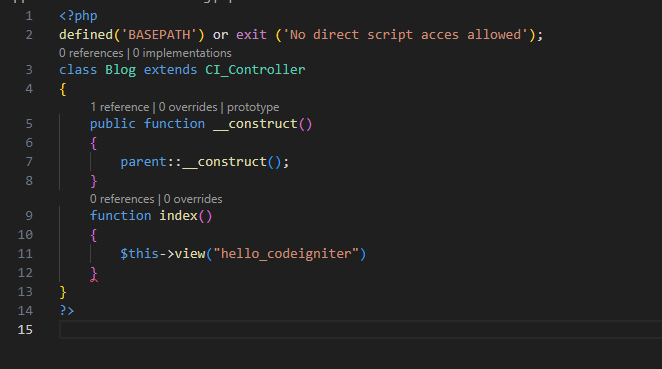
tampilan (View).

- Kode dalam tampilan "hello\_codeigniter.php" digunakan sebagai contoh penerapan

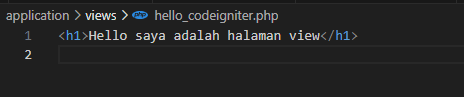
View. View bertugas untuk merepresentasikan data yang diberikan oleh Controller

kepada pengguna.

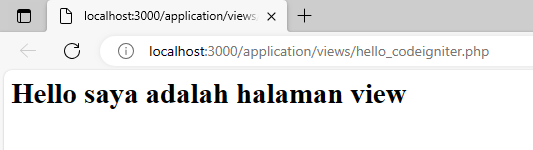
**application/controller/Blog.php**



**application/views/hello\_codeigniter.php**



**Output:**



**Fungsi view memiliki 3 Parameter**

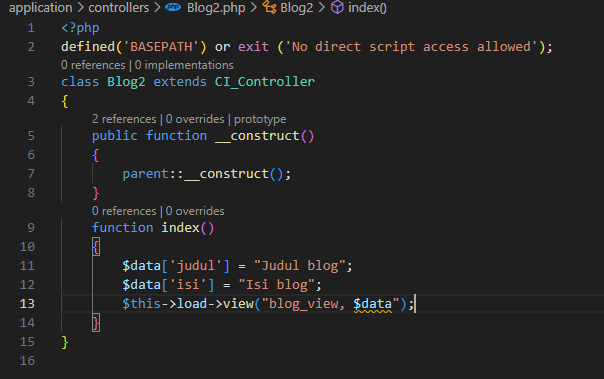
1. **Nama file view**

Nama file view yang hendak diload terletak didalam folder **application/view**

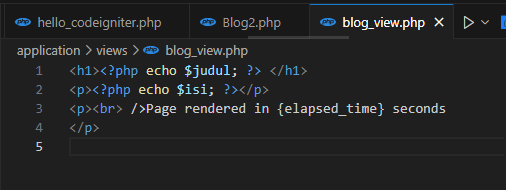
1. **Data Parameter**

Data parameter yaitu parameter yang digunakan untuk melewatkan data dari controller kedalam view contoh nya:

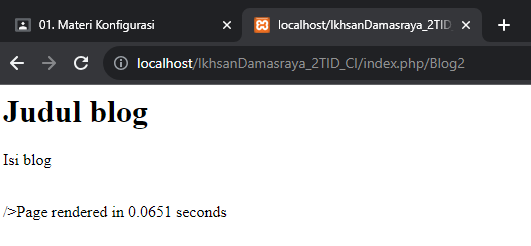
**application/controllers/Blog2.php**



**application/views/blog\_view.php**



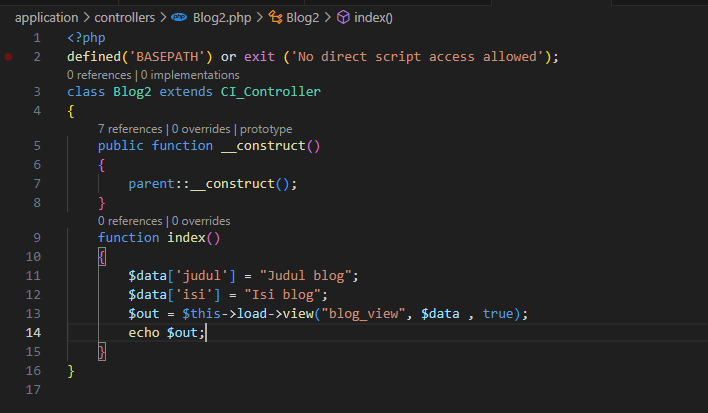
**Output:**



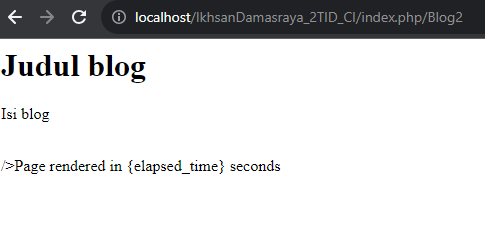
1. **Output Parameter**

Parameter ini akan set true jika kita ingin menyimpan hasil view ke dalam sebuah variabel. Contohnya:

**application/controllers/Blog2.pgp**



**Output:**



**Library dalam CodeIgniter**

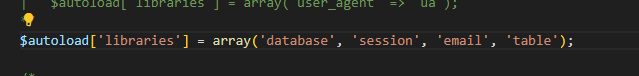
- CodeIgniter menyediakan banyak library yang dapat digunakan untuk tugas-tugas umum dalam pengembangan aplikasi web.

- Contoh beberapa library yang tersedia meliputi database, form validation, email, session, dan lainnya.

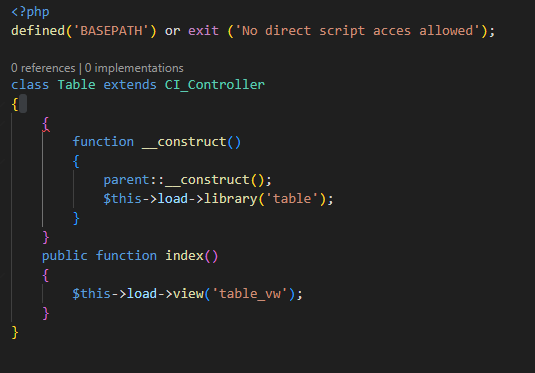
- Kita dapat mengimpor dan menggunakan library ini dalam aplikasi Anda untuk mempercepat pengembangan.

**Ada dua cara untuk menambahkan library pada codeigniter**

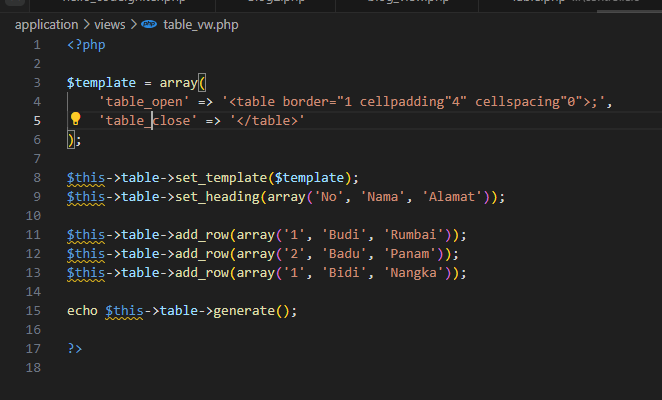
1. Menambahkan file autoupload.php pada bagian libraries



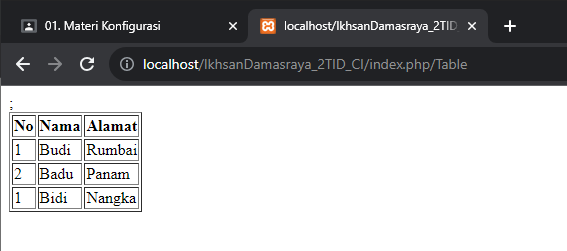
1. Menambahkan pada bagian function **\_\_construct** yang berada pada file controller, contoh:



Lalu, langka berikutnya yaitu membuat halaman view dengan nama **application/views/table\_vw.php**



Beginila tampilan nya sesudah dijalan kan.



**Membuat Library Sendiri**

- CodeIgniter memungkinkan untuk membuat library khusus sesuai

kebutuhan aplikasi.

- Dapat menyimpan library buatan sendiri dalam folder

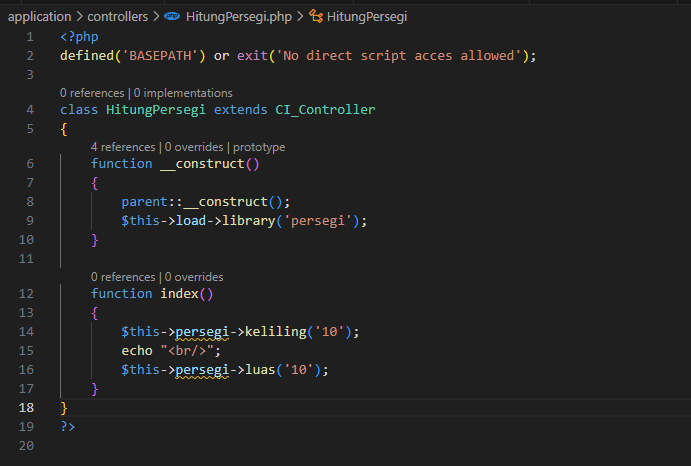
**"application/libraries"** dan mengaksesnya dari Controller.

