

Nama Kelompok : DEV Team

Nama Anggota :

1. Dhimas Rizaldy (20312030)
2. Aldi Anggara (20312021)
3. Adhitya Elga Nalendra (20312156)

## Dokumentasi Tugas Akhir PWEB

Frontend-admin

### A. Penjelasan Singkat

Peningkatan urbanisasi dan mobilitas penduduk telah mendorong permintaan yang tinggi untuk fasilitas perumahan sementara, seperti kost atau rumah kos. Namun, proses pencarian dan penyewaan kost masih sering kali rumit dan memakan waktu bagi penyewa dan pemilik kost. Oleh karena itu, sebuah aplikasi penyewaan kost diusulkan untuk meningkatkan efisiensi dan kemudahan dalam proses penyewaan kost.

Aplikasi ini memanfaatkan teknologi digital dan informasi yang diperbarui secara real-time untuk memfasilitasi penyewaan kost berdasarkan kriteria jenis kamar yang diinginkan oleh calon penyewa. Dengan menggunakan aplikasi ini, calon penyewa dapat memfilter pencarian mereka berdasarkan harga, fasilitas, dan ketersediaan. Selain itu, aplikasi ini juga menyediakan informasi lengkap mengenai setiap kost, termasuk deskripsi, foto, dan ulasan dari penyewa sebelumnya.

Bagi pemilik kost, aplikasi ini menyediakan platform yang mudah digunakan untuk memasarkan dan mengelola properti mereka. Pemilik dapat mengunggah informasi dan foto kost mereka, serta mengatur ketersediaan dan harga sewa. Aplikasi ini juga menyediakan fitur komunikasi yang memungkinkan pemilik dan penyewa untuk berinteraksi, melakukan negosiasi, dan mengatur jadwal kunjungan.

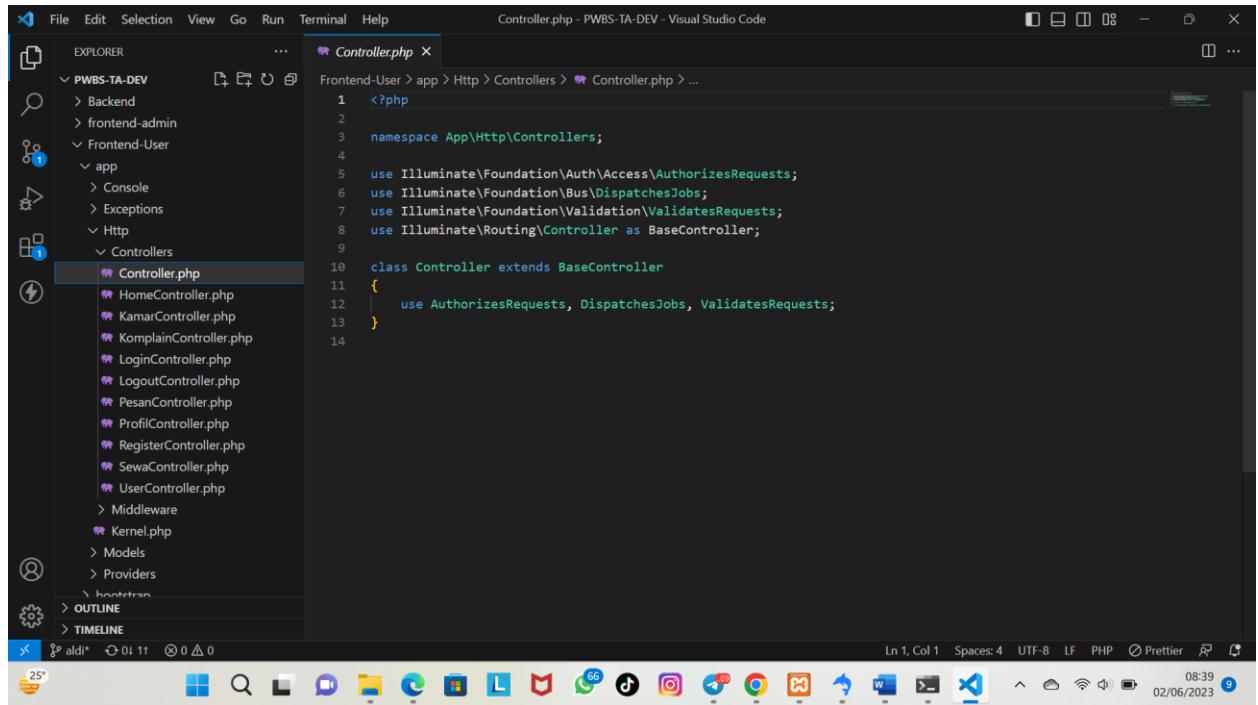
Keuntungan lain dari aplikasi penyewaan kost ini adalah kemampuannya untuk memfasilitasi pembayaran online yang aman dan transparan antara penyewa dan pemilik. Hal ini menghilangkan kebutuhan untuk transaksi tunai, sehingga meningkatkan keamanan dan kenyamanan bagi kedua belah pihak.

Dalam rangka mengembangkan aplikasi ini, perlu dilakukan pengembangan sistem yang andal dan aman, serta pengelolaan data yang baik untuk menjaga privasi dan keamanan pengguna. Selain itu, strategi pemasaran yang efektif juga diperlukan untuk menarik penyewa dan pemilik kost agar agar menggunakan aplikasi ini.

Diharapkan bahwa aplikasi penyewaan kost ini akan memberikan manfaat besar bagi calon penyewa dan pemilik kost dalam hal efisiensi, kemudahan, dan keamanan. Dengan demikian, aplikasi ini dapat menjadi solusi yang inovatif dan relevan dalam mengatasi tantangan dalam proses penyewaan kost di era digital ini.

## Tampilan Coding Controller Frontend-User

### 1. Controller.PHP



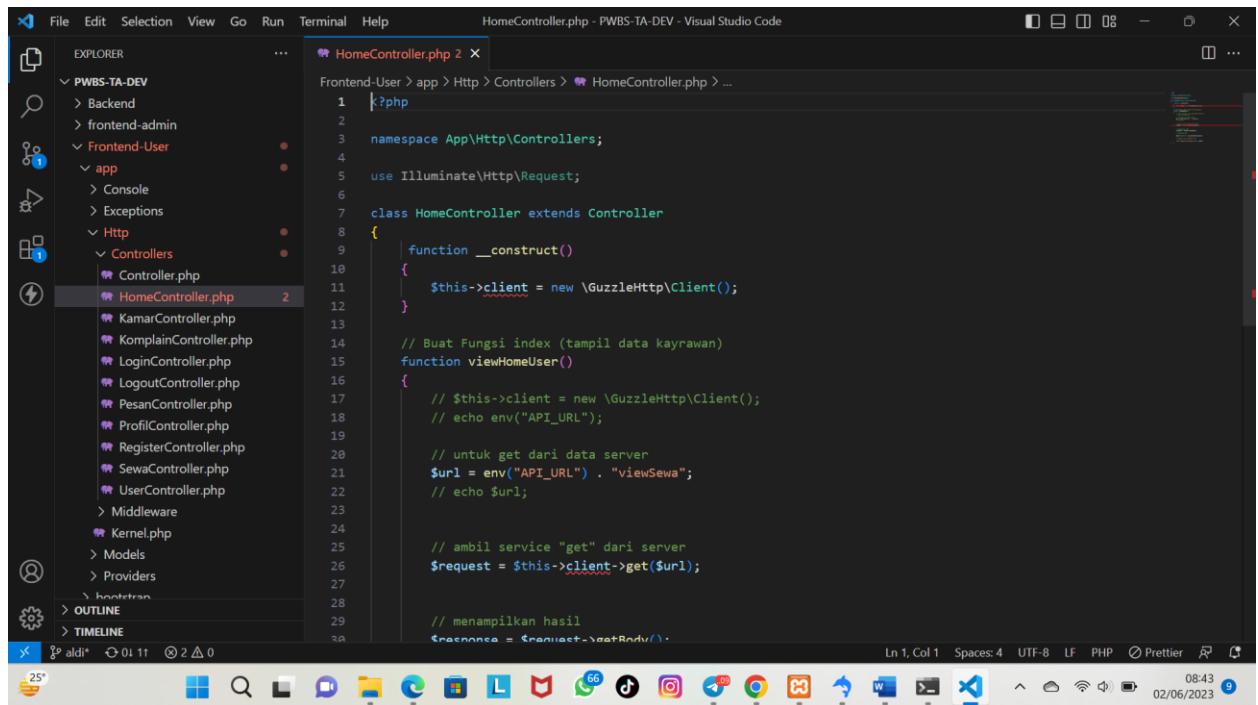
The screenshot shows the Visual Studio Code interface with the following details:

