

# **LAPORAN KERJA PRAKTIK**

## **PEMBUATAN WEBSITE ADMIN AGEN SAMPAH E-TRASHSERVICE**

Diajukan Guna Memenuhi Syarat Kuliah Kerja Praktik

Program Strata 1 Jurusan Teknologi Informasi

Fakultas Teknik



Oleh:

**MUHAMMAD FERDY YOSA**

**20180140054**

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**FAKULTAS TEKNIK**

**UNIVERSITAS MUHAMMADIYAH YOGYAKARTA**

**2021**

# **LEMBAR PENGESAHAN**

## **LAPORAN KERJA PRAKTIK**

Berlokasi di

**Creative Gama Studio**

Periode 26 Juli 2021 – 7 September 2021

**Pembuatan Website Admin Agen Sampah E-TrashService**

**Disusun oleh:**

**Muhammad Ferdy Yosa**

**20180140054**

**Yogyakarta, 24 September 2021**

**Mengesahkan,**

**DOSEN PEMBIMBING KERJA PRAKTIK JUGA SELAKU**

**KETUA JURUSAN TEKNOLOGI INFORMASI**



**Asroni, S.T., M.Eng**

**NIDN. 0526047401**

## KATA PENGANTAR

Alhamdulillah rabbil'aalamiin, segala puji dan syukur penulis panjatkan kepada Allah Subhanahu wa Ta'ala. Karena-Nya pelaksanaan dan penyusunan laporan Kerja Praktik (KP) yang berjudul "Pembuatan Website Admin Agen Sampah *E-TrashService*" dapat terselesaikan dengan lancar.

Laporan Kerja Praktik ini adalah sebagai syarat untuk memenuhi tugas mata kuliah wajib pada Jurusan Teknologi Informasi, Fakultas Teknik, Universitas Muhammadiyah Yogyakarta.

Pada kesempatan ini, penulis ingin mengucapkan terima kasih kepada semua pihak yang telah membantu, khususnya kepada:

1. Allah Subhanahu wa Ta'ala atas segala tuntunan, kemudahan, kesehatan, kelancaran dan keselamatan yang diberikan, sehingga bisa menyelesaikan kerja praktik ini dengan baik.
2. Kedua orang tua dan keluarga yang selalu memberikan dukungan kepada penulis dalam menyelesaikan kerja praktik ini.
3. Bapak Asroni S.T., M.Eng. selaku Ketua Jurusan Teknologi Informasi Universitas Muhammadiyah Yogyakarta serta selaku Dosen Pembimbing Kerja Praktik kali ini.
4. Bapak Syahli Lilipali. selaku Direktur Creative Gama Studio yang telah memberikan izin untuk melaksanakan kerja praktik kali ini pada Creative Gama Studio.
5. Lingga Buwono Jati, Fakhrizar Rifqi Fuad dan Syafira Jafar selaku teman sekelompok yang telah berusaha keras dan berjuang bersama-sama dalam penyelesaian kerja praktik kali ini.
6. Semua pihak yang berperan dalam pelaksanaan penulisan dan penyelesaian laporan yang tidak dapat penulis sebutkan satu persatu.

Akhirnya, penulis menyadari bahwa pelaksanaan kerja praktik dan penyusunan laporan ini masih belum sempurna. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan. Demikianlah pembukaan yang dapat penulis utarakan, semoga

laporan ini dapat bermanfaat bagi penulis pribadi maupun pembacanya.

Yogyakarta, 24 September 2021

Penulis,

A handwritten signature in black ink, appearing to read 'Ferdy'.

Muhammad Ferdy Yosa  
(20180140054)

# DAFTAR ISI

<b>KATA PENGANTAR</b> .....	i
<b>DAFTAR ISI</b> .....	iii
<b>DAFTAR GAMBAR</b> .....	v
<b>BAB I PENDAHULUAN</b> .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	1
1.3 Maksud dan Tujuan .....	1
1.4 Manfaat.....	2
1.4.1 Bagi Mahasiswa .....	2
1.4.2 Bagi Perusahaan.....	2
<b>1.5 Waktu dan Tempat Pelaksanaan Magang</b> .....	2
1.5.1 Lokasi Pelaksanaan Magang.....	2
1.5.2 Waktu Pelaksanaan Magang .....	3
<b>BAB II LANDASAN TEORI</b> .....	4
2.1 <i>PHP (Hypertext Preprocessor)</i> .....	4
2.2 <i>HTML (HyperText Markup Language)</i> .....	4
2.3 <i>CSS (Cascading Style Sheet)</i> .....	4
2.4 <i>VS Code (Visual Studio Code)</i> .....	4
2.6 <i>JS (JavaScript)</i> .....	5
<b>BAB III PERANCANGAN</b> .....	6
3.1 Alat .....	6
3.2 Use Case Diagram .....	6
3.2 Entity Relationship Diagram .....	7
3.3 Class Diagram .....	8
<b>BAB IV IMPLEMENTASI</b> .....	9
4.1 Database .....	9
4.2 Halaman Sambutan .....	9
4.3 Halaman Login .....	10
4.4 Halaman Dashboard .....	10
4.5 Halaman Data Admin.....	11

4.6 Halaman Data Member .....	12
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>20</b>
5.1 Kesimpulan.....	20
5.2 Saran.....	20
<b>LAMPIRAN.....</b>	<b>21</b>
<b>DAFTAR PUSTAKA.....</b>	<b>105</b>

## DAFTAR GAMBAR

<b>Gambar 1.</b> Use Case Diagram .....	6
<b>Gambar 2.</b> Entity Relationship Diagram .....	7
<b>Gambar 3.</b> Class Diagram .....	8
<b>Gambar 4.</b> Database E-TrashService.....	9
<b>Gambar 5.</b> Tampilan Sambutan E-TrashService .....	10
<b>Gambar 6.</b> Tampilan Login E-TrashService .....	10
<b>Gambar 7.</b> Tampilan Dashboard E-TrashService.....	11
<b>Gambar 8.</b> Tampilan Data Admin .....	11
<b>Gambar 9.</b> Update Data Admin .....	12
<b>Gambar 10.</b> Halaman Data Member.....	12
<b>Gambar 11.</b> Halaman Tambah Data Member.....	13
<b>Gambar 12.</b> Halaman Edit Data Member .....	13
<b>Gambar 13.</b> Search Data Member .....	14
<b>Gambar 14.</b> Halaman Data Tipe Pembayaran .....	14
<b>Gambar 15.</b> Halaman Tambah Tipe Pembayaran.....	15
<b>Gambar 16.</b> Halaman Edit Tipe Pembayaran .....	15
<b>Gambar 17.</b> Search Tipe Pembayaran .....	16
<b>Gambar 18.</b> Halaman Data Transaksi.....	16
<b>Gambar 19.</b> Halaman Tambah Data Transaksi.....	17
<b>Gambar 20.</b> Halaman Edit Data Transaksi .....	17
<b>Gambar 21.</b> Search Data Transaksi .....	18
<b>Gambar 22.</b> Halaman Rekap Pembayaran.....	18
<b>Gambar 23.</b> Halaman Hasil Sortir Bulan.....	19

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dewasa ini, penerapan teknologi banyak diterapkan pada berbagai aspek, sehingga dapat mempermudah kehidupan. Namun disisi lain, masih ada aspek yang kurang menerapkan teknologi, salah satunya adalah pengelolaan sampah.

Hal ini dapat berpengaruh terhadap proses pengelolaan sampah itu sendiri baik dari segi kecepatan pengolahan maupun dari segi kepraktisan dalam pengelolaan sampahnya, sehingga dapat terjadi penimbunan sampah yang mencemari lingkungan.

Creative Gama Studio (CGS) merupakan perusahaan yang bergerak dalam bidang kreativitas Informasi Teknologi (IT) yang bertempat di Jl. Tata Bumi Selatan No. 109, Kel. Banyuraden, Kec. Gamping, Sleman, Yogyakarta dan dipimpin oleh bapak Syahli Lilipali. Oleh karena itu, CGS dipilih sebagai tempat pelaksanaan kerja praktik kali ini, sehingga dapat meningkatkan pengalaman dan wawasan serta untuk mempersiapkan diri memasuki dunia kerja.

### **1.2 Rumusan Masalah**

Berdasarkan latar belakang diatas, dapat diangkat permasalahan yang ada, yakni sebagai berikut:

1. Bagaimana meningkatkan peranan teknologi dalam bidang pengelolaan sampah.
2. Bagaimana cara mempermudah proses pengelolaan sampah.

### **1.3 Maksud dan Tujuan**

Adapun maksud dan tujuan yang hendak dicapai dalam pelaksanaan kerja praktik kali ini, antara lain adalah sebagai berikut:



- Membantu pengelolaan sampah dalam menggunakan bidang teknologi.
- Menerapkan teknologi dalam aspek pengelolaan sampah.
- Meningkatkan persiapan dan kemampuan dalam menghadapi dunia kerja.
- Memenuhi syarat untuk menyelesaikan studi/perkuliahan prodi Teknologi Informasi Universitas Muhammadiyah Yogyakarta.

## **1.4 Manfaat**

### 1.4.1 Bagi Mahasiswa

- Mendapatkan pengalaman dan gambaran terkait kondisi pada saat memasuki dunia kerja.
- Mendapatkan kesempatan untuk mengimplementasikan ilmu yang didapat saat berkuliah kedalam dunia kerja.
- Sebagai refleksi diri dalam kemampuan saat ini apabila diterapkan dalam dunia kerja.

### 1.4.2 Bagi Perusahaan

- Mendapatkan bantuan dalam menyelesaikan proyek perusahaan.
- Mendapatkan proyek baru dan lebih mudah terselesaikan dengan tepat waktu.

## **1.5 Waktu dan Tempat Pelaksanaan Magang**

### 1.5.1 Lokasi Pelaksanaan Magang

Keterangan mengenai lokasi magang kali ini adalah sebagai berikut:

Nama Perusahaan : Creative Gama Studio

Alamat : Jl. Tata Bumi Selatan No. 109, Kel. Banyuraden, Kec. Gamping, Sleman, Yogyakarta.

### 1.5.2 Waktu Pelaksanaan Magang

Adapun pelaksanaan kegiatan kerja praktik berlangsung selama 1 bulan, 1 pekan 5 hari. Dimulai pada tanggal 26 Juli 2021 – 7 September 2021. Kerja praktik pada kesempatan kali ini berlangsung secara *Work From Home (WFH)* disebabkan kondisi pandemi *CoVid-19* yang masih dalam kondisi kritis.

*WFH* adalah metode bekerja dari rumah, namun interaksi dengan pihak perusahaan tetap berjalan secara online.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 PHP (*Hypertext Preprocessor*)**

PHP adalah bahasa scripting yang memiliki tujuan umum (*general purpose*). Kode PHP biasanya diproses pada web server oleh penerjemah PHP yang diimplementasikan sebagai module. Pada web server, hasil dari terjemahan dan kode PHP yang sudah dieksekusi yang mungkin menjadi tipe data, seperti HTML yang dihasilkan atau data gambar biner. PHP memiliki kelebihan yang sangat membantu yaitu dapat membuat web menjadi lebih dinamis serta mendukung banyak web server. (*www.php.net-2020*).

#### **2.2 HTML (*HyperText Markup Language*)**

HTML adalah bahasa pemrograman khusus web dan merupakan bahasa standar dalam pembuatan web, yang kemudian diakses untuk memperlihatkan berbagai informasi dalam sebuah penjelajah web Internet (*Browser*). (*World Wide Web Consortium-1997*).

#### **2.3 CSS (*Cascading Style Sheet*)**

CSS bukan merupakan bahasa pemrograman melainkan CSS adalah bahasa *style sheet* yang berfokus pada tampilan website itu sendiri, seperti ukuran, warna, posisi, serta dapat digunakan untuk sarana penyampaian informasi atau konten yang ingin disampaikan kepada pengguna.

#### **2.4 VS Code (*Visual Studio Code*)**

*VS Code* adalah software kode editor yang digunakan untuk membuat website/aplikasi, baik dari sisi tampilan/desainnya maupun dari sisi fungsi website/aplikasinya dirancang untuk Linux, MacOS dan Windows. Fitur yang tersedia pada software ini yaitu penyorotan sintaks, merefaktor kode, dan sebagainya. (*McBreen, Sean-2015*).

## **2.5 PHPMYAdmin**

*PHPMYAdmin* adalah software database yang berfungsi untuk mengatur basis data (*database*) pada suatu website. *PHPMYAdmin* mendukung berbagai operasi yang diantaranya berfungsi untuk mengelola basis data, tabel-tabel, bidang, relasi, index, pengguna, perizinan, dan lain-lain. (*phpMyAdmin.net-2010*)

## **2.6 JS (JavaScript)**

JS berfungsi untuk membuat website agar lebih interaktif. JS itu sendiri merupakan bahasa pemrograman tingkat tinggi serta dinamis. JS sangat populer di dunia internet dan dapat bekerja di sebagian besar penjelajah web yang juga populer seperti *Google Chrome*, *Mozilla Firefox*, *Netscape*, dan *Opera*. Disisi lain, JS juga merupakan salah satu teknologi inti *World Wide Web* selain *HTML* dan *CSS*. (*Flanagan, David-2011. Java Script: The Definitive Guide (edisi ke 6-th)*).

# BAB III

## PERANCANGAN

### 3.1 Alat

Alat yang digunakan dalam pembuatan website admin ini, yakni sebagai berikut:

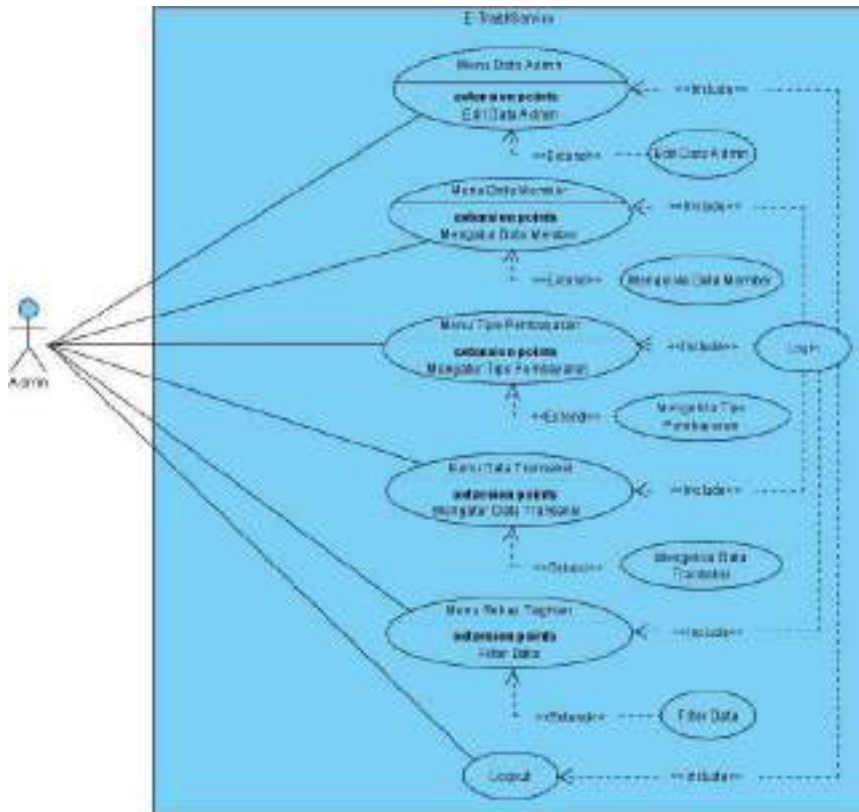
a. Perangkat Keras (*Hardware*)

- Processor Intel Ci7-6700HQ 2.6-3.5 GHz.
- 8 GB RAM.
- 583 GB HDD.

b. Perangkat Lunak (*Software*)

- XAMPP.
- Visual Studio Code.
- OS Windows 10.

