

TUTORIAL MERANCANG ENGINE API ANDA SENDIRI MENGUNAKAN CODEIGNITER 4

Modul Praktikum Teknologi Sistem Terintegrasi
Versi 0.1-30.11.2023



HAI, SELAMAT
BERSENANG-SENANG
DENGAN CODEIGNITER



11/16/2024

Dr. Fadhil Hidayat (@2persen)
II3160 – 01 : Teknologi Sistem Terintegrasi
Prodi Sistem dan Teknologi Infomasi ITB



CODEIGNITER 4

BAGIAN 1- INSTALASI

Instalasi XAMPP dan Codeigniter 4

SOURCE

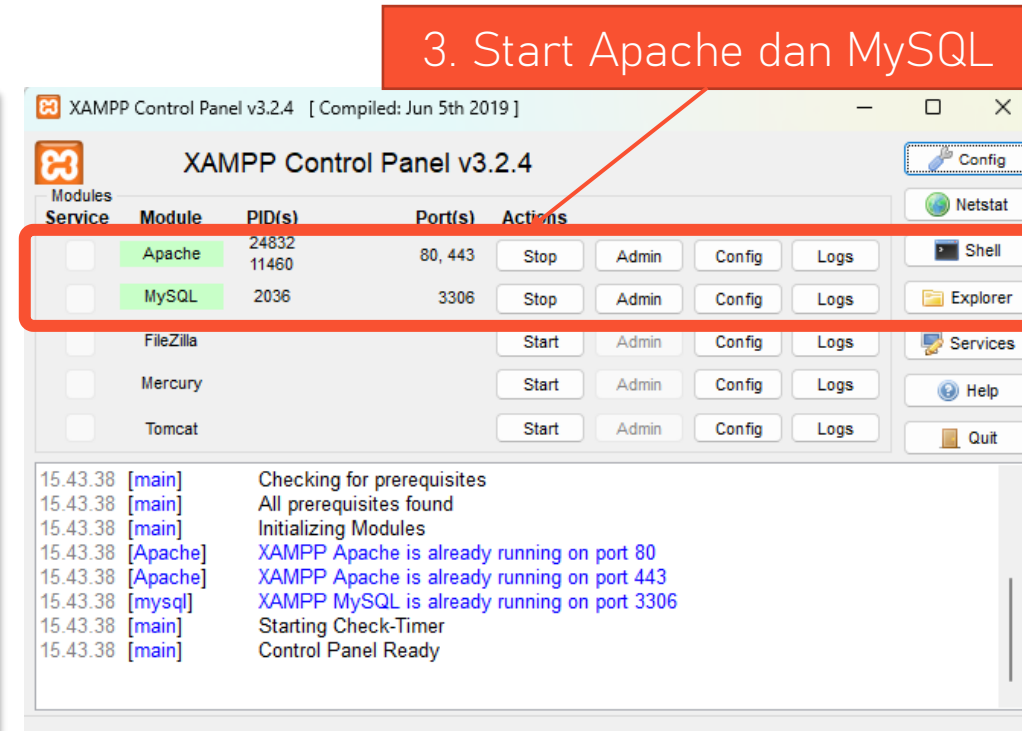
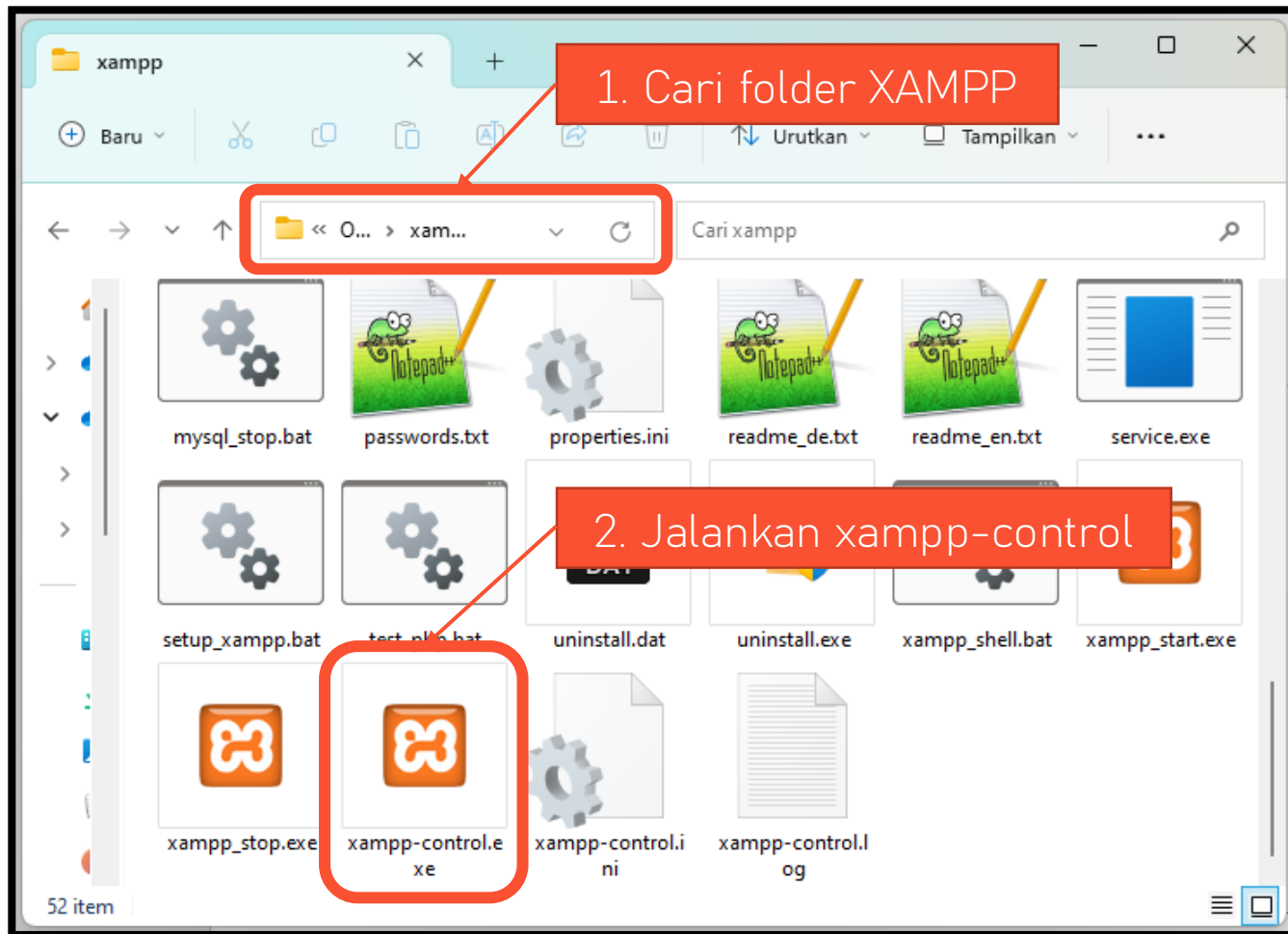
NOTE: VERSI TUTORIAL MUNGKIN BERBEDA DENGAN VERSI YANG ANDA DONLOAD

- XAMPP Apache + MariaDB + PHP + Perl : <https://www.apachefriends.org/>
- Codeigniter 4 : <https://github.com/codeigniter4/CodeIgniter4>
- Visual Studio Code : <https://code.visualstudio.com/download>

Instruksi:

1. Download dan install XAMPP
2. Download dan install Visual Studio Code
3. Download source **Codeigniter 4**, simpan di **folder** manapun yang Anda inginkan kemudian **extract zip** filenya di folder tersebut. Anda boleh **mengganti nama foldernya** dengan nama apapun selama **tidak menggunakan spasi**.

JALANKAN XAMPP



MEMBUAT DATABASE KOSONG

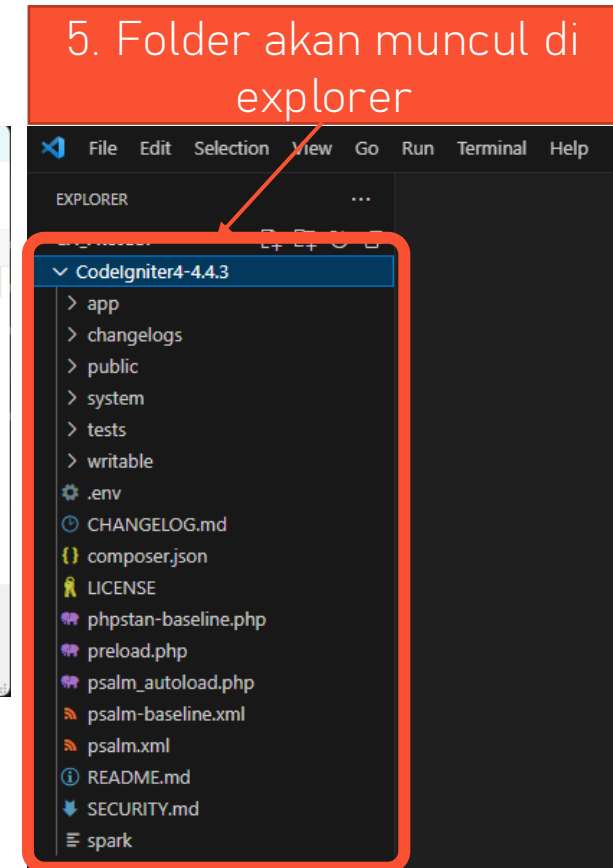
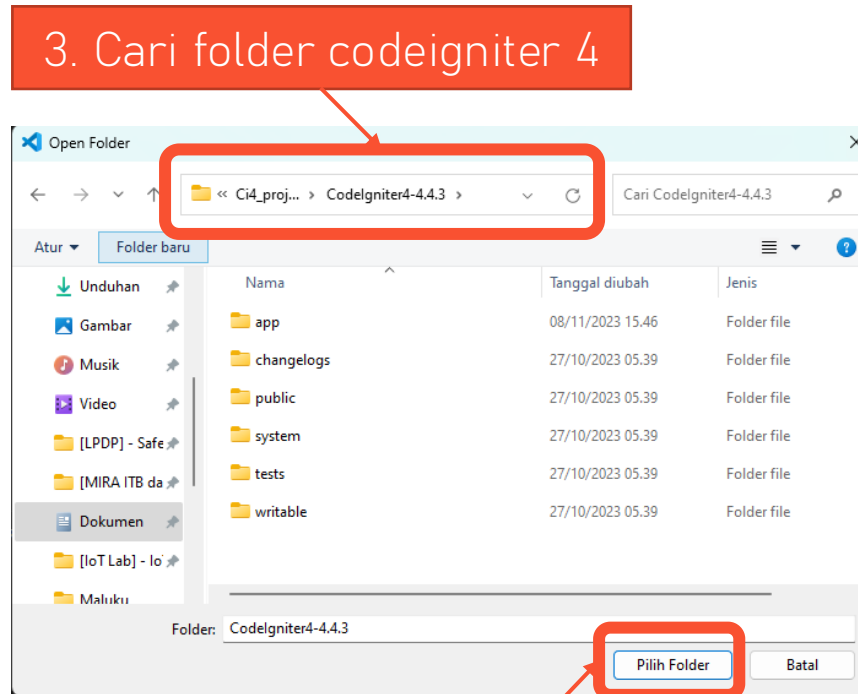
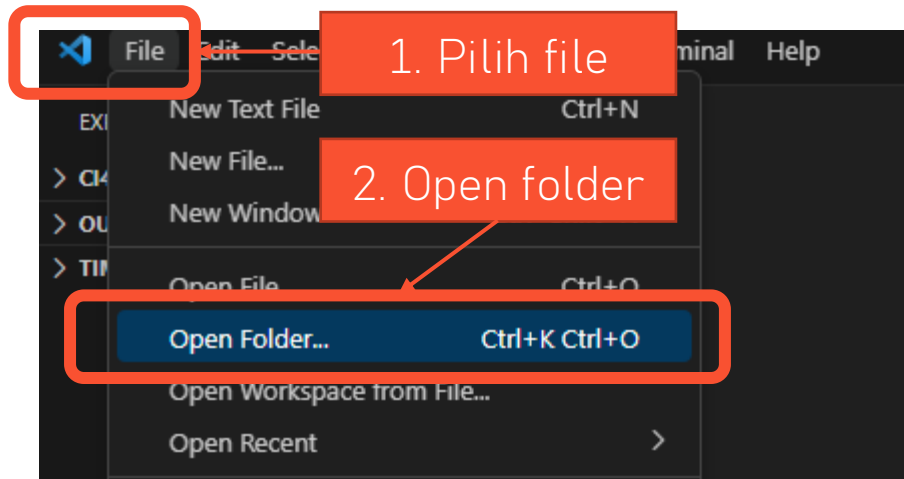
Akses <http://localhost/phpmyadmin/> menggunakan browser

The screenshot shows the phpMyAdmin web interface. The browser address bar displays `localhost/phpmyadmin/server_databases.php?server=1`. The interface includes a top navigation bar with tabs for 'Databases', 'SQL', 'Status', 'User accounts', 'Export', and 'Import'. On the left sidebar, there are 'Recent' and 'Favorites' sections. The 'Databases' section shows a 'New' button and a list of existing databases, including 'codeigniter4_db'. The main content area is titled 'Databases' and contains a 'Create database' section. This section has a text input field for 'Database name', a dropdown menu for the character set (currently set to 'utf8mb4_general_ci'), and a 'Create' button. Red arrows and boxes highlight the following steps:

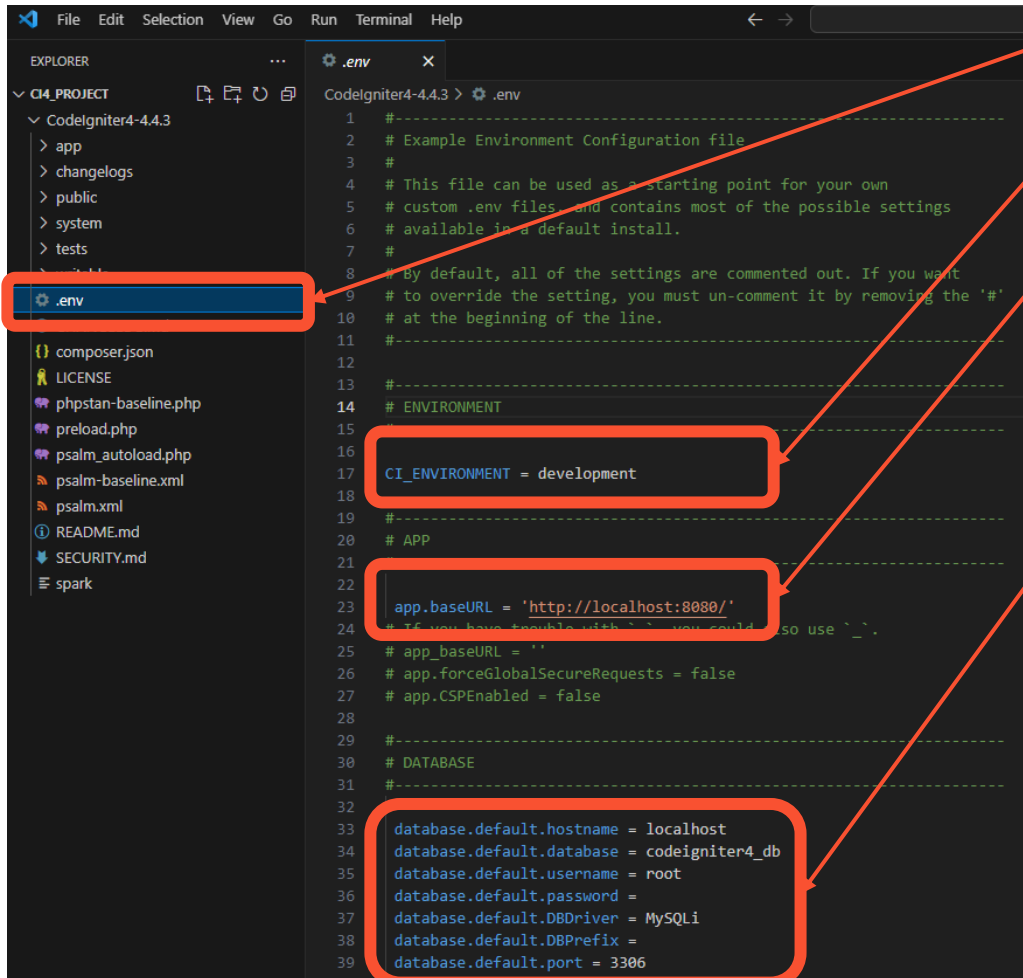
1. Akses phpmyadmin (points to the browser address bar)
2. Klik New (points to the 'New' button in the sidebar)
3. Buat nama database, misalnya `codeigniter4_db` (points to the 'Database name' input field)
4. Klik Create, kemudian nama DB yang baru akan muncul di sidebar (points to the 'Create' button)

MENGGUNAKAN VISUAL STUDIO CODE

- Buka Visual Studio Code



SETUP CODEIGNITER 4



```
1 #-----
2 # Example Environment Configuration file
3 #
4 # This file can be used as a starting point for your own
5 # custom .env files, and contains most of the possible settings
6 # available in a default install.
7 #
8 # By default, all of the settings are commented out. If you want
9 # to override the setting, you must un-comment it by removing the '#'
10 # at the beginning of the line.
11 #-----
12 #
13 #-----
14 # ENVIRONMENT
15 #
16 CI_ENVIRONMENT = development
17 #
18 #-----
19 # APP
20 #
21 app.baseURL = 'http://localhost:8080/'
22 # If you have trouble with 'http://localhost:8080/' you could also use 'http://localhost:8080/'
23 # app_baseURL = ''
24 # app.forceGlobalSecureRequests = false
25 # app.CSPEnabled = false
26 #
27 #-----
28 # DATABASE
29 #
30 database.default.hostname = localhost
31 database.default.database = codeigniter4_db
32 database.default.username = root
33 database.default.password =
34 database.default.DBDriver = MySQLi
35 database.default.DBPrefix =
36 database.default.port = 3306
```

1. Rename file env menjadi .env

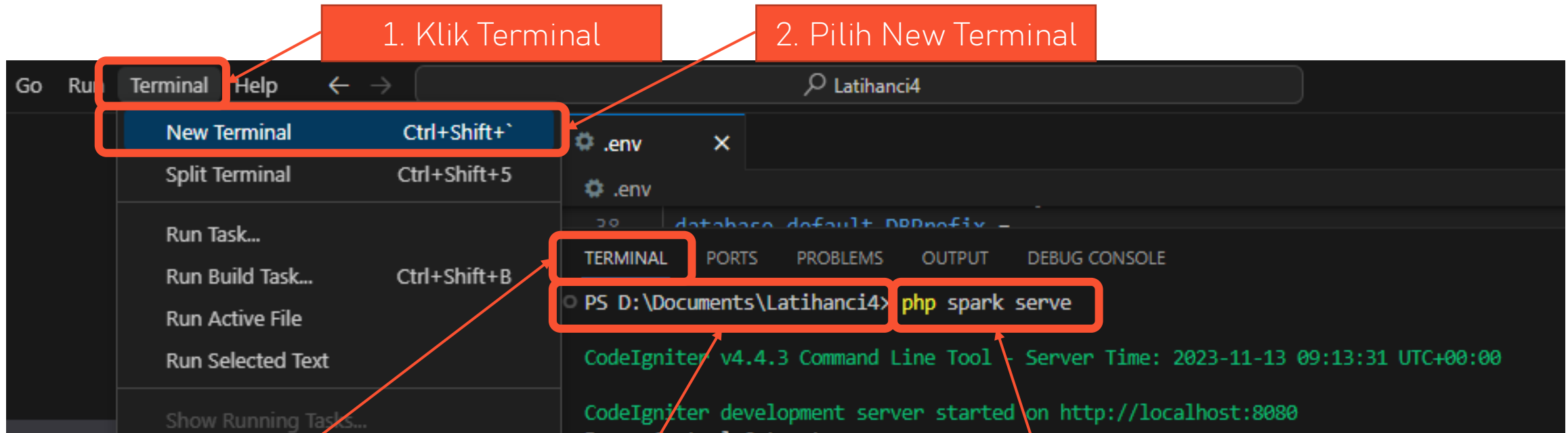
2. CI_ENVIRONMENT = development

3. app.baseURL = 'http://localhost:8080/'

4. database.default.hostname = localhost
database.default.database = codeigniter4_db
database.default.username = root
database.default.password =
database.default.DBDriver = MySQLi
database.default.DBPrefix =
database.default.port = 3306

Nama database “codeigniter4_db” di atas diganti dengan nama database yang Anda buat.

JALANKAN SERVICE CODEIGNITER 4



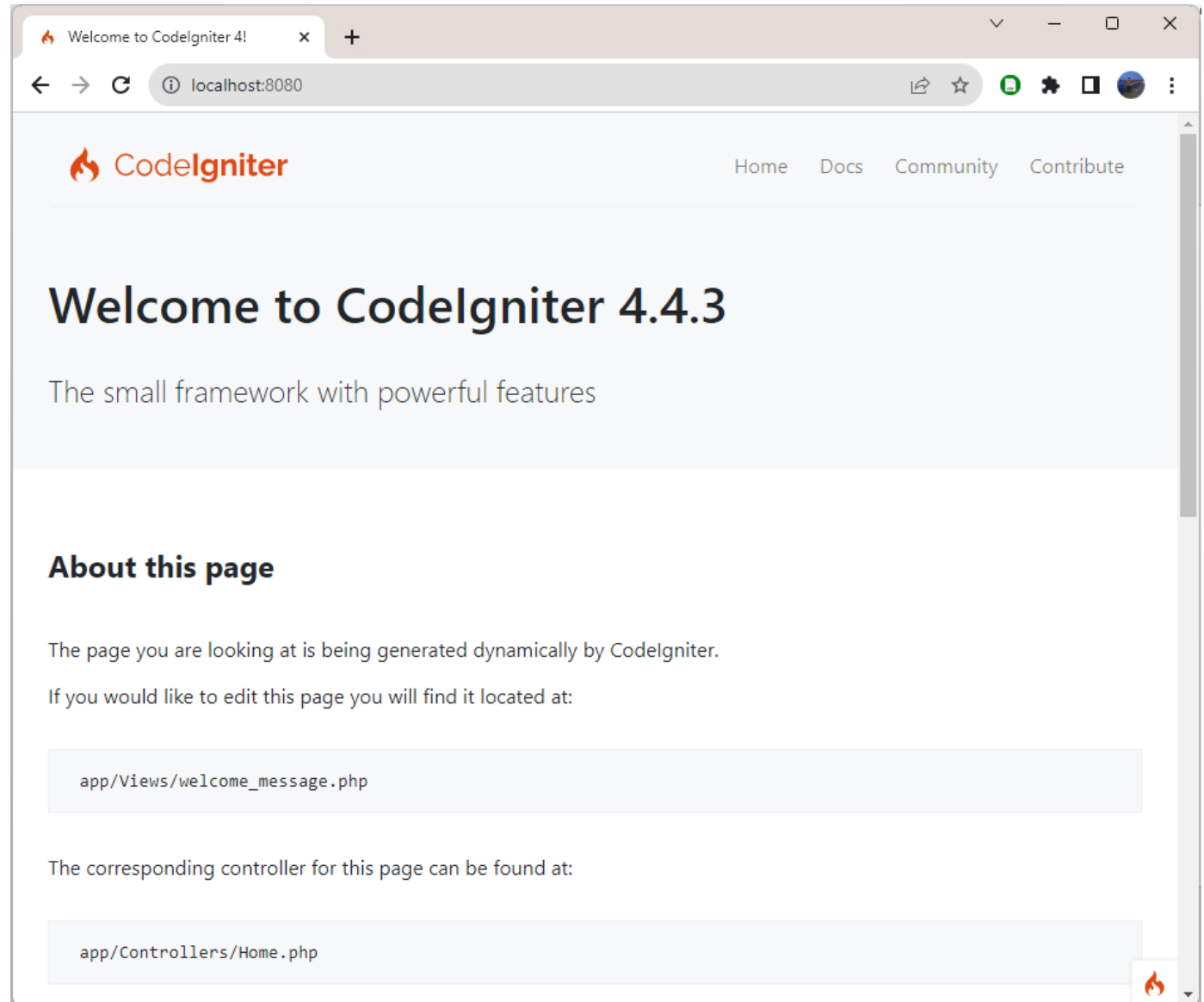
3. Cari Tab terminal di bagian bawah

4. Pastikan folder Anda berada pada folder yang memiliki file spark

5. Ketik php spark serve

JALANKAN DI BROWSER

Buka browser, ketik
localhost:8080 dan Enter



JIKA PHP SPARK SERVE TIDAK JALAN

The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORER:** The file explorer on the left shows the project structure. The 'app' folder is selected. At the bottom, the 'spark' folder is highlighted with an orange box.
- .env File:** The file is open in the editor, showing database configuration:

```
33 database.default.hostname = localhost
34 database.default.database = codeigniter4_db
35 database.default.username = root
36 database.default.password =
```
- TERMINAL:** The terminal at the bottom shows the output of the development server:

```
CodeIgniter development server started on http://localhost:8080
Press Control-C to stop.
[Mon Nov 13 16:13:32 2023] PHP 7.4.1 Development Server (http://localhost:8080) started
[Mon Nov 13 16:13:41 2023] [::1]:64758 Accepted
[Mon Nov 13 16:13:42 2023] [::1]:64758 Closing
[Mon Nov 13 16:13:42 2023] [::1]:64760 Accepted
[Mon Nov 13 16:13:42 2023] [::1]:64760 [200]: GET /index.php?debugbar
[Mon Nov 13 16:13:42 2023] [::1]:64760 Closing
[Mon Nov 13 16:13:42 2023] [::1]:64762 Accepted
[Mon Nov 13 16:13:42 2023] [::1]:64762 [200]: GET /index.php?debugbar_time=1699866821.968708
[Mon Nov 13 16:13:42 2023] [::1]:64762 Closing
```

An orange text box with the text "Pastikan kembali folder pada terminal berada pada folder yang sama dengan file spark" has two arrows pointing to the 'spark' folder in the Explorer and the terminal prompt.