- File Path:** Frontend-User > app > Http > Controllers > Controller.php
- Code Content:**

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
6 use Illuminate\Foundation\Bus\DispatchesJobs;
7 use Illuminate\Foundation\Validation\ValidatesRequests;
8 use Illuminate\Routing\Controller as BaseController;
9
10 class Controller extends BaseController
11 {
12     use AuthorizesRequests, DispatchesJobs, ValidatesRequests;
13 }
14
```

- Bottom Status Bar:** Ln 1, Col 1 | Spaces: 4 | UTF-8 | LF | PHP | Prettier | 08:39 | 02/06/2023

### 2. HomeController.php

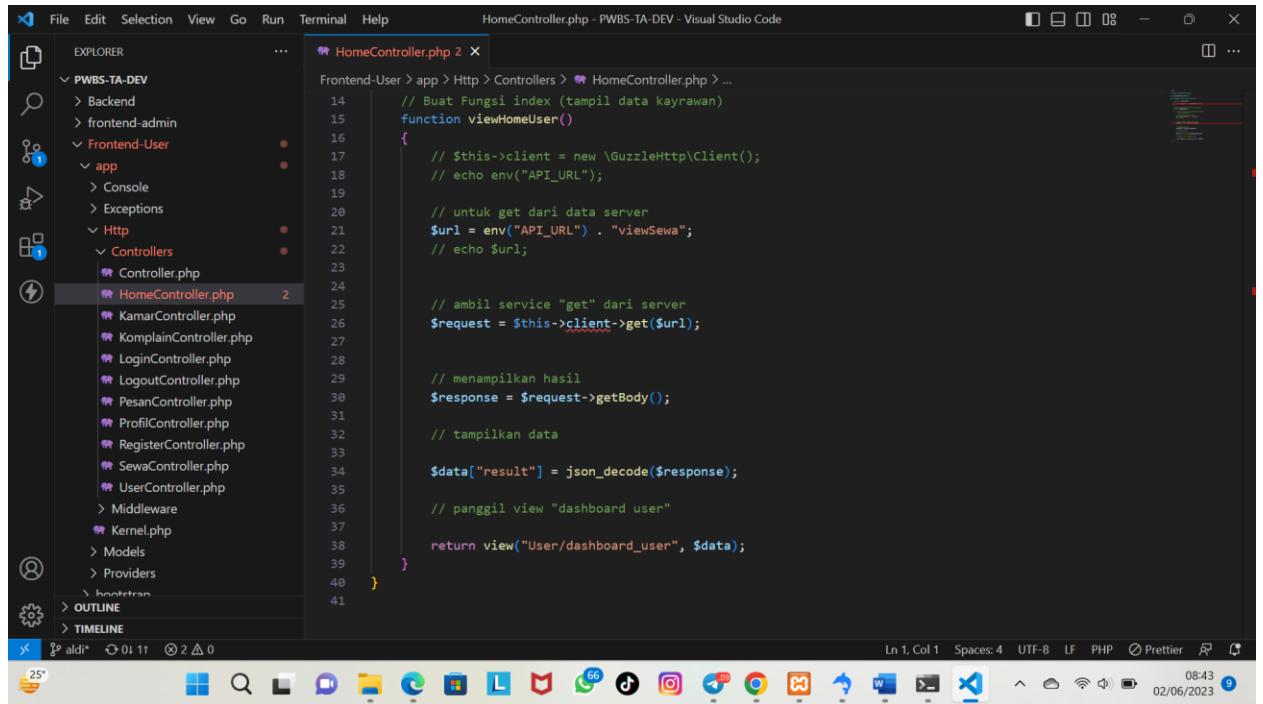


The screenshot shows the Visual Studio Code interface with the following details:

- File Path:** Frontend-User > app > Http > Controllers > HomeController.php
- Code Content:**

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class HomeController extends Controller
8 {
9     function __construct()
10    {
11        $this->client = new \GuzzleHttp\Client();
12    }
13
14    // Buat Fungsi index (tampilkan data kayawan)
15    function viewHomeUser()
16    {
17        // $this->client = new \GuzzleHttp\Client();
18        // echo env("API_URL");
19
20        // untuk get dari data server
21        $url = env("API_URL") . "viewSewa";
22        // echo $url;
23
24
25        // ambil service "get" dari server
26        $request = $this->client->get($url);
27
28
29        // menampilkan hasil
30        $response = $request->getBody();
31    }
32}
```

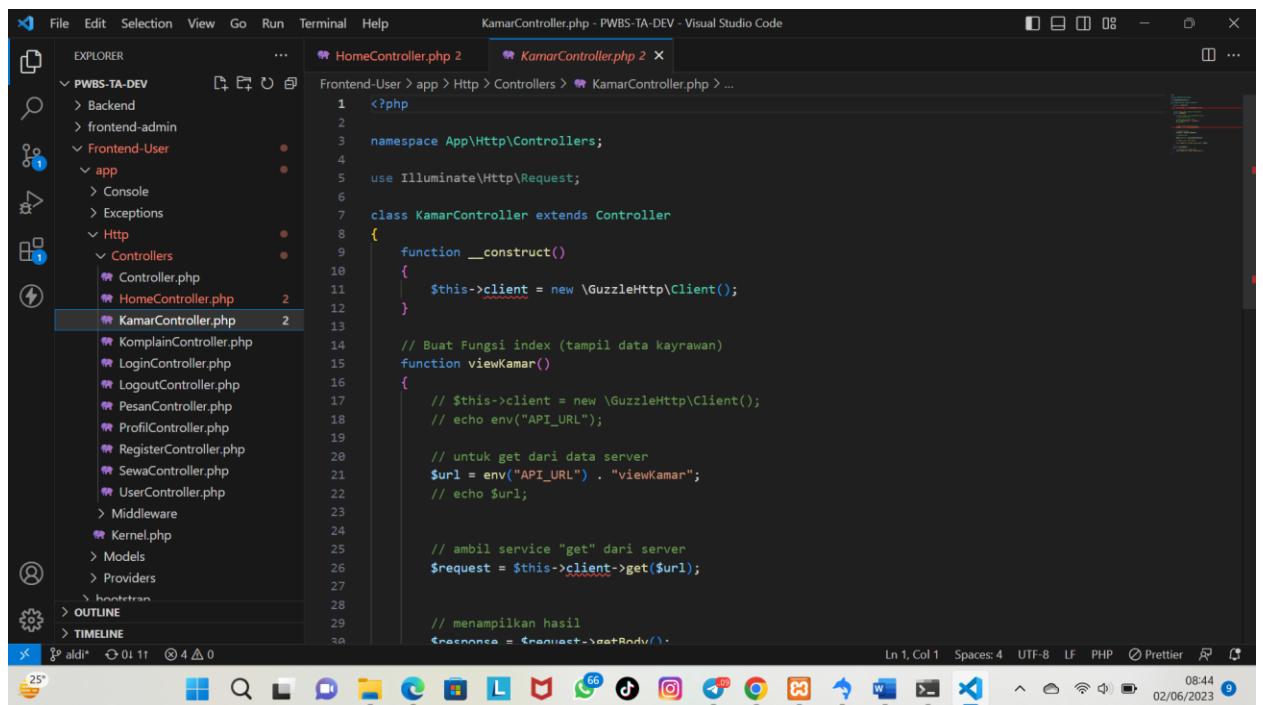
- Bottom Status Bar:** Ln 1, Col 1 | Spaces: 4 | UTF-8 | LF | PHP | Prettier | 08:43 | 02/06/2023