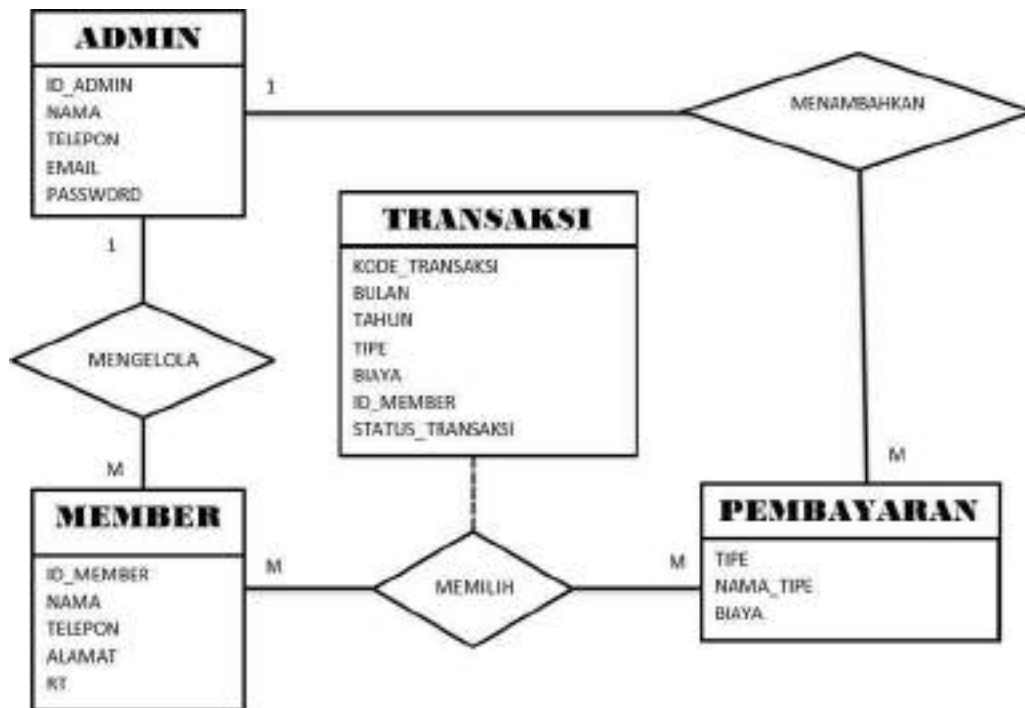
### 3.2 Use Case Diagram



Gambar 1. Use Case Diagram

**Gambar 1** adalah Use Case Diagram dari aplikasi website *E-TrashService* dimana admin dapat melakukan login, mengelola data admin, member, pembayaran, transaksi, dan tagihan serta dapat logout. Sebelum mengakses menu data admin, member, pembayaran, transaksi, dan tagihan, admin diharuskan untuk login terlebih dahulu, setelah itu dapat melakukan pengolahan data tersebut.

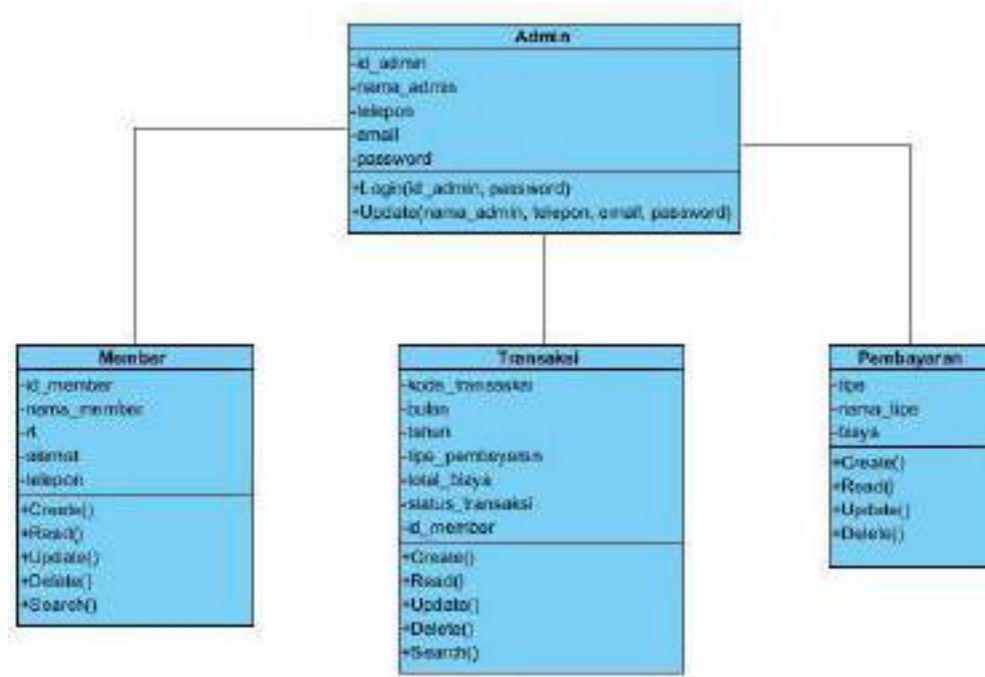
### 3.2 Entity Relationship Diagram



**Gambar 2.** Entity Relationship Diagram

**Gambar 2** adalah Entity Relationship Diagram/ERD dari aplikasi website *E-TrashService* dimana entitas admin berelasi dengan entitas pembayaran dengan kardinalitas 1 to Many, entitas pembayaran dengan entitas member dengan kardinalitas Many to Many yang menghasilkan entitas baru Bernama transaksi, kemudian entitas Member berelasi dengan entitas admin dengan kardinalitas 1 to Many.

### 3.3 Class Diagram



**Gambar 3.** Class Diagram

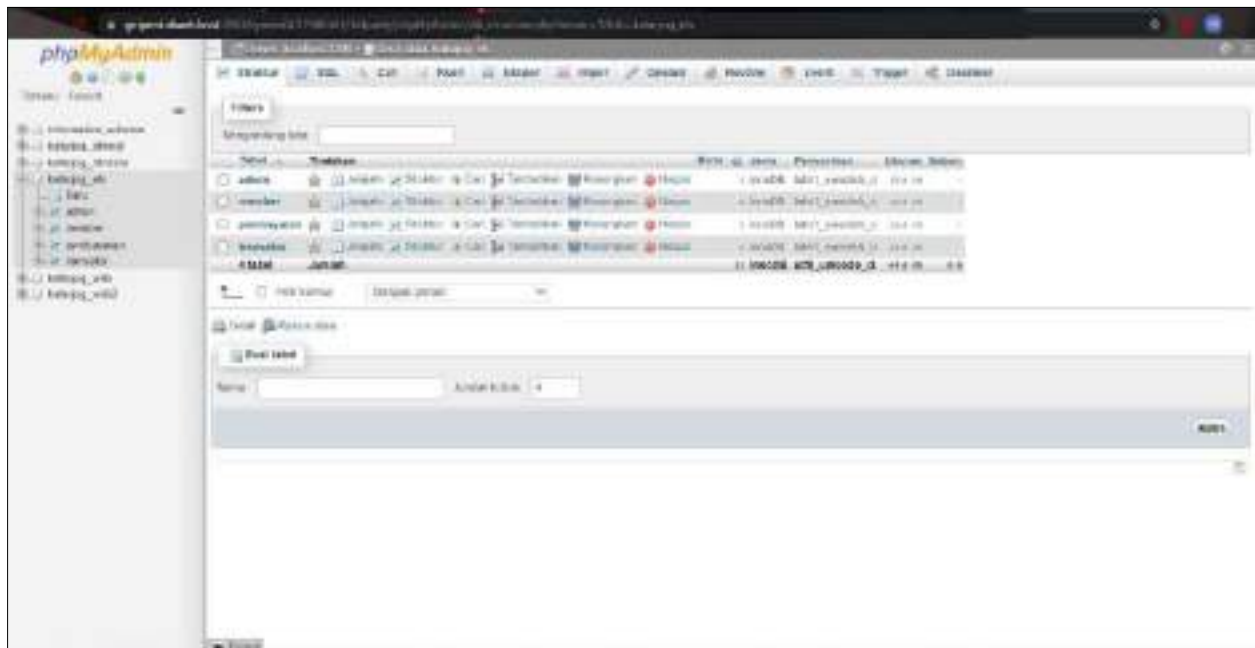
**Gambar 3** adalah Class Diagram dari aplikasi website *E-TrashService*, terdapat 4 class yakni Admin, Member, Transaksi dan Pembayaran. Class Admin memiliki operasi Login dan Update Data Admin. Class Admin juga dapat mengelola data Member yakni CRUDS data member, seperti yang ditunjukkan pada operasi class Member. Class Admin juga dapat melakukan CRUDS terhadap data Transaksi, seperti yang ditunjukkan pada operasi class Transaksi. Serta, class Admin juga dapat melakukan CRUD terhadap data Pembayaran seperti yang ditunjukkan pada operasi class Pembayaran.

## BAB IV

### IMPLEMENTASI

#### 4.1 Database

**Gambar 4** adalah database dari website *E-TrashService*. Terdapat 4 tabel didalamnya, yang masing-masing memiliki fungsi yang berbeda. Yaitu tabel admin, member, pembayaran dan transaksi. Tabel admin digunakan untuk memperlihatkan data admin pada websitenya, serta digunakan pada saat mengupdate data admin dan melakukan login. Tabel member, pembayaran dan transaksi secara berurutan digunakan untuk pengolahan data member, pembayaran dan transaksi.

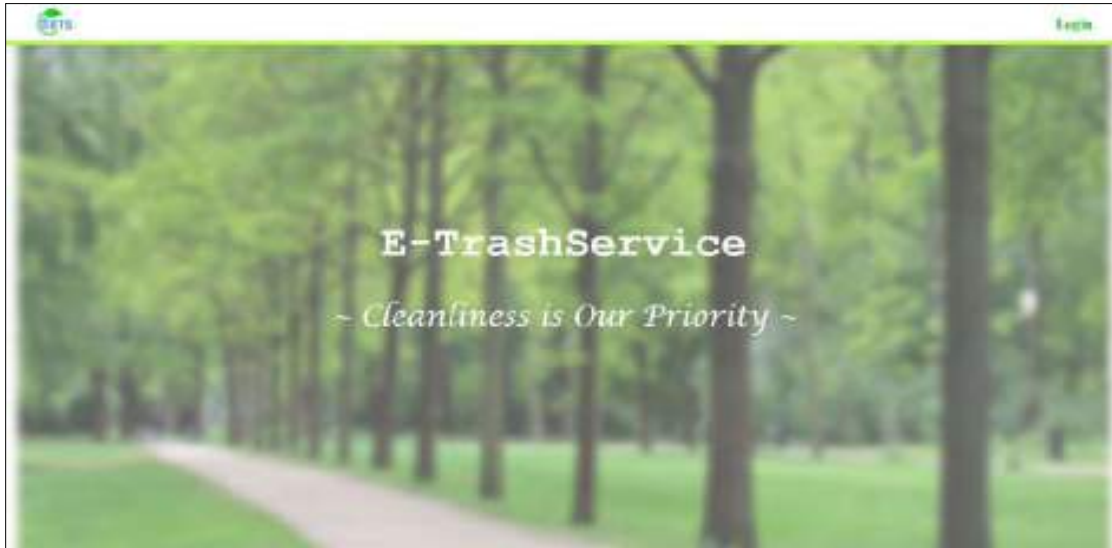


**Gambar 4.** Database *E-TrashService*

#### 4.2 Halaman Sambutan

**Gambar 5** adalah halaman dari website *E-TrashService* yang berfungsi sebagai sambutan kepada *user*.





**Gambar 5.** Tampilan Sambutan *E-TrashService*

#### 4.3 Halaman Login

**Gambar 6** adalah tampilan login dari website *E-TrashService*. Login dapat dilakukan oleh Admin.



**Gambar 6.** Tampilan Login *E-TrashService*

#### 4.4 Halaman Dashboard

**Gambar 7** adalah halaman beranda/home dari website *E-TrashService*, serta menampilkan jumlah data dari masing-masing kategori.



**Gambar 7.** Tampilan Dashboard *E-TrashService*

#### 4.5 Halaman Data Admin

**Gambar 8** dan **Gambar 9** adalah halaman website *E-TrashService* bagian data Admin dan berfungsi menampilkan data Admin secara rinci, serta digunakan untuk mengupdate data Admin.