Membuat file **persegi.php** dan simpan didalam folder **application/libraries**

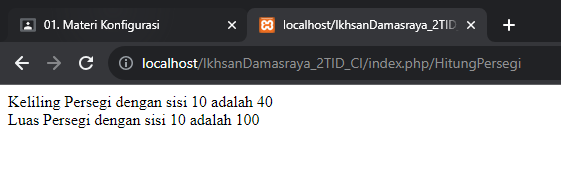


Berikut nya buat file **Hitungpersegi.php** yang disimpan didalam

**application/controlles**



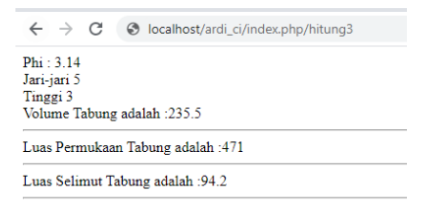
**Output:**



1. **Tugas**
   1. **Soal**

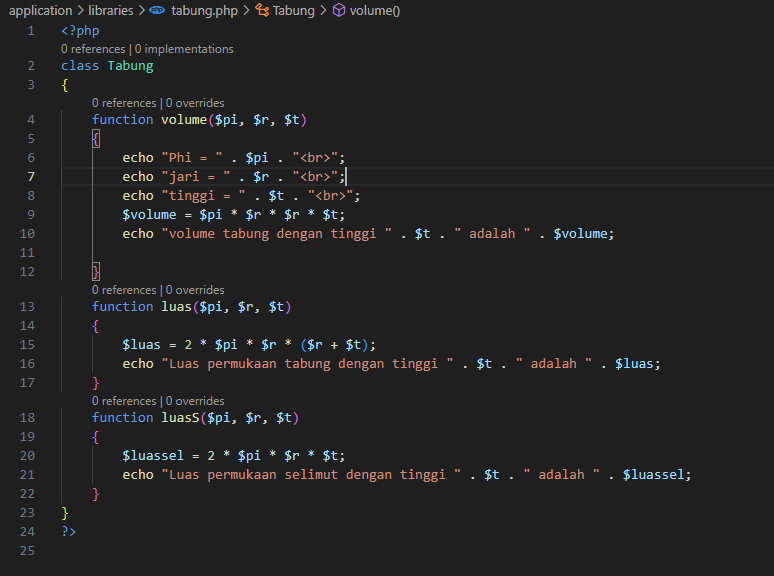
Silahkan buat library sendiri untuk perhitungan berikut.

| **Volume Tabung (V)** | **V = phi x r^2 x t** |
| --- | --- |
| **Luas Permukaan Tabung (L)** | **L = 2 x phi x r x (r + t)** |
| **Luas Selimut Ls** | **Ls = 2 x phi x r x t** |

****

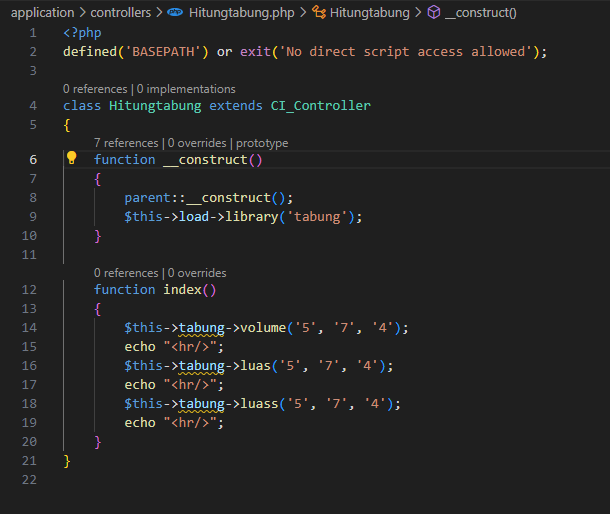
**application/libarires/tabung.php**

Membuat library menggunakan rumus menghitung volume tabung, luas permukaan tabung dan luas selimut tabung.



**application/controllers/Hitungtabung.php**

Mencetak output menggunakan php ini.



**Output:**