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

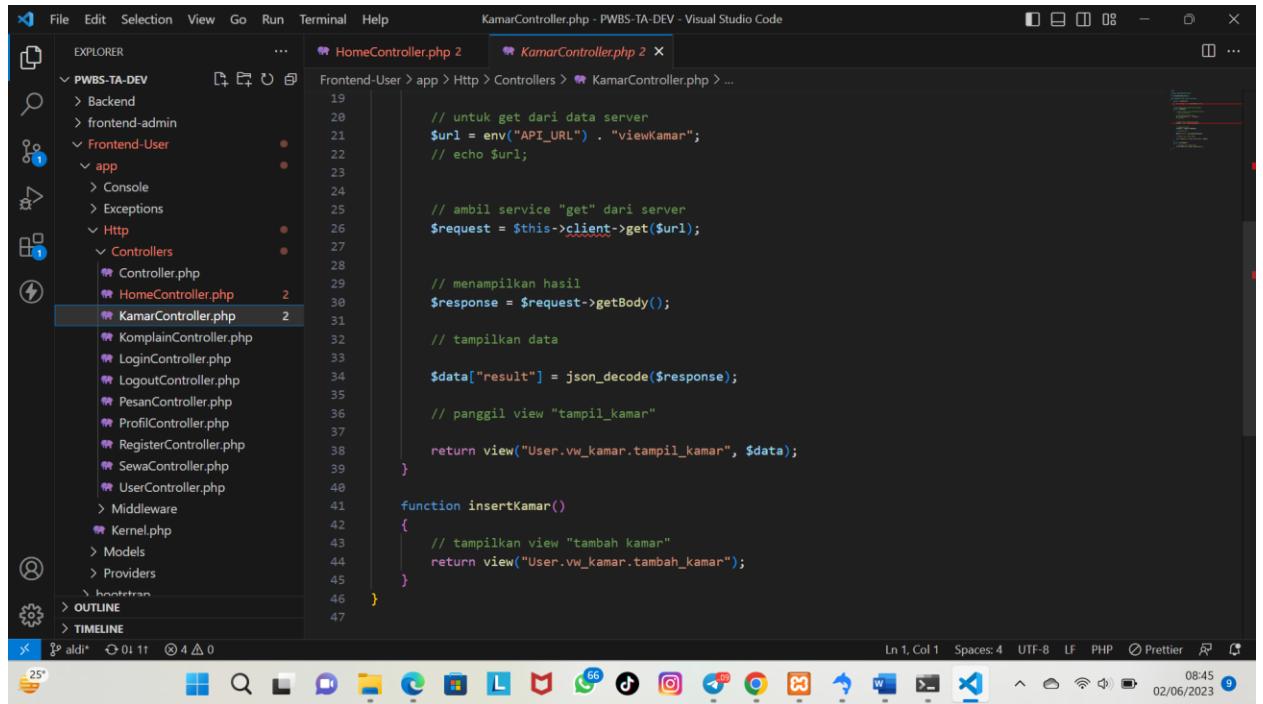
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** The active file is `HomeController.php` (version 2). The code implements a `viewHomeUser()` function that sends a GET request to an API endpoint to retrieve data about users.
- Explorer:** Shows the project structure under `PWBS-TA-DEV`, including `Backend`, `frontend-admin`, `Frontend-User` (with `app` and `Http` subfolders), and various controller files like `Controller.php`, `KamarController.php`, etc.
- Bottom Status Bar:** Shows the current file is `HomeController.php`, version 2, with 41 lines of code. It also displays system information like battery level (25%), date (02/06/2023), and time (08:43).

### 3. KamarController.php



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

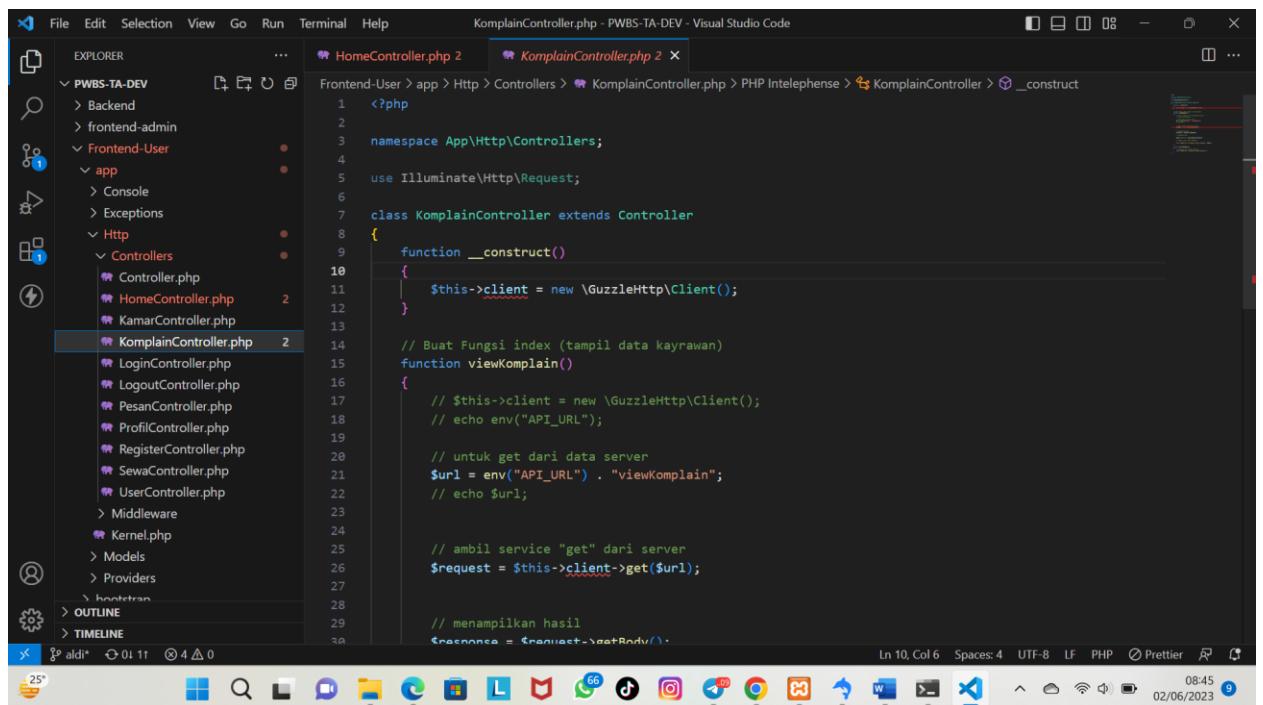
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** The active file is `KamarController.php` (version 2). The code defines a `KamarController` class that extends the `Controller` class and contains a `viewKamar()` method.
- Explorer:** Shows the project structure under `PWBS-TA-DEV`, including `Backend`, `frontend-admin`, `Frontend-User` (with `app` and `Http` subfolders), and various controller files like `Controller.php`, `HomeController.php`, etc.
- Bottom Status Bar:** Shows the current file is `KamarController.php`, version 2, with 39 lines of code. It also displays system information like battery level (25%), date (02/06/2023), and time (08:44).



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

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** The active file is `KamarController.php` located at `Frontend-User > app > Http > Controllers > KamarController.php`. The code implements a `viewKamar` method and an `insertKamar` method.
- Explorer:** Shows the project structure under `PWBS-TA-DEV`, including `Backend`, `frontend-admin`, `Frontend-User` (selected), `app`, `Http`, and `Controllers` (selected). Inside `Controllers`, files like `Controller.php`, `HomeController.php`, and `KamarController.php` are listed.
- Bottom Status Bar:** Shows the current file is `KamarController.php`, line 1, column 1, spaces: 4, encoding: UTF-8, LF, PHP, and Prettier status.

#### 4. KomplainController.php



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

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** The active file is `KomplainController.php` located at `Frontend-User > app > Http > Controllers > KomplainController.php`. It contains a `__construct` method and a `viewKomplain` method.
- Explorer:** Shows the project structure under `PWBS-TA-DEV`, including `Backend`, `frontend-admin`, `Frontend-User` (selected), `app`, `Http`, and `Controllers` (selected). Inside `Controllers`, files like `Controller.php`, `HomeController.php`, `KamarController.php`, and `KomplainController.php` are listed.
- Bottom Status Bar:** Shows the current file is `KomplainController.php`, line 10, column 6, spaces: 4, encoding: UTF-8, LF, PHP, and Prettier status.

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

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** KomplainController.php - PWBS-TA-DEV - Visual Studio Code.
- Explorer:** Shows the project structure under PWBS-TA-DEV, including Frontend-User, app, Http, and Controllers. The KomplainController.php file is selected.
- Editor:** Displays the code for KomplainController.php. The code handles API requests to get complain data and insert new complaints.
- Bottom Status Bar:** Ln 10, Col 6, Spaces: 4, UTF-8, LF, PHP, Prettier, 08:46, 02/06/2023.

```
// untuk get dari data server
$url = env("API_URL") . "viewKomplain";
// echo $url;

// ambil service "get" dari server
$request = $this->client->get($url);

// menampilkan hasil
$response = $request->getBody();

// tampilkan data

$data["result"] = json_decode($response);

// panggil view "tampil_komplain"
return view("User.vw_komplain.tampil_komplain", $data);

}

function insertKomplain()
{
    // tampilkan view "tambah komplain"
    return view("User.vw_komplain.tambah_komplain");
}
```

## 5. LoginController.php

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

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** LoginController.php - PWBS-TA-DEV - Visual Studio Code.
- Explorer:** Shows the project structure under PWBS-TA-DEV, including Frontend-User, app, Http, and Controllers. The LoginController.php file is selected.
- Editor:** Displays the code for LoginController.php. It defines a class LoginController that extends Controller. It includes methods for logging in users and handling login API requests.
- Bottom Status Bar:** Ln 1, Col 1, Spaces: 4, UTF-8, LF, PHP, Prettier, 08:46, 02/06/2023.