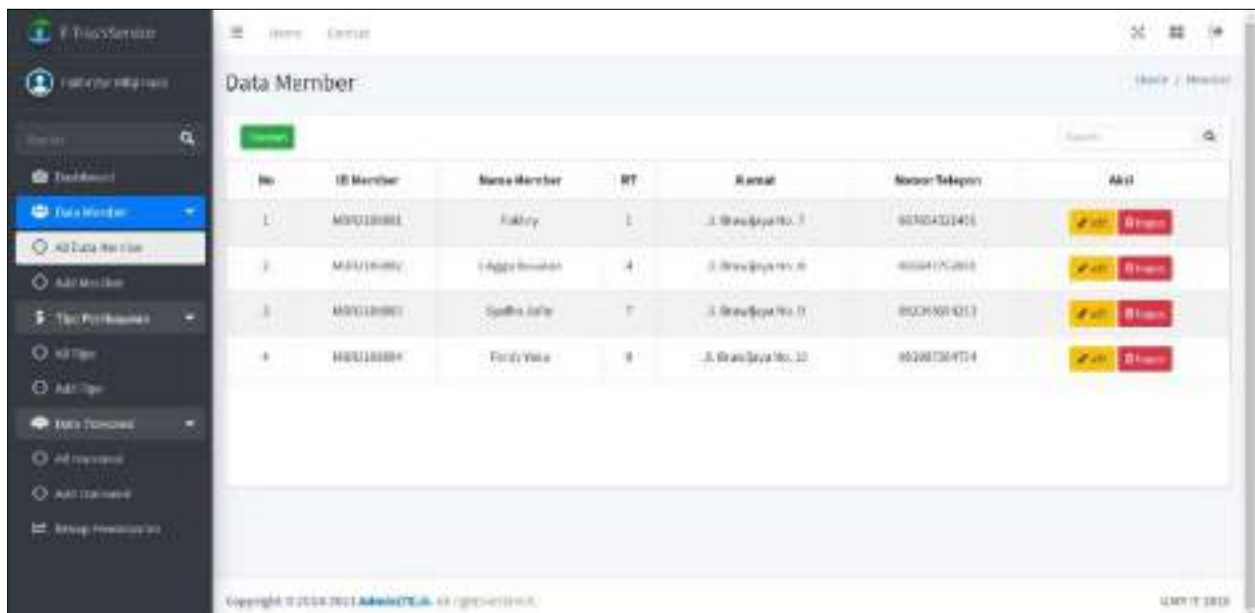
**Gambar 8.** Tampilan Data Admin



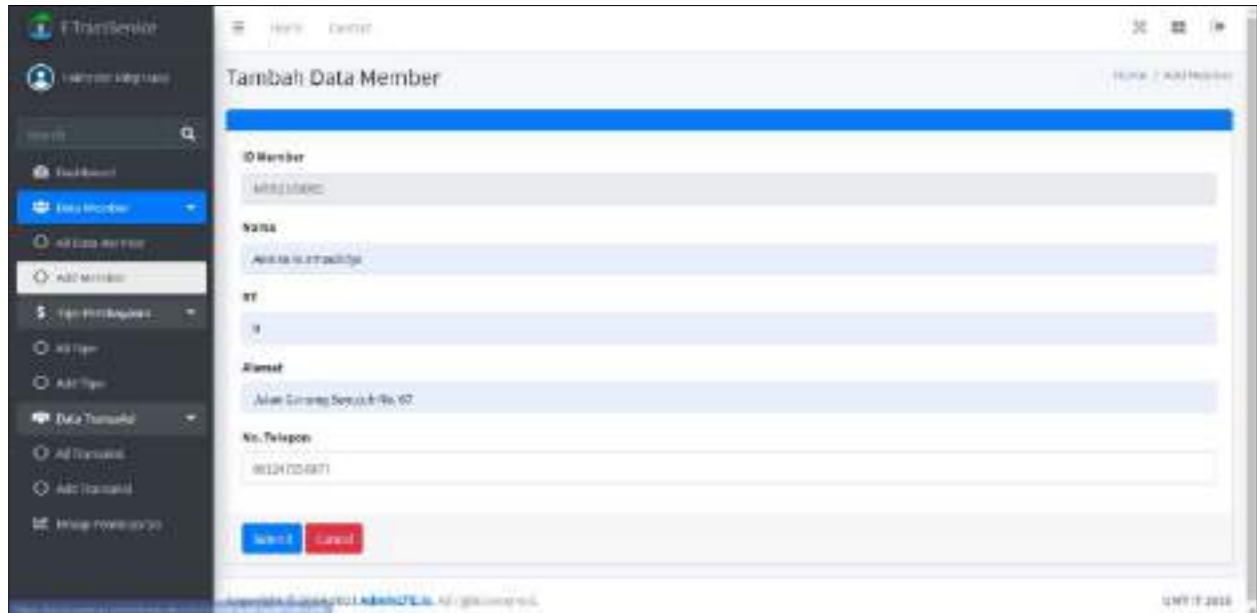
**Gambar 9.** Update Data Admin

#### 4.6 Halaman Data Member

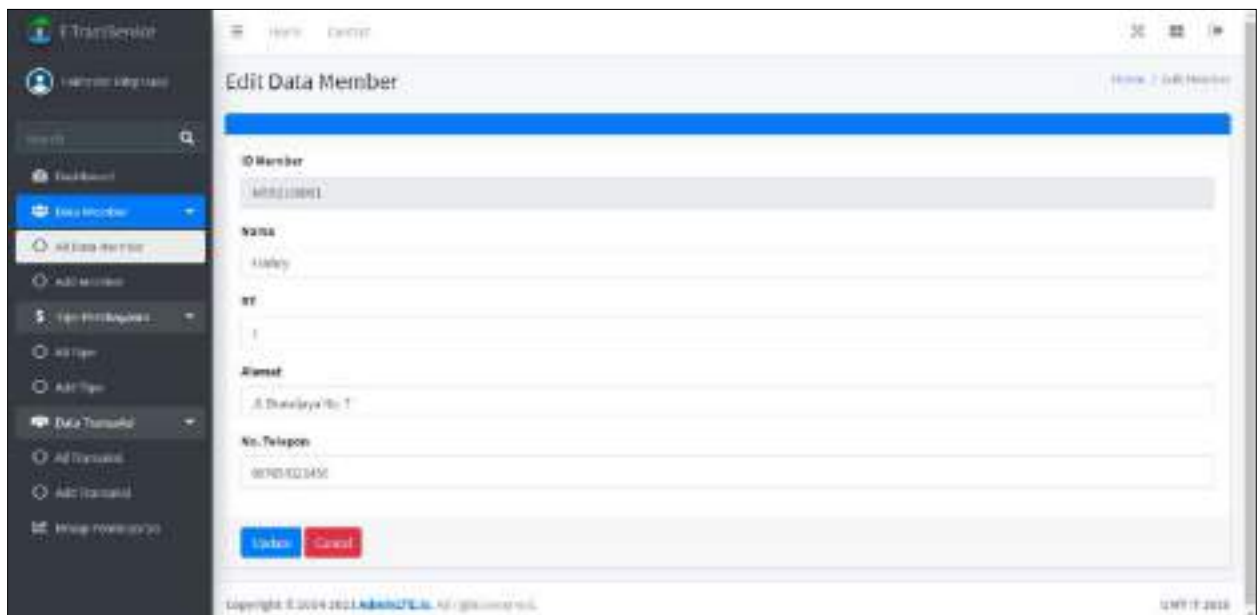
**Gambar 10** sampai dengan **Gambar 13** adalah halaman tampilan menu member. Halaman ini berfungsi untuk melakukan CRUDS (Create, Read, Update, Delete, Search) pada data member.



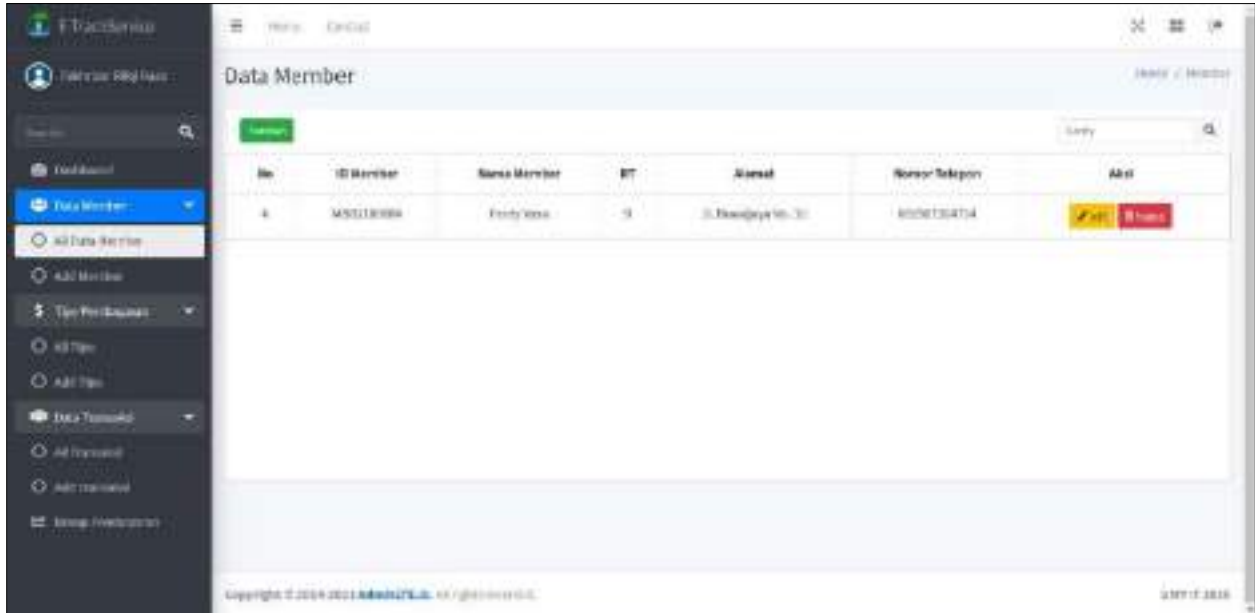
**Gambar 10.** Halaman Data Member



**Gambar 11.** Halaman Tambah Data Member



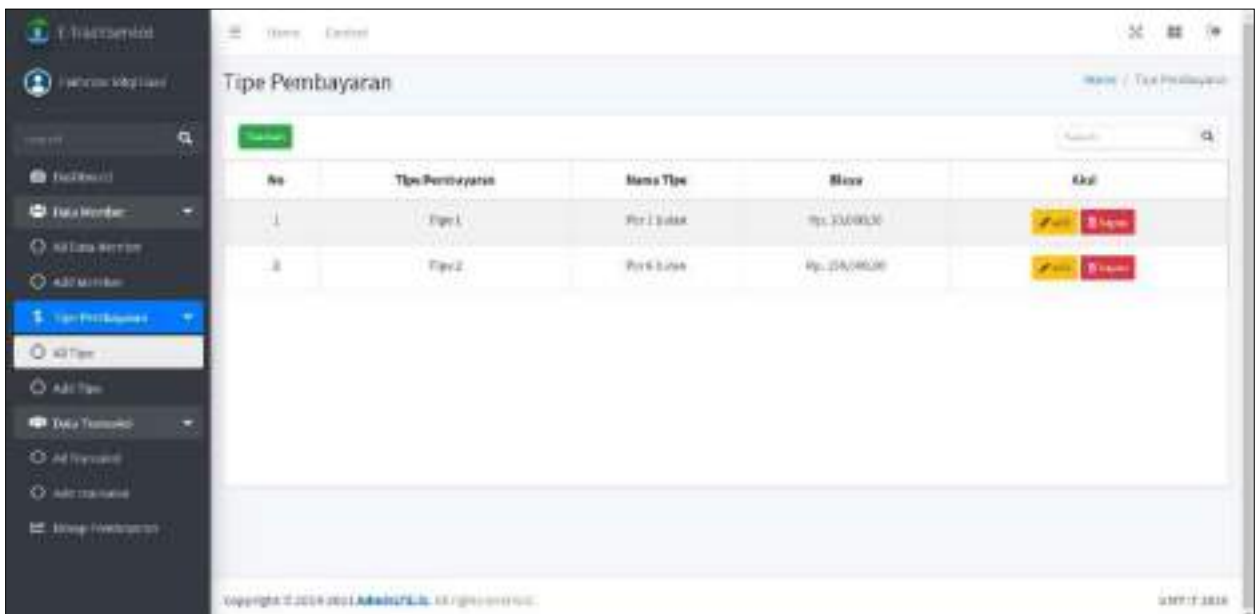
**Gambar 12.** Halaman Edit Data Member



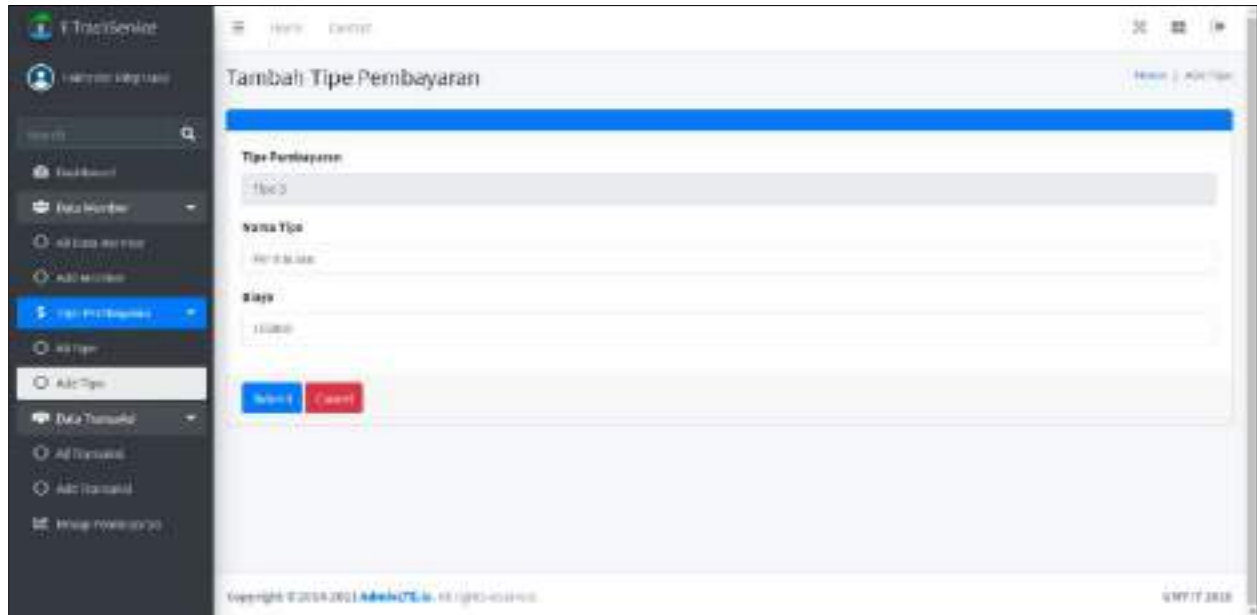
**Gambar 13.** Search Data Member

#### 4.7 Halaman Tipe Pembayaran

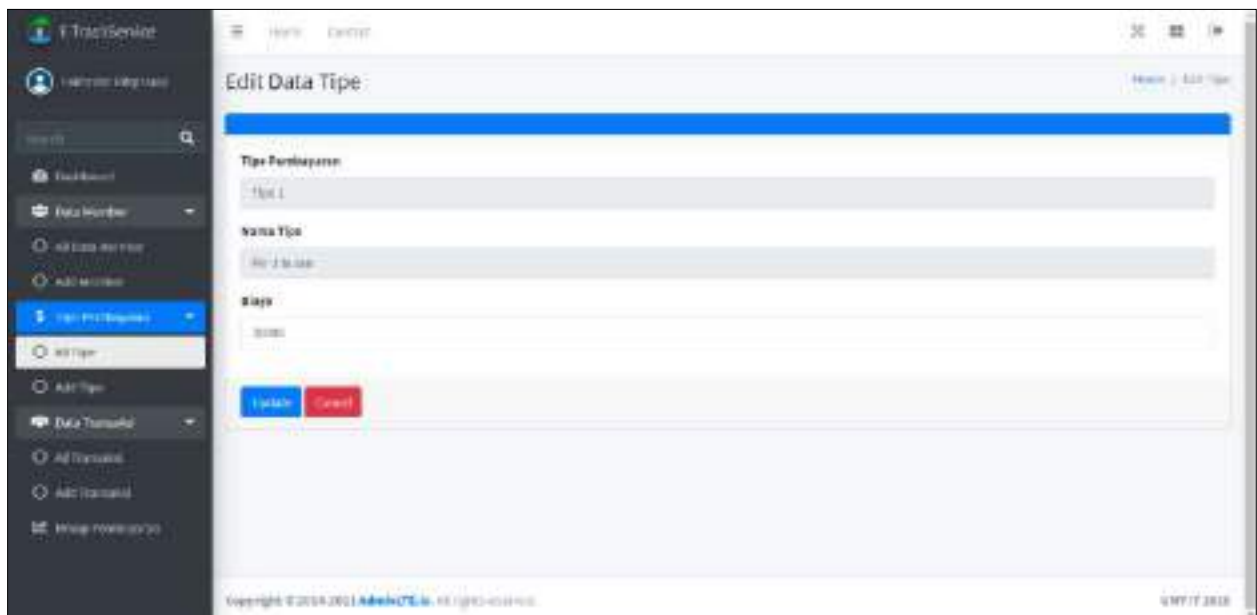
**Gambar 14** sampai dengan **Gambar 17** adalah halaman tampilan menu data tipe pembayaran. Halaman ini berfungsi untuk melakukan CRUDS (Create, Read, Update, Delete, Search) pada data tipe pembaayaran.