History restored

PS D:\Documents\Latihanci4>

AKHIR BAGIAN 1

Lanjutkan ke Bagian 2 - Controller

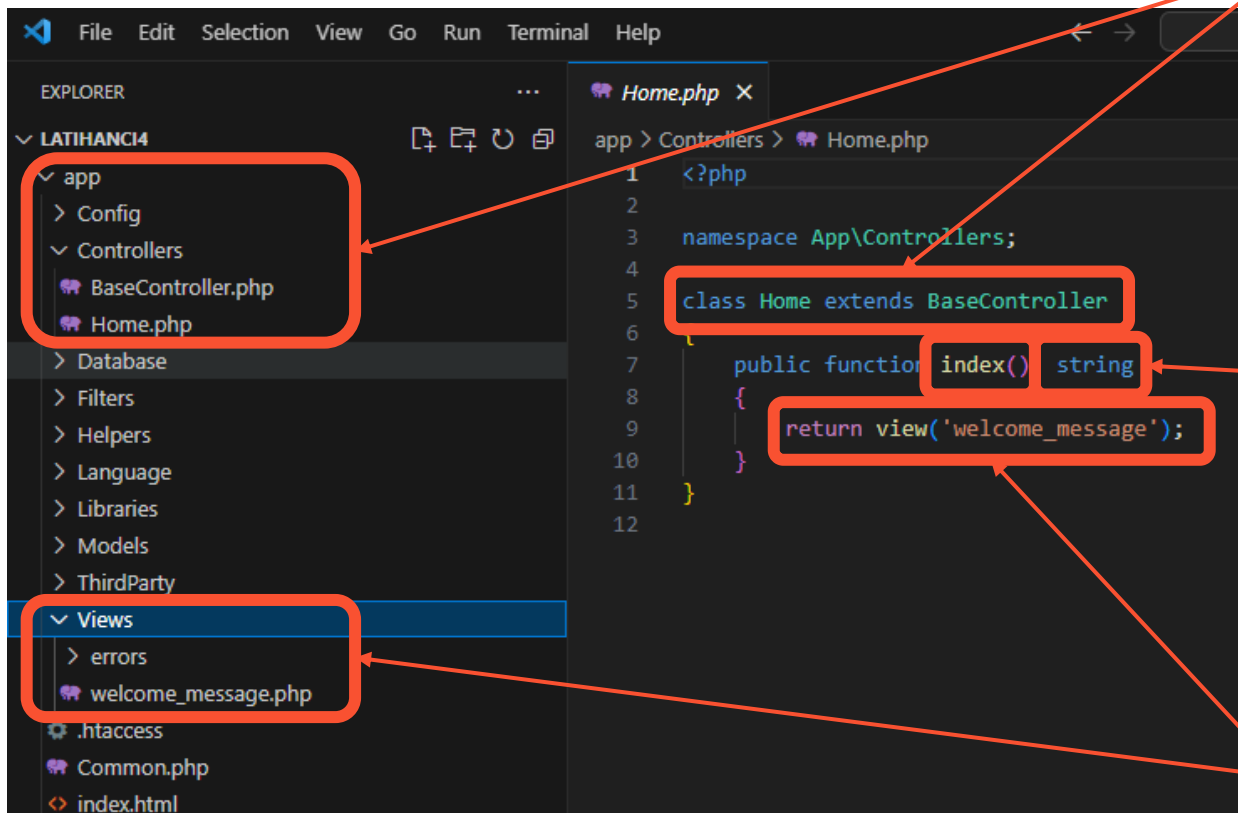
The background of the slide features a complex network of white dots connected by thin white lines, creating a web-like or molecular structure. The dots vary in size, and the lines are of varying lengths, creating a sense of depth and connectivity. The overall color palette is a gradient of warm tones, from a deep orange at the top to a lighter, almost white, at the bottom.

CODEIGNITER 4

BAGIAN 2- CONTROLLER

Hello World, Controller, Method, dan Routes

CONTROLLER

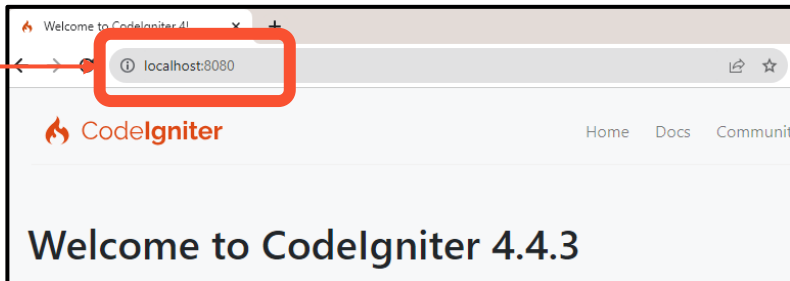


1. Controller ada pada folder app > Controllers. Terdiri atas 2 file yaitu BaseController.php dan Home.php yang merupakan turunan dari BaseController. Buka Home.php, nama class harus sama dengan nama controller.

2. Class Home memiliki method indeks(). Ingat jika method berupa function maka harus ada return variable seperti pada contoh, return variable adalah string. Namun jika berupa prosedur, maka return tidak diperlukan (hapus : string).

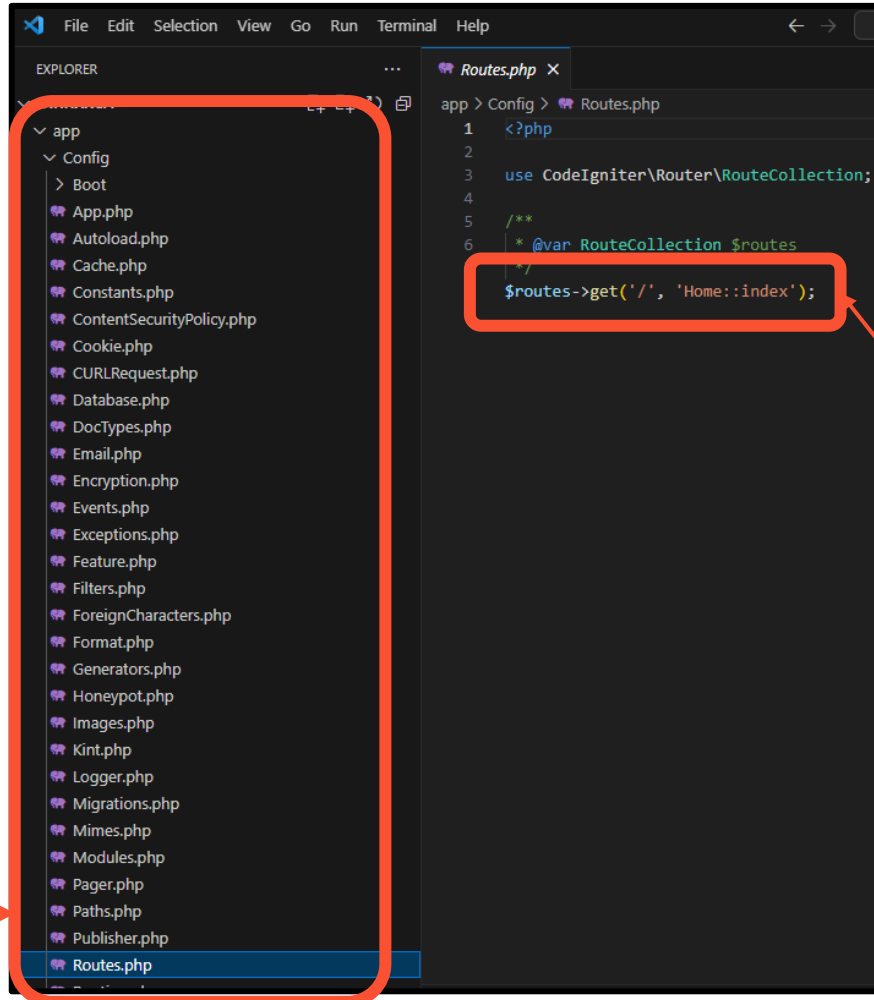
3. Pada method indeks() : string, return memanggil halaman pada folder view dengan nama file welcome_message.php.

ROUTES



1. Saat pertama kali mengakses localhost:8080, maka diarahkan ke controller Home dan method indeks() karena diatur oleh file Routed.php

2. File Routes.php berada pada folder app > Config > Routes.php



3. Pada baris ini '/' berarti indeks atau halaman yang pertama kali diakses. 'Home::index' berarti memanggil controller Home dan method indeks(). Selanjutnya untuk mengakses method lain dengan nama linknya, maka ditambahkan nama setelah tanda /. Misal link: localhost:8080/about, maka tambahkan **\$routes->get('/about', 'About::about');**

MENGGANTI HALAMAN INDEKS DENGAN HELLO WORLD

CARA 1: GANTI PADA VIEW

- Buka file app > views > welcome_message.php.
- Hapus semua kode pada halaman tersebut.
- Tulis ulang dengan kode HTML biasa, misal:

```
<html>
    <body>
        <b>Hello World</b>
    </body>
</html>
```

- Jalankan kembali localhost:8080 pada browser

CARA 2: GANTI PADA CONTROLLER

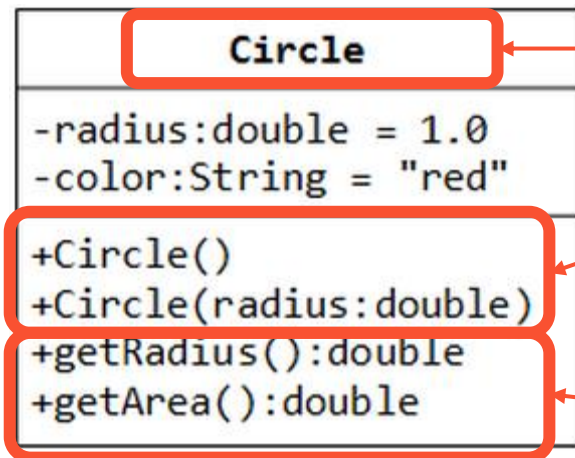
- Buka file app > Controllers > Home.php.
- Buang : string pada method indeks() dan hapus return variable ganti dengan echo 'Hello World';

```
public function index(){
    echo 'Hello World';
}
```

- Jalankan kembali localhost:8080 pada browser
- Dengan cara ini tidak memanggil view.

CLASS, CONSTRUCTOR, DAN METHOD

Ingat kembali konsep class pada OOP, konsep ini dapat digunakan untuk merancang aplikasi menggunakan C#



1. Bagian nama class akan menjadi nama Controller

2. Constructor method adalah method yang memiliki nama yang sama dengan nama class, jika class dipanggil dengan parameter input maka constructor yang otomatis dijalankan adalah yang memiliki parameter input.

3. Method bukan constructor tidak otomatis dijalankan.

AKHIR BAGIAN 2

Lanjutkan ke Bagian 3 - View



CODEIGNITER 4

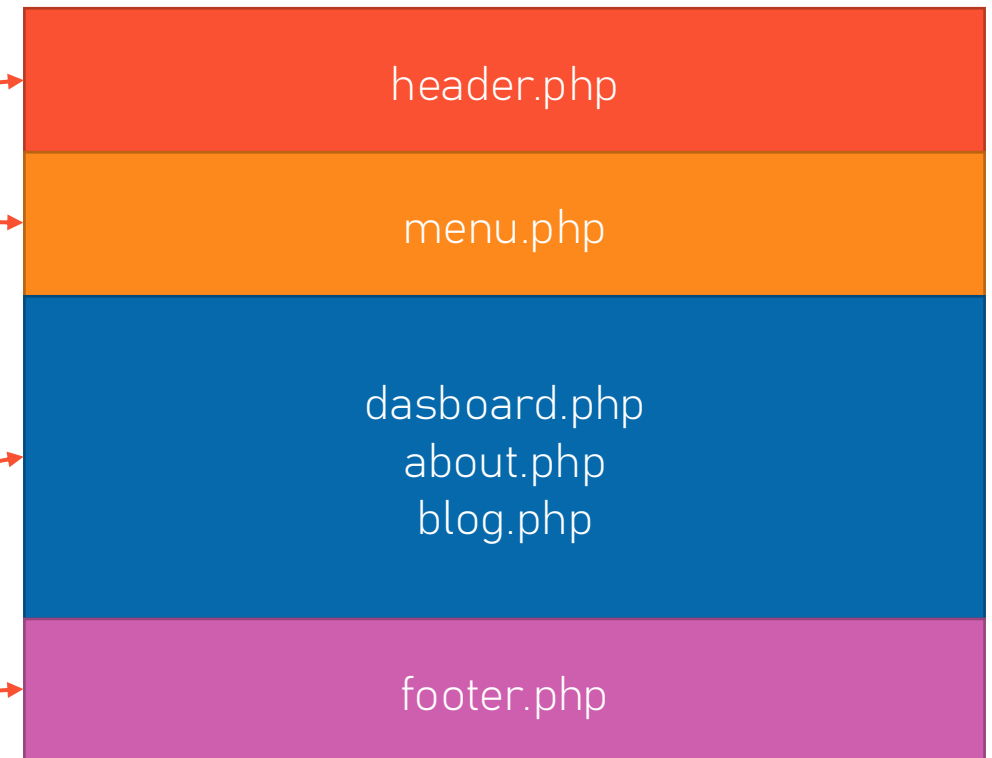
BAGIAN 3- VIEW

Load Multiple View, dan Static Web Page

IDE VIEW

View yang akan dirancang terdiri atas 4 bagian, yaitu:

- File header.php : berisi judul aplikasi
- File menu.php : berisi link menu
- File konten, yang terdapat di dalam folder app > views :
 - File dashboard.php : berisi dashboard
 - File about.php : berisi informasi about
 - File blog.php : berisi informasi blog
- File footer.php : berisi copyright footer



HEADER.PHP

- Buat file baru pada app > Views > header.php
- Ketik kode HTML berikut:

```
<HTML>
```

```
<body>
```

```
<h1>Website Beta</h1>
```

- Biarkan tag <HTML> dan <body> tidak ditutup dengan </HTML> dan </body>, karena akan ditutup pada bagian footer.php

FOOTER.PHP

- Buat file baru pada app > Views > footer.php
- Ketik kode HTML berikut:

```
<p>Copyright 2023 STI ITB</p>
```

```
</body>
```

```
</HTML>
```

- Tutup tag <HTML> dan <body> dengan </HTML> dan </body> pada bagian footer.php ini

MENU.PHP

- Buat file baru pada app > Views > menu.php
- Ketik kode HTML berikut:

```
<p> Menu :
```

```
<a href="/">Dashboard</a> |
```

```
<a href="/about">About</a> |
```

```
<a href="/blog">Blog</a> </p>
```

KONTEN

- Buat 3 file pada folder app > Views sebagai berikut:

- File dashboard.php berisi kode berikut:

```
<h2>Dashboard</h2>
```

```
<p>Ini halaman dashboard</p>
```

- File about.php berisi kode berikut:

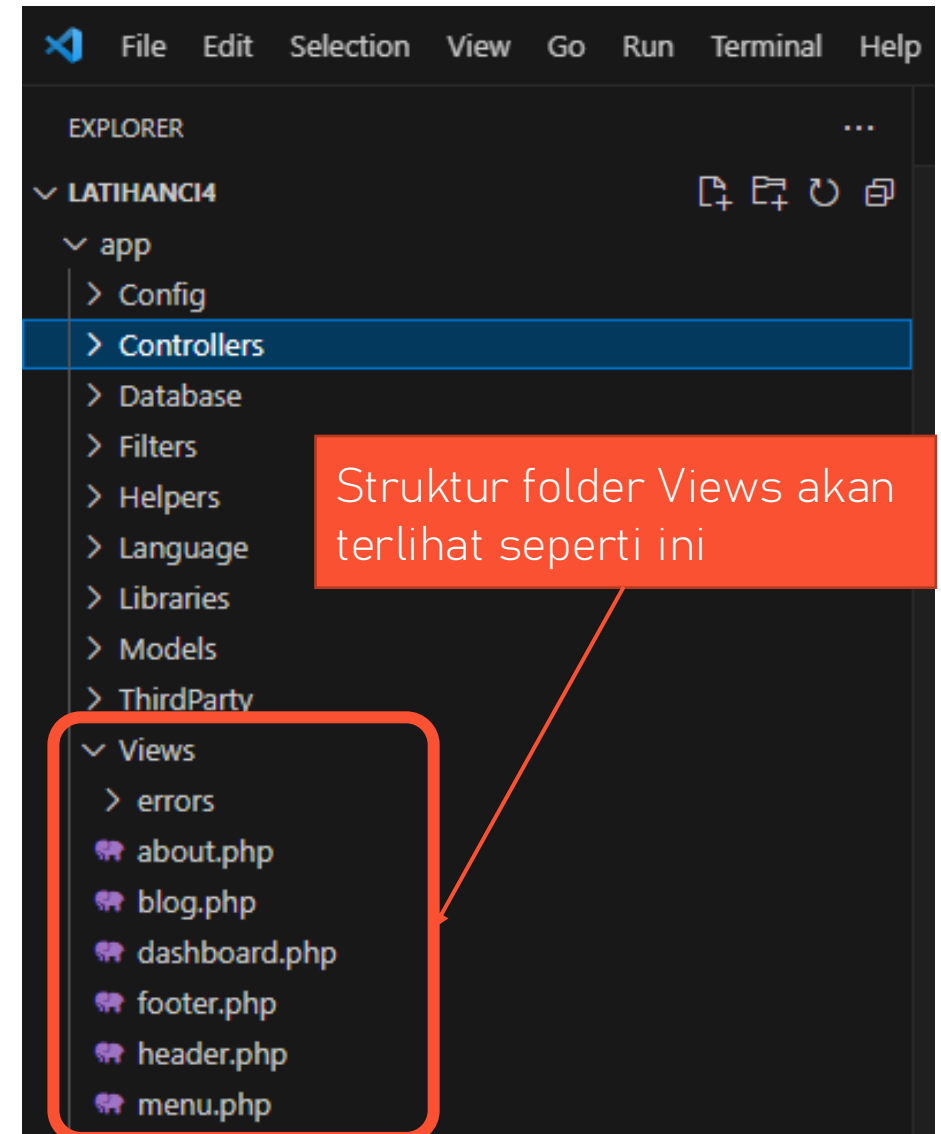
```
<h2>About</h2>
```

```
<p>Ini halaman about</p>
```

- File blog.php berisi kode berikut:

```
<h2>Blog</h2>
```

```
<p>Ini halaman blog</p>
```

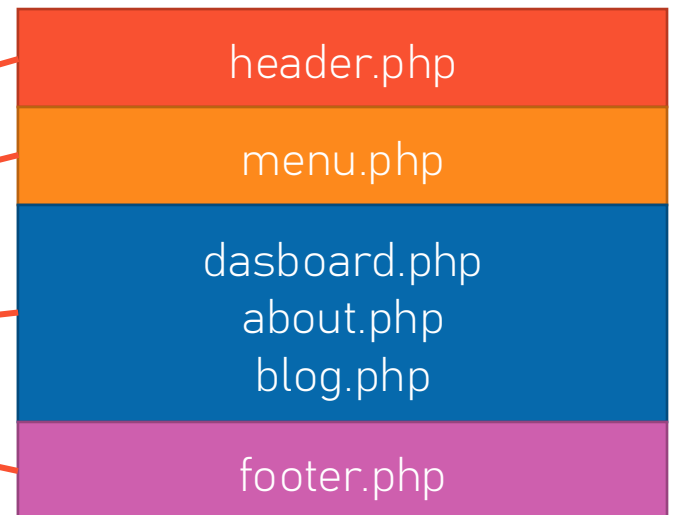


UPDATE CONTROLLER HOME.PHP

Update file apps > Controllers > Home.php pada method index() menjadi:

```
public function index() : string {  
    return view('header').view('menu').view('dashboard').view('footer');  
}
```

```
app > Controllers > Home.php  
1  <?php  
2  
3  namespace App\Controllers;  
4  
5  class Home extends BaseController  
6  {  
7      public function index() : string  
8      {  
9          return view('header').view('menu').view('dashboard').view('footer');  
10     }  
11 }  
12
```



CONTROLLER ABOUT.PHP

Buat file apps > Controllers > About.php dan ketik kode berikut:

```
<?php
namespace App\Controllers;
class About extends BaseController
{
    public function index() : string
    {
        return view('header').view('menu').view('about').view('footer');
    }
}
```

CONTROLLER BLOG.PHP

Buat file apps > Controllers > Blog.php dan ketik kode berikut:

```
<?php
namespace App\Controllers;
class Blog extends BaseController
{
    public function index() : string
    {
        return view('header').view('menu').view(blog').view('footer');
    }
}
```

UPDATE ROUTES

Update file apps > Config > Routes.php dengan menambahkan kode berikut:

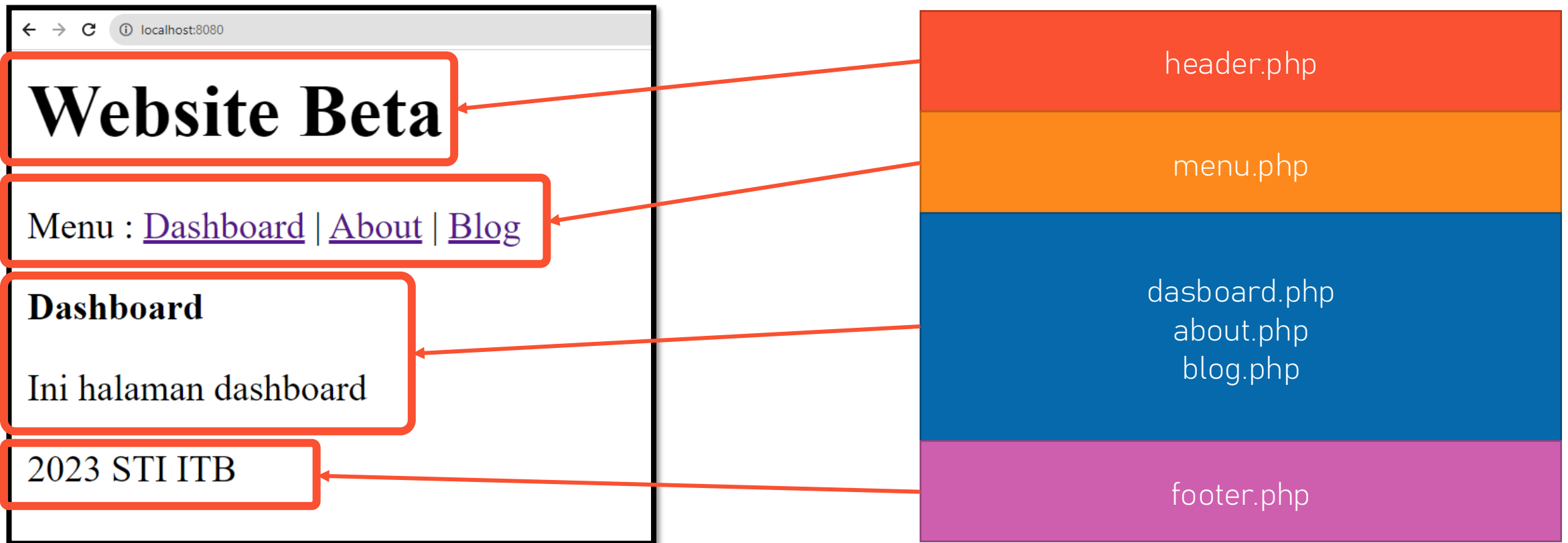
```
$routes->get('/about', 'About::index');  
$routes->get('/blog', 'Blog::index');
```

```
app > Config > Routes.php  
1  <?php  
2  
3  use CodeIgniter\Router\RouteCollection;  
4  
5  /**  
6   * @var RouteCollection $routes  
7   */  
8  $routes->get('/', 'Home::index');  
9  $routes->get('/about', 'About::index');  
10 $routes->get('/blog', 'Blog::index');
```

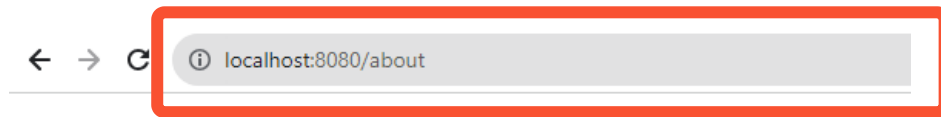
Ingat kode link href pada menu:
Dashboard
About
Blog

JALANKAN WEB STATIS

Akses localhost:8080 maka akan diarahkan ke halaman dashboard.



PERHATIKAN URL DI ATAS
SAAT MENGAKSES LINK



Website Beta

Menu : [Dashboard](#) | [About](#) | [Blog](#)

About

Ini halaman about

2023 STI ITB

```
app > Config > Routes.php
1  <?php
2
3  use CodeIgniter\Router\RouteCollection;
4
5  /**
6   * @var RouteCollection $routes
7   */
8  $routes->get('/', 'Home::index');
9  $routes->get('/about', 'About::index');
10 $routes->get('/blog', 'Blog::index');
```

Ingat kode link href pada menu:

```
<a href="/">Dashboard</a>
```

```
<a href="/about">About</a>
```

```
<a href="/blog">Blog</a>
```

AKHIR BAGIAN 3

Lanjutkan ke Bagian 4 - Model

The background of the slide features a complex network of white dots connected by thin white lines, creating a web-like or molecular structure. The dots vary in size, and the lines are of uniform thickness. The overall color scheme is a gradient of warm tones, from a deep orange at the top to a lighter, almost white, at the bottom.