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Http;
use GuzzleHttp\Client;
use Auth;
use Session;

class LoginController extends Controller
{
    // Function Login
    public function login()
    {
        return view('login');
    }

    public function loginApi(Request $request)
    {

        $request->validate([
            'email' => 'required|email',
            'password' => 'required|min:8',
        ]);

        try {

            $http = new \GuzzleHttp\Client();
            $email = $request->email;
```

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

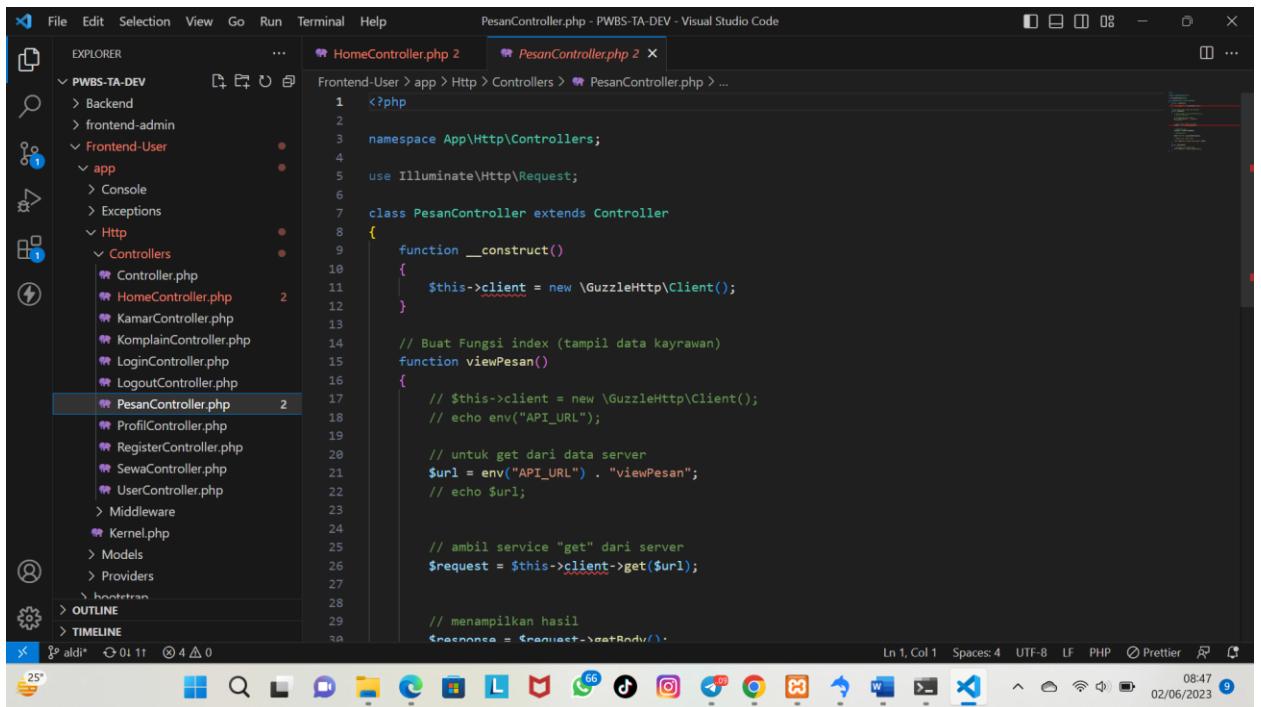
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** LoginController.php - PWBS-TA-DEV - Visual Studio Code.
- Explorer:** Shows the project structure under PWBS-TA-DEV, including Frontend-User, app, Http, and Controllers. The LoginController.php file is selected.
- Editor:** Displays the code for LoginController.php. The code uses GuzzleHttp\Client to make a POST request to a local API endpoint to log in a user by email and password. It handles the response and returns a redirect to the dashboard user if successful, or an error message if there's a catched exception.
- Bottom Status Bar:** Shows the current file (LoginController.php), line 1, column 1, spaces: 4, encoding: UTF-8, line separator: LF, PHP, and Prettier status.
- Taskbar:** Shows various pinned icons for Microsoft Office, social media, and developer tools.

## 6. LogoutController.php

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

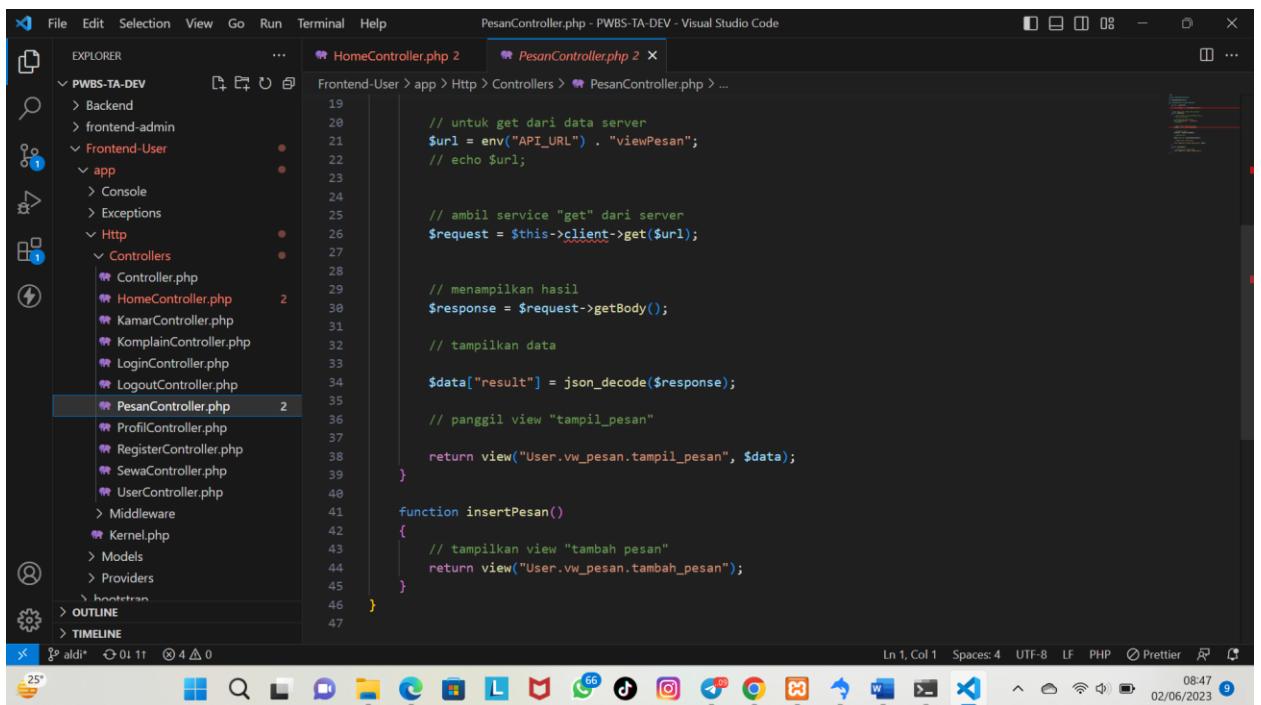
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** LogoutController.php - PWBS-TA-DEV - Visual Studio Code.
- Explorer:** Shows the project structure under PWBS-TA-DEV, including Frontend-User, app, Http, and Controllers. The LogoutController.php file is selected.
- Editor:** Displays the code for LogoutController.php. It starts with a PHP opening tag, defines a namespace App\Http\Controllers, uses Illuminate\Http\Request, and extends the Controller class. The class definition ends with a closing brace on line 10.
- Bottom Status Bar:** Shows the current file (LogoutController.php), line 1, column 1, spaces: 4, encoding: UTF-8, line separator: LF, PHP, and Prettier status.
- Taskbar:** Shows various pinned icons for Microsoft Office, social media, and developer tools.

## 7. PesanController.php



The screenshot shows the Visual Studio Code interface with the PesanController.php file open in the editor. The code implements a PesanController class that extends the Illuminate\Foundation\Http\Controller class. It contains two methods: \_\_construct() and viewPesan(). The viewPesan() method uses a GuzzleHttp\Client to fetch data from an API URL and then returns a view named 'tampil\_pesanan'.

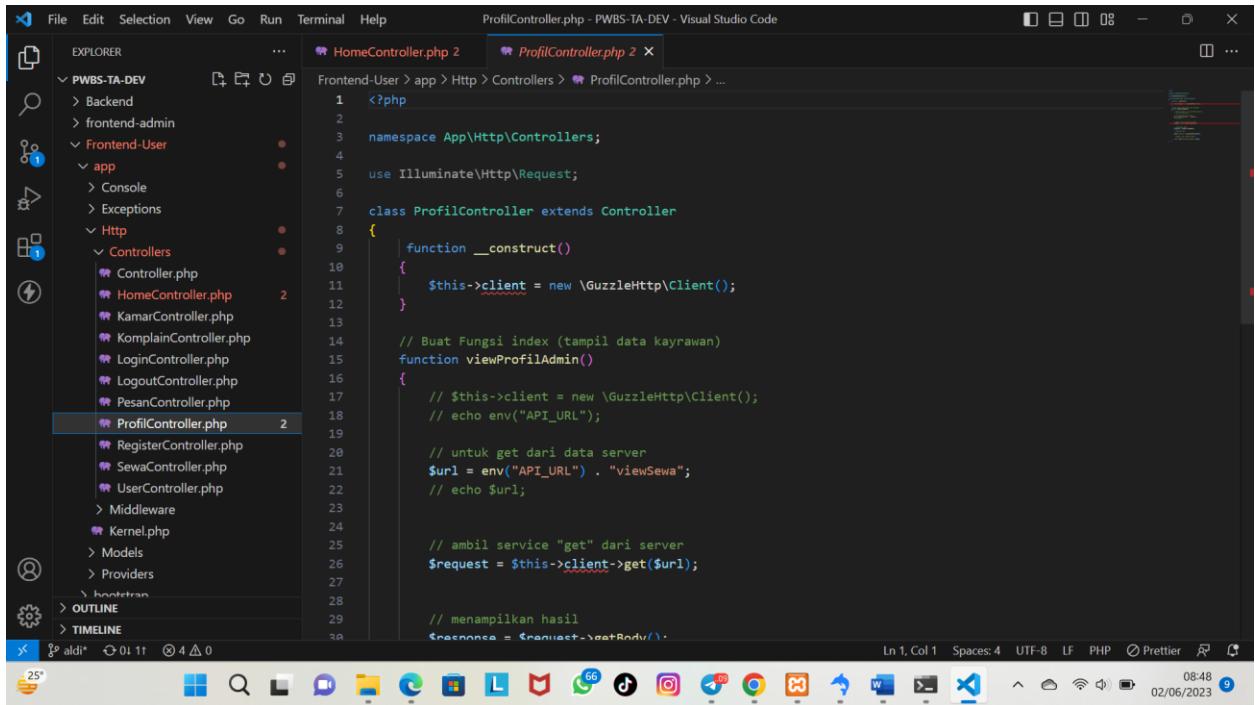
```
<?php  
namespace App\Http\Controllers;  
  