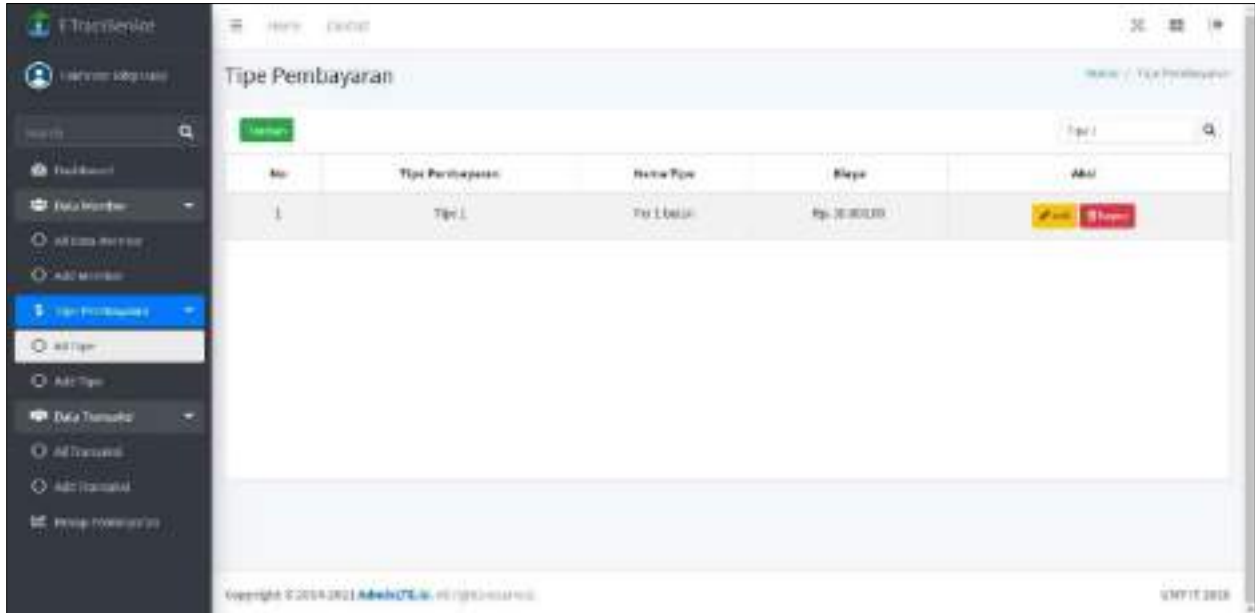
**Gambar 14.** Halaman Data Tipe Pembayaran



**Gambar 15.** Halaman Tambah Tipe Pembayaran



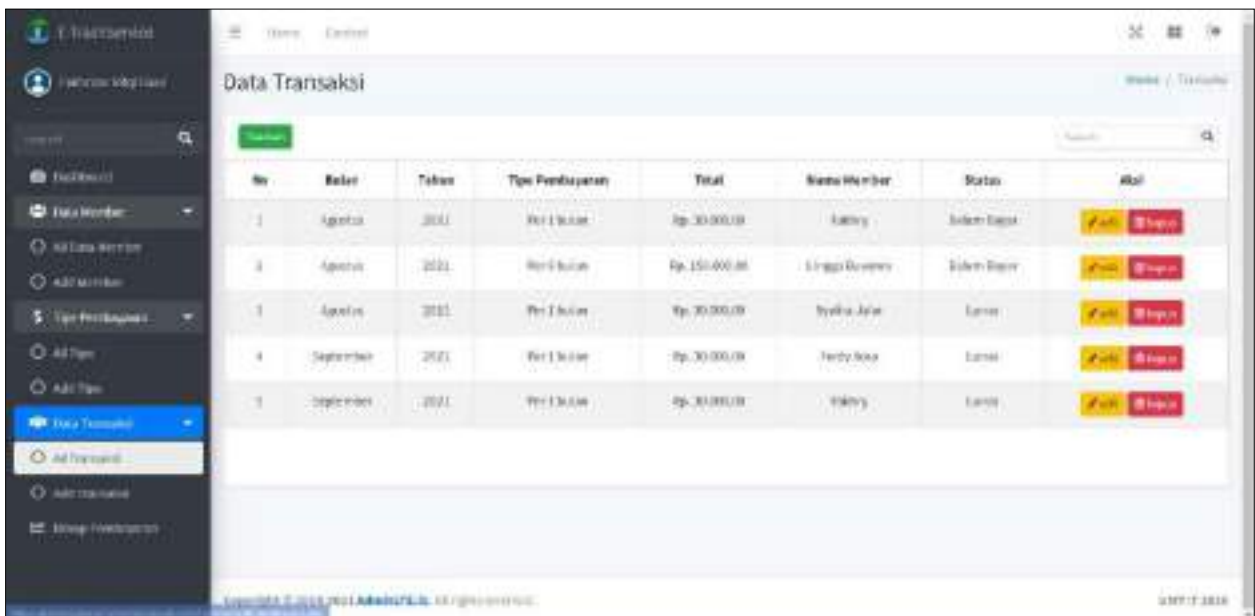
**Gambar 16.** Halaman Edit Tipe Pembayaran



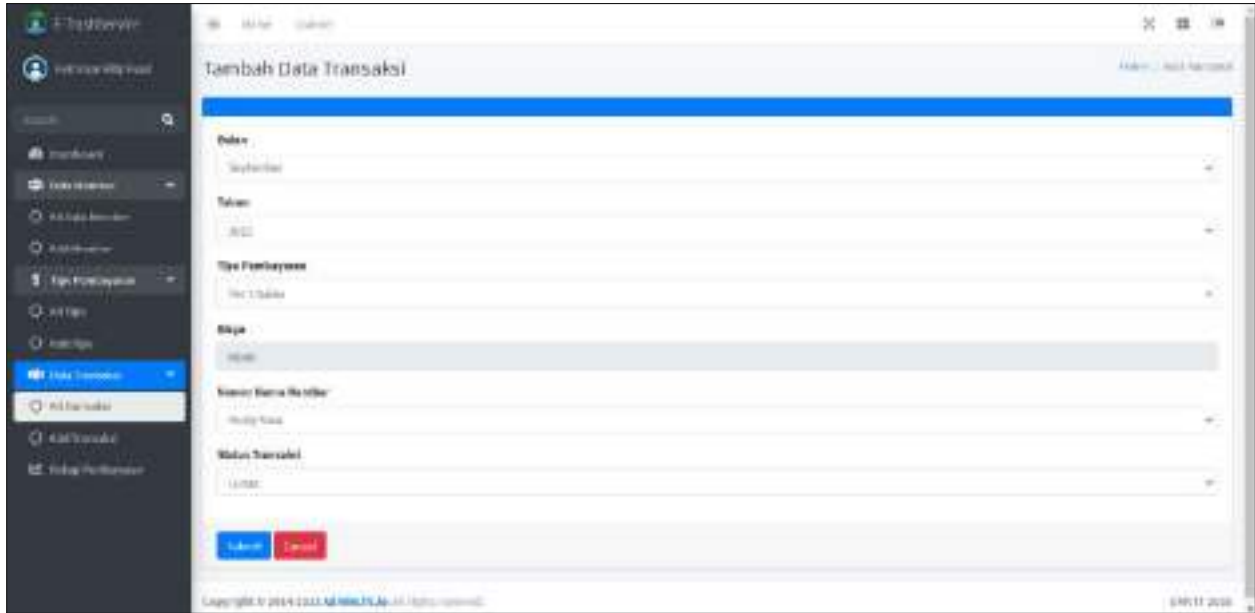
**Gambar 17.** Search Tipe Pembayaran

#### 4.8 Halaman Transaksi

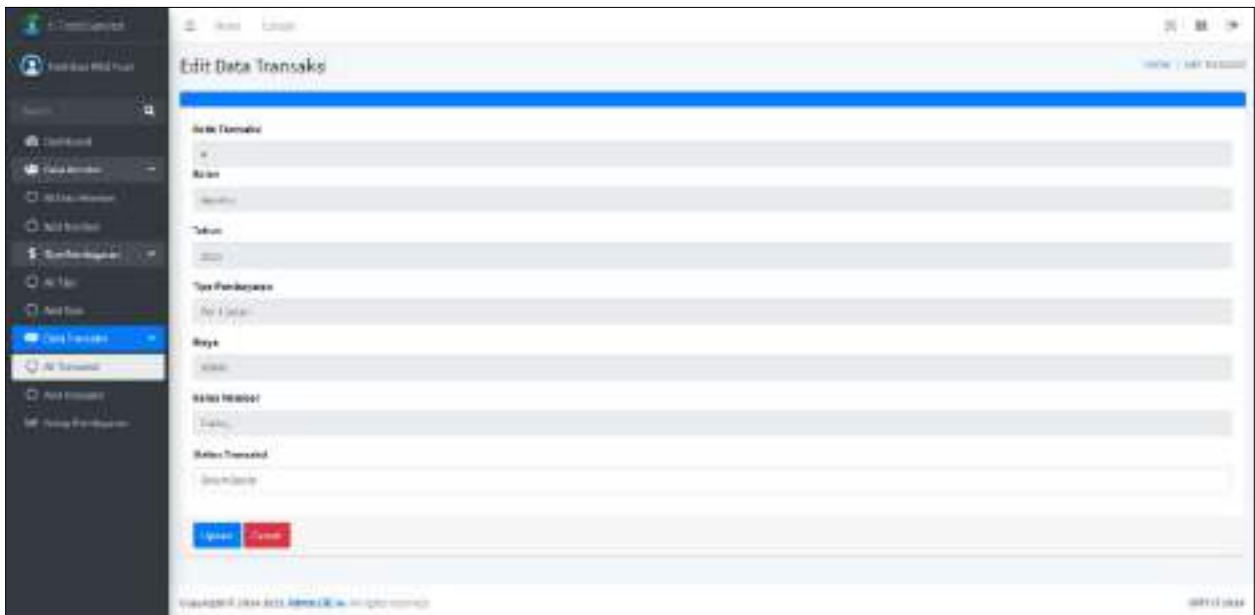
**Gambar 18** sampai dengan **Gambar 21** adalah halaman tampilan menu data transaksi. Halaman ini berfungsi untuk melakukan CRUDS (Create, Read, Update, Delete, Search) pada data transaksi.



**Gambar 18.** Halaman Data Transaksi

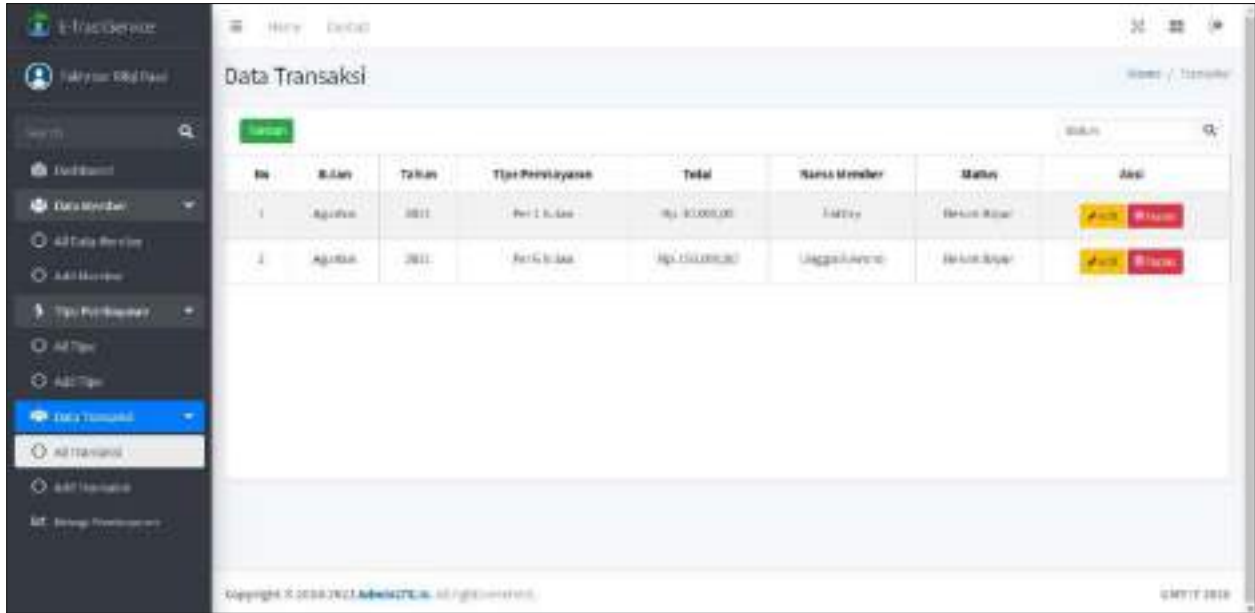


**Gambar 19.** Halaman Tambah Data Transaksi



**Gambar 20.** Halaman Edit Data Transaksi

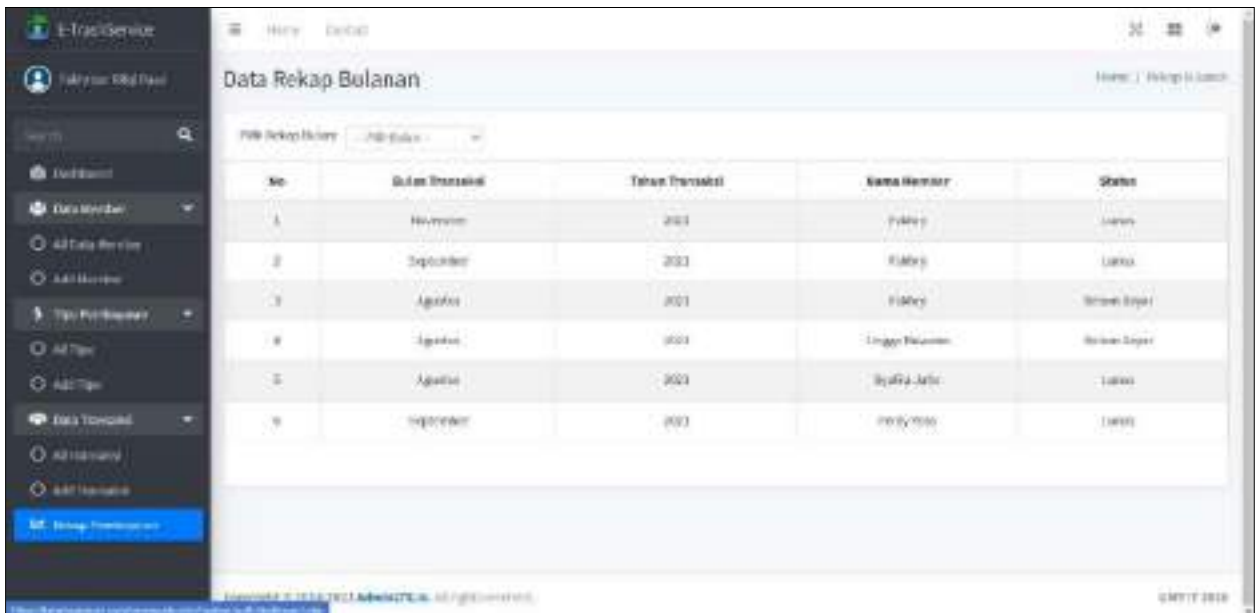




**Gambar 21.** Search Data Transaksi

#### 4.9 Halaman Rekap Pembayaran

**Gambar 22** dan **Gambar 23** adalah halaman tampilan rekap pembayaran. Halaman ini berfungsi untuk melihat rekap pembayaran bulanan serta mensortirnya.