CODEIGNITER 4

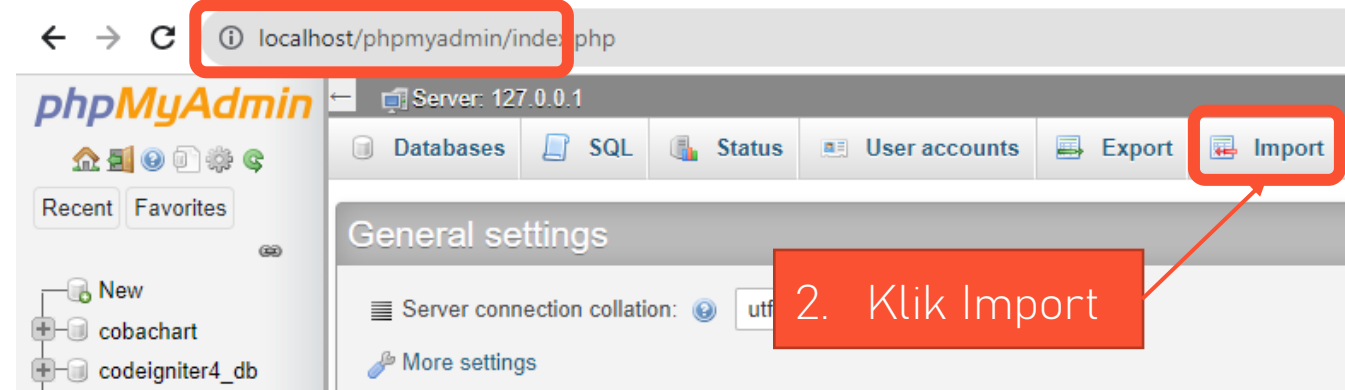
BAGIAN 4- MODEL

Import Database, Finding Data, dan Display Result

IMPORT DATABASE

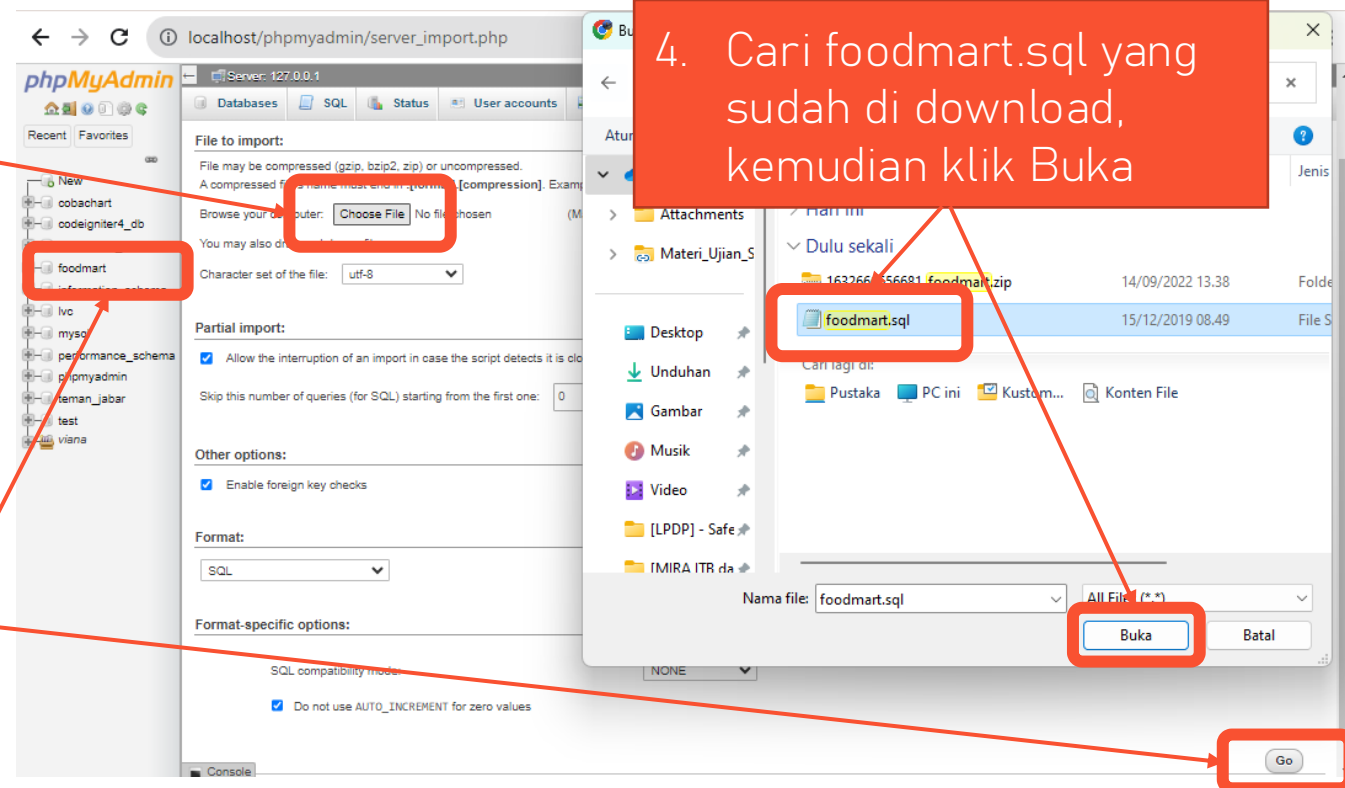
Download foodmart.sql di
<https://github.com/arunkjn/foodmart-mysql/blob/master/foodmart.sql>

1. Akses localhost/phpmyadmin



2. Klik Import

3. Klik Choose File



4. Cari foodmart.sql yang sudah di download, kemudian klik Buka

5. Klik Go, jika berhasil maka database foodmart akan muncul di sidebar

STRUKTUR TABEL FOODMART

1. Database foodmart.sql berisi tabel fakta sales_fact_1997. Klik di sales_fact_1997 untuk menampilkan record tabel

2. Total ada 86.868 baris, mungkin berbeda tergantung source download database

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> customer		10,281	InnoDB	utf8_general_ci	4.0 MiB	-
<input type="checkbox"/> product		1,560	InnoDB	utf8_general_ci	480.0 KiB	-
<input type="checkbox"/> product_class		110	InnoDB	utf8_general_ci	16.0 KiB	-
<input checked="" type="checkbox"/> sales_fact_1997		~86,868	InnoDB	utf8_general_ci	19.1 MiB	-
<input type="checkbox"/> time_by_day		730	InnoDB	utf8_general_ci	144.0 KiB	-
<input type="checkbox"/> view_sales_fact_1997		~0	View	---	-	-
6 tables	Sum	~99,549	InnoDB	utf8mb4_general_ci	23.8 MiB	0 B

☐ Check all With selected:

Showing rows 0 - 24 (86868 total, Query took 0.0003 seconds.)

```
SELECT * FROM `sales_fact_1997`
```

1 > >> | Number of rows: 25 | Filter rows: Search this table | Sort by key:

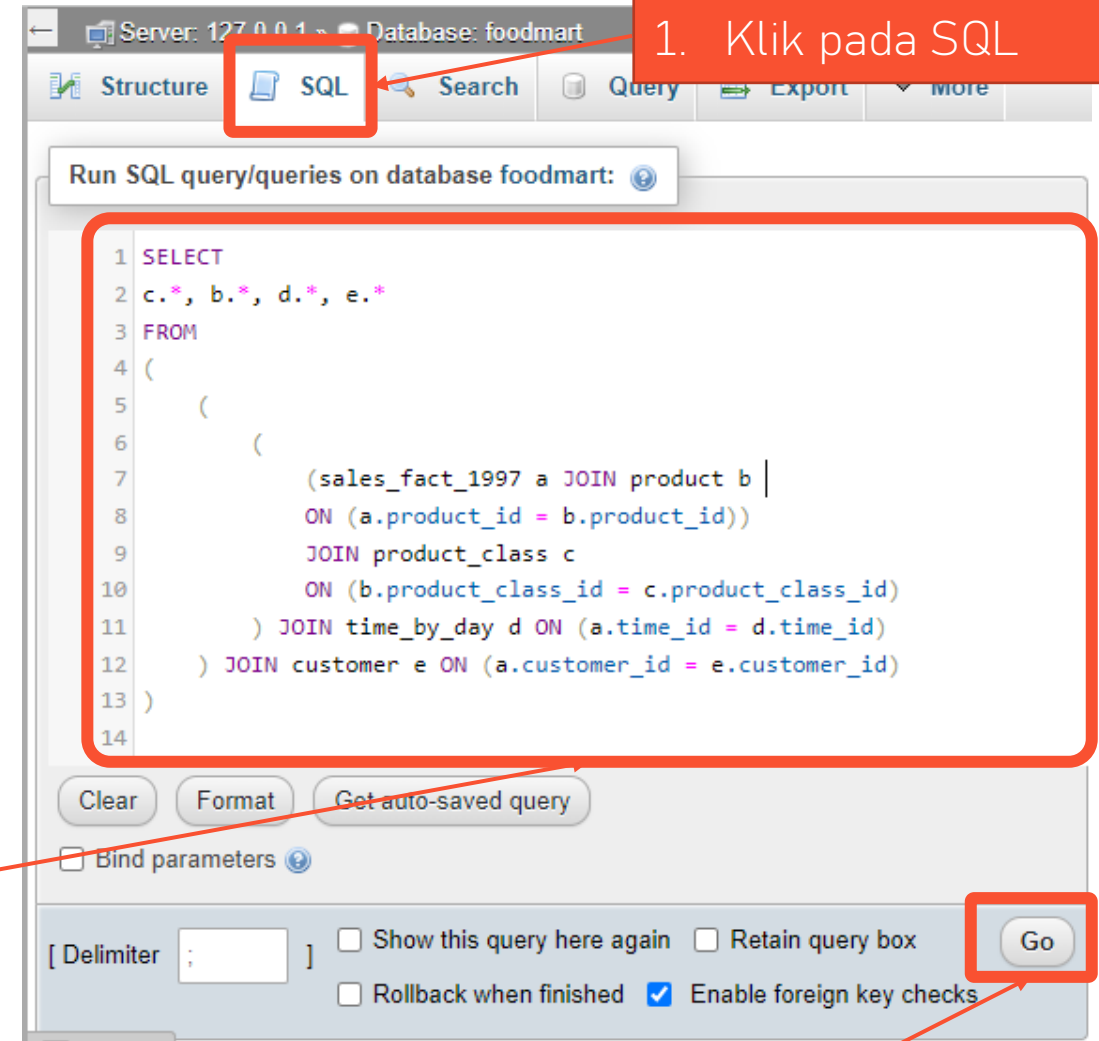
product_id	time_id	customer_id	promotion_id	store_id	store_sales	store_cost	unit_sales
337	371	6280	0	2	1.5000	0.5100	2.0000
1512	371	6280	0	2	1.6200	0.6318	3.0000
963	371	4018	0	2	2.4000	0.7200	1.0000
181	371	4018	0	2	2.7900	1.0323	3.0000
1383	371	4018	0	2	5.1800	2.1756	2.0000
1306	371	4018	0	2	7.4100	2.7417	3.0000
1196	371	1418	0	2	5.8400	1.9856	2.0000
360	371	1418	0	2	2.6200	1.0480	2.0000
1242	371	1418	0	2	3.9600	1.7028	2.0000
154	371	1418	0	2	1.9600	0.7252	1.0000
483	371	4382	0	2	3.8800	1.7072	2.0000
77	371	1293	0	2	5.6000	2.8000	2.0000
533	371	1293	0	2	4.8400	2.3232	2.0000
310	371	1293	0	2	0.7600	0.3420	1.0000
1392	371	1293	0	2	0.8300	0.3735	1.0000
1303	394	9305	0	2	1.3600	0.4352	2.0000
748	394	9305	0	2	4.4000	1.9360	2.0000
1270	394	9305	0	2	2.6900	1.0491	1.0000
311	394	5649	0	2	6.4500	2.3865	3.0000
194	394	5649	0	2	3.3600	1.5120	1.0000
544	394	5649	0	2	2.9500	1.0325	1.0000
610	394	6319	0	2	2.2700	0.7491	1.0000
854	394	6319	0	2	1.6000	0.6880	1.0000
538	394	6319	0	2	1.8600	0.7440	2.0000
72	394	1771	0	2	5.4800	2.0824	2.0000

3. Tabel ini masih menggunakan data id, faktanya id ini tidak berarti apa-apa dan untuk mengetahui maknanya dapat mengacu pada tabel lain yang memiliki id yang sama dengan yang ada pada tabel fakta. Untuk mengubah id menjadi data maka gunakan perintah join pada query.

JOIN TABEL

2. Isikan query berikut:

```
SELECT
    c.*, b.*, d.*, e.*
FROM
(
    (
        (
            (sales_fact_1997 a JOIN product b
            ON (a.product_id = b.product_id))
            JOIN product_class c
            ON (b.product_class_id = c.product_class_id)
        ) JOIN time_by_day d ON (a.time_id = d.time_id)
    ) JOIN customer e ON (a.customer_id = e.customer_id)
)
```



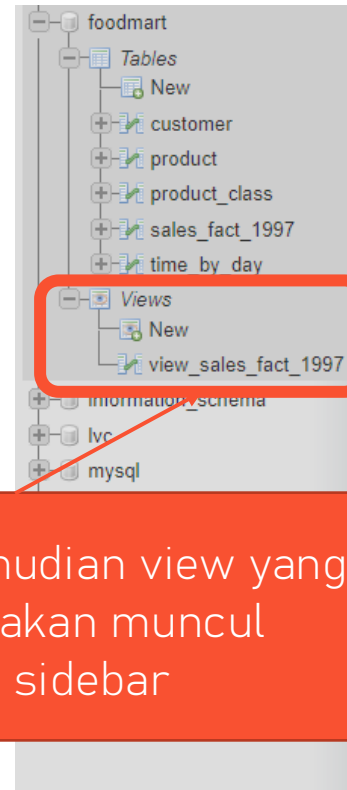
3. Klik pada Go untuk menampilkan hasil SQL

SAVE VIEW

product_class_id	product_subcategory	product_category	product_department	product_family	product_class_id
1	Nuts	Specialty	Produce	Food	1
1	Nuts	Specialty	Produce	Food	1
1	Nuts	Specialty	Produce	Food	1
1	Nuts	Specialty	Produce	Food	1
1	Nuts	Specialty	Produce	Food	1

Query results operations

Print Copy to clipboard Export Display chart Create view



1. Tabel akan menampilkan informasi tambahan yang terdapat pada tabel lain

3. Isi nama view dengan view_sales_fact_1997

Create view

Details

OR REPLACE ☐

ALGORITHM

Definer

SQL SECURITY

VIEW name

Column names

AS

```
1 SELECT
2 c.*, b.*, d.*, e.*
3 FROM
4 /
```

Format

WITH CHECK OPTION

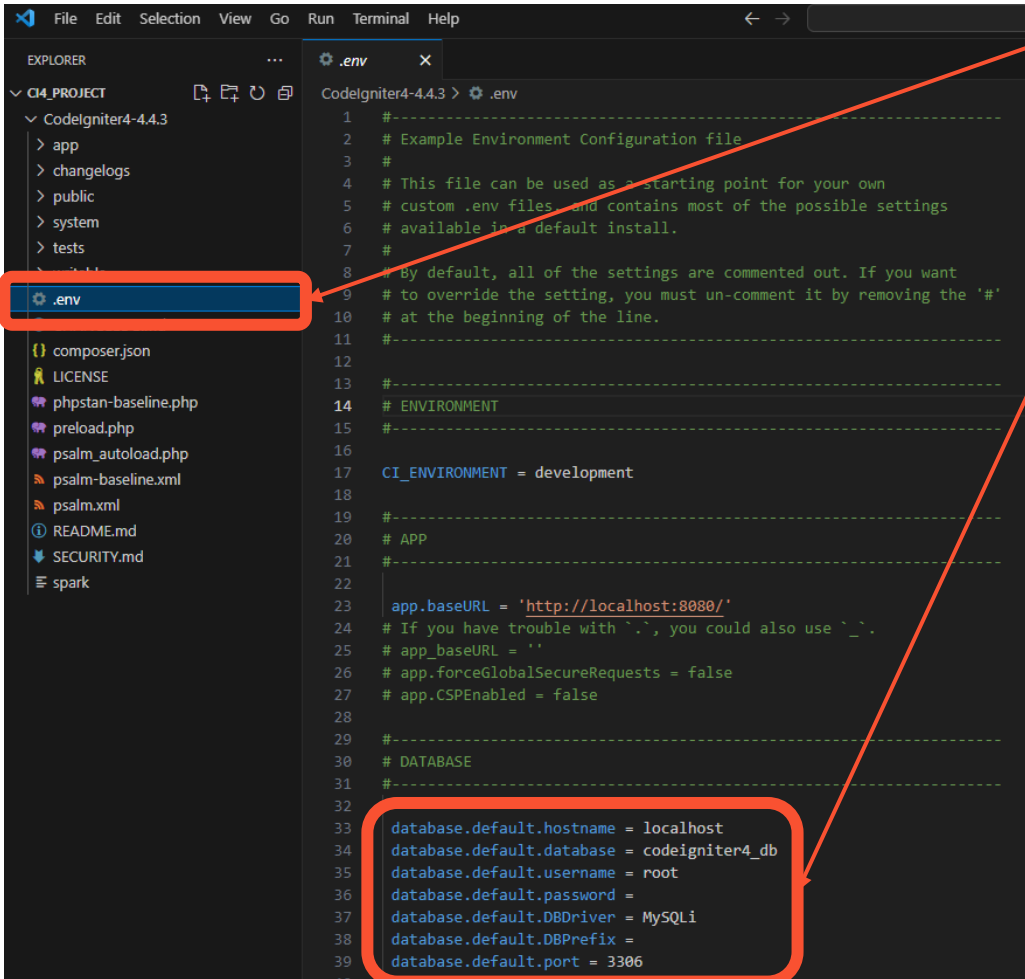
Go Close

2. Klik pada Create view

4. Klik Go, kemudian view yang baru dibuat akan muncul pada bagian sidebar

Notes: Selanjutnya tabel yang akan digunakan adalah tabel **view_sales_fact_1997**

GANTI DATABASE



1. Buka file .env

```
2. database.default.hostname = localhost
   database.default.database = foodmart
   database.default.username = root
   database.default.password =
   database.default.DBDriver = MySQLi
   database.default.DBPrefix =
   database.default.port = 3306
```

Nama database “codeigniter4_db” di atas diganti dengan nama **foodmart** yang Anda download.

MODEL FOODMART

- Buat file baru pada app > Models > Foodmart.php

- Model Foodmart.php berisi kode:

```
<?php
namespace App\Models;
use CodeIgniter\Model;
class Foodmart extends Model{
    protected $table = 'view_sales_fact_1997';
    public function getDataFoodmart(){
        return $this->findAll(1000);
    }
}
```

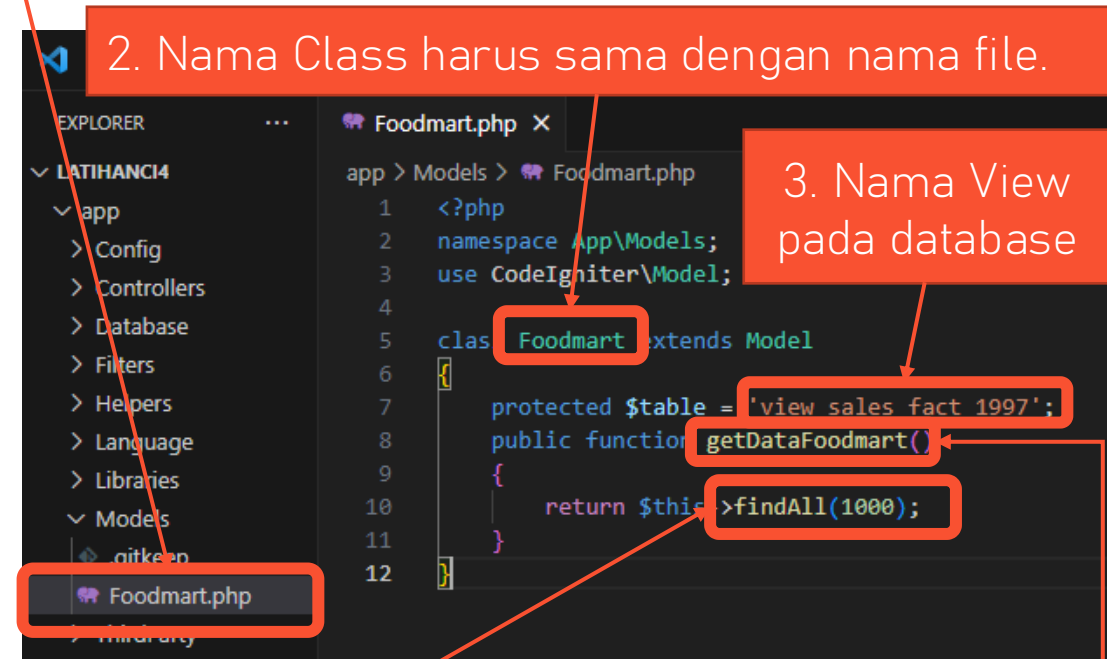
1. File Foodmart.php

2. Nama Class harus sama dengan nama file.

3. Nama View pada database

4. Nama Method

5. Fungsi findAll(1000) berarti Select * From view_sales_fact_1997 Limit 1000. Anda bisa menambahkan offset (dimulai dari row tertentu misal row 25) dengan cara findAll(1000,25). Offset bisa digunakan untuk memindahkan page.



CONTROLLER

FOODMARTCONTROLLER

- Buat file baru pada app > Controllers > FoodmartController.php
- Controller FoodmartController.php berisi kode:

```
<?php
namespace App\Controllers;
use App\Models\Foodmart;
class FoodmartController extends BaseController
{
    public function index()
    {
        $model = model(Foodmart::class);
        $data['foodmart'] = $model->getDataFoodmart();
        return view('header', $data).view('menu').
            view('foodmart').view('footer');
    }
}
```

1. File FoodmartController.php

2. Memanggil Model Foodmart

3. Nama Class harus sama dengan nama file.

4. Pemanggilan Model Foodmart dimasukkan ke Variabel \$model untuk penyederhanaan kode program.

5. Hasil dari method getDataFoodmart pada Model Foodmart disimpan dalam variabel 'foodmart'. Kemudian dikirimkan ke view header.