use Illuminate\Http\Request;  
  
class PesanController extends Controller  
{  
    function __construct()  
    {  
        $this->client = new \GuzzleHttp\Client();  
    }  
  
    // Buat Fungsi index (tampil data kayawan)  
    function viewPesan()  
    {  
        // $this->client = new \GuzzleHttp\Client();  
        // echo env("API_URL");  
  
        // untuk get dari data server  
        $url = env("API_URL") . "viewPesan";  
        // echo $url;  
  
        // ambil service "get" dari server  
        $request = $this->client->get($url);  
  
        // menampilkan hasil  
        $response = $request->getBody();  
    }  
}
```



This screenshot shows the same PesanController.php file with additional code added. The viewPesan() method now includes json\_decode() to parse the API response and return a view named 'tampil\_pesanan'. The insertPesan() method is also defined.

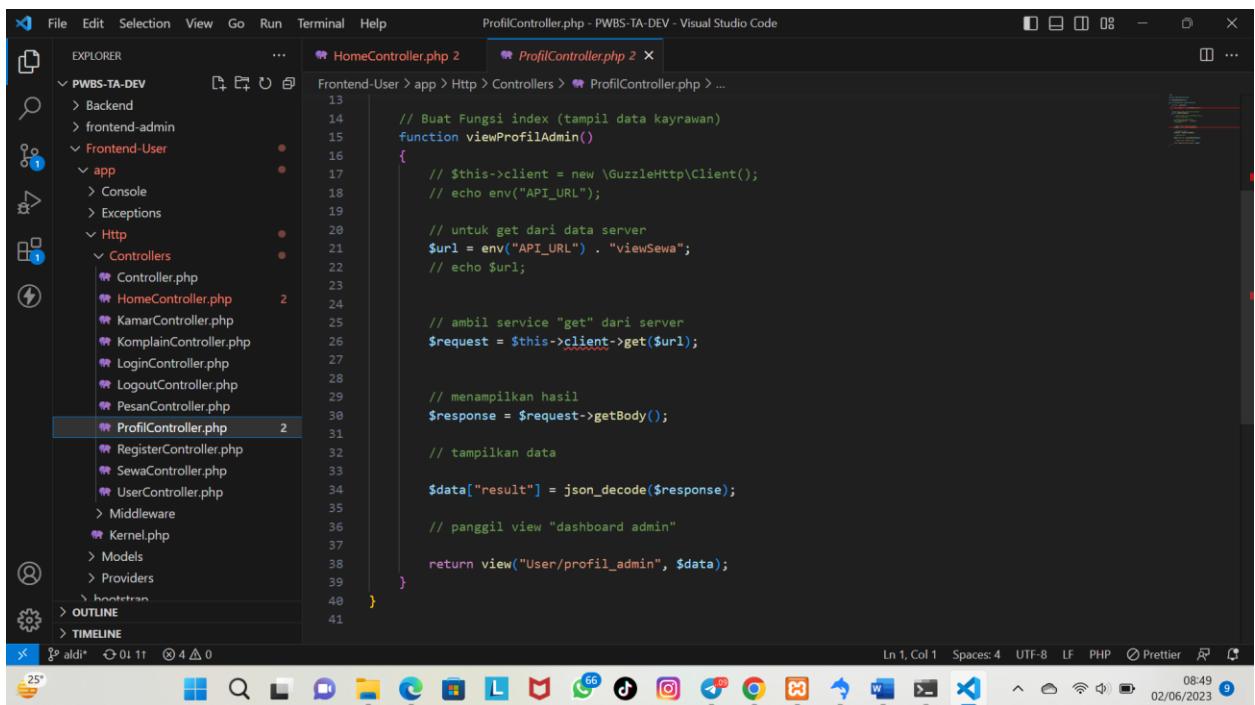
```
// untuk get dari data server  
$url = env("API_URL") . "viewPesan";  
// echo $url;  
  
// ambil service "get" dari server  
$request = $this->client->get($url);  
  
// menampilkan hasil  
$response = $request->getBody();  
  
// tampilkan data  
$data["result"] = json_decode($response);  
  
// panggil view "tampil_pesanan"  
return view("User.vw_pesanan.tampil_pesanan", $data);  
}  
  