**Gambar 22.** Halaman Rekap Pembayaran

The screenshot shows a web application interface with a sidebar on the left and a main content area. The main content area is titled "Data Rekap Bulanan" and contains a table with the following data:

No	Bulan Tersebut	Tahun Tersebut	Nama Member	Status
3	Agustus	2021	Fahri	Selesai Super
4	Agustus	2021	Angga Fauzan	Selesai Super
5	Agustus	2021	Syafiq Jafir	Lunas

The sidebar on the left contains navigation options such as "Dashboard", "Data Member", "Tipe Pembelian", and "Data Transaksi". The main content area also includes a search bar and a "Data Rekap Bulanan" dropdown menu. The footer of the page contains copyright information: "Copyright © 2024-2021 AdminLTE. All rights reserved." and the version number "v5.0.11 (2021)".

**Gambar 23.** Halaman Hasil Sortir Bulan

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil kerja praktik perancangan dan pembuatan website *E-TrashService* selama 1 bulan 1 pekan 5 hari di Creative Gama Studio dapat ditarik kesimpulan yakni sebagai berikut:

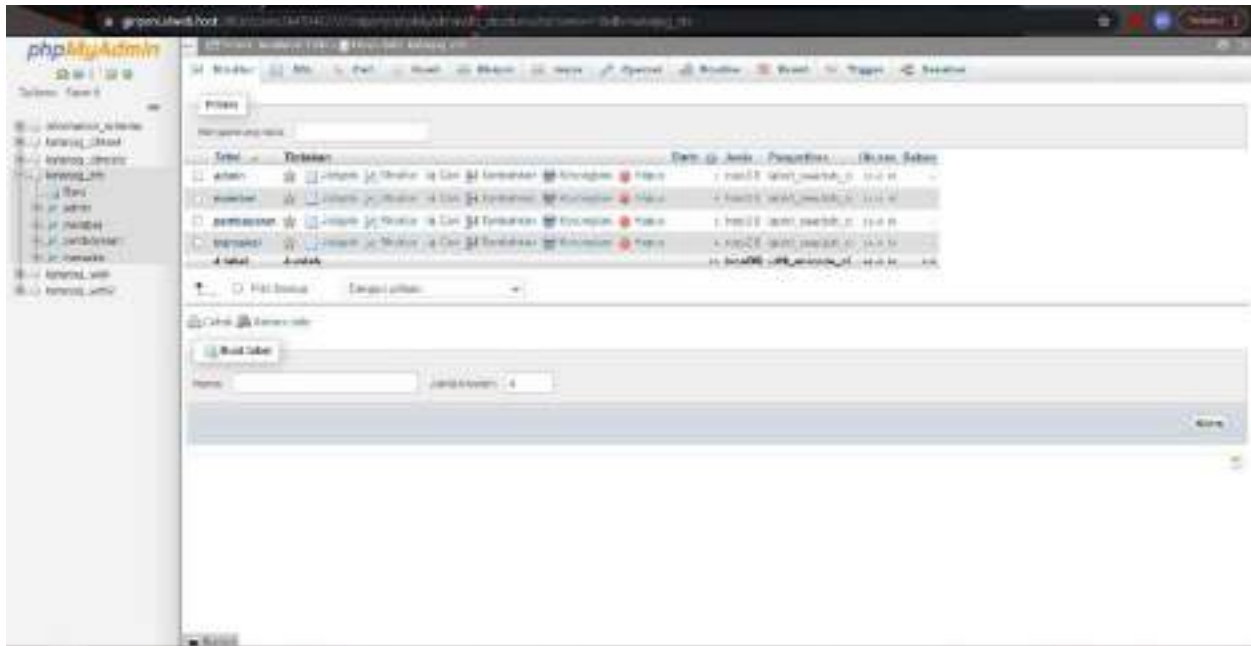
1. Komunikasi antara client dan developer sangat dibutuhkan.
2. Perlu waktu dan usaha yang lebih dalam mengerjakan proyek berskala besar dibandingkan dengan tugas matakuliah.
3. Mendapatkan pengalaman, serta meningkatkan mental dan wawasan dalam persiapan bekerja dibidang web developer.

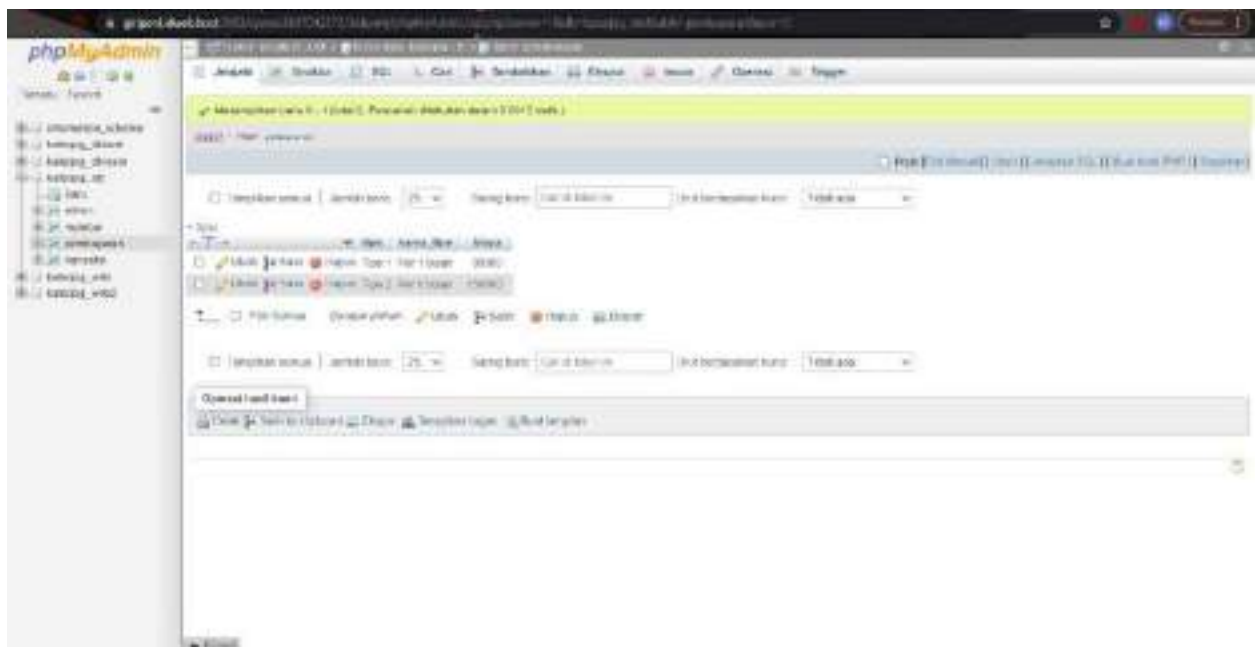
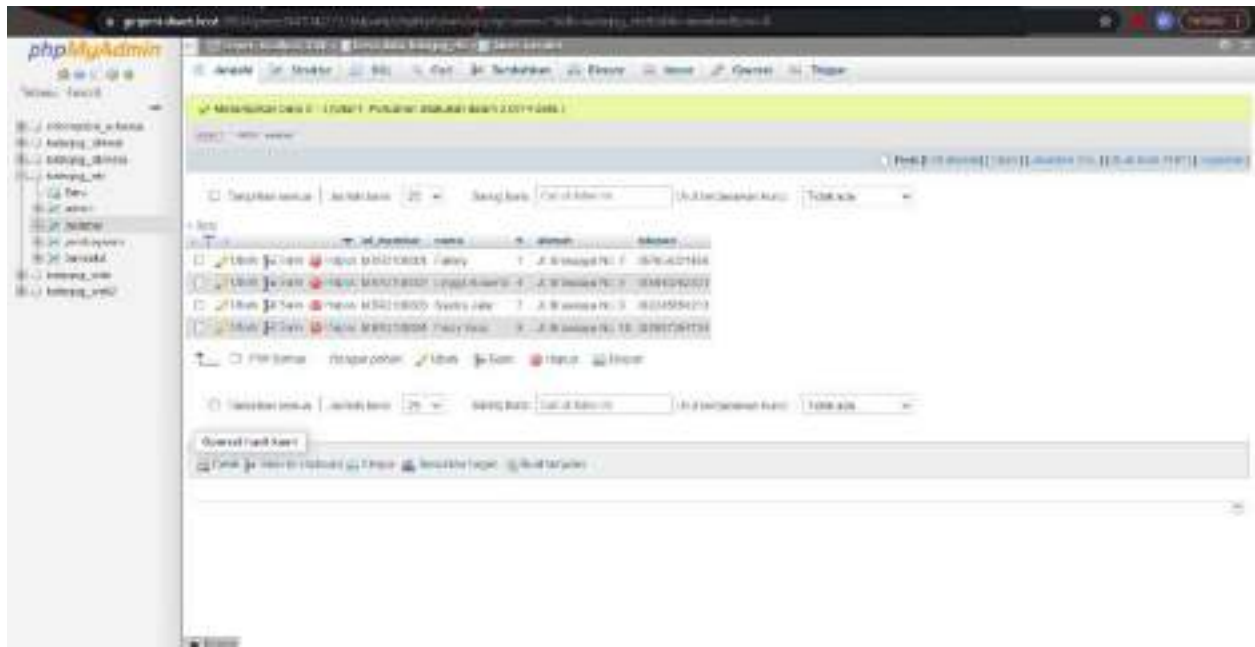
#### **5.2 Saran**

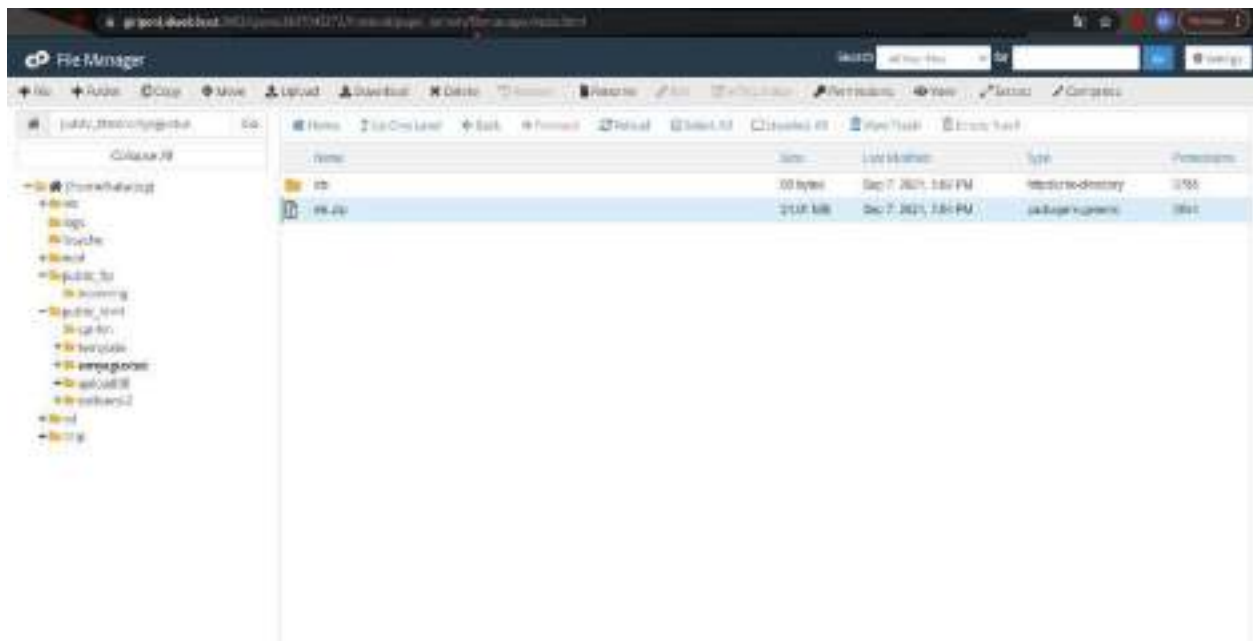
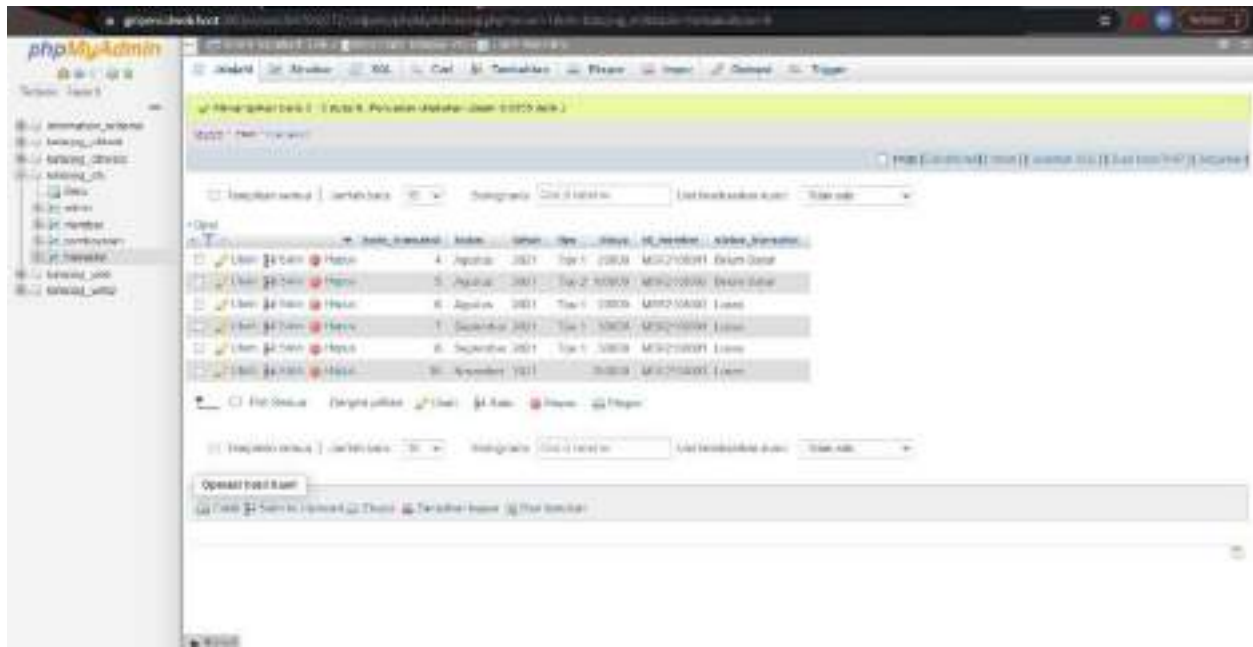
Ada beberapa pendapat kepada Creatuve Gama Studio dalam pengembangan website admin *E-TrashService* yang sudah dibuat, yakni sebagai berikut:

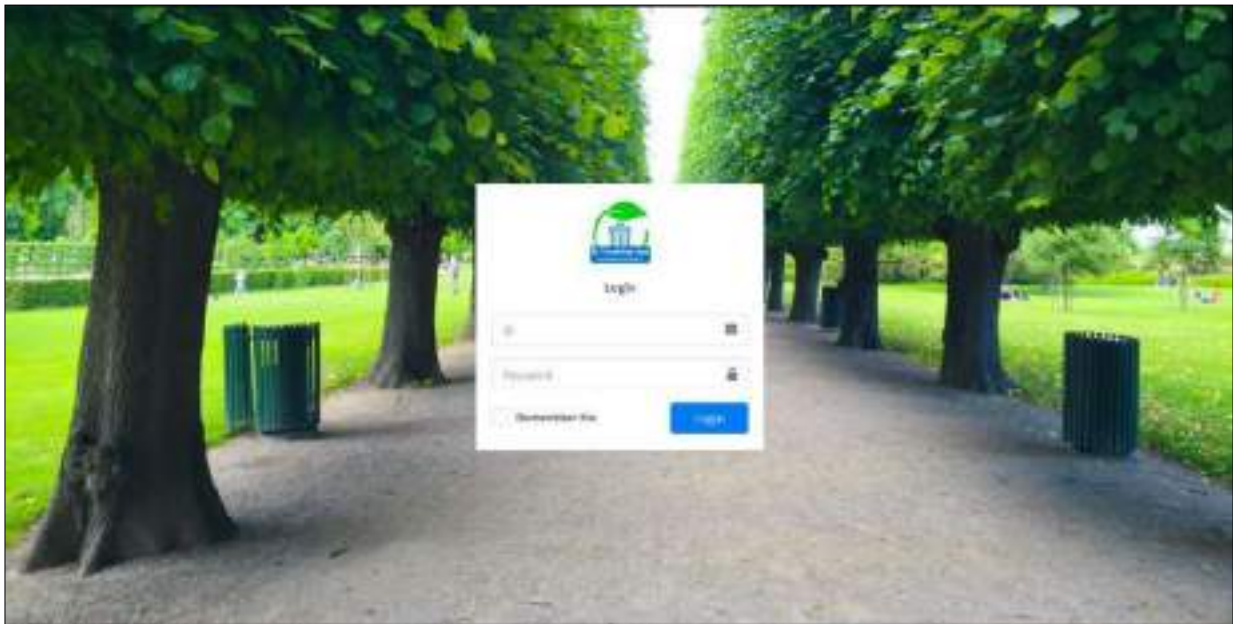
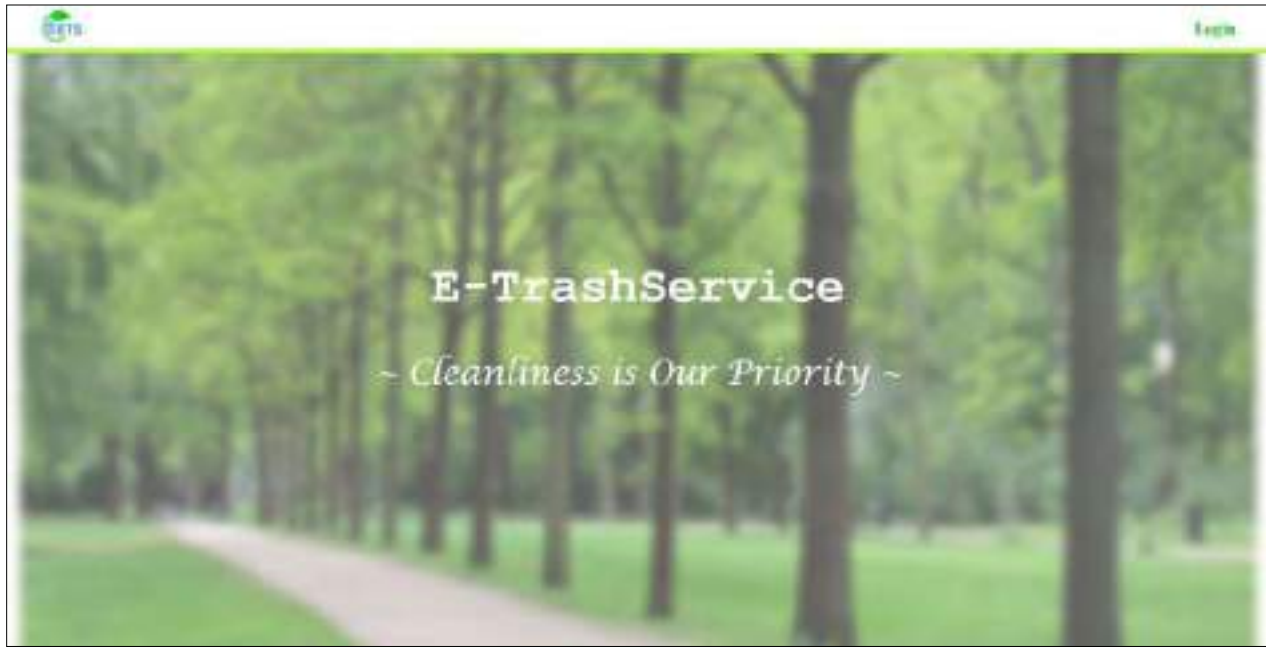
1. Dapat dilanjutkan dengan penambahan fitur untuk user.
2. Penambahan fitur agar website memiliki cakupan daerah yang lebih luas/global.

# LAMPIRAN















```
class Node {
public:
    Node(int data) {
        data = data;
        left = NULL;
        right = NULL;
    }
};

Node* insert(Node* root, int data) {
    if (root == NULL)
        return new Node(data);
    if (data < root->data)
        root->left = insert(root->left, data);
    else
        root->right = insert(root->right, data);
    return root;
}

Node* search(Node* root, int data) {
    if (root == NULL)
        return NULL;
    if (root->data == data)
        return root;
    if (data < root->data)
        return search(root->left, data);
    else
        return search(root->right, data);
}

Node* deleteNode(Node* root, int data) {
    if (root == NULL)
        return NULL;
    if (root->data == data) {
        if (root->left == NULL)
            return root->right;
        else if (root->right == NULL)
            return root->left;
        else {
            Node* temp = root->right;
            while (temp->left != NULL)
                temp = temp->left;
            root->data = temp->data;
            root->right = deleteNode(root->right, temp->data);
        }
    }
    else if (data < root->data)
        root->left = deleteNode(root->left, data);
    else
        root->right = deleteNode(root->right, data);
    return root;
}

Node* levelOrder(Node* root) {
    if (root == NULL)
        return NULL;
    queue<Node*> q;
    q.push(root);
    while (!q.empty()) {
        Node* temp = q.front();
        cout << temp->data << " ";
        if (temp->left != NULL)
            q.push(temp->left);
        if (temp->right != NULL)
            q.push(temp->right);
        q.pop();
    }
}

int main() {
    Node* root = NULL;
    int data;
    while (1) {
        cout << "Enter a number: ";
        int n;
        if (n == 1)
            root = insert(root, data);
        else if (n == 2)
            search(root, data);
        else if (n == 3)
            deleteNode(root, data);
        else if (n == 4)
            levelOrder(root);
        else
            break;
    }
}
```

```
class Node {
public:
    Node(int data) {
        data = data;
        left = NULL;
        right = NULL;
    }
};

Node* insert(Node* root, int data) {
    if (root == NULL)
        return new Node(data);
    if (data < root->data)
        root->left = insert(root->left, data);
    else
        root->right = insert(root->right, data);
    return root;
}

Node* search(Node* root, int data) {
    if (root == NULL)
        return NULL;
    if (root->data == data)
        return root;
    if (data < root->data)
        return search(root->left, data);
    else
        return search(root->right, data);
}

Node* deleteNode(Node* root, int data) {
    if (root == NULL)
        return NULL;
    if (root->data == data) {
        if (root->left == NULL)
            return root->right;
        else if (root->right == NULL)
            return root->left;
        else {
            Node* temp = root->right;
            while (temp->left != NULL)
                temp = temp->left;
            root->data = temp->data;
            root->right = deleteNode(root->right, temp->data);
        }
    }
    else if (data < root->data)
        root->left = deleteNode(root->left, data);
    else
        root->right = deleteNode(root->right, data);
    return root;
}

Node* levelOrder(Node* root) {
    if (root == NULL)
        return NULL;
    queue<Node*> q;
    q.push(root);
    while (!q.empty()) {
        Node* temp = q.front();
        cout << temp->data << " ";
        if (temp->left != NULL)
            q.push(temp->left);
        if (temp->right != NULL)
            q.push(temp->right);
        q.pop();
    }
}

int main() {
    Node* root = NULL;
    int data;
    while (1) {
        cout << "Enter a number: ";
        int n;
        if (n == 1)
            root = insert(root, data);
        else if (n == 2)
            search(root, data);
        else if (n == 3)
            deleteNode(root, data);
        else if (n == 4)
            levelOrder(root);
        else
            break;
    }
}
```





```
class Game {
public:
    Game(int width, int height) {
        width_ = width;
        height_ = height;
        player_ = new Player(50, 50);
        enemy_ = new Enemy(90, 90);
        score_ = 0;
        lives_ = 3;
        level_ = 1;
        powerups_ = new Powerups();
        background_ = new Background();
        music_ = new Music();
        sound_ = new Sound();
        ui_ = new UI();
        state_ = STATE_GAME;
    }

    void update() {
        player_->update();
        enemy_->update();
        powerups_->update();
        background_->update();
        music_->update();
        sound_->update();
        ui_->update();
        state_ = STATE_GAME;
    }

    void render() {
        player_->render();
        enemy_->render();
        powerups_->render();
        background_->render();
        music_->render();
        sound_->render();
        ui_->render();
    }

    void start() {
        player_ = new Player(50, 50);
        enemy_ = new Enemy(90, 90);
        score_ = 0;
        lives_ = 3;
        level_ = 1;
        powerups_ = new Powerups();
        background_ = new Background();
        music_ = new Music();
        sound_ = new Sound();
        ui_ = new UI();
        state_ = STATE_GAME;
    }

    void stop() {
        delete player_;
        delete enemy_;
        delete powerups_;
        delete background_;
        delete music_;
        delete sound_;
        delete ui_;
    }
};
```

```
class Game {
public:
    Game(int width, int height) {
        width_ = width;
        height_ = height;
        player_ = new Player(50, 50);
        enemy_ = new Enemy(90, 90);
        score_ = 0;
        lives_ = 3;
        level_ = 1;
        powerups_ = new Powerups();
        background_ = new Background();
        music_ = new Music();
        sound_ = new Sound();
        ui_ = new UI();
        state_ = STATE_GAME;
    }

    void update() {
        player_->update();
        enemy_->update();
        powerups_->update();
        background_->update();
        music_->update();
        sound_->update();
        ui_->update();
        state_ = STATE_GAME;
    }

    void render() {
        player_->render();
        enemy_->render();
        powerups_->render();
        background_->render();
        music_->render();
        sound_->render();
        ui_->render();
    }

    void start() {
        player_ = new Player(50, 50);
        enemy_ = new Enemy(90, 90);
        score_ = 0;
        lives_ = 3;
        level_ = 1;
        powerups_ = new Powerups();
        background_ = new Background();
        music_ = new Music();
        sound_ = new Sound();
        ui_ = new UI();
        state_ = STATE_GAME;
    }

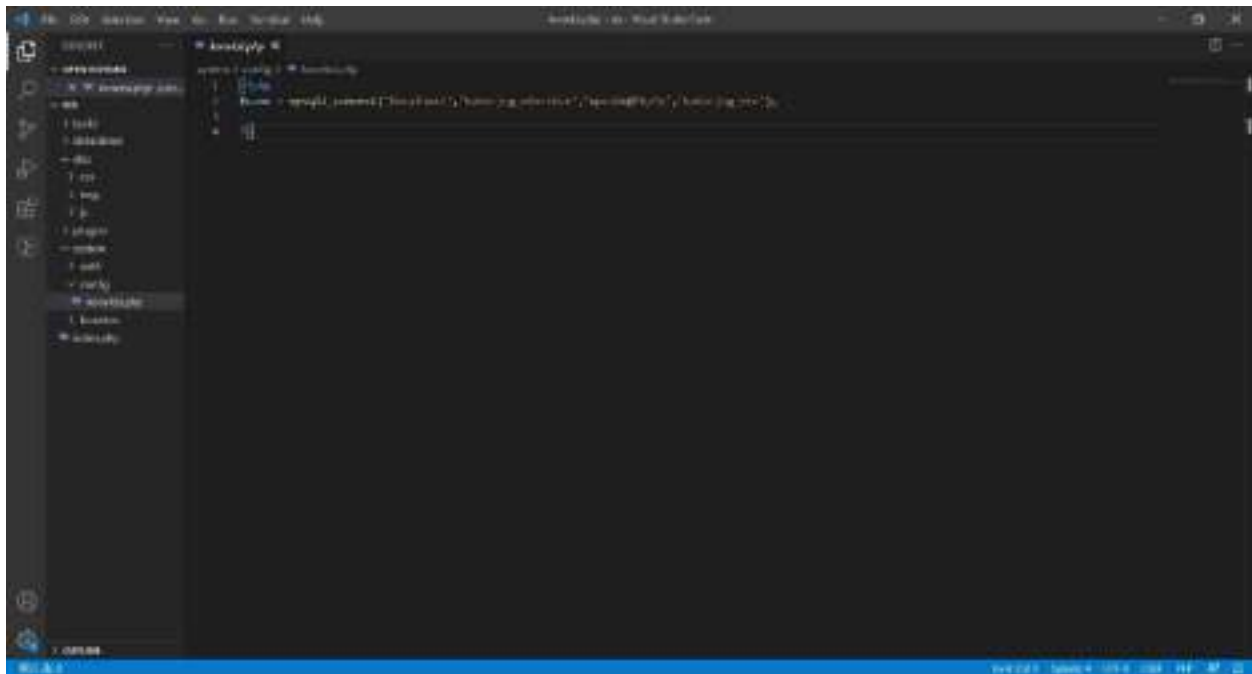
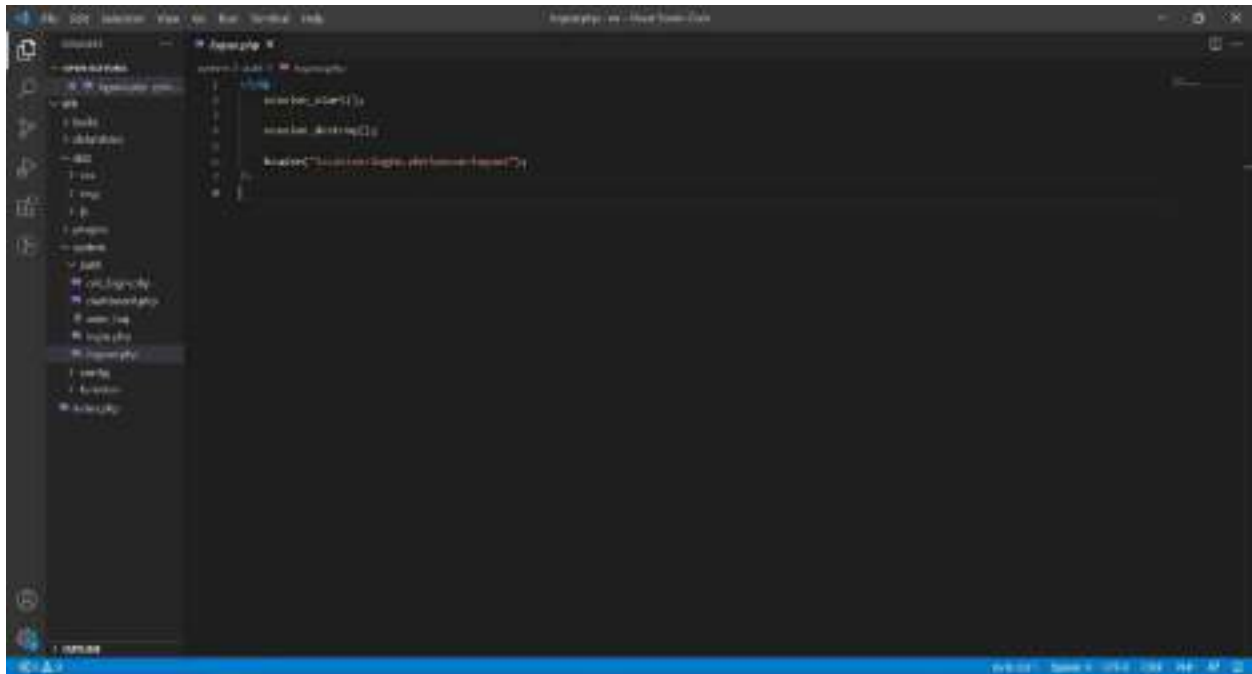
    void stop() {
        delete player_;
        delete enemy_;
        delete powerups_;
        delete background_;
        delete music_;
        delete sound_;
        delete ui_;
    }
};
```











```
class Animal:
    def __init__(self, name, species, weight, height, age):
        self.name = name
        self.species = species
        self.weight = weight
        self.height = height
        self.age = age

class Dog(Animal):
    def __init__(self, name, breed, weight, height, age):
        super().__init__(name, species, weight, height, age)
        self.breed = breed

class Cat(Animal):
    def __init__(self, name, breed, weight, height, age):
        super().__init__(name, species, weight, height, age)
        self.breed = breed

class Snake(Animal):
    def __init__(self, name, species, weight, height, age):
        super().__init__(name, species, weight, height, age)

class Bird(Animal):
    def __init__(self, name, species, weight, height, age):
        super().__init__(name, species, weight, height, age)
```