VIEW

FOODMART.PHP

- Buat file baru pada app > Views > foodmart.php

- View foodmart.php berisi kode:

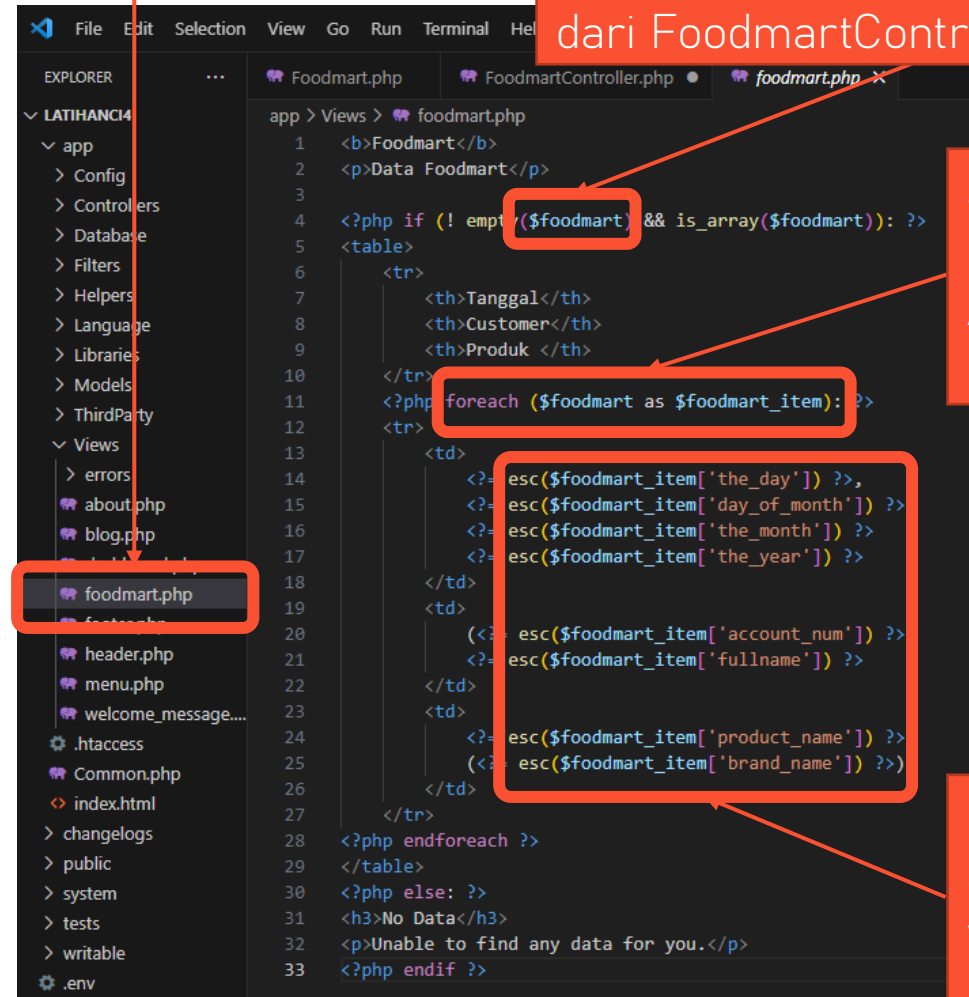
```
<b>Foodmart</b> <p>Data Foodmart</p>
<?php if (! empty($foodmart) && is_array($foodmart)): ?>
<table>
  <tr><th>Tanggal</th><th>Customer</th><th>Produk</th></tr>
  <?php foreach ($foodmart as $foodmart_item): ?>
  <tr>
    <td> <?= esc($foodmart_item['the_day']) ?>,
    <?= esc($foodmart_item['day_of_month']) ?>
    <?= esc($foodmart_item['the_month']) ?>
    <?= esc($foodmart_item['the_year']) ?> </td>
    <td> (<?= esc($foodmart_item['account_num']) ?>)
    <?= esc($foodmart_item['fullname']) ?> </td>
    <td> <?= esc($foodmart_item['product_name']) ?>
    (<?= esc($foodmart_item['brand_name']) ?>) </td>
  </tr>
<?php endforeach ?> </table>
<?php else: ?>
<h3>No Data</h3>
<p>Unable to find any data for you.</p>
<?php endif ?>
```

1. File foodmart.php

2. Variabel foodmart yang dikirim dari FoodmartController.php

3. Data foodmart masih berbentuk Array, maka harus diparsing

4. Gunakan nama atribut pada tabel view_sales_fact_1997

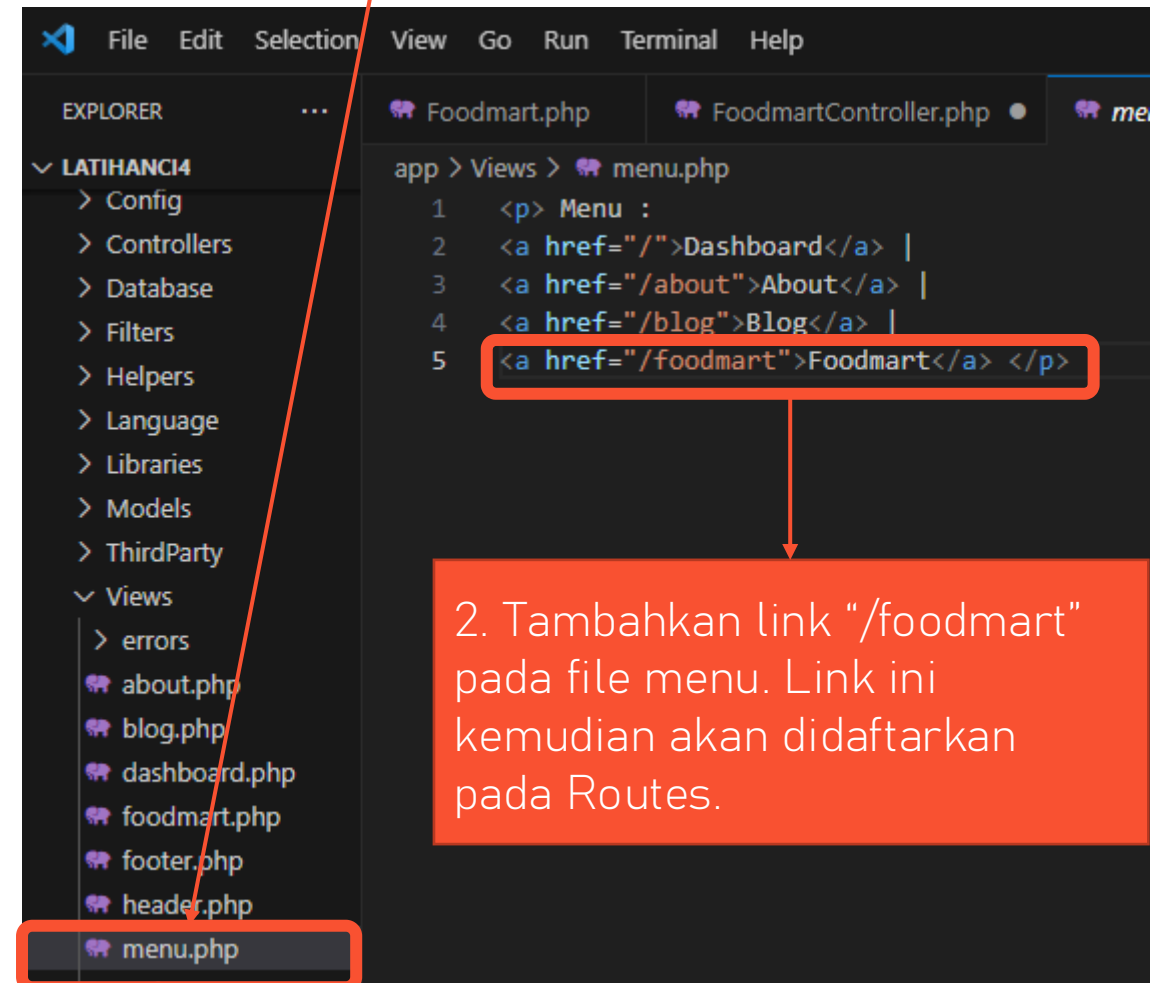


UPDATE VIEW MENU.PHP

- Buka app > Views > menu.php
- Update kode pada menu.php menjadi:

```
<p> Menu :  
<a href="/">Dashboard</a> |  
<a href="/about">About</a> |  
<a href="/blog">Blog</a> |  
<a href="/foodmart">Foodmart</a> </p>
```

1. File menu.php

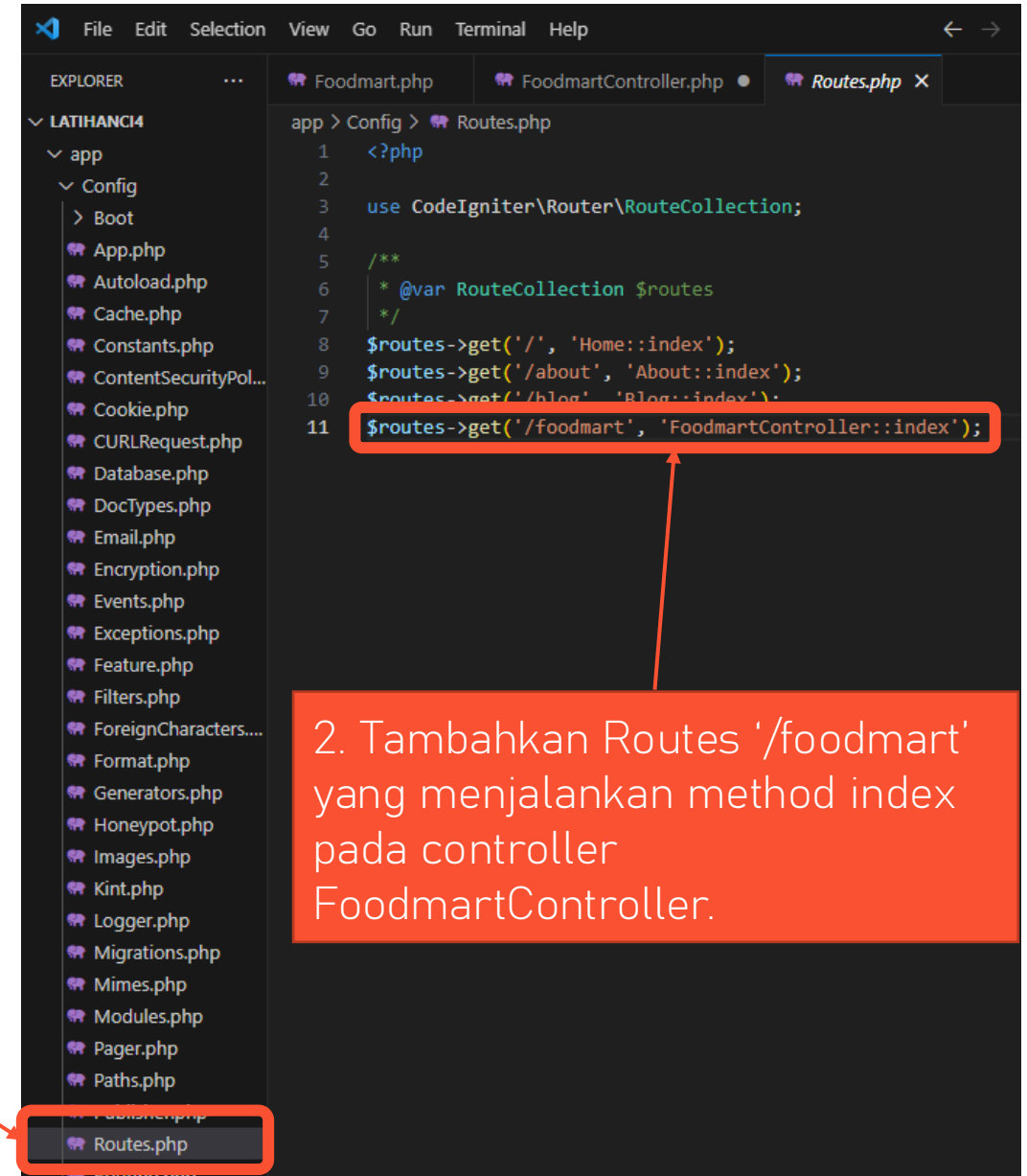


UPDATE ROUTES

- Buka app > Config > Routes.php
- Tambahkan kode pada Routes.php di bagian bawah:

```
$routes->get('/foodmart',  
    'FoodmartController::index');
```

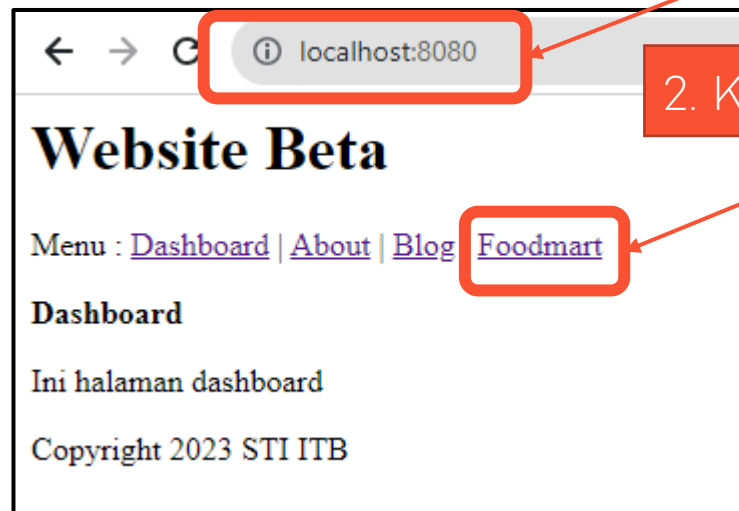
1. File Routes.php



JALANKAN

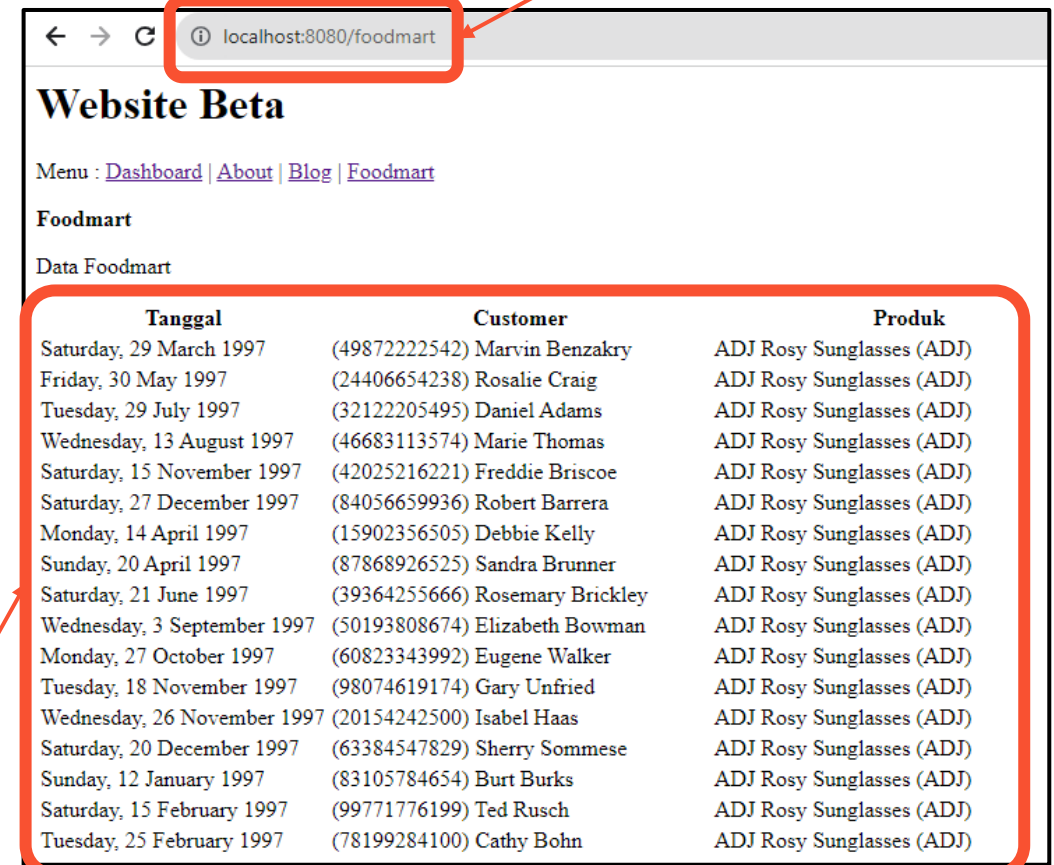
LOCALHOST:8080/FOODMART

1. Akses localhost:8080 pada browser.



2. Klik link Foodmart

3. Link '/foodmart' yang sudah didaftarkan pada Routes, diakses melalui menu.

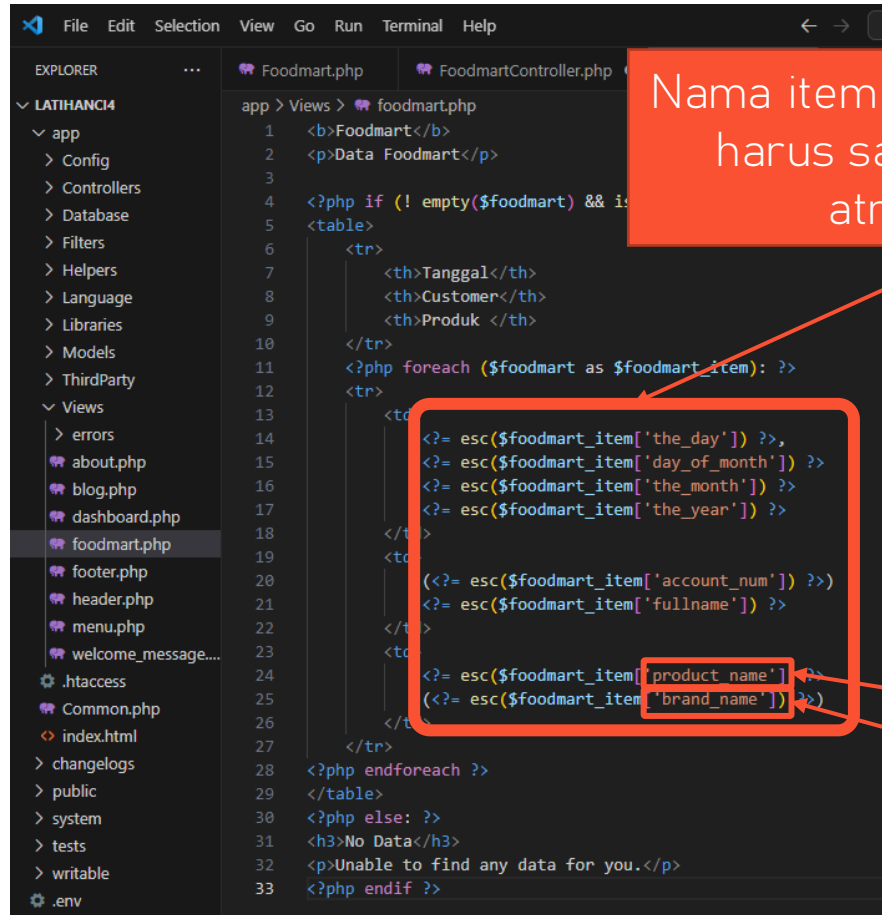


4. Hasil findAll(1000) yang sudah dikirimkan ke view foodmart.php dapat ditampilkan dalam format tabel. Jika halaman web Anda menghasilkan error 505 maka kecilkan limit dari 1000 menjadi 100. Hal ini terjadi akibat resource komputasi Anda tidak memadai atau PHP membatasi besaran cache memory server.

LOCALHOST:8080/FOODMART

TIDAK JALAN

Selain waspada terhadap error 505 Anda juga harus waspada terhadap **typo** pada nama atribut tabel. Typo ini tidak dapat dideteksi oleh Visual Studio Code.



```
1 <b>Foodmart</b>
2 <p>Data Foodmart</p>
3
4 <?php if (! empty($foodmart) && is_array($foodmart)) :>
5 <table>
6
7 <tr>
8 <th>Tanggal</th>
9 <th>Customer</th>
10 <th>Produk </th>
11 </tr>
12 <?php foreach ($foodmart as $foodmart_item): ?>
13 <tr>
14 <td>
15 <?= esc($foodmart_item['the_day']) ?>,
16 <?= esc($foodmart_item['day_of_month']) ?>
17 <?= esc($foodmart_item['the_month']) ?>
18 <?= esc($foodmart_item['the_year']) ?>
19 </td>
20 <td>
21 (<?= esc($foodmart_item['account_num']) ?>)
22 <?= esc($foodmart_item['fullname']) ?>
23 </td>
24 <td>
25 <?= esc($foodmart_item['product_name']) ?>
26 <?= esc($foodmart_item['brand_name']) ?>
27 </td>
28 </tr>
29 <?php endforeach ?>
30 </table>
31 <?php else: ?>
32 <h3>No Data</h3>
33 <p>Unable to find any data for you.</p>
34 <?php endif ?>
```

Nama item ini tidak boleh salah, harus sama dengan nama atribut di tabel.

product_family	brand_name	product_name
Non-Consumable	ADJ	ADJ Rosy Sunglasses
Non-Consumable	ADJ	ADJ Rosy Sunglasses
Non-Consumable	ADJ	ADJ Rosy Sunglasses

AKHIR BAGIAN 4

Lanjutkan ke Bagian 5 – Building Response

CODEIGNITER 4

BAGIAN 5- BUILDING RESPONSE

Read API, Testing API menggunakan Postman

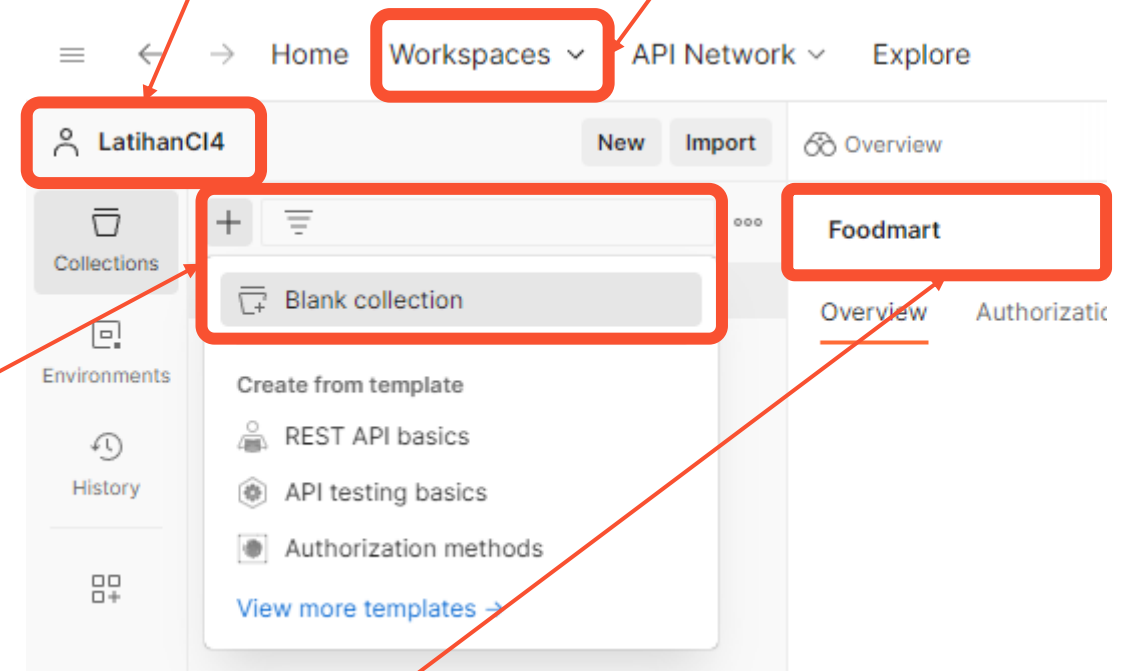
INSTALL POSTMAN

- Download Postman di :
<https://www.postman.com/downloads/>
- Install di sistem Anda.
- Buka Postman yang sudah diinstall.

3. Klik tanda + dan pilih Blank collection

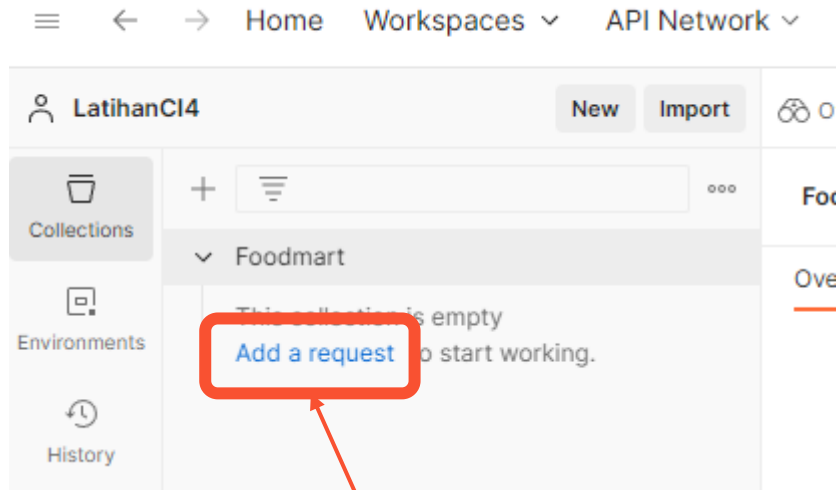
2. Beri nama misalnya 'LatihanCI4'

1. Create Workspace baru.



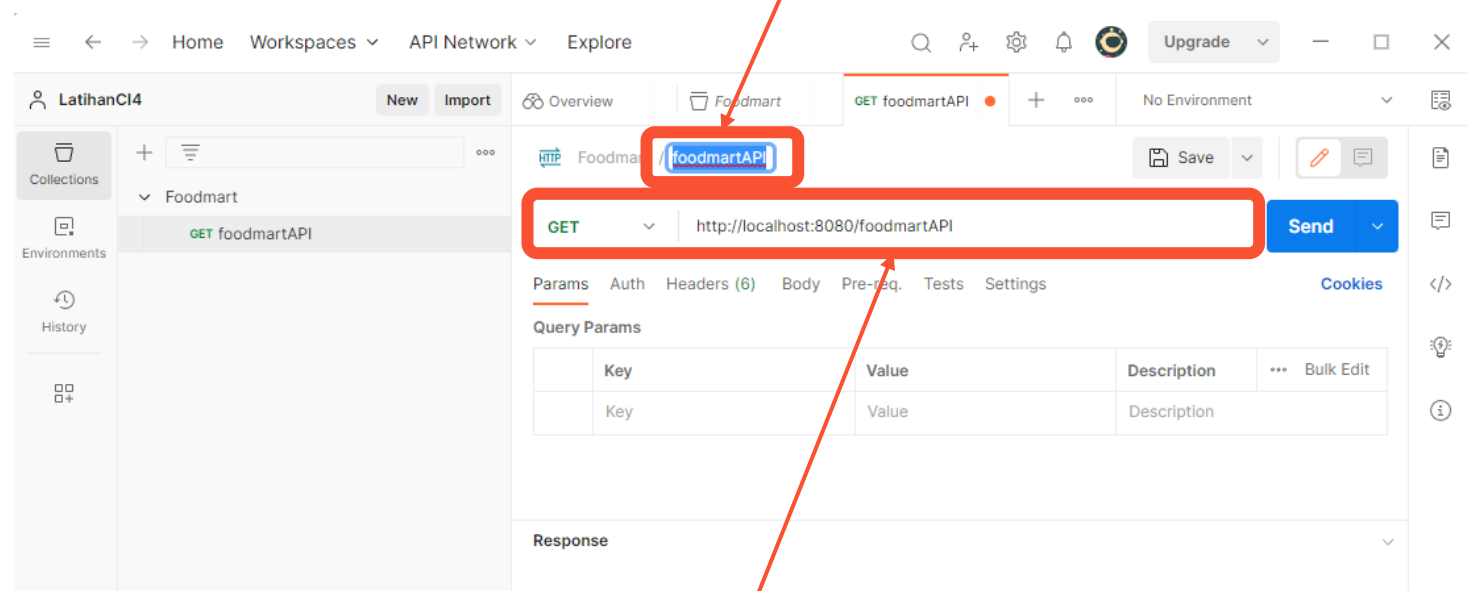
4. Beri nama collection baru misalnya dengan nama 'Foodmart'.

ADD GET REQUEST PADA POSTMAN



1. Klik pada Add a request

2. Beri nama request baru misalnya dengan 'foodmartAPI'



3. Pilih jenis request GET dan tuliskan alamat request Anda di sini, misalnya 'http://localhost:8080/foodmartAPI'. Saat ini jika tombol Send diklik maka akan memberikan Response error. Request ini akan dibangun pada tahap selanjutnya

CONTROLLER

FOODMARTAPI

- Buat file controller baru pada app > Controllers > FoodmartAPI.php
- Ketikkan kode berikut pada FoodmartAPI.php :

```
<?php
namespace App\Controllers;
use CodeIgniter\RESTful\ResourceController;
use App\Models\Foodmart;
class FoodmartAPI extends ResourceController
{
    public function index(){
        $model = model(Foodmart::class);
        $data = [ 'message' => 'success',
                  'foodmart' => $model->getDataFoodmart()];
        return $this->respond($data,200);
    }
}
```

- Kode ini mirip dengan controller FoodmartController.php, penambahannya pada **ResourceController** dan **respond(\$data,200)**.