function insertPesan()  
{  
    // tampilkan view "tambah pesan"  
    return view("User.vw_pesanan.tambah_pesanan");  
}
```

## 8. ProfilController.php



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

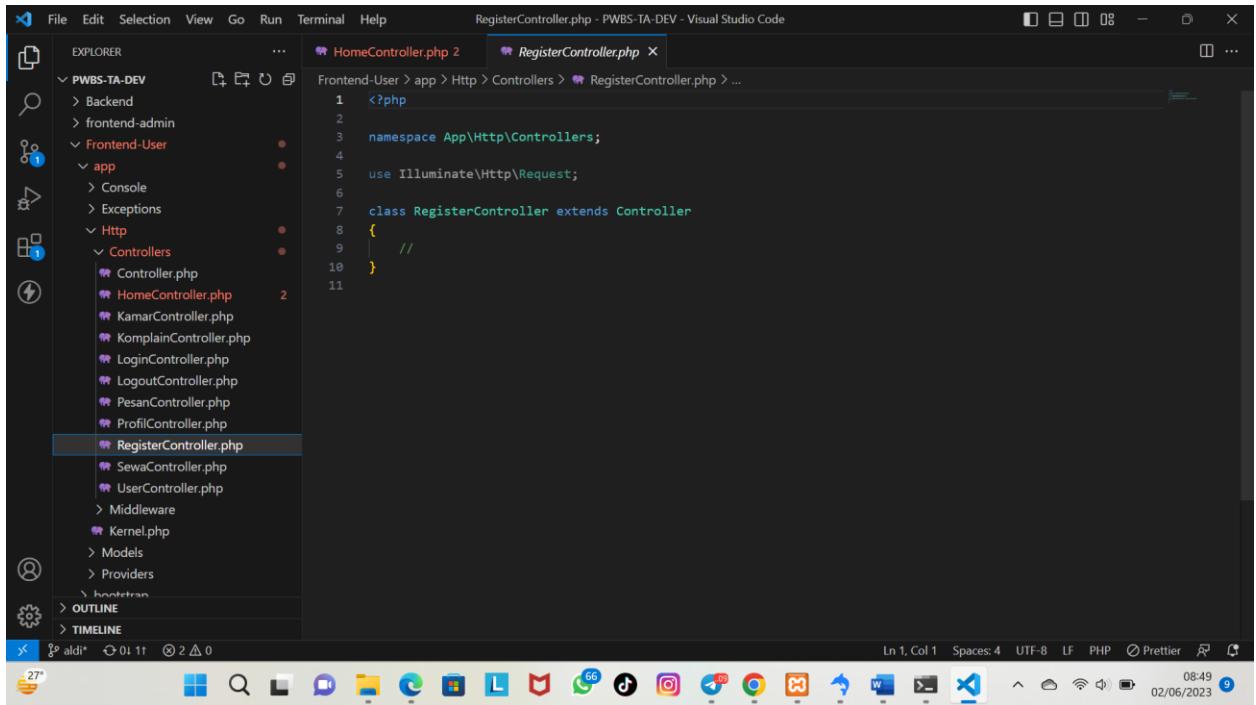
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** Two tabs are open: HomeController.php 2 and ProfilController.php 2.
- Explorer:** Shows the project structure under PWBS-TA-DEV. The Frontend-User folder contains app, Http, and Controllers. The Controllers folder contains several files including ProfilController.php, which is currently selected.
- Code:** The ProfilController.php code is displayed. It includes a constructor that initializes a GuzzleHttp\Client, a viewProfilAdmin() function that makes a GET request to a specified URL, and a return statement that likely renders a view.
- Bottom Bar:** Includes icons for file operations, terminal, and various extensions. The status bar shows "Ln 1, Col 1" and "02/06/2023".



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

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** Two tabs are open: HomeController.php 2 and ProfilController.php 2.
- Explorer:** Shows the project structure under PWBS-TA-DEV. The Frontend-User folder contains app, Http, and Controllers. The Controllers folder contains several files including ProfilController.php, which is currently selected.
- Code:** The ProfilController.php code is displayed. It includes a viewProfilAdmin() function that makes a GET request to a specified URL, processes the response body, and returns a view named "User/profil\_admin" with the processed data.
- Bottom Bar:** Includes icons for file operations, terminal, and various extensions. The status bar shows "Ln 1, Col 1" and "02/06/2023".

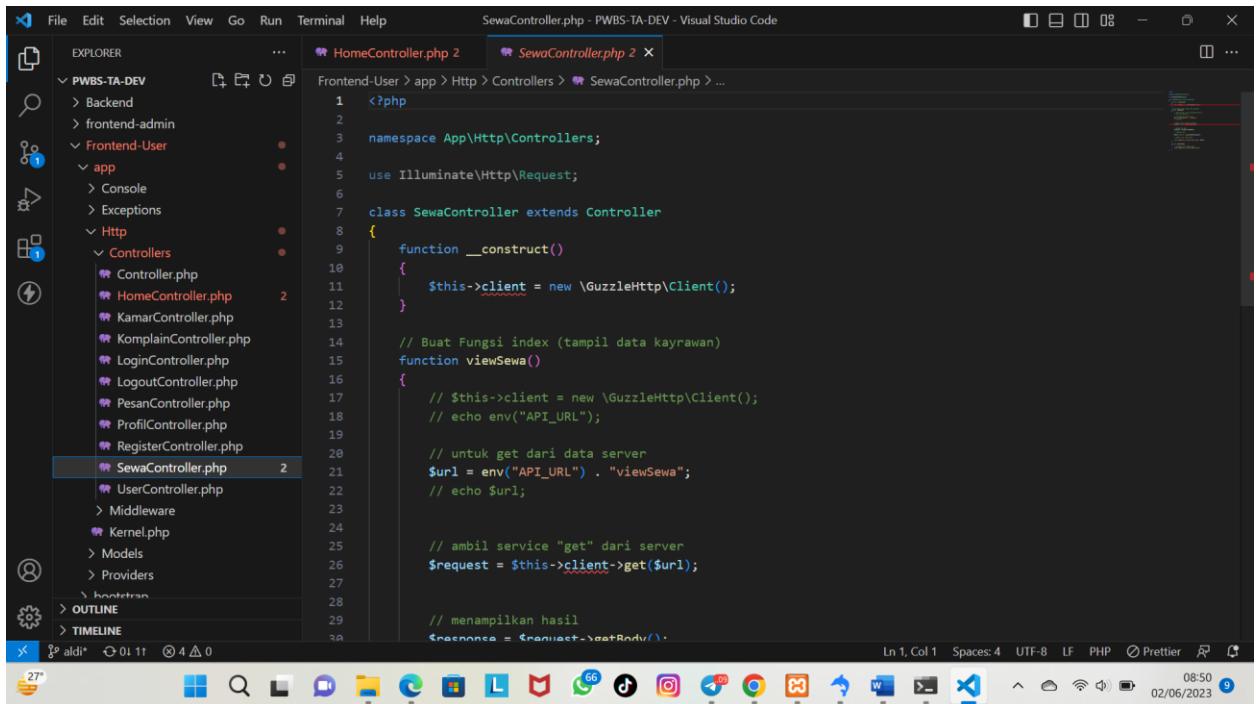
## 9. RegisterController.php



The screenshot shows the Visual Studio Code interface with the file `RegisterController.php` open. The code defines a controller class `RegisterController` that extends `Controller`. The file is located in the `Frontend-User/app/Http/Controllers` directory. The code is as follows:

```
<?php  
namespace App\Http\Controllers;  
use Illuminate\Http\Request;  
class RegisterController extends Controller  
{  
}  
//
```

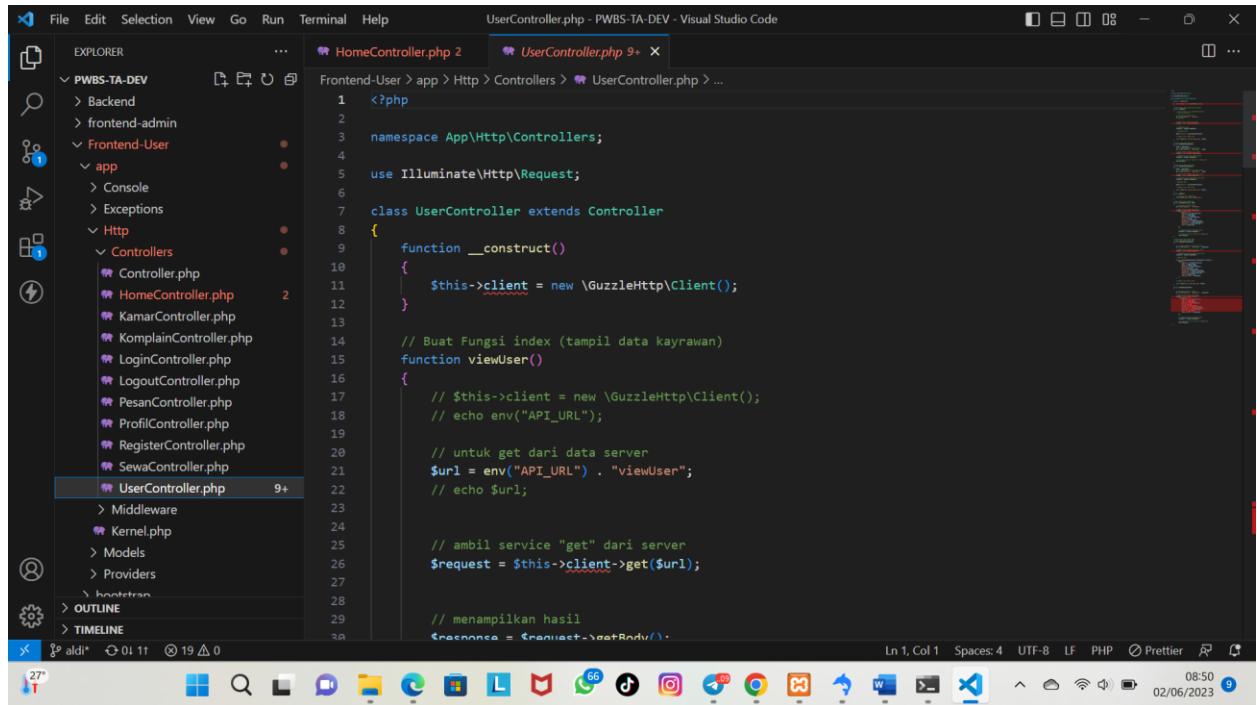
## 10. SewaController.php



The screenshot shows the Visual Studio Code interface with the file `SewaController.php` open. The code defines a controller class `SewaController` that extends `Controller`. It includes a constructor that initializes a Guzzle client and several methods for handling requests. The file is located in the `Frontend-User/app/Http/Controllers` directory. The code is as follows:

```
<?php  
namespace App\Http\Controllers;  
use Illuminate\Http\Request;  
class SewaController extends Controller  
{  
    function __construct()  
    {  
        $this->client = new \GuzzleHttp\Client();  
    }  
  
    // Buat Fungsi index (tampilkan data kayawan)  
    function viewSewa()  
    {  
        // $this->client = new \GuzzleHttp\Client();  
        // echo env("API_URL");  
  
        // untuk get dari data server  
        $url = env("API_URL") . "viewSewa";  
        // echo $url;  
  
        // ambil service "get" dari server  
        $request = $this->client->get($url);  
  
        // menampilkan hasil  
        $response = $request->getBody();  
    }  
}
```

## 11. UserController.php



The screenshot shows the Visual Studio Code interface with the file `UserController.php` open. The code implements a `UserController` class that extends `Controller`. It uses a `GuzzleHttp\Client` to interact with an API. The `viewUser()` function sends a GET request to the API to retrieve user data. The `deleteUser($parameter)` function sends a DELETE request to the API to delete a user by their ID.

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;

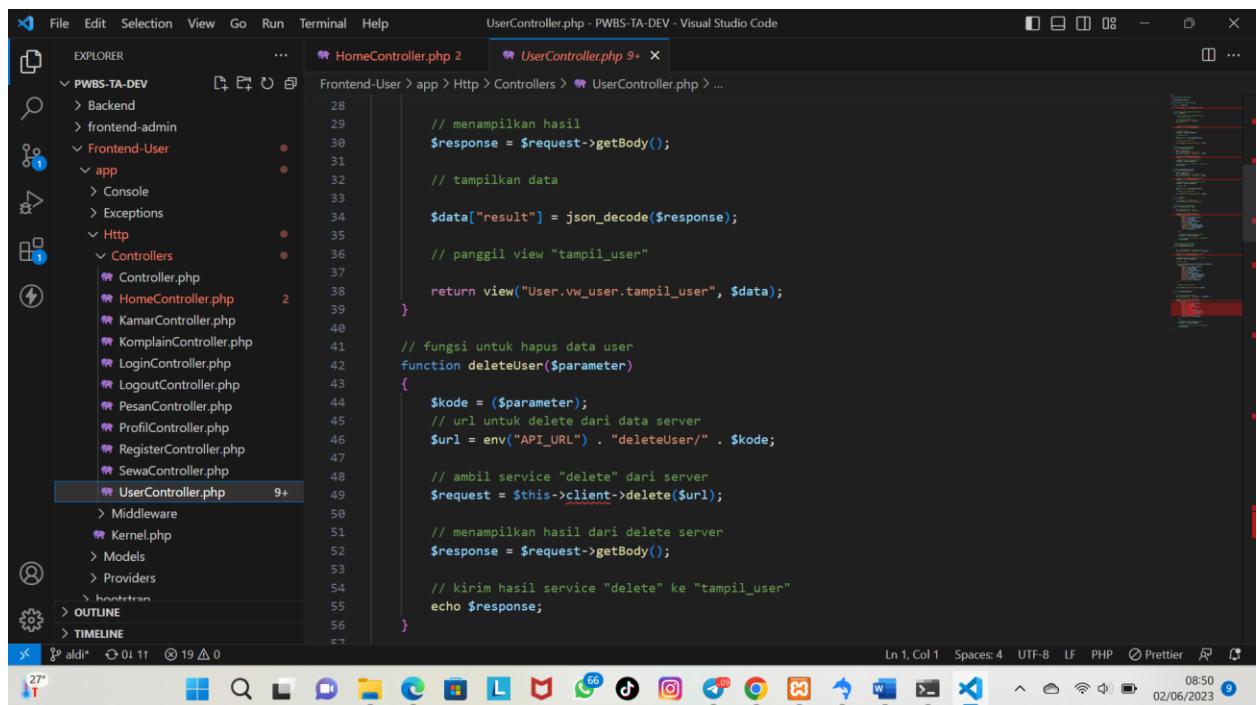
class UserController extends Controller
{
    function __construct()
    {
        $this->client = new \GuzzleHttp\Client();
    }

    // Buat Fungsi index (tampil data kayrawan)
    function viewUser()
    {
        // $this->client = new \GuzzleHttp\Client();
        // echo env("API_URL");

        // untuk get dari data server
        $url = env("API_URL") . "viewUser";
        // echo $url;

        // ambil service "get" dari server
        $request = $this->client->get($url);

        // menampilkan hasil
        $response = $request->getBody();
    }
}
```



The screenshot shows the Visual Studio Code interface with the file `UserController.php` open. The code has been updated to include a `getBody()` call after the `get()` request, and it now decodes the JSON response into an array. The `deleteUser($parameter)` function has also been modified to echo the response from the delete request.

```
// menampilkan hasil
$response = $request->getBody();

// tampilkan data

$data["result"] = json_decode($response);

// panggil view "tampil_user"

return view("User.vw_user.tampil_user", $data);

// fungsi untuk hapus data user
function deleteUser($parameter)
{
    $kode = ($parameter);
    // url untuk delete dari data server
    $url = env("API_URL") . "deleteUser/" . $kode;

    // ambil service "delete" dari server
    $request = $this->client->delete($url);

    // menampilkan hasil dari delete server
    $response = $request->getBody();

    // kirim hasil service "delete" ke "tampil_user"
    echo $response;
}
```

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

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** The main area displays the code for `UserController.php`. The code handles user detail retrieval and addition via an API.
- Explorer:** On the left, the file tree shows the project structure under `PWBS-TA-DEV`, including `Frontend-User`, `app`, `Http`, and `Controllers`.
- Bottom Status Bar:** Shows file paths, line numbers (e.g., 27), and system status like battery level (27%), time (08:51), and date (02/06/2023).

```
// fungsi untuk detail data user
function detailUser($parameter)
{
    $kode = ($parameter);
    // url untuk detail_user dari data server
    $url = env("API_URL") . "detailUser/" . $kode;

    // ambil service "detail_user" dari server
    $request = $this->client->get($url);

    // menampilkan hasil dari detail_user server
    $response = $request->getBody();

    // tampilkan data

    $data["result"] = json_decode($response);

    // panggil view "detail_user"

    return view("User.vw_user.detail_user", $data);
}

function addUser()
{
    // tampilkan view "tambah_user"
    return view("User.vw_user.tambah_user");
}
```

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

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** The main area displays the code for `UserController.php`, which includes a function to insert new users.
- Explorer:** On the left, the file tree shows the project structure under `PWBS-TA-DEV`, including `Frontend-User`, `app`, `Http`, and `Controllers`.
- Bottom Status Bar:** Shows file paths, line numbers (e.g., 27), and system status like battery level (27%), time (08:51), and date (02/06/2023).

```
// buat fungsi untuk simpan data
function insertUser(Request $req)
{
    // untuk post dari data server
    $url = env("API_URL") . "insertUser";

    // ambil service "post" dari server
    $request = $this->client->post($url,[
        "form_params" => [
            "kode_user" => $req->kode_user,
            "nama" => $req->nama,
            "email" => $req->email,
            "password" => $req->password,
            "jk" => $req->jk,
            "alamat" => $req->alamat,
            "no_telepon" => $req->no_telepon,
            "status_user" => $req->status_user,
            "foto_user" => $req->foto_user,
            "role" => $req->role,
        ]
    ]);

    // menampilkan hasil dari post server
    $response = $request->getBody();

    // kirim hasil service "post" ke "tambah_user"
    echo $response;
}
```