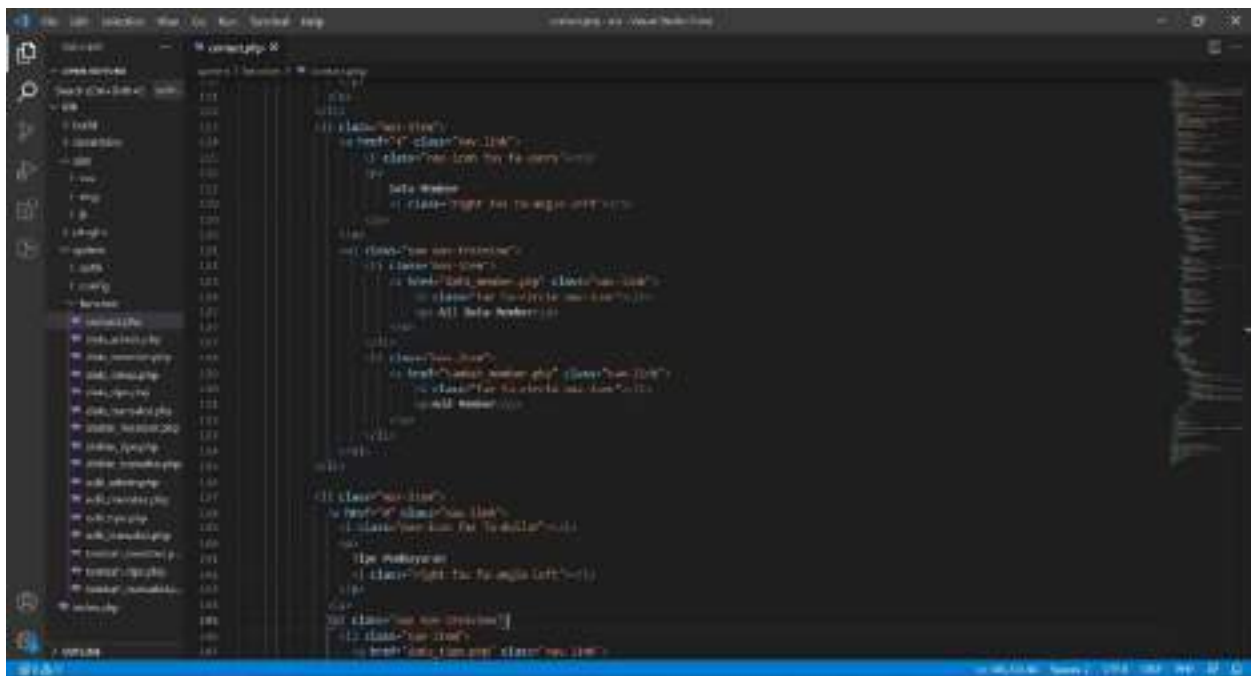
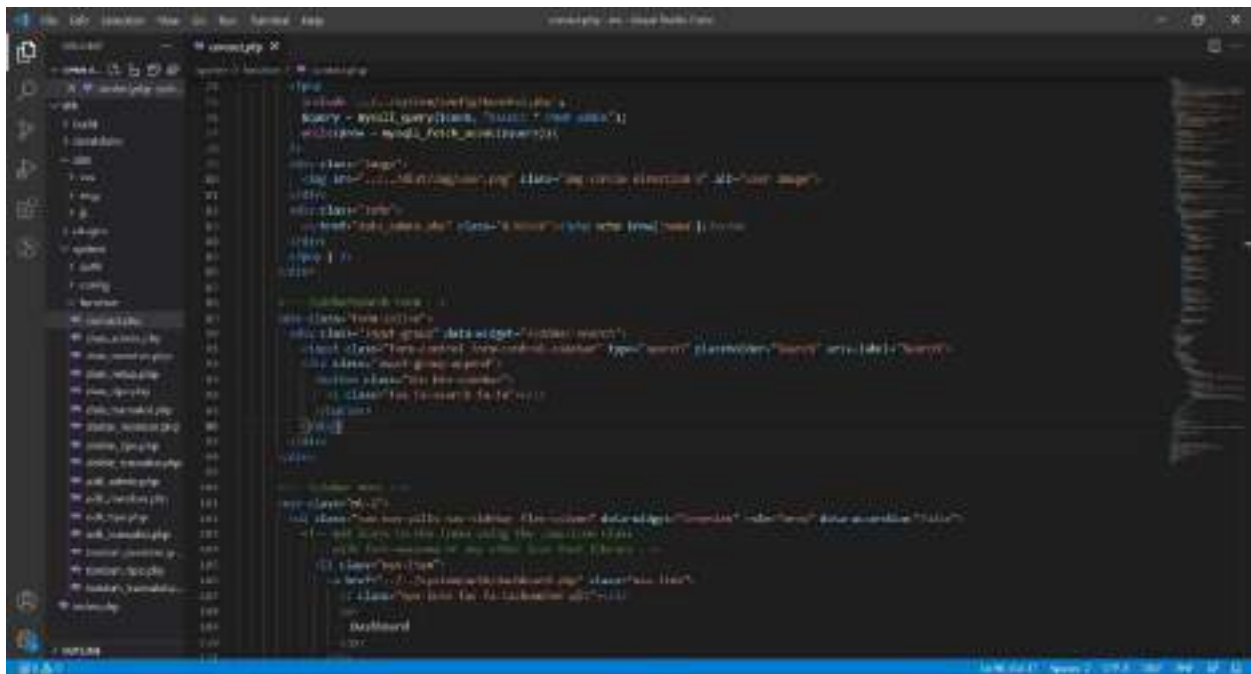
```
class Animal:
    def __init__(self, name, species, weight, height, age):
        self.name = name
        self.species = species
        self.weight = weight
        self.height = height
        self.age = age

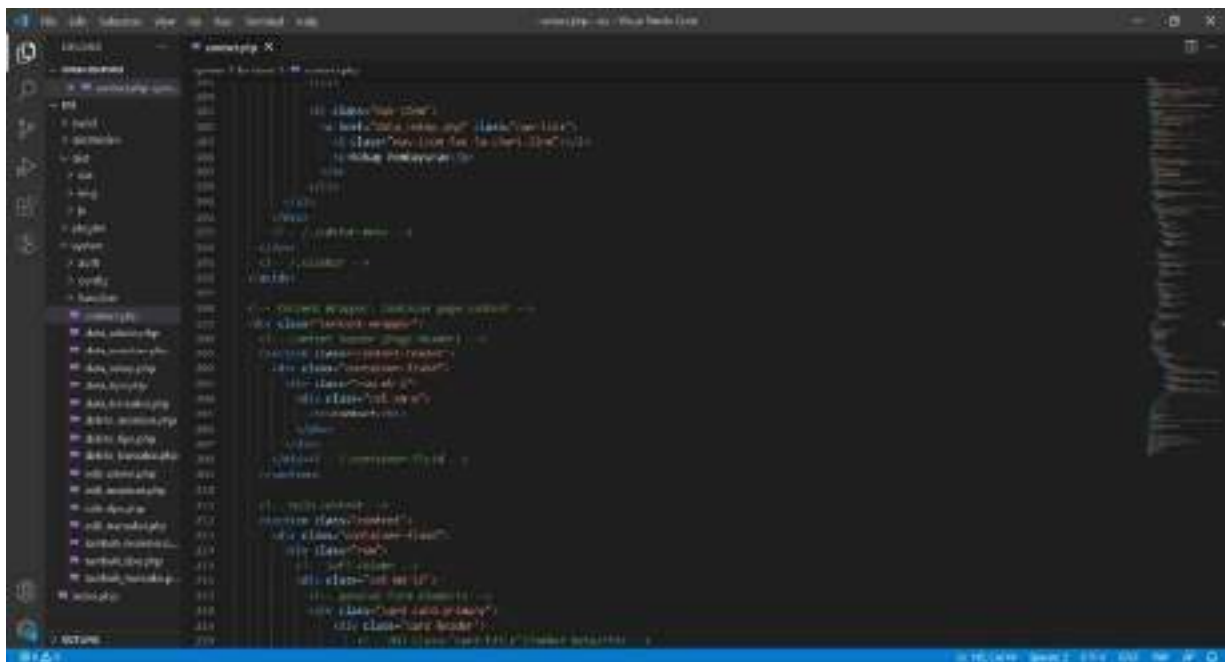
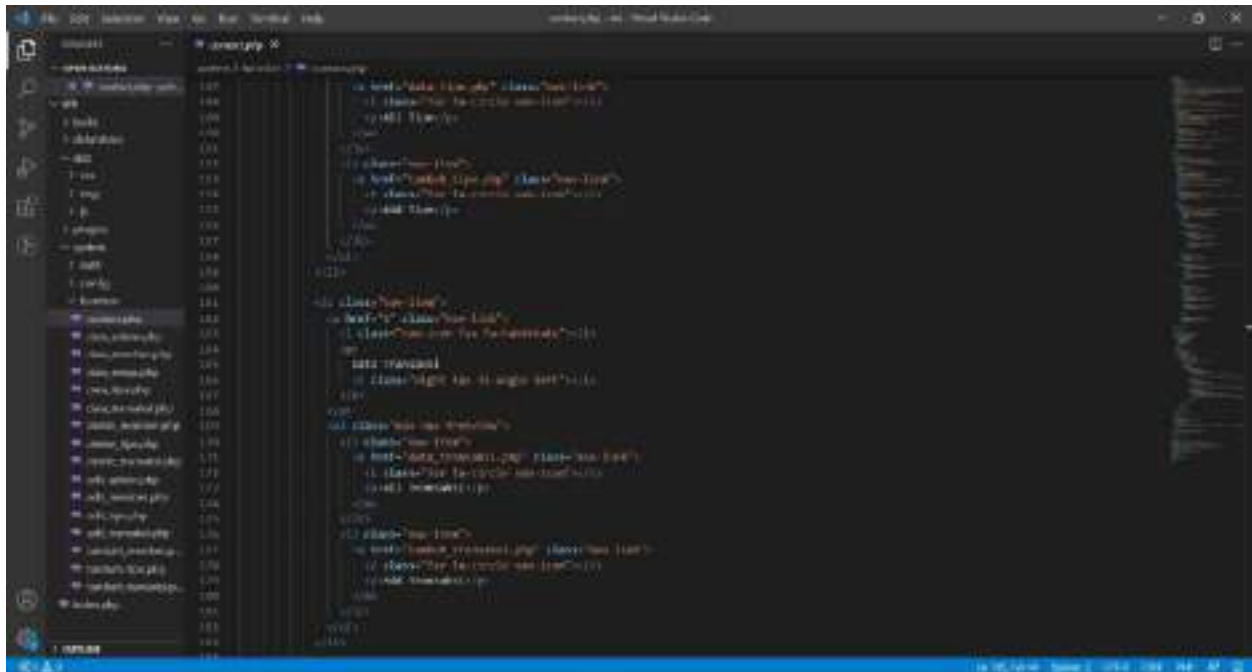
class Dog(Animal):
    def __init__(self, name, breed, weight, height, age):
        super().__init__(name, species, weight, height, age)
        self.breed = breed

class Cat(Animal):
    def __init__(self, name, breed, weight, height, age):
        super().__init__(name, species, weight, height, age)
        self.breed = breed

class Snake(Animal):
    def __init__(self, name, species, weight, height, age):
        super().__init__(name, species, weight, height, age)

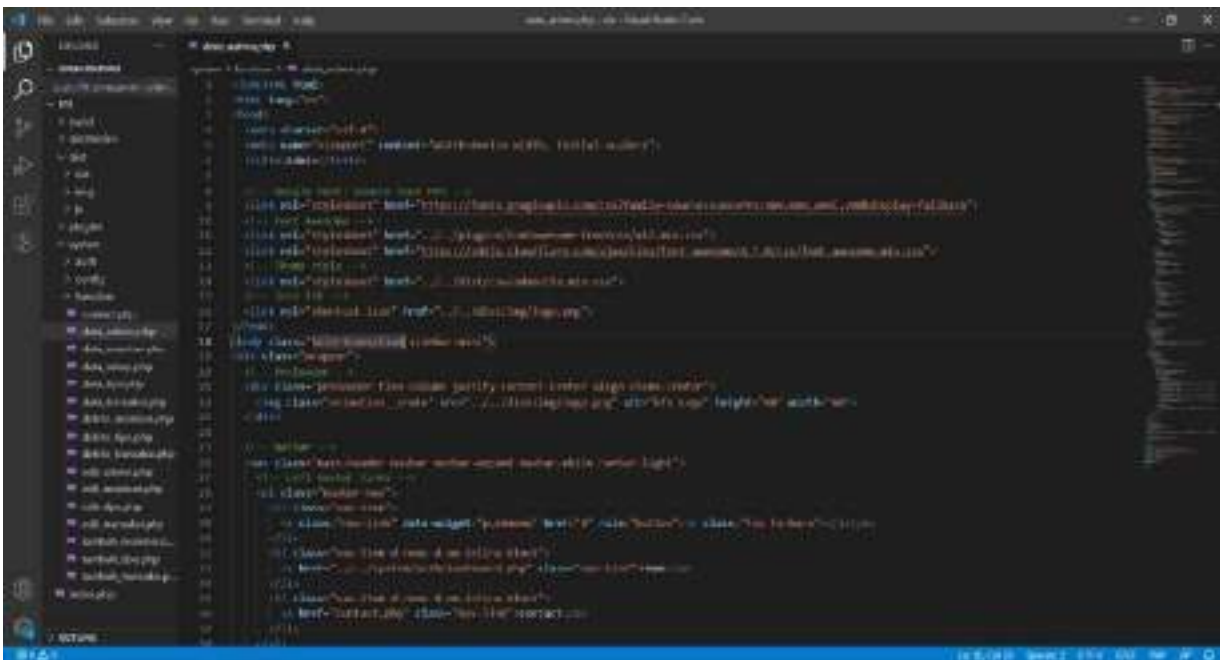
class Bird(Animal):
    def __init__(self, name, species, weight, height, age):
        super().__init__(name, species, weight, height, age)
```

















```

class Person {
    public $name;
    public $age;
    public $gender;

    public function __construct($name, $age, $gender) {
        $this->name = $name;
        $this->age = $age;
        $this->gender = $gender;
    }

    public function displayInfo() {
        echo "Name: " . $this->name . ", Age: " . $this->age . ", Gender: " . $this->gender . "\n";
    }
}

class Student extends Person {
    public $school;

    public function __construct($name, $age, $gender, $school) {
        parent::__construct($name, $age, $gender);
        $this->school = $school;
    }

    public function displayInfo() {
        echo "Name: " . $this->name . ", Age: " . $this->age . ", Gender: " . $this->gender . ", School: " . $this->school . "\n";
    }
}

class Teacher extends Person {
    public $subject;

    public function __construct($name, $age, $gender, $subject) {
        parent::__construct($name, $age, $gender);
        $this->subject = $subject;
    }

    public function displayInfo() {
        echo "Name: " . $this->name . ", Age: " . $this->age . ", Gender: " . $this->gender . ", Subject: " . $this->subject . "\n";
    }
}

class Employee extends Person {
    public $company;

    public function __construct($name, $age, $gender, $company) {
        parent::__construct($name, $age, $gender);
        $this->company = $company;
    }

    public function displayInfo() {
        echo "Name: " . $this->name . ", Age: " . $this->age . ", Gender: " . $this->gender . ", Company: " . $this->company . "\n";
    }
}