1. Buat file controller baru pada app > Controllers > FoodmartAPI.php

2. Penggunaan RESTful\ResourceController

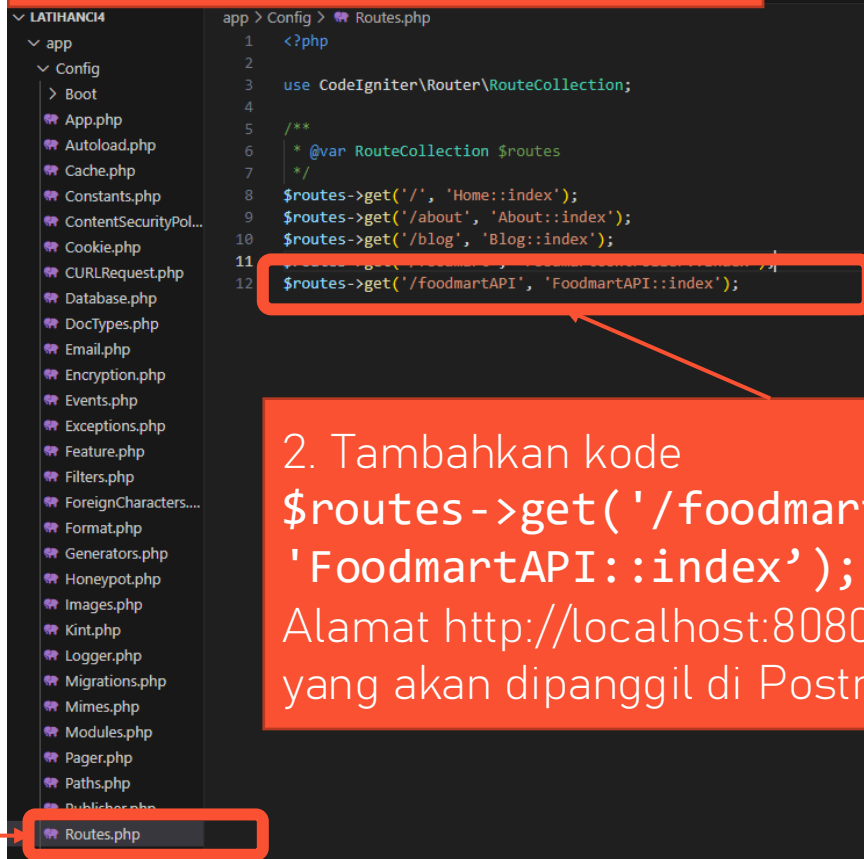
```
File Edit Selection View Go Run Terminal Help
EXPLORER
LATIHANCI4
  app
    Config
    Controllers
      About.php
      BaseController.php
      Blog.php
      FoodmartAPI.php
      FoodmartController.php
      Home.php
    Database
    Filters
    Helpers
    Language
    Libraries
    Models
    ThirdParty
    Views
FoodmartAPI.php X
app > Controllers > FoodmartAPI.php
1 <?php
2
3 namespace App\Controllers;
4 use CodeIgniter\RESTful\ResourceController;
5 use App\Models\Foodmart;
6
7 class FoodmartAPI extends ResourceController
8 {
9     public function index()
10     {
11         $model = model(Foodmart::class);
12         $data = [
13             'message' => 'success',
14             'foodmart' => $model->getDataFoodmart()
15         ];
16         return $this->respond($data,200);
17     }
18 }
19
```

3. Return tidak ke view melainkan berupa API response

UPDATE ROUTES

DAN UPDATE MENU.PHP

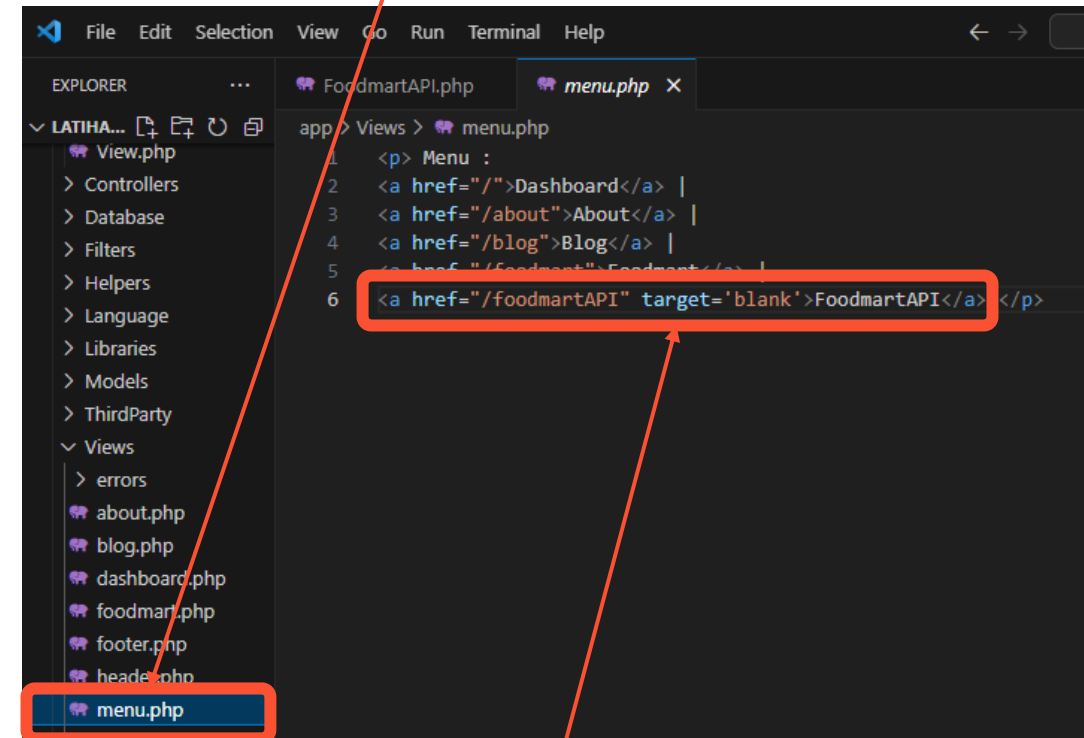
1. Buka app > Config > Routes.php



```
1 <?php
2
3 use CodeIgniter\Router\RouteCollection;
4
5 /**
6  * @var RouteCollection $routes
7  */
8 $routes->get('/', 'Home::index');
9 $routes->get('/about', 'About::index');
10 $routes->get('/blog', 'Blog::index');
11
12 $routes->get('/foodmartAPI', 'FoodmartAPI::index');
```

2. Tambahkan kode
`$routes->get('/foodmartAPI',
'FoodmartAPI::index');`
Alamat `http://localhost:8080/foodmartAPI` ini
yang akan dipanggil di Postman.

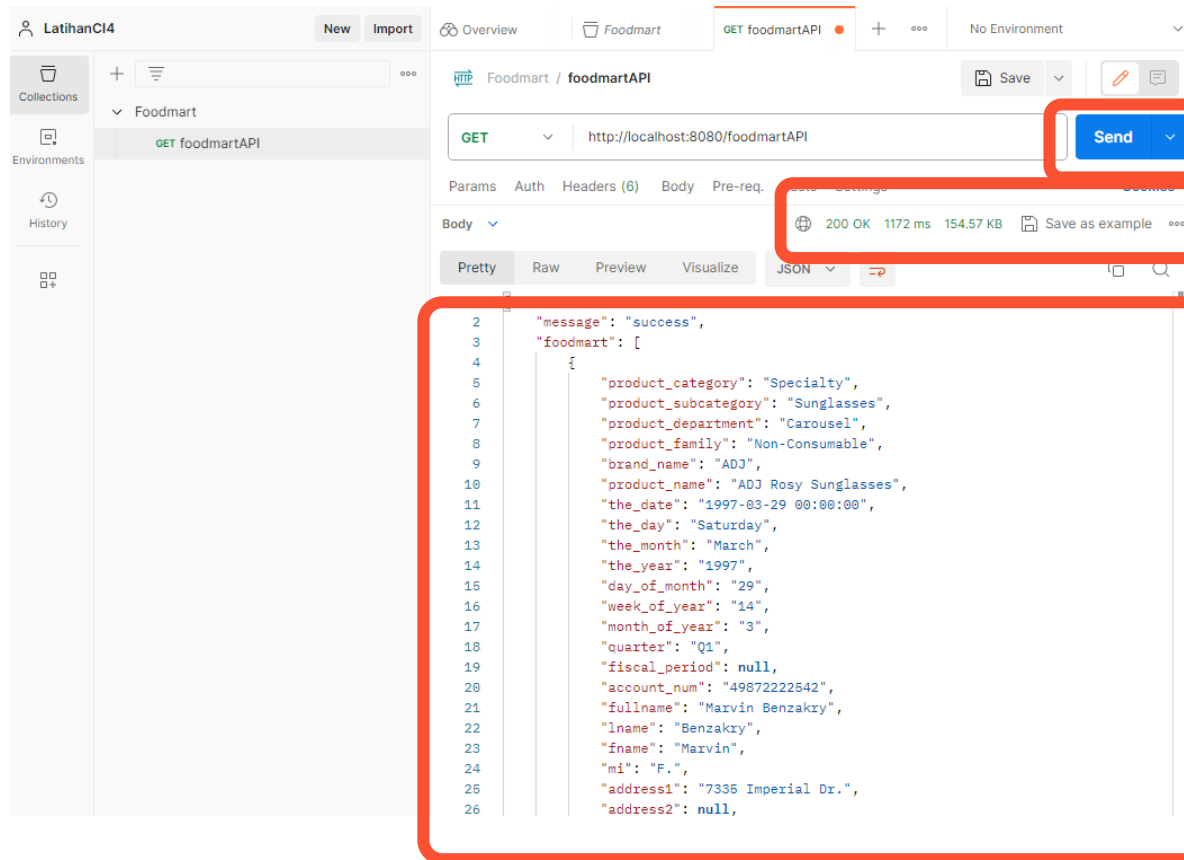
3. Buka app > View > menu.php



```
1 <p> Menu :
2 <a href="/">Dashboard</a> |
3 <a href="/about">About</a> |
4 <a href="/blog">Blog</a> |
5 <a href="/foodmartAPI">FoodmartAPI</a> |
6 <a href="/foodmartAPI" target='blank'>FoodmartAPI</a> </p>
```

4. Tambahkan link
`<a href="/foodmartAPI"
target='blank'>FoodmartAPI`

TEST API PADA POSTMAN



1. Klik Send

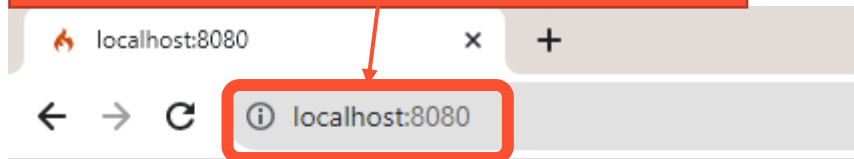
2. Informasi bahwa response berhasil

3. Respond berupa data yang dikirimkan dari sistem

JALANKAN

LOCALHOST:8080/FOODMARTAPI

1. Akses localhost:8080



Website Beta

Menu : [Dashboard](#) | [About](#) | [Blog](#) | [Foodmart](#) | [FoodmartAPI](#)

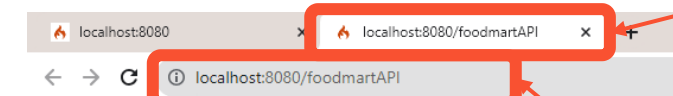
Dashboard

Ini halaman dashboard

Copyright 2023 STI ITB

2. Klik link FoodmartAPI

3. Target ke blank new page



4. Link yang sudah didaftarkan pada Routes

```
{
  "message": "success",
  "foodmart": [
    {
      "product_category": "Specialty",
      "product_subcategory": "Sunglasses",
      "product_department": "Carousel",
      "product_family": "Non-Consumable",
      "brand_name": "ADJ",
      "product_name": "ADJ Rosy Sunglasses",
      "the_date": "1997-03-29 00:00:00",
      "the_day": "Saturday",
      "the_month": "March",
      "the_year": "1997",
      "day_of_month": "29",
      "week_of_year": "14",
      "month_of_year": "3",
      "quarter": "Q1",
      "fiscal_period": null,
      "account_num": "49872222542",
      "fullname": "Marvin Benzakry",
      "lname": "Benzakry",
      "fname": "Marvin",
      "mi": "F.",
      "address1": "7335 Imperial Dr.",
      "address2": null,
      "address3": null,
      "address4": null,
      "city": "Sedro Woolley",
      "state_province": "WA",
      "postal_code": "10751",
      "country": "USA",
      "phone1": "553-555-7479",
      "phone2": "403-555-8615",
      "birthdate": "1945-05-22",
      "marital_status": "S",
      "yearly_income": "$110K - $130K",
      "gender": "M",
      "total_children": "0",
      "num_children_at_home": "0",
      "education": "Bachelors Degree",
      "date_acct_opened": "1994-08-18",
      "member_card": "Bronze",
      "occupation": "Management",
      "houseowner": "N",
      "num_cars_owned": "4"
    }
  ],
}
```

4. Respond berupa data yang dikirimkan dari sistem dicetak pada halaman ini.

AKHIR BAGIAN 5

Lanjutkan ke Bagian 6 – Authentication

The background of the slide features a complex network of white dots connected by thin white lines, creating a web-like or molecular structure. The dots vary in size, and the lines are of varying thickness, giving it a dynamic and interconnected appearance. The overall color scheme is a gradient of warm tones, from a deep orange at the top to a lighter, almost white, at the bottom.

CODEIGNITER 4

BAGIAN 6- AUTHENTICATION

Membuat Tabel Users, Login Form, Session

BUAT TABEL USERS

1. Klik New pada Tables di phpmyadmin

2. Ketikkan nama tabel misalnya 'users'

3. Buat 3 kolom atribut beri nama 'name', 'email', dan 'password'. Tipe data setiap atribut adalah VARCHAR dengan panjang 225 karakter

4. Set email sebagai PRIMARY key

5. Klik Save

6. Setelah di save maka tabel users akan muncul di sidebar

The screenshot shows the phpMyAdmin 'New Table' wizard. The 'Table name' field is 'users'. The 'Columns' table has three rows: 'name' (VARCHAR, 225), 'email' (VARCHAR, 225), and 'password' (VARCHAR, 225). The 'email' column is set as the 'PRIMARY' key. The 'Storage Engine' is set to 'InnoDB'. The 'Save' button is highlighted at the bottom right.

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index
name	VARCHAR	225	None				
email	VARCHAR	225	None				PRIMARY
password	VARCHAR	225	None				

MEMBUAT USER DUMMY

- Akses link <https://www.md5hashgenerator.com/>
- Masukkan password dummy Anda, misalnya 'password', maka akan menghasilkan MD5 hash adalah '5f4dcc3b5aa765d61d8327deb882cf99'

The screenshot shows the MD5 Hash Generator website. The browser address bar shows the URL <https://www.md5hashgenerator.com/>. The page title is "MD5 Hash Generator". Below the title, there is a text input field containing the word "password". A "Generate" button is visible. Below the input field, the results are displayed: "Your String" is "password", "MD5 Hash" is "5f4dcc3b5aa765d61d8327deb882cf99", and "SHA1 Hash" is "5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8".

1. Akses link <https://www.md5hashgenerator.com/>

2. Masukkan plaintext misalnya 'password'

3. Hasil hash MD5 untuk plaintext 'password.'

4. Klik pada tabel users

5. Masukkan Nama

6. Masukkan email

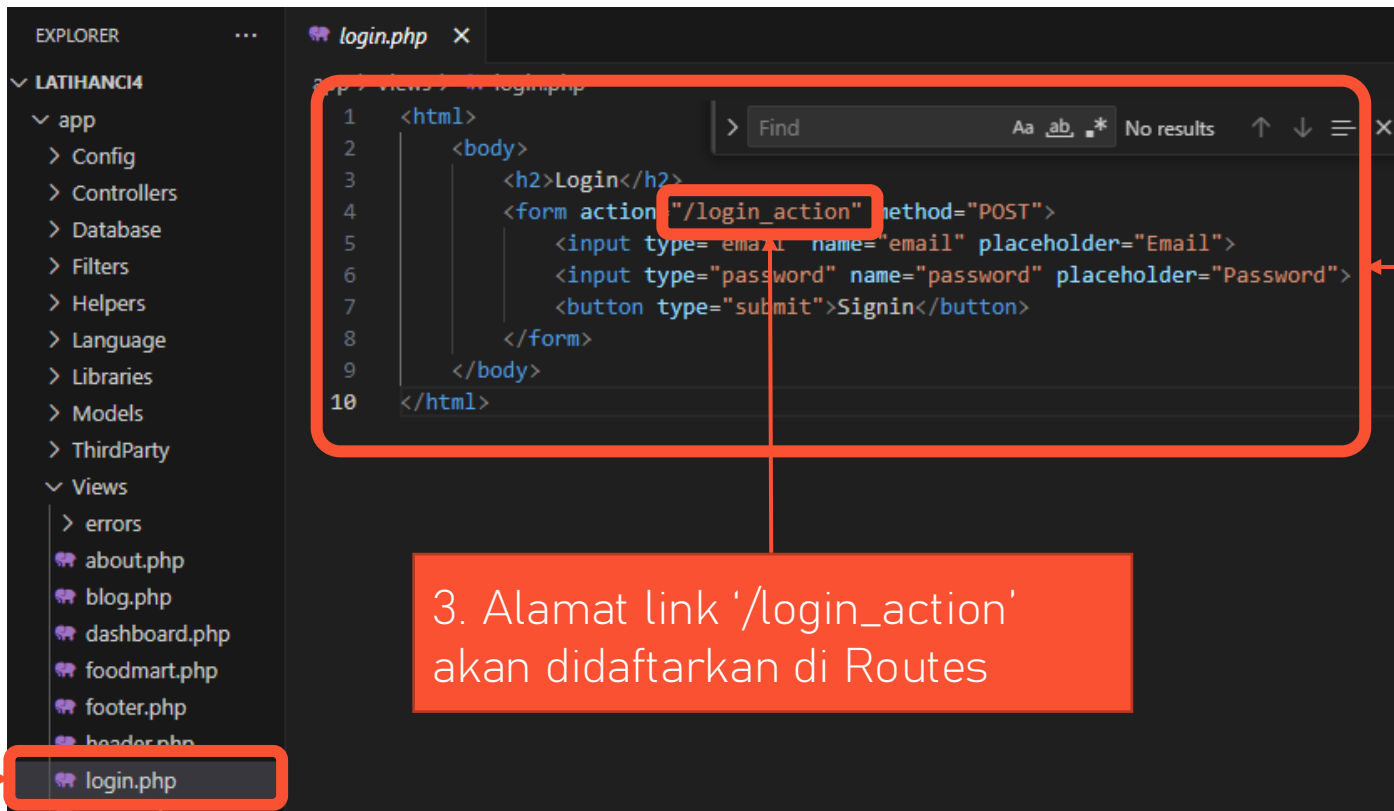
7. Masukkan MD5 password Anda

8. Klik Go

The screenshot shows the phpMyAdmin interface. The left sidebar shows the database structure, with the 'users' table selected. The main area shows the 'Table: users' form. The 'name' field is filled with 'Fadhil Hidayat', the 'email' field is filled with 'fadhil@email.com', and the 'password' field is filled with the MD5 hash '5f4dcc3b5aa765d61d8327deb882cf99'. A 'Go' button is at the bottom right of the form.

VIEW LOGIN

1. Buat file view baru pada app > Views > login.php



3. Alamat link '/login_action' akan didaftarkan di Routes

2. Ketikkan kode berikut:

```
<html>
<body>
  <h2>Login</h2>
  <form action="/login_action"
    method="POST">
    <input type="email" name="email"
      placeholder="Email">
    <input type="password"
      name="password"
      placeholder="Password">
    <button type="submit">
      Signin</button>
  </form>
</body>
</html>
```

MODEL LOGIN

1. Buat file view baru pada
app > Models > Login.php

2. Ketikkan kode berikut:

```
<?php
namespace App\Models;
use CodeIgniter\Model;
class Login extends Model
{
    public function getDataUsers($email, $password)
    {
        $db = \Config\Database::connect();
        $queryString = 'SELECT name FROM users WHERE
                        email = "'. $email. '"
                        AND password = "'. $password. '"';
        $query = $db->query($queryString);
        $results = $query->getResult();
        return count($results);
    }
}
```

```
1 <?php
2 namespace App\Models;
3 use CodeIgniter\Model;
4 class Login extends Model
5 {
6     public function getDataUsers($email, $password)
7     {
8         $db = \Config\Database::connect();
9         $queryString = 'SELECT name FROM users WHERE
10 email = "'. $email. '"
11 AND password = "'. $password. '"';
12 $query = $db->query($queryString);
13 $results = $query->getResult();
14 return count($results);
15 }
16 }
```

3. Method getDataUsers membutuhkan
input variabel \$email, dan \$password

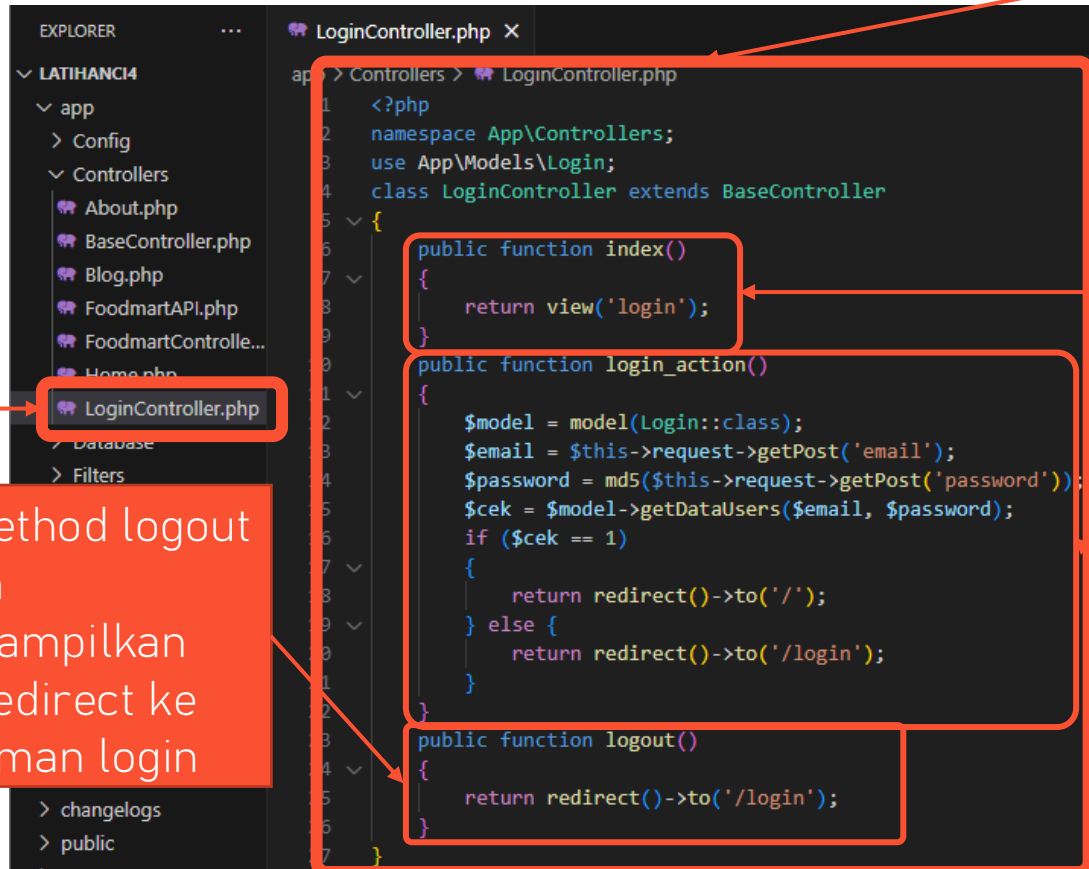
4. Query untuk mencari name pada tabel
users yang cocok dengan input variabel
\$email dan \$password

5. Return berupa jumlah data (integer)
yang cocok dengan query di atas.

CONTROLLER

LOGINCONTROLLER

1. Buat file controller baru pada app > Controllers > LoginController.php



```
1 <?php
2 namespace App\Controllers;
3 use App\Models\Login;
4 class LoginController extends BaseController
5 {
6     public function index()
7     {
8         return view('login');
9     }
10    public function login_action()
11    {
12        $model = model(Login::class);
13        $email = $this->request->getPost('email');
14        $password = md5($this->request->getPost('password'));
15        $cek = $model->getDataUsers($email, $password);
16        if ($cek == 1)
17        {
18            return redirect()->to('/');
19        } else {
20            return redirect()->to('/login');
21        }
22    }
23    public function logout()
24    {
25        return redirect()->to('/login');
26    }
27 }
```

5. Method logout akan menampilkan meredirect ke halaman login

2. Ketikkan kode berikut:

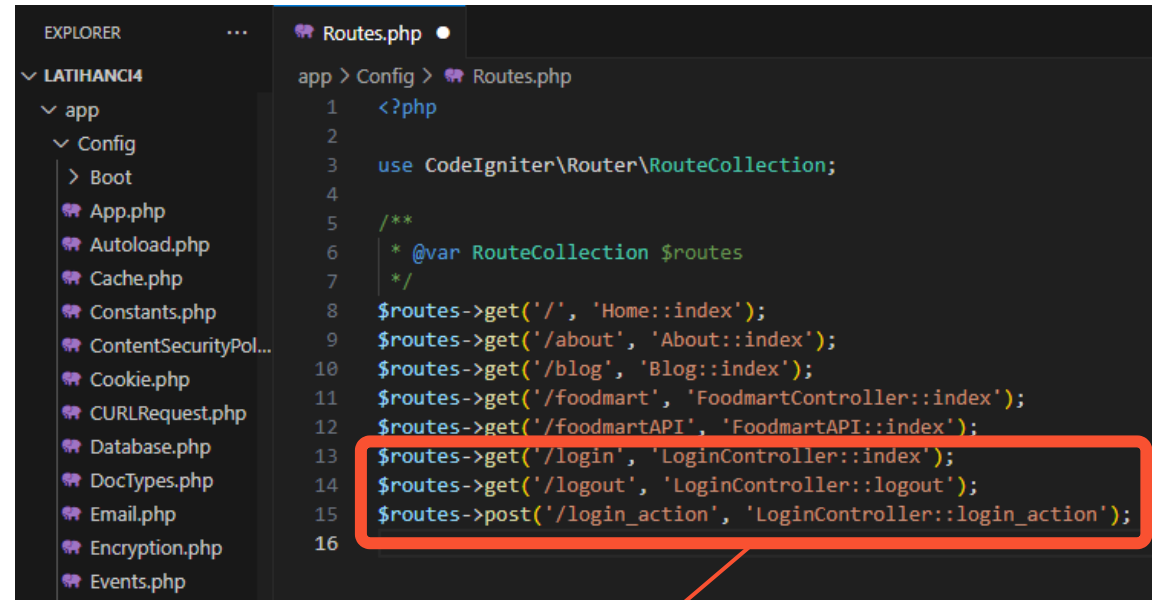
```
<?php
namespace App\Controllers;
use App\Models\Login;
class LoginController extends BaseController{
    public function index(){
        return view('login');
    }
    public function login_action(){
        $model = model(Login::class);
        $email = $this->request->getPost('email');
        $password = md5($this->request->getPost('password'));
        $cek = $model->getDataUsers($email, $password);
        if ($cek == 1){
            return redirect()->to('/');
        } else {
            return redirect()->to('/login');
        }
    }
    public function logout() {
        return redirect()->to('/login');
    }
}
```

3. Method index akan menampilkan view login

4. Method login_action, form login akan diarahkan ke mehtod ini. Password di hash menggunakan MD5 sebelum dikirimkan ke Model Login Method getDataUsers.