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

- File Path:** Frontend-User > app > Http > Controllers > UserController.php
- Code Content (UserController.php):**

```
114 // buat fungsi untuk update data
115 // fungsi untuk detail data user
116 function updateUser($parameter)
{
    // url untuk detail_user dari data server
    $url = env("API_URL") . "detailUser/" . $parameter;

    // ambil service "detail_user" dari server
    $request = $this->client->get($url);

    // menampilkan hasil dari detail_user server
    $response = $request->getBody();

    // tampilkan data

    foreach (json_decode($response)->DetailUser as $hasil) {
        $data = [
            "kode_user" => $hasil->kode_user,
            "nama" => $hasil->nama,
            "email" => $hasil->email,
            "password" => $hasil->password,
            "jk" => $hasil->jk,
            "alamat" => $hasil->alamat,
            "no_telepon" => $hasil->no_telepon,
            "status_user" => $hasil->status_user,
            "foto_user" => $hasil->foto_user,
            "role" => $hasil->role,
            "kode_user_lama" => $parameter
        ];
    }
}
```

- Bottom Status Bar:** Ln 1, Col 1 | Spaces: 4 | UTF-8 | LF | PHP | Prettier | 08:51 | 02/06/2023

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

- File Path:** Frontend-User > app > Http > Controllers > UserController.php
- Code Content (UserController.php):**

```
128 // tampilkan data
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157

    foreach (json_decode($response)->DetailUser as $hasil) {
        $data = [
            "kode_user" => $hasil->kode_user,
            "nama" => $hasil->nama,
            "email" => $hasil->email,
            "password" => $hasil->password,
            "jk" => $hasil->jk,
            "alamat" => $hasil->alamat,
            "no_telepon" => $hasil->no_telepon,
            "status_user" => $hasil->status_user,
            "foto_user" => $hasil->foto_user,
            "role" => $hasil->role,
            "kode_user_lama" => $parameter
        ];
    }

    // panggil view "detail_user"
    return view("User.vw_user.edit_user", $data);
}

function editData($parameter)
{
    // untuk post dari data server
    $url = env("API_URL") . "updateUser/" . $parameter;
}
```

- Bottom Status Bar:** Ln 1, Col 1 | Spaces: 4 | UTF-8 | LF | PHP | Prettier | 08:52 | 02/06/2023

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

- File Explorer:** On the left, the file structure for "PWBS-TA-DEV" is displayed under "Frontend-User". The "UserController.php" file is currently selected.
- Code Editor:** The main area shows the code for "UserController.php". The code handles a POST request to update a user's information. It uses the "client" object to make a PUT request to the API URL, passing form parameters like "kode\_user", "nama", "email", "password", "jk", "alamat", "no\_telepon", "status\_user", "foto\_user", "role", and "kode\_user\_baru". It then retrieves the response body from the request.
- Status Bar:** At the bottom, the status bar shows "Line 1, Col 1", "Spaces: 4", "UTF-8", "LF", "PHP", and "Prettier" icons.
- Bottom Icons:** A row of icons for file operations (New, Open, Save, Find, Copy, Paste, etc.) and system status (Battery, Network, etc.).

## Tampilan Web Frontend-User

## 1. Dasboard E-KOS

The screenshot shows a web-based dashboard titled "E-KOS". On the left, a sidebar lists navigation items: "Dashboard" (highlighted in green), "Data Kamar", "Data Komplain", "Data Booking Kamar", and "Data Riwayat Booking". The main content area displays a welcome message "Hallo User, Selamat Datang di Dashboard Kami." Below this, a section titled "Dashboard" contains three cards: "1 Belum Bayar" (Unpaid) with a document icon, "2 Sudah Bayar" (Paid) with a dollar sign icon, and "1 Komplain" (Complaint) with a wrench icon.

## 2. Tampil Data Kamar

The screenshot shows a web-based application interface for managing room data. The top navigation bar includes links to WhatsApp, YouTube, TA-PWB, Adhiyaya, Sistem C, Course, Application, Google translate, and other system-related items. Below the bar, a search bar and a user profile icon are visible.

The main header "E-KOS" is displayed above a sidebar menu. The sidebar contains the following items:

- Dashboard (selected)
- Data Kamar
- Data Komplain
- Data Booking Kamar
- Data Riwayat Booking

The main content area is titled "Tampil Data Kamar". It features a table with the following columns: No, Kode Kamar, Nama kamar, Lantai, Deskripsi, Status, Harga, Foto, and Aksi. Two entries are listed:

No	Kode Kamar	Nama kamar	Lantai	Deskripsi	Status	Harga	Foto	Aksi
1	kmr01	Kamar A01	Lantai 1	Kamar Lengkap	Kosong	750000	kmr01.jpg	<a href="#">+ Booking</a>
2	kmr02	Kamar A02	Lantai 1	Kamar Lengkap	Kosong	750000	kmr02.jpg	<a href="#">+ Booking</a>

Below the table, there are buttons for "Search" and "Show" (set to 10 entries). The bottom of the screen shows a Windows taskbar with various pinned icons and the date/time (02/06/2023, 20:59).

## 3. Tampil Data Komplain

The screenshot shows the same web-based application interface, now displaying the "Tampil Data Komplain" page. The top navigation bar and sidebar are identical to the previous screenshot.

The main content area is titled "Tampil Data Komplain". It features a table with the following columns: No, Kode Komplain, Kode User, Kode Kamar, Perihal, and Isi Pesan. Two entries are listed:

No	Kode Komplain	Kode User	Kode Kamar	Perihal	Isi Pesan
1	KDKMP01	10102939	kmr01	Keran Air Rusak dan Genteng Bocor	Pak Tolong Diperbaiki Pak dikam
2	KDKMP02	23432424	kmr02	Token Listrik Habis	Pak Tolong dikar saya Toker

Below the table, there are buttons for "Search" and "Show" (set to 10 entries). At the bottom, a message indicates "Showing 1 to 2 of 2 entries" and includes "Previous", "Next", and a page number "1". The bottom of the screen shows a Windows taskbar with various pinned icons and the date/time (02/06/2023, 21:00).

#### 4. Tampil Data Booking

The screenshot shows a web browser window with the URL [127.0.0.1:8001/User/vw\\_pesan/tampil\\_pesan](http://127.0.0.1:8001/User/vw_pesan/tampil_pesan). The page title is "Tampil Data Booking". On the left, there is a sidebar with navigation links: "Dashboard", "Data Kamar", "Data Komplain", "Data Booking Kamar", and "Data Riwayat Booking". The main content area displays a table titled "Show" with a dropdown menu set to "10 entries". The table has columns: No, Kode Pesan, Kode kamar, Kode User, Tanggal Masuk, Lama Tinggal, Status, and Aksi. Two entries are listed:

No	Kode Pesan	Kode kamar	Kode User	Tanggal Masuk	Lama Tinggal	Status	Aksi
1	KPSN01	kmr01	10102939	2023-05-31	2 Bulan		
2	KPSN02	kmr02	23432424	2023-05-31	3 Bulan		

At the bottom, it says "Showing 1 to 2 of 2 entries" with "Previous" and "Next" buttons. The browser toolbar at the top includes various icons like WhatsApp, YouTube, Maps, and Gmail. The taskbar at the bottom shows several pinned applications.

#### 5. Tampil Data Riwayat Booking

The screenshot shows a web browser window with the URL [127.0.0.1:8001/User/vw\\_sewa/tampil\\_sewa](http://127.0.0.1:8001/User/vw_sewa/tampil_sewa). The page title is "Tampil Data Riwayat Booking". On the left, there is a sidebar with navigation links: "Dashboard", "Data Kamar", "Data Komplain", "Data Booking Kamar", and "Data Riwayat Booking". The main content area displays a table titled "Show" with a dropdown menu set to "10 entries". The table has columns: No, Kode Sewa, Kode Pesan, Kode User, Kode kamar, Tanggal Keluar, Pembayaran, Status, and Aksi. Two entries are listed:

No	Kode Sewa	Kode Pesan	Kode User	Kode kamar	Tanggal Keluar	Pembayaran	Status	Aksi
1	KDSW01	KPSN01	10102939	kmr01	2023-07-31	650000	DiCicil	
2	KDSW02	KPSN02	23432424	kmr02	2023-05-31	650000	DiCicil	

At the bottom, it says "Showing 1 to 2 of 2 entries" with "Previous" and "Next" buttons. The browser toolbar at the top includes various icons like WhatsApp, YouTube, Maps, and Gmail. The taskbar at the bottom shows several pinned applications.