UPDATE ROUTES

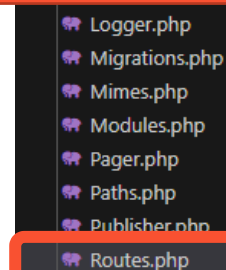
1. Buka file Routes.php baru pada
app > Config > Routes.php



```
1 <?php
2
3 use CodeIgniter\Router\RouteCollection;
4
5 /**
6  * @var RouteCollection $routes
7  */
8 $routes->get('/', 'Home::index');
9 $routes->get('/about', 'About::index');
10 $routes->get('/blog', 'Blog::index');
11 $routes->get('/foodmart', 'FoodmartController::index');
12 $routes->get('/foodmartAPI', 'FoodmartAPI::index');
13 $routes->get('/login', 'LoginController::index');
14 $routes->get('/logout', 'LoginController::logout');
15 $routes->post('/login_action', 'LoginController::login_action');
16
```

2. Update file Routes.php dengan menambahkan kode:
`$routes->get('/login', 'LoginController::index');`
`$routes->get('/logout', 'LoginController::logout');`
`$routes->post('/login_action', 'LoginController::login_action');`

***Notes :** pada tahap ini Anda sudah dapat mencoba form login Anda meskipun tanpa melakukan registrasi session.



Routes.php

AKSES

LOCALHOST:8080/LOGIN

1. Akses localhost:8080/login pada browser

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/login'. The page content includes a heading 'Login' and a form with two input fields: 'Email' and 'Password', and a 'Signin' button. Red annotations highlight the address bar, the input fields, and the button, with arrows pointing to the numbered instructions.

2. Masukkan alamat email yang sudah didaftarkan pada tabel users di database Foordmart

3. Masukkan password berupa plaintext (bukan MD5) yang sebelumnya MD5 nya sudah didaftarkan pada tabel users di database Foordmart

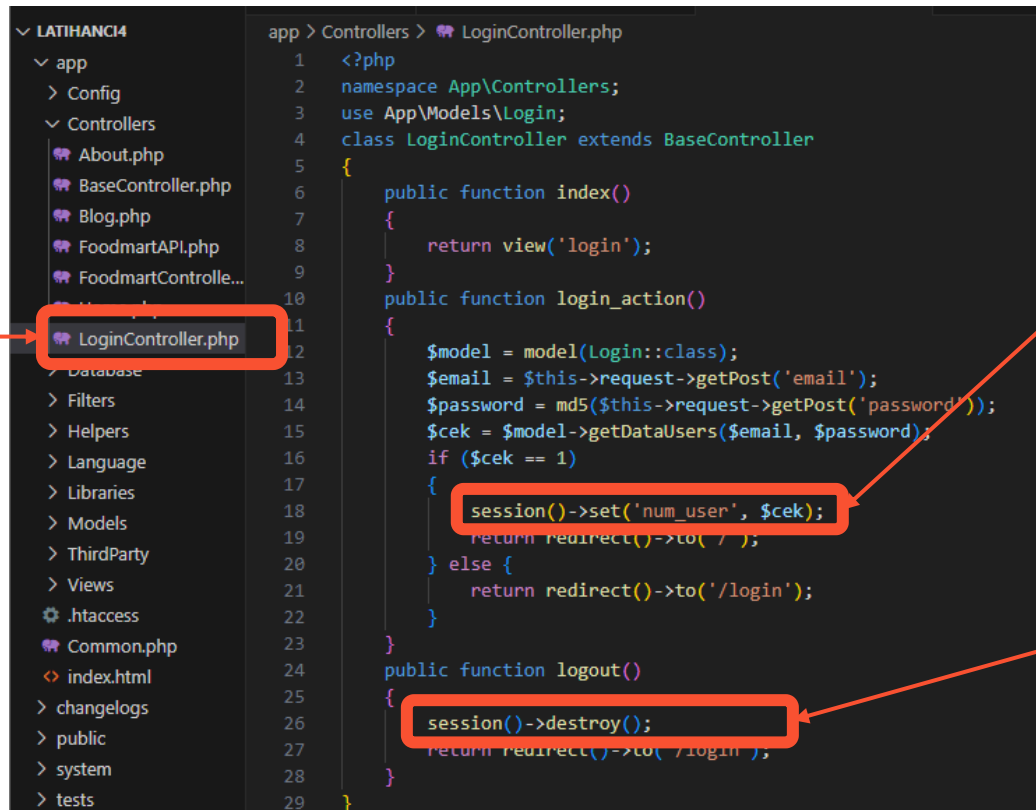
4. Klik pada tombol Signin, jika email dan password tidak sesuai dengan yang ada di database maka akan dikembalikan ke form login, jika cocok maka diteruskan ke halaman indeks.

***Notes :** Meskipun sudah menggunakan form login, namun Anda masih dapat langsung mengakses localhost:8080 tanpa login.

UPDATE CONTROLLER

LOGINCONTROLLER

1. Buat file controller baru pada app > Controllers > LoginController.php



```
app > Controllers > LoginController.php
1  <?php
2  namespace App\Controllers;
3  use App\Models\Login;
4  class LoginController extends BaseController
5  {
6      public function index()
7      {
8          return view('login');
9      }
10     public function login_action()
11     {
12         $model = model(Login::class);
13         $email = $this->request->getPost('email');
14         $password = md5($this->request->getPost('password'));
15         $cek = $model->getDataUsers($email, $password);
16         if ($cek == 1)
17         {
18             session()->set('num_user', $cek);
19             return redirect()->to('/');
20         } else {
21             return redirect()->to('/login');
22         }
23     }
24     public function logout()
25     {
26         session()->destroy();
27         return redirect()->to('/login');
28     }
29 }
```

2. Tambahkan kode `session()->set('num_user', $cek);` Pada method login_action. Anda dapat set beberapa session sekaligus. Pada contoh ini session yang digunakan hanya jumlah row output query.

3. Tambahkan kode `session()->destroy();` Pada method logout. Tujuannya adalah menghapus variabel session pada saat logout.

TAMBAHKAN PENGECEKAN SESSION PADA CONTROLLERS

Tambahkan kode berikut pada setiap controllers:
`if (session()->get('num_user') == '') {
 return redirect()->to('/login');`
`}`

```
app > Controllers > About.php
1 <?php
2 namespace App\Controllers;
3 class About extends BaseController
4 {
5     public function index()
6     {
7         if (session()->get('num_user') == '') {
8             return redirect()->to('/login');
9         }
10    return view('header').view('menu').view('about').view('footer');
11 }
12 }
```

```
app > Controllers > Home.php
1 <?php
2 namespace App\Controllers;
3 class Home extends BaseController
4 {
5     public function index()
6     {
7         if (session()->get('num_user') == '') {
8             return redirect()->to('/login');
9         }
10    return view('header').view('menu').view('dashboard').view('footer');
11 }
12 }
```

```
app > Controllers > FoodmartController.php
1 <?php
2 namespace App\Controllers;
3 use App\Models\Foodmart;
4 class FoodmartController extends BaseController
5 {
6     public function index()
7     {
8         if (session()->get('num_user') == '') {
9             return redirect()->to('/login');
10        }
11    $model = model(Foodmart::class);
12    $data['foodmart'] = $model->getDataFoodmart();
13    return view('header', $data).view('menu').
14        view('foodmart').view('footer');
15 }
16 }
```

```
app > Controllers > Blog.php
1 <?php
2
3 namespace App\Controllers;
4
5 class Blog extends BaseController
6 {
7     public function index()
8     {
9         if (session()->get('num_user') == '') {
10            return redirect()->to('/login');
11        }
12    return view('header').view('menu').view('blog').view('footer');
13 }
14 }
```

***Notes :** Sekarang Anda tidak dapat mengakses langsung ke halaman About, Dashboard, Blog, dan Foodmart tanpa login. Namun halaman foodmartAPI masih dibiarkan dapat diakses tanpa login.

AKHIR BAGIAN 6

Lanjutkan ke Bagian 7 – API Authentication

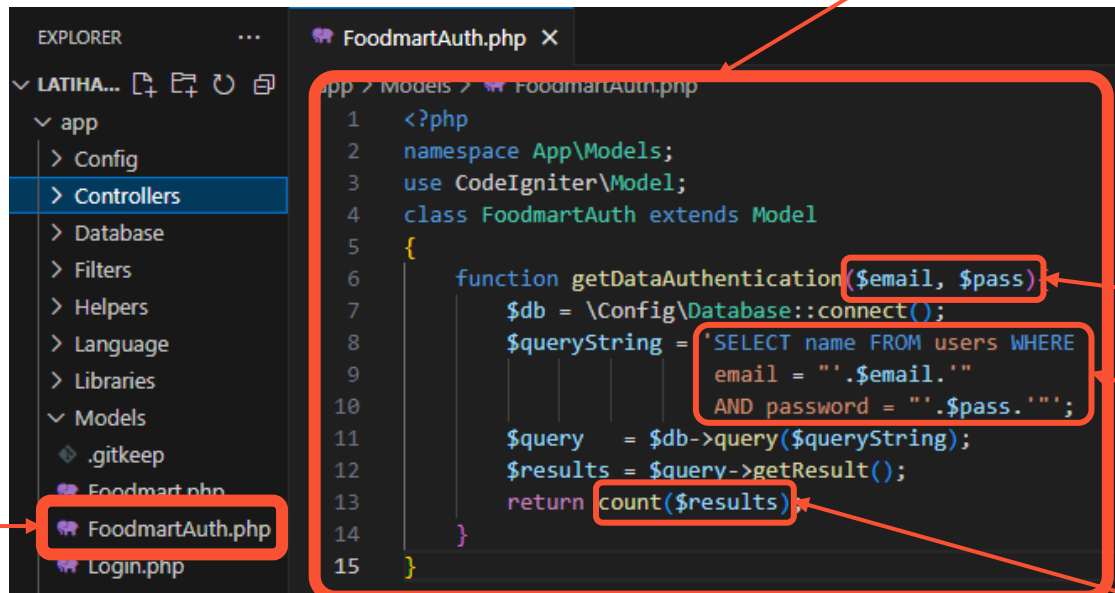
CODEIGNITER 4 BAGIAN 7- API AUTHENTICATION

Simple Authentication, Testing pada Postman

MODEL

FOODMARTAUTH

1. Buat file model baru pada app > Models > FoodmartAuth.php



```
2. Ketikkan kode berikut pada FoodmartAuth.php:
<?php
namespace App\Models;
use CodeIgniter\Model;
class FoodmartAuth extends Model{
    function getDataAuthentication($email, $pass){
        $db = \Config\Database::connect();
        $queryString = 'SELECT name FROM users
        WHERE email = "'. $email. "'"
        AND password = "'. $pass. '"';
        $query    = $db->query($queryString);
        $results = $query->getResult();
        return count($results);
    }
}
```

3. Input parameter berupa \$email dan \$pass

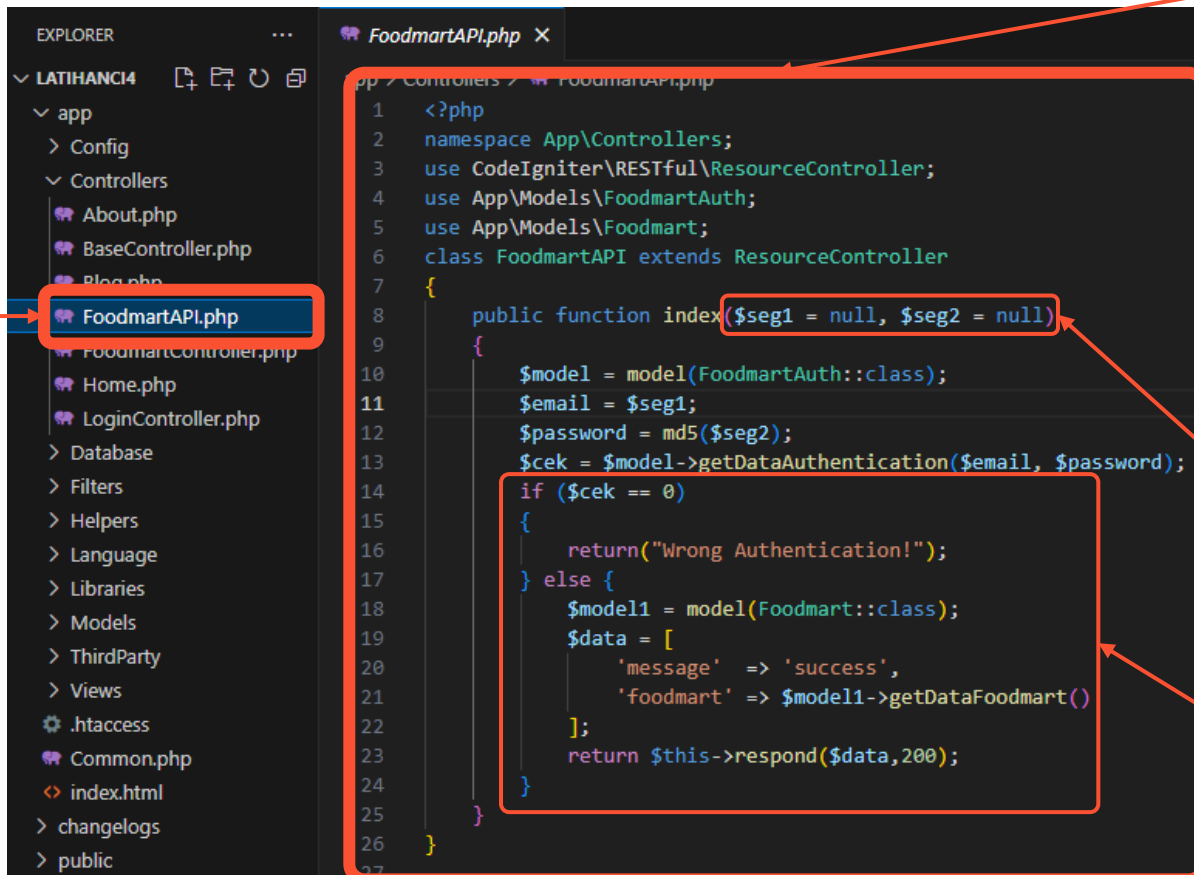
4. Query MySQL untuk mencari data yang cocok.

5. Return hasil query, bila ditemukan user yang cocok maka nilainya akan 1 dengan asumsi atribut email adalah primary key.

UPDATE CONTROLLER

FOODMARTAPI

1. Buka file app > Controllers > FoodmartAPI.php



```
1 <?php
2 namespace App\Controllers;
3 use CodeIgniter\RESTful\ResourceController;
4 use App\Models\FoodmartAuth;
5 use App\Models\Foodmart;
6 class FoodmartAPI extends ResourceController
7 {
8     public function index($seg1 = null, $seg2 = null)
9     {
10         $model = model(FoodmartAuth::class);
11         $email = $seg1;
12         $password = md5($seg2);
13         $cek = $model->getDataAuthentication($email, $password);
14         if ($cek == 0)
15         {
16             return("Wrong Authentication!");
17         } else {
18             $model1 = model(Foodmart::class);
19             $data = [
20                 'message' => 'success',
21                 'foodmart' => $model1->getDataFoodmart()
22             ];
23             return $this->respond($data, 200);
24         }
25     }
26 }
```

2. Update kode pada file app > Controllers > FoodmartAPI.php menjadi:

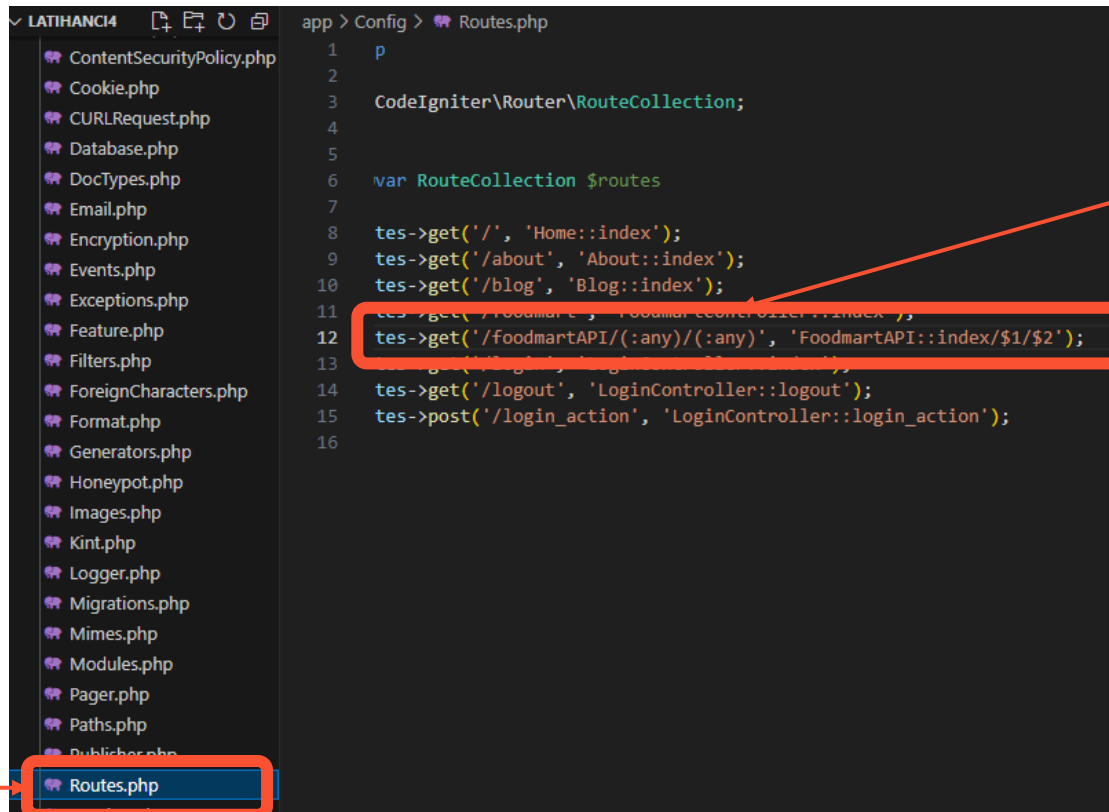
```
<?php
namespace App\Controllers;
use CodeIgniter\RESTful\ResourceController;
use App\Models\FoodmartAuth;
use App\Models\Foodmart;
class FoodmartAPI extends ResourceController{
    public function index($seg1 = null, $seg2 = null){
        $model = model(FoodmartAuth::class);
        $email = $seg1;
        $password = md5($seg2);
        $cek = $model->getDataAuthentication($email, $password);
        if ($cek == 0) {
            return("Wrong Authentication!");
        } else {
            $model1 = model(Foodmart::class);
            $data = ['message' => 'success',
                    'foodmart' => $model1->getDataFoodmart()];
            return $this->respond($data, 200);
        }
    }
}
```

3. Method index menerima input variable dari segmen URI berisi username untuk \$seg1 dan password pada \$seg2

4. Jika username dan password salah maka akan menampilkan pesan 'wrong authentication', dan jika benar maka akan menampilkan data.

UPDATE ROUTES

1. Buka file app > Config > Routes.php



```
1 p
2
3 CodeIgniter\Router\RouteCollection;
4
5
6 var RouteCollection $routes
7
8 tes->get('/', 'Home::index');
9 tes->get('/about', 'About::index');
10 tes->get('/blog', 'Blog::index');
11 tes->get('/foodmart', 'FoodmartController::index');
12 tes->get('/foodmartAPI/(:any)/(:any)', 'FoodmartAPI::index/$1/$2');
13
14 tes->get('/logout', 'LoginController::logout');
15 tes->post('/login_action', 'LoginController::login_action');
16
```

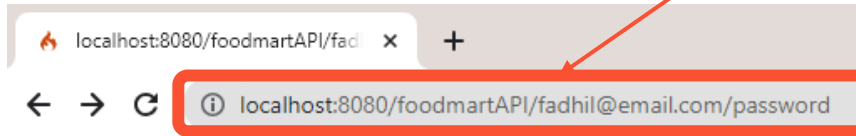
2. Update kode
`$routes->get('/foodmartAPI',
'FoodmartAPI::index');`
Menjadi
`$routes->get('/foodmartAPI/(:any)/(:any)',
'FoodmartAPI::index/$1/$2');`

*Notes : Pada tahap ini, link localhost/foodmartAPI sudah tidak dapat diakses lagi dan akan menghasilkan error 404.

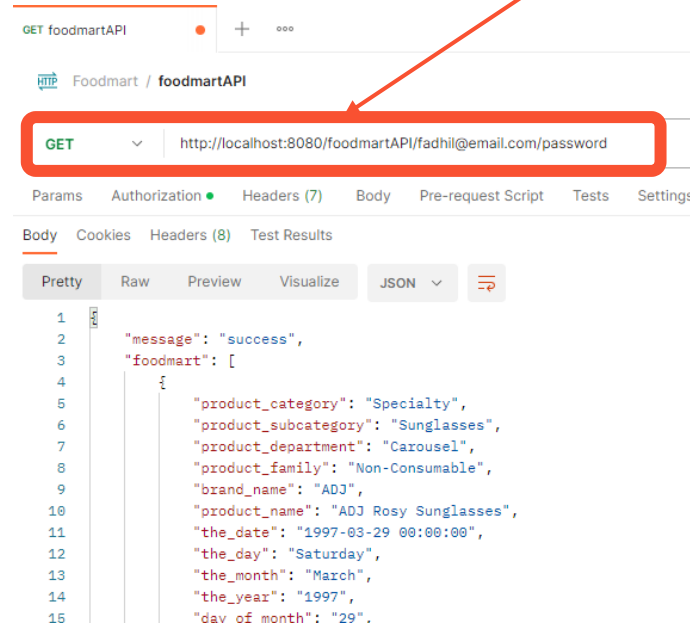
AKSES

FOODMARTAPI

1. Ketik alamat localhost:8080/foodmartAPI/<username>/<password> pada browser atau pada postman, misalnya http://localhost:8080/foodmartAPI/fadhil@email.com/password untuk mengakses API.



```
{
  "message": "success",
  "foodmart": [
    {
      "product_category": "Specialty",
      "product_subcategory": "Sunglasses",
      "product_department": "Carousel",
      "product_family": "Non-Consumable",
      "brand_name": "ADJ",
      "product_name": "ADJ Rosy Sunglasses",
      "the_date": "1997-03-29 00:00:00",
      "the_day": "Saturday",
      "the_month": "March",
      "the_year": "1997",
      "day_of_month": "29",
      "week_of_year": "14",
      "month_of_year": "3",
      "quarter": "Q1",
      "fiscal_period": null,
      "account_num": "49872222542",
      "fullname": "Marvin Benzakry",
      "lname": "Benzakry",
      "fname": "Marvin",
      "mi": "F.",
      "address1": "7335 Imperial Dr.",
      "address2": null,
      "address3": null,
      "address4": null,
      "city": "Sedro Woolley",
      "state_province": "WA",
      "postal_code": "10751",
      "country": "USA",
      "phone1": "553-555-7479",
      "phone2": "403-555-8615".
    }
  ]
}
```



2. Jika username dan password salah maka akan menampilkan pesan 'Wrong Authentication'.

AKHIR BAGIAN 7

Lanjutkan ke Bagian 8 – Request API

The background of the slide features a complex network of white dots connected by thin white lines, creating a web-like or molecular structure. The dots vary in size, and the lines are of varying thickness, giving it a dynamic and interconnected appearance. The overall color scheme is a gradient of warm tones, from a deep orange at the top to a lighter, almost white, at the bottom.

CODEIGNITER 4

BAGIAN 8- REQUEST API

Access API, Parsing Data

ANTARA NEWS RSS

1. Akses open RSS dari Antara New di alamat <https://www.antaranews.com/rss/terkini.xml>

← → ↻ 🔒 antaranews.com/rss/terkini.xml

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:content="http://purl.org/rss/1.0/modules/content/"
  xmlns:atom="http://www.w3.org/2005/Atom" xmlns:sy="http://purl.org/rss/1.0/modules/syndication/"
  xmlns:slash="http://purl.org/rss/1.0/modules/slash/" version="2.0">
  <channel>
    <title>Berita Terkini - ANTARA News</title>
    <description>Berita Terkini dan Terpercaya Indonesia</description>
    <link>https://www.antaranews.com</link>
    <language>id</language>
    <copyright>Copyright 2023, ANTARA News</copyright>
    <category>Berita Terkini</category>
    <lastBuildDate>Thu, 23 Nov 2023 13:41:01 +0700</lastBuildDate>
  </channel>
  <item>
    <title>Kemacetan di tol akibat penutupan pintu tol kawasan industri MM 2100</title>
    <link>https://www.antaranews.com/foto/3837054/kemacetan-di-tol-akibat-penutupan-pintu-tol-kawasan-industri-mm-2100</link>
    <pubDate>Thu, 23 Nov 2023 13:39:29 +0700</pubDate>
    <description>
      <![CDATA[ Foto udara kemacetan truk barang yang akan menuju kawasan MM 2100 di Tol Jakarta- Cikampek,
        Cibitung, Kabupaten Bekasi, Jawa Barat, Kamis (23/11/2023). &nbsp;Kemacetan tersebut terjadi akibat buruh menutup pintu
        tol menuju dan ... ]]>
    </description>
    <content:encoded>
      <![CDATA[ Foto udara kemacetan truk barang yang akan menuju kawasan MM 2100 di Tol Jakarta- Cikampek, Cibitung,
        Kabupaten Bekasi, Jawa Barat, Kamis (23/11/2023). &nbsp;Kemacetan tersebut terjadi akibat buruh menutup pintu tol menuju
        dan ... ]]>
    </content:encoded>
    <guid>https://www.antaranews.com/foto/3837054/kemacetan-di-tol-akibat-penutupan-pintu-tol-kawasan-industri-mm-2100</guid>
    <enclosure length="0" type="image/jpeg" url="https://img.antaranews.com/cache/800x533/2023/11/23/Kemacetan-Tol-Japek-di-
      Cibitung-4.jpg"/>
  </item>
```

2. Perhatikan struktur XML feed Antara News, data berita terdapat di dalam tag `<item></item>`, di dalamnya terdapat judul (`<title></title>`), link ke halaman berita (`<link></link>`), tanggal publikasi (`<pubDate></pubDate>`), dan isi berita yang didalamnya ada gambar (`<description></description>`). Item dibatasi hanya sebanyak 20 item. Data ini akan direquest melalui website yang dibangun,

CONTROLLER

NEWSCONTROLLER.PHP

Buat controller baru pada app > Controllers > NewsController.php, dan ketikkan kode berikut:

```
<?php

namespace App\Controllers;

class NewsController extends BaseController{

    public function index(){

        if (session()->get('num_user') == '') {

            return redirect()->to('/login');

        }

        $url = 'https://www.antaranews.com/rss/terkini.xml';

        $doc = new \DOMDocument(); $doc->load($url);

        $newsData = $doc->getElementsByTagName("item");

        foreach($newsData as $newsDataParse){

            $titles = $newsDataParse->getElementsByTagName( "title" ); $title = $titles->item(0)->nodeValue;

            $links = $newsDataParse->getElementsByTagName( "link" ); $link = $links->item(0)->nodeValue;

            $pubDates = $newsDataParse->getElementsByTagName( "pubDate" ); $pubDate = $pubDates->item(0)->nodeValue;

            $descriptions = $newsDataParse->getElementsByTagName( "description" ); $description = $descriptions->item(0)->nodeValue;

            $data['title'][] = $title; $data['link'][] = $link; $data['pubDate'][] = $pubDate;
            $data['description'][] = $description;

            continue;

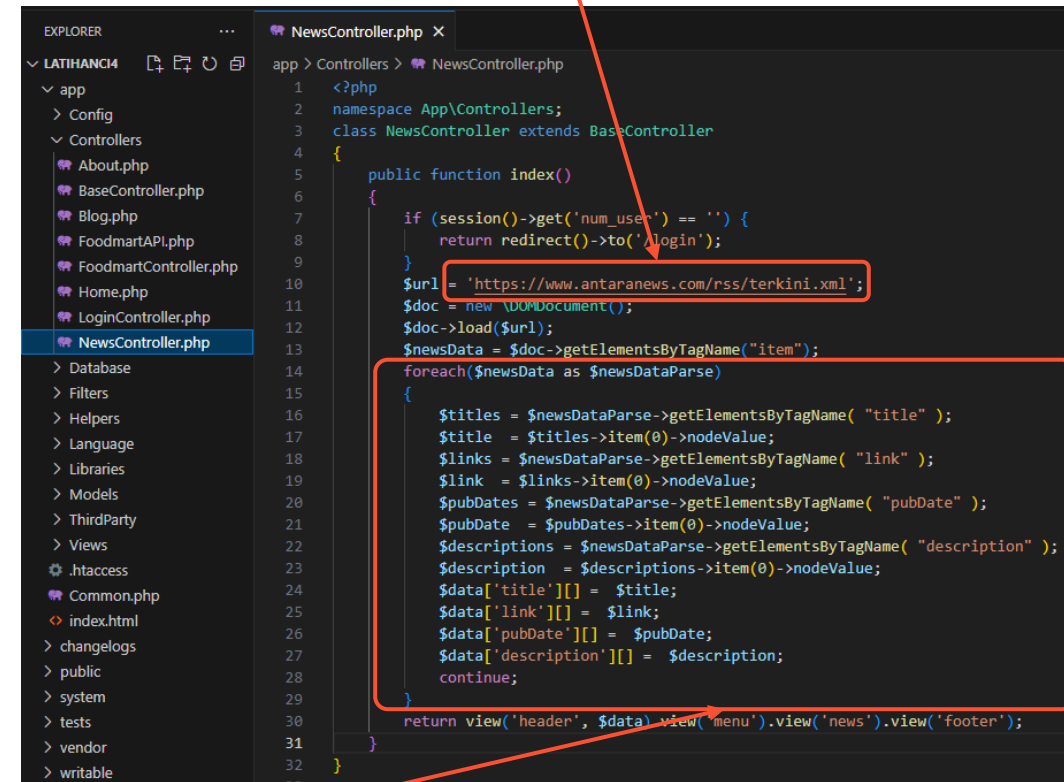
        }

        return view('header', $data).view('menu').view('news').view('footer');

    }

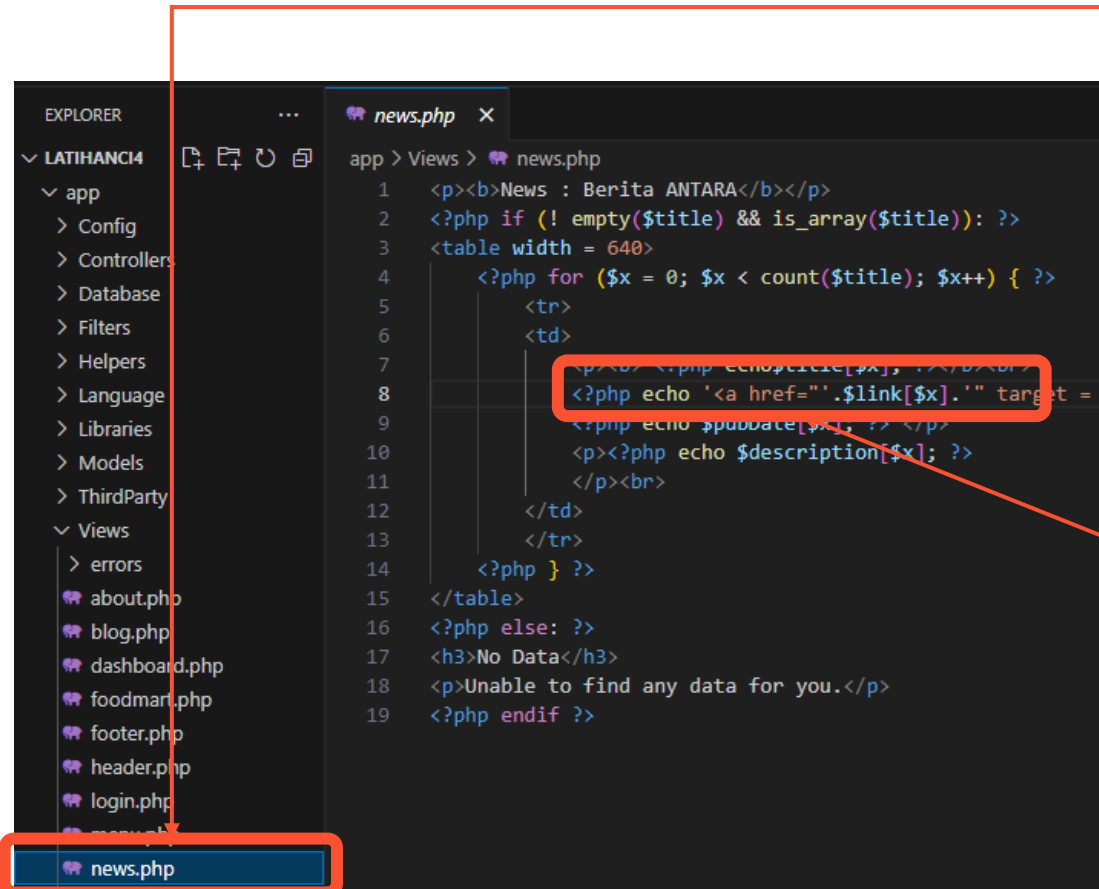
}
```

1. Akses open RSS dari Antara New di alamat <https://www.antaranews.com/rss/terkini.xml>



2. Parsing data pada tag <item></item>. Node value berarti nilai di dalam tag, misal <title>Judul</title>, maka node valuenya adalah 'Judul'.

VIEW NEWS.PHP



1. Buat file view pada app > Views > news.php dan ketikkan kode berikut:

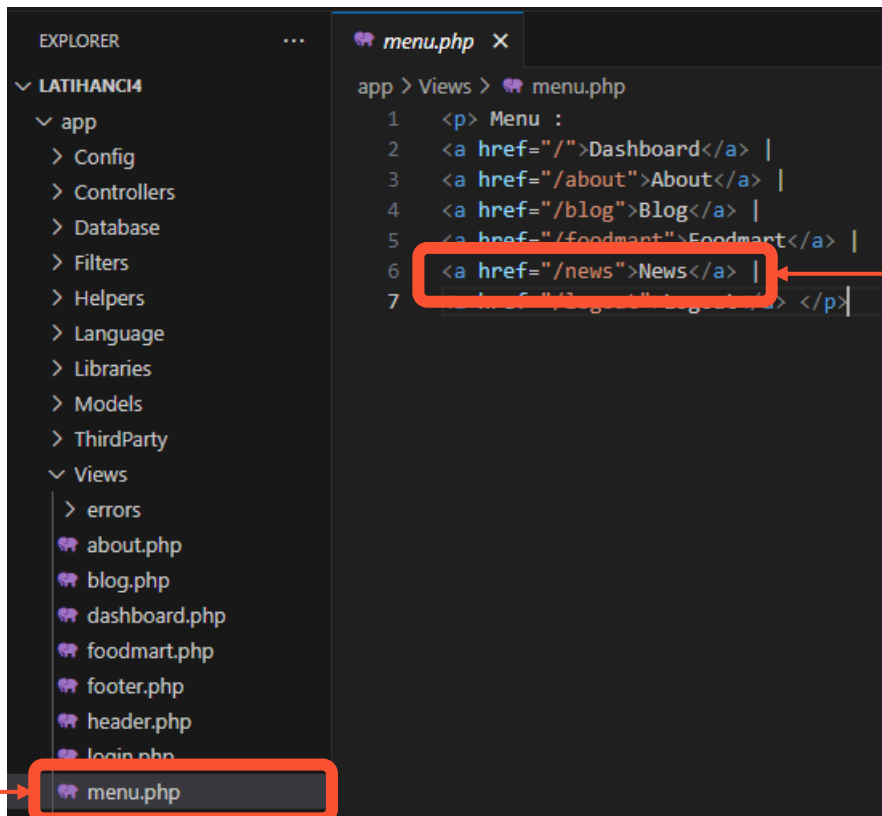
```
<p><b>News : Berita ANTARA</b></p>
<?php if (! empty($title) && is_array($title)): ?>
<table width = 640>
    <?php for ($x = 0; $x < count($title); $x++) { ?>
        <tr>
            <td>
                <p><b> <?php echo $title[$x]; ?></b><br>
                <?php echo '<a href="'. $link[$x].'" target = "blank">'
                . $link[$x]. '</a>'; ?> <br>
                <?php echo $pubDate[$x]; ?> </p>
                <p><?php echo $description[$x]; ?>
                </p><br>
            </td>
        </tr>
    <?php } ?>
</table>
<?php else: ?>
<h3>No Data</h3>
<p>Unable to find any data for you.</p>
<?php endif ?>
```

2. Pada tak <link></link> berisi link ke alamat web antara news.

UPDATE VIEW

MENU.PHP DAN ROUTES

1. Buka file app > Views > menu.php



```
1 <p> Menu :
2 <a href="/">Dashboard</a> |
3 <a href="/about">About</a> |
4 <a href="/blog">Blog</a> |
5 <a href="/foodmart">Foodmart</a> |
6 <a href="/news">News</a> |
7 <a href="/login">Login</a> </p>
```

2. Tambahkan kode:

`News |`

3. Buka file app > Config > Routes.php

4. Tambahkan kode:

`$routes->get('/news', 'NewsController::index');`



```
1 use Illuminate\Routing\Router;
2 use Illuminate\Support\Facades\Route;
3 use CodeIgniter\Router\RouteCollection;
4
5 /**
6  * @var RouteCollection $routes
7  */
8 $routes->get('/', 'Home::index');
9 $routes->get('/about', 'About::index');
10 $routes->get('/blog', 'Blog::index');
11 $routes->get('/foodmart', 'FoodmartController::index');
12 $routes->get('/foodmartAPI/(:any)/(:any)', 'FoodmartAPI::index/$1/$2');
13 $routes->get('/login', 'LoginController::index');
14 $routes->get('/logout', 'LoginController::logout');
15 $routes->get('/news', 'NewsController::index');
16 $routes->post('/login_action', 'LoginController::login_action');
```

AKSES LOCALHOST:8080/NEWS (LOGIN TERLEBIH DAHULU)



1. Akses localhost:8080/news melalui menu News, Anda memerlukan login terlebih dahulu.

Website Beta

Menu : [Dashboard](#) | [About](#) | [Blog](#) | [Foodmart](#) | [News](#) | [Logout](#)

News : Berita ANTARA

Yamali TB apresiasi ratusan kader TBC melalui jambore

<https://www.antaranews.com/berita/3843456-yamali-tb-apresiasi-ratusan-kader-tbc-melalui-jambore>

Mon, 27 Nov 2023 20:27:53 +0700



Yayasan Masyarakat Peduli TB (Yamali TB) Sulawesi Selatan mengapresiasi ratusan kader TBC (Tuberkolosis) melalui penyelenggaraan Jambore Kader di Kawasan Hutan Pinus Bulutana, Malino, Kabupaten Gowa, 24-26 November ...

2. Data pada bagian <title></title>

3. Data pada bagian <link></link>

4. Data pada bagian <pubDate></pubDate>

5. Data pada bagian <description></description>

Notes: Silakan eksplorasi bagaimana melakukan parsing data JSON. Anda dapat mengolah dan menyimpan data yang diterima.

AKHIR BAGIAN 8

Lanjutkan ke Bagian 9 – Visualization

The background of the slide features a complex network of white dots connected by thin white lines, creating a web-like pattern. The dots vary in size, and the lines are of varying thickness, giving it a dynamic and interconnected appearance. The background color is a gradient of orange and brown tones.

CODEIGNITER 4

BAGIAN 9- VISUALIZATION

Report, Chart-Js

VIEW

REPORT_SALES

1. Akses localhost/phpmyadmin pada browser, akses database foodmart, klik pada tombol SQL.

2. Ketikkan query berikut:
`SELECT time_id, SUM(store_sales) as total_sales, SUM(store_cost) as total_store_cost FROM `sales_fact_1997` GROUP BY `time_id` ORDER BY `time_id` ASC`

3. Klik Go

4. Klik Create View untuk membuat view baru.

5. Isikan VIEW name menjadi "view_report_sales"

6. Klik Go untuk menyimpan view

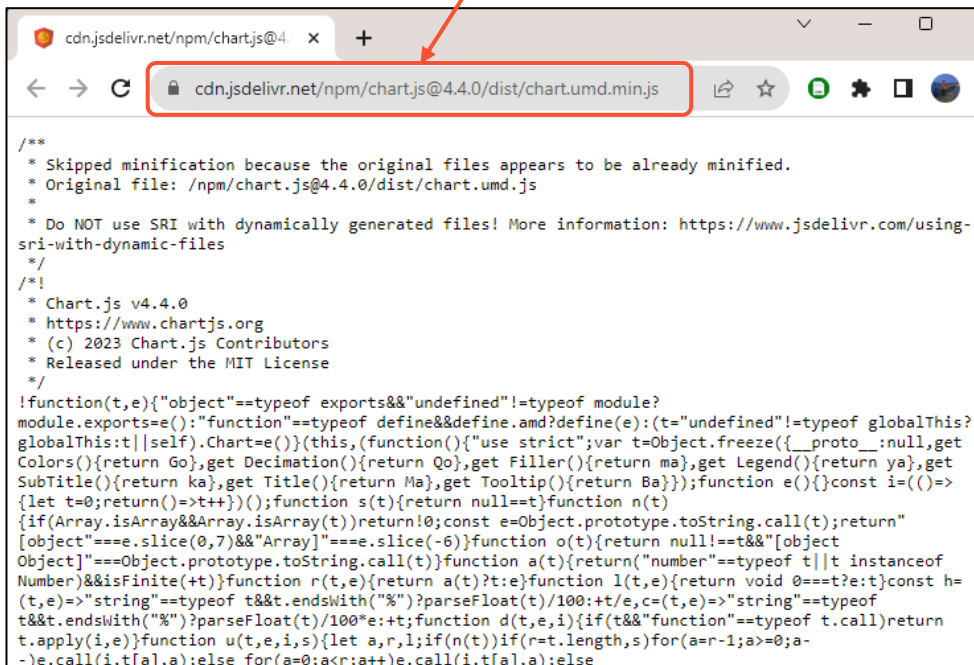
The screenshot shows the phpMyAdmin interface with the 'SQL' tab selected. The query entered is: `SELECT time_id, SUM(store_sales), SUM(store_cost) FROM `sales_fact_1997` GROUP BY `time_id` ORDER BY `time_id` ASC`. The 'Go' button is highlighted. Below the query, the 'Options' section shows the 'time_id' column selected. The 'Query results operations' section at the bottom has the 'Create view' button highlighted.

time_id	SUM(store_sales)	SUM(store_cost)
367	706.3400	280.4990
368	1304.5300	525.8396
369	1294.1200	515.2609
370	42.8700	17.6873
371	1987.1900	809.6743
372	2162.3400	864.9502
373	2696.6100	1078.2984
374	69.3000	25.9894
375	1090.6500	431.3554
376	1022.0800	403.1699
377	5550.8400	2214.3738
378	859.6200	338.2163
379	3527.4500	1404.4789
380	542.0800	213.8537
381	1084.6500	443.2269
382	1232.1900	487.4430
383	2538.5300	1000.8063
384	856.4400	341.9665
385	1445.7100	578.9351
386	4371.0500	1744.0010
387	2455.6300	982.0348
389	1659.2700	667.4021
390	54.0100	20.8004
391	2353.3900	930.5615
392	1241.5900	507.4709

The 'Create view' dialog box is shown with the 'VIEW name' field set to 'view_report_sales'. The 'Column names' field is empty. The 'SQL SECURITY' dropdown is set to 'VIEW'. The 'WITH CHECK OPTION' checkbox is checked. The 'Go' button is highlighted.

CHART JS

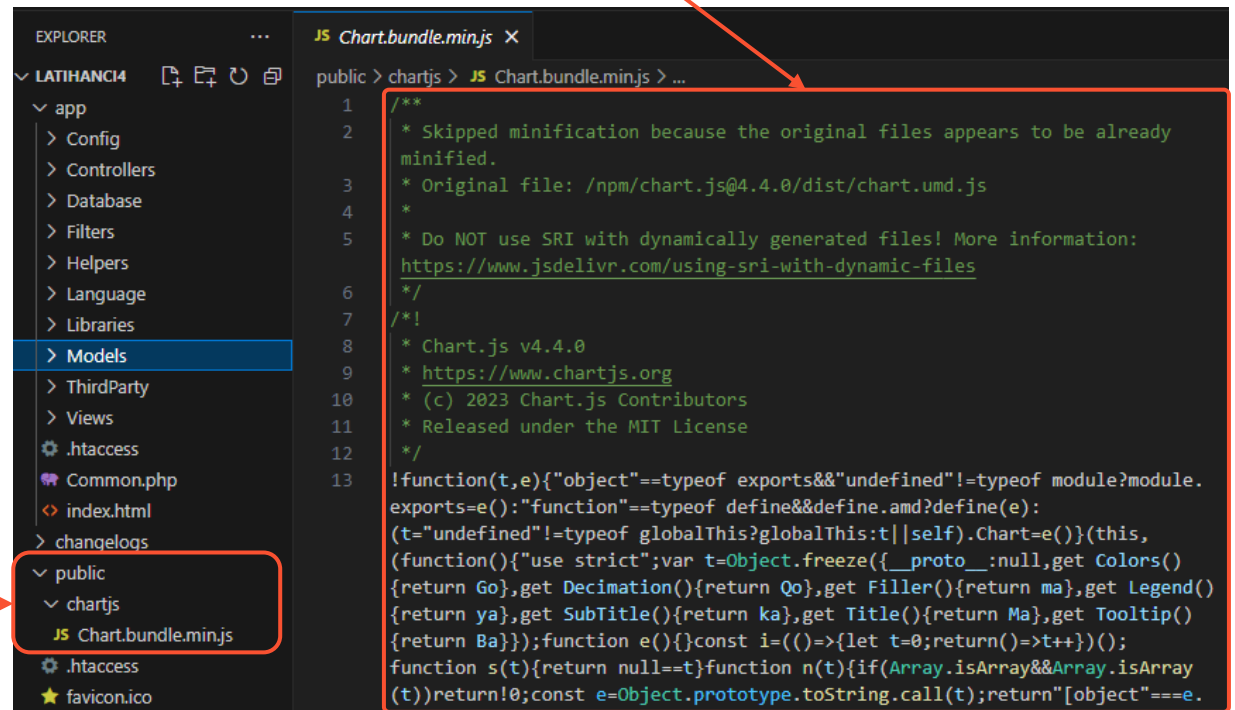
1. Akses sourcecode Chart JS di halaman: <https://cdn.jsdelivr.net/npm/chart.js@4.4.0/dist/chart.umd.min.js>
Copy semua sourcecode Chart JS



```
/**
 * Skipped minification because the original files appears to be already minified.
 * Original file: /npm/chart.js@4.4.0/dist/chart.umd.js
 *
 * Do NOT use SRI with dynamically generated files! More information: https://www.jsdelivr.com/using-sri-with-dynamic-files
 */
/*!
 * Chart.js v4.4.0
 * https://www.chartjs.org
 * (c) 2023 Chart.js Contributors
 * Released under the MIT License
 */
!function(t,e){"object"==typeof exports&&"undefined"!=typeof module?module.exports=e():"function"==typeof define&&define.amd?define(e):(t="undefined"!=typeof globalThis?globalThis:t||self).Chart=e()}(this,(function(){var t=Object.freeze({__proto__:null,get Colors(){return Go},get Decimation(){return Qo},get Filler(){return ma},get Legend(){return ya},get SubTitle(){return ka},get Title(){return Ma},get Tooltip(){return Ba}});function e(){const i=()=>{let t=0;return()=>t++}};function s(t){return null==t?function n(t){if(Array.isArray&&Array.isArray(t))return!0;const e=Object.prototype.toString.call(t);return"object"===e.slice(0,7)&&"Array"===e.slice(-6)}function o(t){return null!=t&&[Object]===Object.prototype.toString.call(t)?function a(t){return"number"==typeof t||t instanceof Number&&isFinite(+t)}function r(t,e){return a(t)?t:e}function l(t,e){return void 0===t?e:t}const h=(t,e)=>"string"==typeof t&&t.endsWith("%")?parseFloat(t)/100:+t/e,c=(t,e)=>"string"==typeof t&&t.endsWith("%")?parseFloat(t)/100:+t;function d(t,e,i){if(t&&"function"==typeof t.call)return t.apply(i,e);function u(t,e,i,s){let a,r,l;if(n(t))if(r=t.length,s)for(a=r-1;a>=0;a--){e.call(i,t[a],a);else for(a=0;a<r;a++){e.call(i,t[a],a);else
```

2. Buat Folder baru pada app > public > chartjs. Di dalam folder tersebut buat file Chart.bundle.min.js.

3. Paste source code Chart JS yang sebelumnya di copy ke dalam file Chart.bundle.min.js, kemudian Save.

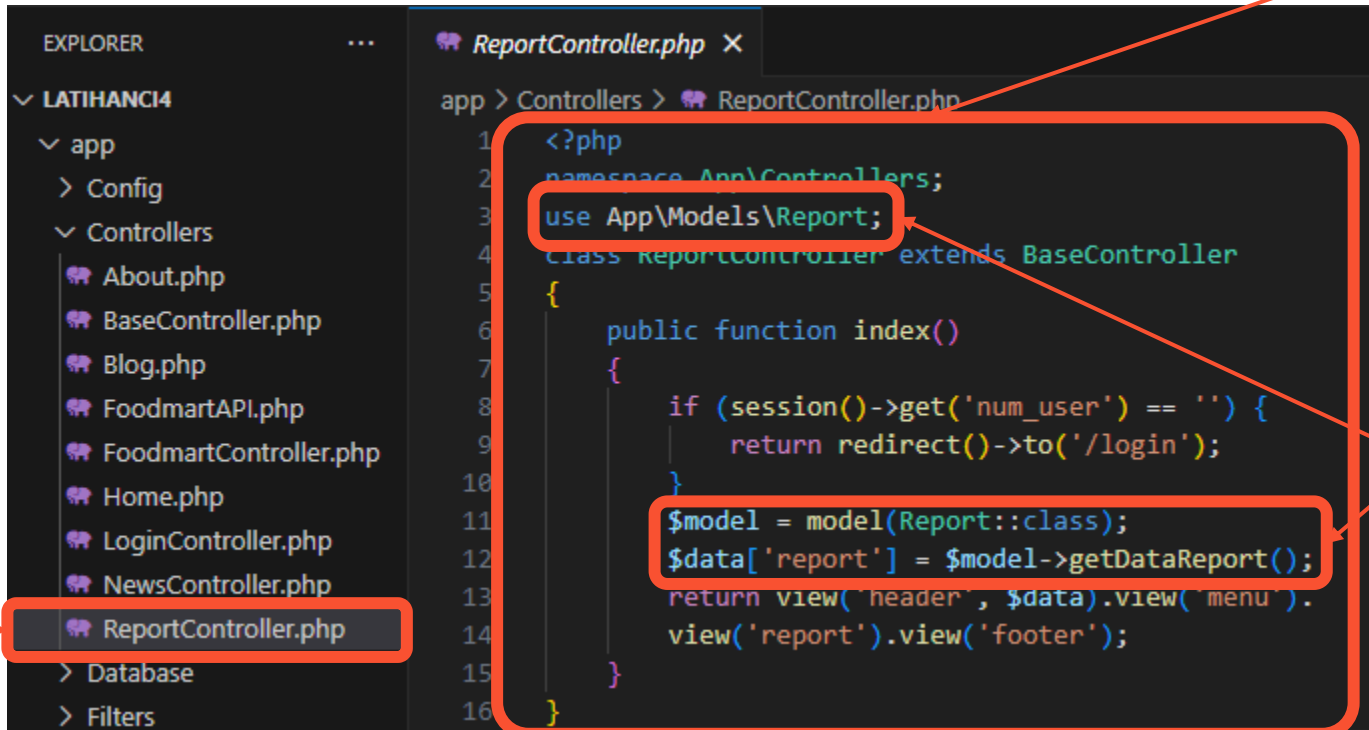


```
/**
 * Skipped minification because the original files appears to be already
minified.
 * Original file: /npm/chart.js@4.4.0/dist/chart.umd.js
 *
 * Do NOT use SRI with dynamically generated files! More information:
https://www.jsdelivr.com/using-sri-with-dynamic-files
 */
/*!
 * Chart.js v4.4.0
 * https://www.chartjs.org
 * (c) 2023 Chart.js Contributors
 * Released under the MIT License
 */
!function(t,e){"object"==typeof exports&&"undefined"!=typeof module?module.
exports=e():"function"==typeof define&&define.amd?define(e):(t="undefined"!=typeof globalThis?globalThis:t||self).Chart=e()}(this,
(function(){var t=Object.freeze({__proto__:null,get Colors()
{return Go},get Decimation(){return Qo},get Filler(){return ma},get Legend()
{return ya},get SubTitle(){return ka},get Title(){return Ma},get Tooltip()
{return Ba}});function e(){const i=()=>{let t=0;return()=>t++}};
function s(t){return null==t?function n(t){if(Array.isArray&&Array.isArray
(t))return!0;const e=Object.prototype.toString.call(t);return"object"===e.
```


CONTROLLER

REPORTCONTROLLER.PHP

1. Buat file controller baru pada app > Controllers > ReportController.php



The screenshot shows the VS Code interface. On the left, the 'EXPLORER' sidebar displays the project structure under 'LATIHANCI4'. The 'app' directory is expanded, showing 'Controllers'. The file 'ReportController.php' is highlighted. On the right, the code editor shows the content of 'ReportController.php'. The code is as follows:

```
1 <?php
2 namespace App\Controllers;
3 use App\Models\Report;
4 class ReportController extends BaseController{
5     {
6         public function index()
7         {
8             if (session()->get('num_user') == '') {
9                 return redirect()->to('/login');
10            }
11            $model = model(Report::class);
12            $data['report'] = $model->getDataReport();
13            return view('header', $data).view('menu').
14                view('report').view('footer');
15        }
16    }
```

2. Ketikkan kode berikut:

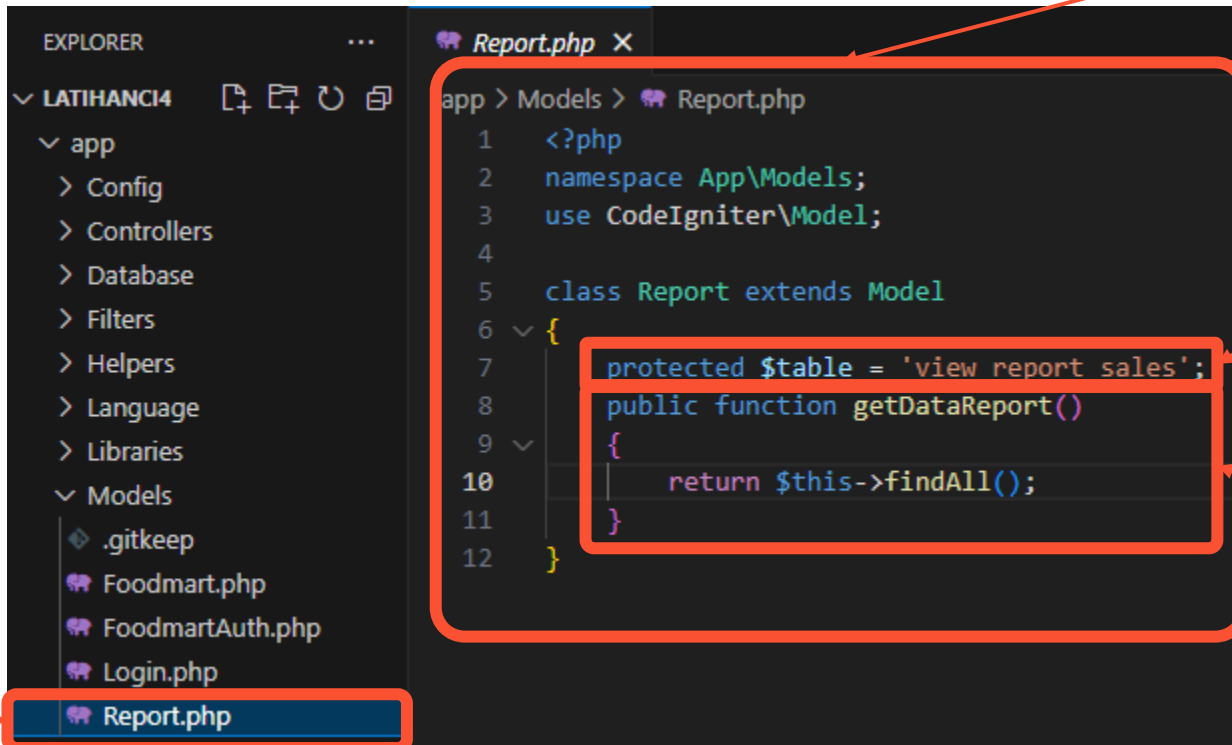
```
<?php
namespace App\Controllers;
use App\Models\Report;
class ReportController extends BaseController{
    public function index(){
        if (session()->get('num_user') == '') {
            return redirect()->to('/login');}
        $model = model(Report::class);
        $data['report'] = $model->getDataReport();
        return view('header', $data).view('menu').
            view('report').view('footer');
    }
}
```

3. Controller ReportController memanggil model Report, Model Report memiliki method getDataReport() untuk mengambil data dari tabel "view_report_sales". Model ini akan dibuat selanjutnya.

MODEL

REPORT.PHP

1. Buat file model baru pada app > Models > Report.php



```
1 <?php
2 namespace App\Models;
3 use CodeIgniter\Model;
4
5 class Report extends Model
6 {
7     protected $table = 'view_report_sales';
8     public function getDataReport()
9     {
10         return $this->findAll();
11     }
12 }
```

2. Ketikkan kode berikut:

```
<?php
namespace App\Models;
use CodeIgniter\Model;
class Report extends Model{
    protected $table = 'view_report_sales';
    public function getDataReport(){
        return $this->findAll();
    }
}
```

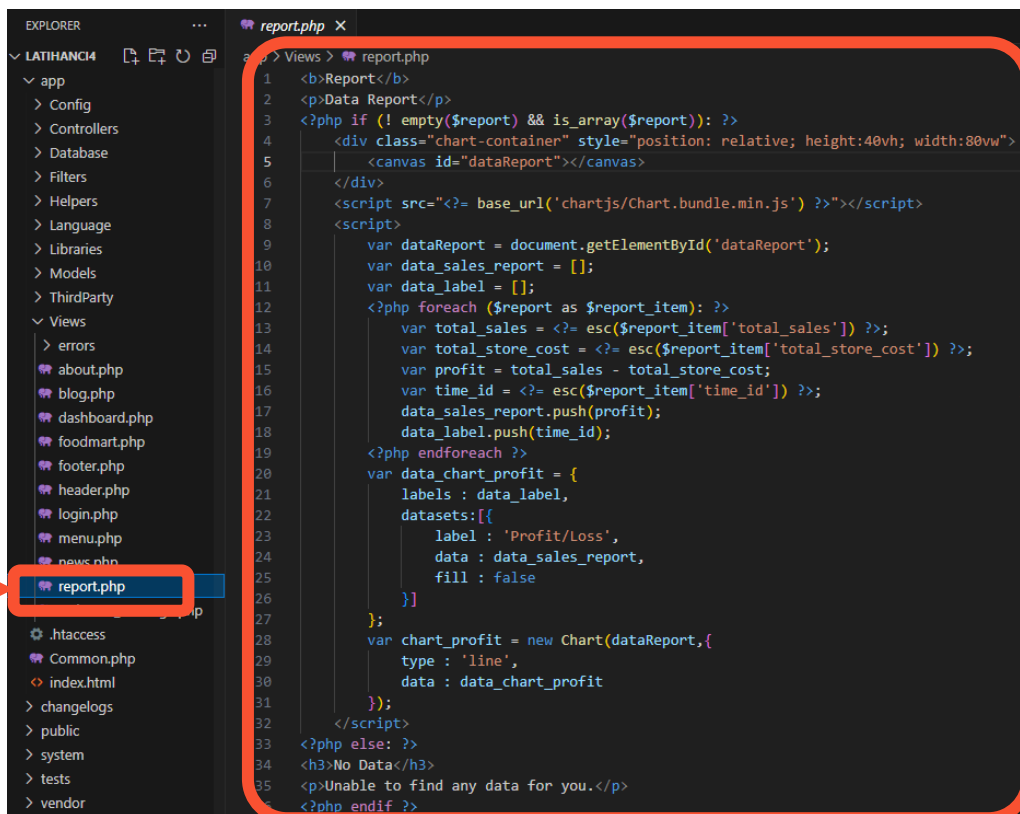
3. Memanggil tabel 'view_report_sales' dari database.

4. Method getDataReport() berisi perintah findAll(), fungsi ini sama dengan query "select * from view_report_sales"

VIEW

REPORT.PHP

1. Buat file view baru pada app > Views > report.php



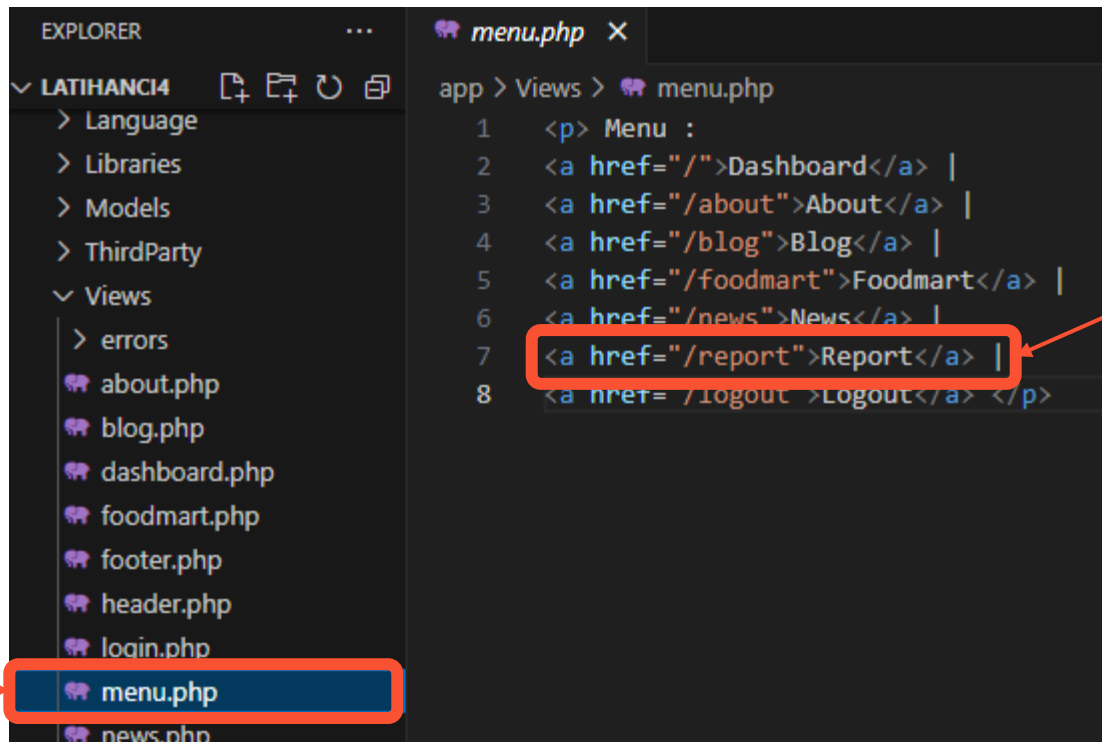
2. Ketikkan kode berikut, kombinasi HTML, PHP dan Javascript. Perhatikan kode javascript di dalam tag <script></script>, kode Chart JS dituliskan di dalam tag tersebut. Sedangkan Chart akan ditampilkan pada tag <canvasid="dataReport"></canvas>.

```
<b>Report</b>
<p>Data Report</p>
<?php if (! empty($report) && is_array($report)): ?>
  <div class="chart-container" style="position: relative; height:40vh; width:80vw">
    <canvas id="dataReport"></canvas>
  </div>
  <script src="<? = base_url('chartjs/Chart.bundle.min.js') ?>"></script>
  <script>
    var dataReport = document.getElementById('dataReport');
    var data_sales_report = [];
    var data_label = [];
    <?php foreach ($report as $report_item): ?>
      var total_sales = <? = esc($report_item['total_sales']) ?>;
      var total_store_cost = <? = esc($report_item['total_store_cost']) ?>;
      var profit = total_sales - total_store_cost;
      var time_id = <? = esc($report_item['time_id']) ?>;
      data_sales_report.push(profit);
      data_label.push(time_id);
    <?php endforeach ?>
    var data_chart_profit = {
      labels : data_label,
      datasets:[{
        label : 'Profit/Loss',
        data : data_sales_report,
        fill : false
      }]
    };
    var chart_profit = new Chart(dataReport,{
      type : 'line',
      data : data_chart_profit
    });
  </script>
<?php else: ?>
<h3>No Data</h3>
<p>Unable to find any data for you.</p>
<?php endif ?>
```

UPDATE VIEW

MENU.PHP

1. Buka file view baru pada app > Views > menu.php



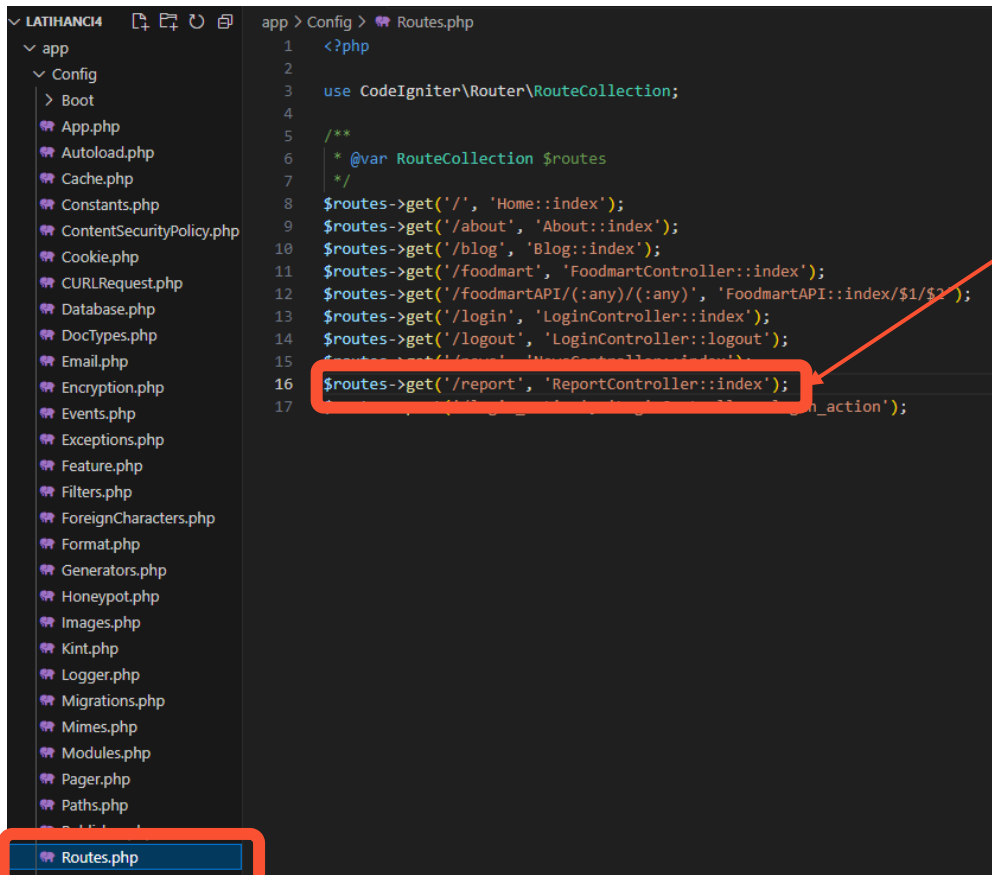
```
EXPLORER
└─ LATIHANCI4
   └─ Views
       └─ menu.php
          └─ errors
             └─ about.php
                └─ blog.php
                   └─ dashboard.php
                      └─ foodmart.php
                         └─ footer.php
                            └─ header.php
                               └─ login.php
                                  └─ menu.php
                                     └─ news.php
```

```
menu.php
1  <p> Menu :
2  <a href="/">Dashboard</a> |
3  <a href="/about">About</a> |
4  <a href="/blog">Blog</a> |
5  <a href="/foodmart">Foodmart</a> |
6  <a href="/news">News</a> |
7  <a href="/report">Report</a> |
8  <a href="/logout">Logout</a> </p>
```

2. Tambahkan menu Report dengan mengetikkan kode: `Report |`. Selanjutnya `/report` akan didaftarkan pada Routes.

UPDATE ROUTES.PHP

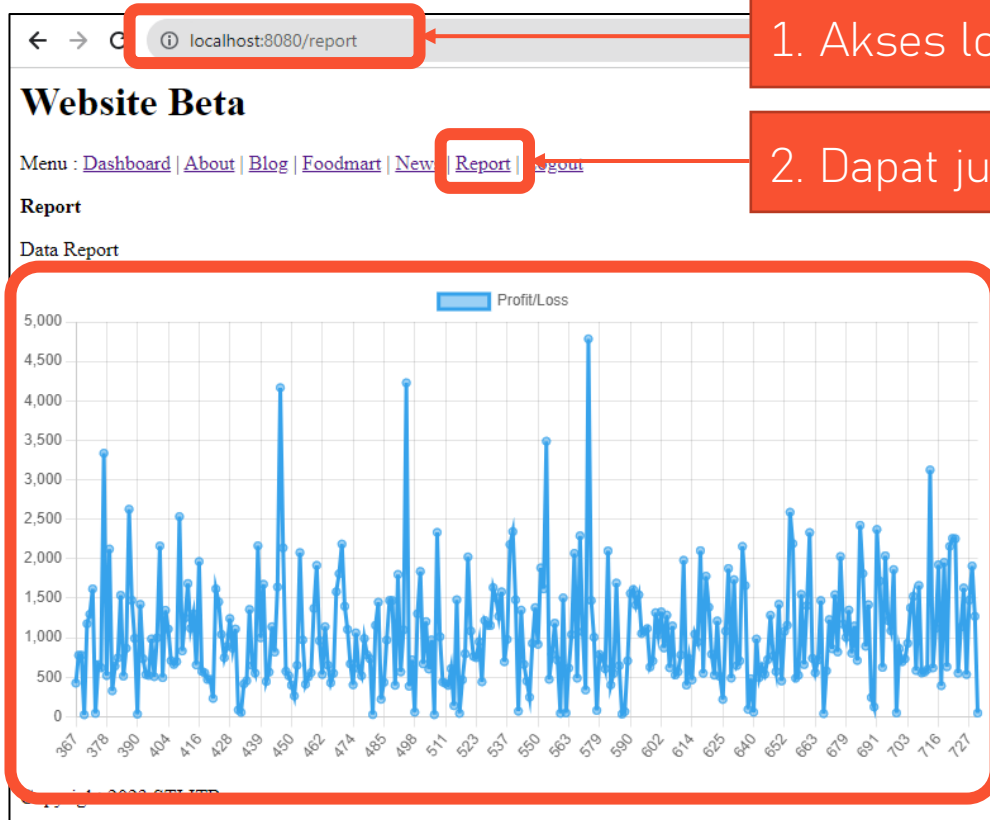
1. Buka file view baru pada app > Config > Routes.php



```
1 <?php
2
3 use CodeIgniter\Router\RouteCollection;
4
5 /**
6  * @var RouteCollection $routes
7  */
8 $routes->get('/', 'Home::index');
9 $routes->get('/about', 'About::index');
10 $routes->get('/blog', 'Blog::index');
11 $routes->get('/foodmart', 'FoodmartController::index');
12 $routes->get('/foodmartAPI/(:any)/(:any)', 'FoodmartAPI::index/$1/$2');
13 $routes->get('/login', 'LoginController::index');
14 $routes->get('/logout', 'LoginController::logout');
15 $routes->get('/report', 'ReportController::index');
16 $routes->get('/report', 'ReportController::index');
17 $routes->get('/report', 'ReportController::index');
```

2. Pada Routes.php tambahkan kode:
`$routes->get('/report', 'ReportController::index');`
Link ini akan memanggil class ReportController method index() yang berisi perintah findAll().

AKSES LOCALHOST:8080/REPORT (PERLU LOGIN TERLEBIH DAHULU)



1. Akses localhost:8080/report pada browser

2. Dapat juga dengan klik pada menu Report

3. Chart JS menampilkan data profit pada tabel "view_sales_report" ke dalam line chart. Ada beberapa chart lain yang dapat dibuat menggunakan Chart JS. Sumbu Y menampilkan besaran profit dalam US Dollar, sedangkan sumbu X menampilkan time_id. Legenda dapat di klik untuk menyembunyikan/menampilkan data, hal ini akan berguna jika Anda menampilkan banyak data sekaligus dalam satu chart, namun ingin memfilter untuk menampilkan salah satu kategori data saja.

AKHIR BAGIAN 9

Lanjutkan ke Bagian 10 – Ajax

CODEIGNITER 4

TUGAS BESAR UAS- DESIGN

SISTEM TERINTEGRASI

Berkelompok 2 orang

PEMBAGIAN KELOMPOK

- Kelompok tugas besar UAS terdiri atas 2 sampai 3 orang.
- Jika ada masalah dengan anggota kelompok Anda, misal teman sekelompok Anda tidak responsi atau jarang hadir kelas, silakan laporkan ke dosen untuk mengganti kelompok.
- Kelompok dibuat secara random, lihat edunex untuk mengetahui anggota kelompok Anda.

TOOLS TUGAS BESAR UAS

- Rational Rose atau sejenis untuk memodelkan UML dan DDD
- MS Word atau sejenis untuk membuat laporan.
- GitHub atau sejenis untuk menyimpan source code Anda.

Implementasi Sistem

- PHP server
- MySQL
- Codeigniter 4
- Postman

Bonus Poin:

- Deploy menggunakan Docker

TUGAS PERANCANGAN

- Rancang 2 buah sistem yang terintegrasi satu sama lain.
- Rancang Class Diagram (DDD low-level) masing-masing sistem.
- Rancang integrasi masing-masing sistem menggunakan DDD high level.
- Rancang minimal 1 Core Subdomain pada masing-masing sistem.

TUGAS PEMROGRAMAN

- Implementasikan rancangan sistem Anda menggunakan Codeigniter 4,
- Masing-masing sistem melakukan akses terhadap core subdomain sistem lainnya.
- Implementasikan model otentikasi pada masing-masing sistem.
- Buat rancangan antarmuka sistem pada Codeigniter.

Bonus:

- Deploy sistem Anda menggunakan Docker
- Deploy di online server.

LAPORAN

- Makalah yang berisi desain sistem Anda yang dikerjakan per Kelompok. Pada makalah ini harus terdapat:
 - Deskripsi masing-masing sistem
 - Desain masing-masing sistem, Class Diagram, DDD high-level, desain antarmuka
 - Cara kerja masing-masing sistem (flowchart)
 - Link source code dan youtube presentasi.
- Link GitHub source code Anda dan Database Anda
- Video presentasi yang berisi Anda mempresentasikan sistem yang Anda buat beserta cara kerjanya. Diupload ke youtube.