```

```

class Person {
    public $name;
    public $age;
    public $gender;

    public function __construct($name, $age, $gender) {
        $this->name = $name;
        $this->age = $age;
        $this->gender = $gender;
    }

    public function displayInfo() {
        echo "Name: " . $this->name . ", Age: " . $this->age . ", Gender: " . $this->gender . "\n";
    }
}

class Student extends Person {
    public $school;

    public function __construct($name, $age, $gender, $school) {
        parent::__construct($name, $age, $gender);
        $this->school = $school;
    }

    public function displayInfo() {
        echo "Name: " . $this->name . ", Age: " . $this->age . ", Gender: " . $this->gender . ", School: " . $this->school . "\n";
    }
}

class Teacher extends Person {
    public $subject;

    public function __construct($name, $age, $gender, $subject) {
        parent::__construct($name, $age, $gender);
        $this->subject = $subject;
    }

    public function displayInfo() {
        echo "Name: " . $this->name . ", Age: " . $this->age . ", Gender: " . $this->gender . ", Subject: " . $this->subject . "\n";
    }
}

class Employee extends Person {
    public $company;

    public function __construct($name, $age, $gender, $company) {
        parent::__construct($name, $age, $gender);
        $this->company = $company;
    }

    public function displayInfo() {
        echo "Name: " . $this->name . ", Age: " . $this->age . ", Gender: " . $this->gender . ", Company: " . $this->company . "\n";
    }
}

```



Admin Dashboard

Home | Dashboard

### Data Member

Search

No	ID Member	Nama Member	BT	Alamat	Nomer Telepon	Aksi
1	HR120001	Fahry	L	Basa	8054171136	<a href="#">Edit</a> <a href="#">Hapus</a>
2	HR120002	Grega	L	Togoyarto	8054174233	<a href="#">Edit</a> <a href="#">Hapus</a>
3	HR120003	Safar	P	Majin Laka	8054161413	<a href="#">Edit</a> <a href="#">Hapus</a>
4	HR120004	Fendi	L	Kuldesan Bani	8058730434	<a href="#">Edit</a> <a href="#">Hapus</a>

Copyright © 2018-2021 AdminLTE Inc. All rights reserved. v3.0.4

```

1  @RequestMapping("/api")
2  public class ApiController {
3      private static final Logger logger = LoggerFactory.getLogger(ApiController.class);
4      private final UserService userService;
5      private final ProductService productService;
6
7      public ApiController(UserService userService, ProductService productService) {
8          this.userService = userService;
9          this.productService = productService;
10     }
11
12     @GetMapping("/users")
13     public List<User> getUsers() {
14         logger.info("Retrieving all users");
15         return userService.getAllUsers();
16     }
17
18     @GetMapping("/products")
19     public List<Product> getProducts() {
20         logger.info("Retrieving all products");
21         return productService.getAllProducts();
22     }
23
24     @PostMapping("/users")
25     public User createUser(@RequestBody User user) {
26         logger.info("Creating new user: {}", user);
27         return userService.createUser(user);
28     }
29
30     @PutMapping("/users/{id}")
31     public User updateUser(@PathVariable Long id, @RequestBody User user) {
32         logger.info("Updating user with ID: {}", id);
33         return userService.updateUser(id, user);
34     }
35
36     @DeleteMapping("/users/{id}")
37     public boolean deleteUser(@PathVariable Long id) {
38         logger.info("Deleting user with ID: {}", id);
39         return userService.deleteUser(id);
40     }
41
42     @GetMapping("/products/{id}")
43     public Product getProduct(@PathVariable Long id) {
44         logger.info("Retrieving product with ID: {}", id);
45         return productService.getProduct(id);
46     }
47
48     @PostMapping("/products")
49     public Product createProduct(@RequestBody Product product) {
50         logger.info("Creating new product: {}", product);
51         return productService.createProduct(product);
52     }
53
54     @PutMapping("/products/{id}")
55     public Product updateProduct(@PathVariable Long id, @RequestBody Product product) {
56         logger.info("Updating product with ID: {}", id);
57         return productService.updateProduct(id, product);
58     }
59
60     @DeleteMapping("/products/{id}")
61     public boolean deleteProduct(@PathVariable Long id) {
62         logger.info("Deleting product with ID: {}", id);
63         return productService.deleteProduct(id);
64     }
65
66     @ExceptionHandler({IllegalArgumentException.class, RuntimeException.class})
67     public ResponseEntity<ErrorResponse> handleException(Exception e) {
68         logger.error("Unhandled exception: {}", e);
69         ErrorResponse errorResponse = new ErrorResponse(e.getMessage());
70         return ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).body(errorResponse);
71     }
72 }

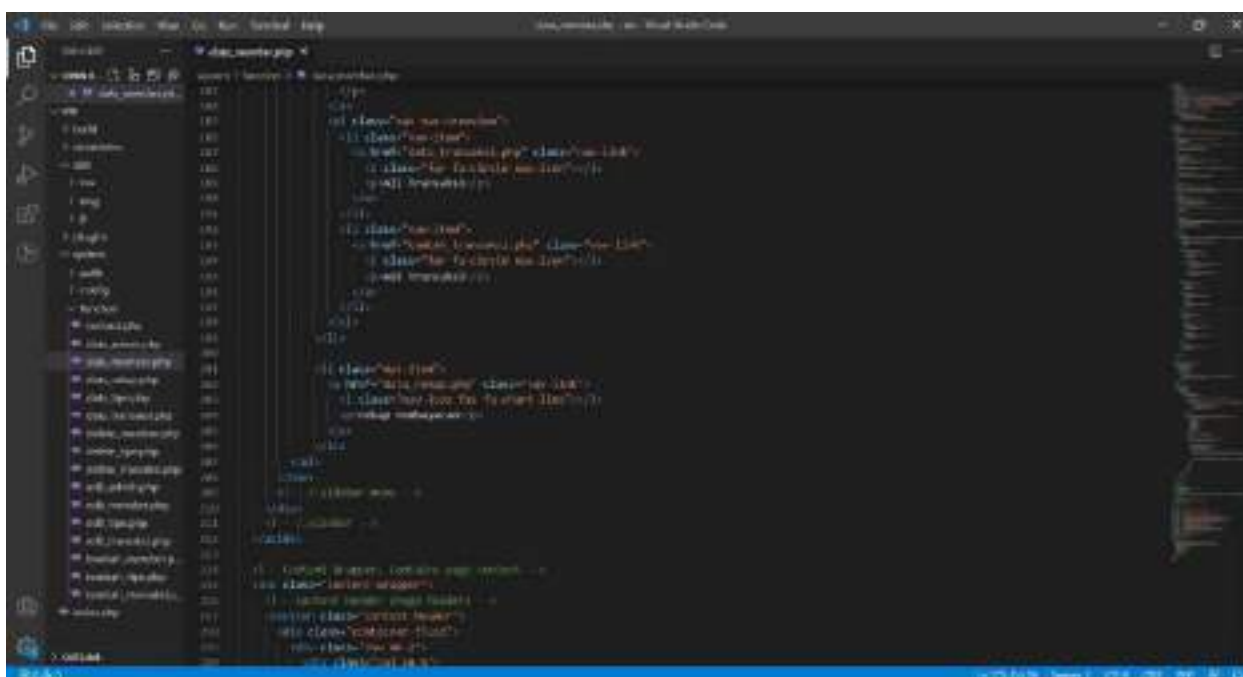
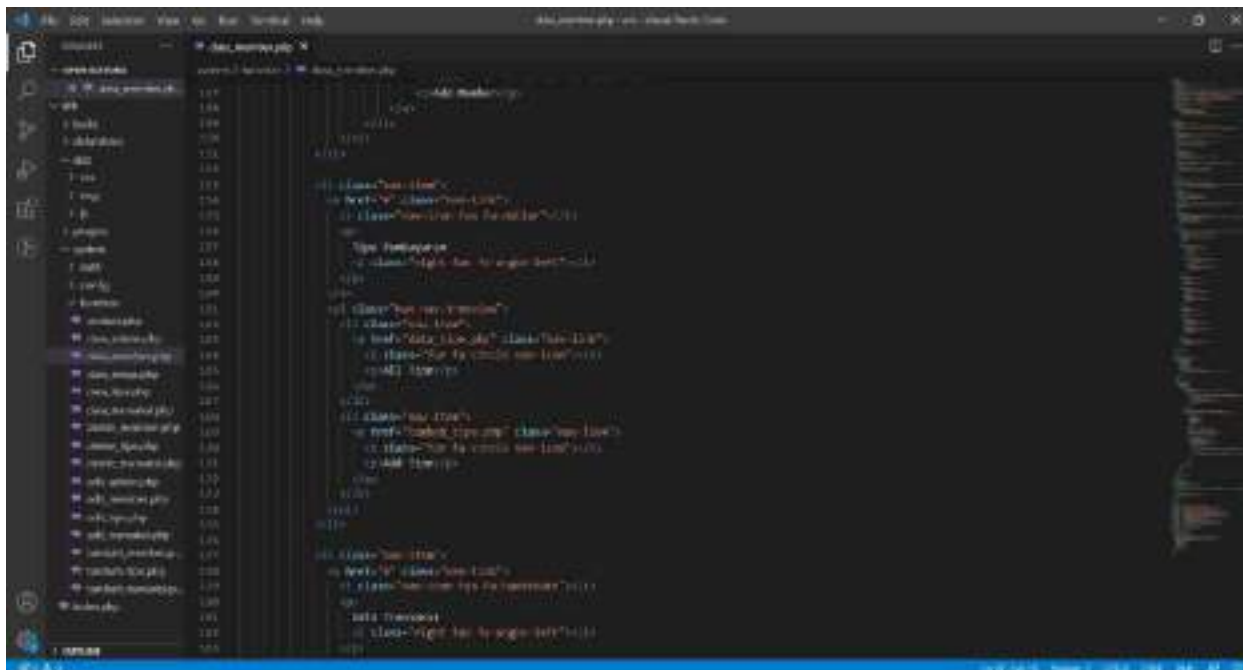
```

```

1  @RestController
2  public class ProductController {
3      private static final Logger logger = LoggerFactory.getLogger(ProductController.class);
4      private final ProductService productService;
5
6      public ProductController(ProductService productService) {
7          this.productService = productService;
8      }
9
10     @GetMapping("/products")
11     public List<Product> getAllProducts() {
12         logger.info("Retrieving all products");
13         return productService.getAllProducts();
14     }
15
16     @GetMapping("/products/{id}")
17     public Product getProduct(@PathVariable Long id) {
18         logger.info("Retrieving product with ID: {}", id);
19         return productService.getProduct(id);
20     }
21
22     @PostMapping("/products")
23     public Product createProduct(@RequestBody Product product) {
24         logger.info("Creating new product: {}", product);
25         return productService.createProduct(product);
26     }
27
28     @PutMapping("/products/{id}")
29     public Product updateProduct(@PathVariable Long id, @RequestBody Product product) {
30         logger.info("Updating product with ID: {}", id);
31         return productService.updateProduct(id, product);
32     }
33
34     @DeleteMapping("/products/{id}")
35     public boolean deleteProduct(@PathVariable Long id) {
36         logger.info("Deleting product with ID: {}", id);
37         return productService.deleteProduct(id);
38     }
39
40     @ExceptionHandler({IllegalArgumentException.class, RuntimeException.class})
41     public ResponseEntity<ErrorResponse> handleException(Exception e) {
42         logger.error("Unhandled exception: {}", e);
43         ErrorResponse errorResponse = new ErrorResponse(e.getMessage());
44         return ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).body(errorResponse);
45     }
46 }

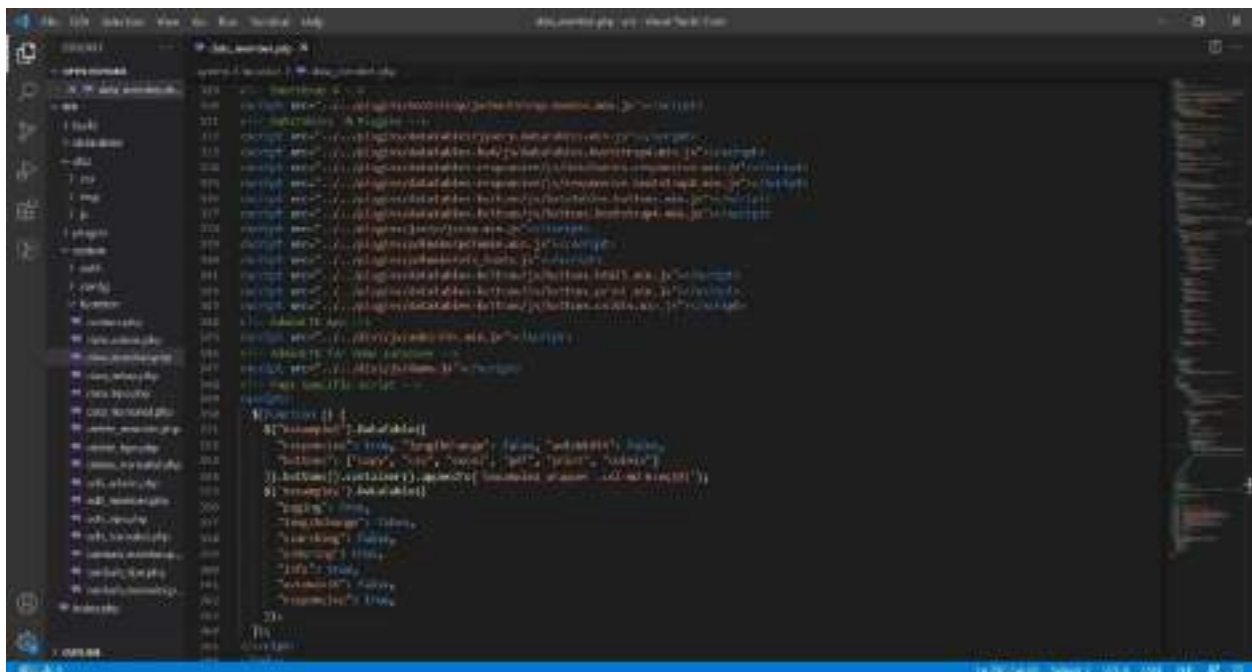
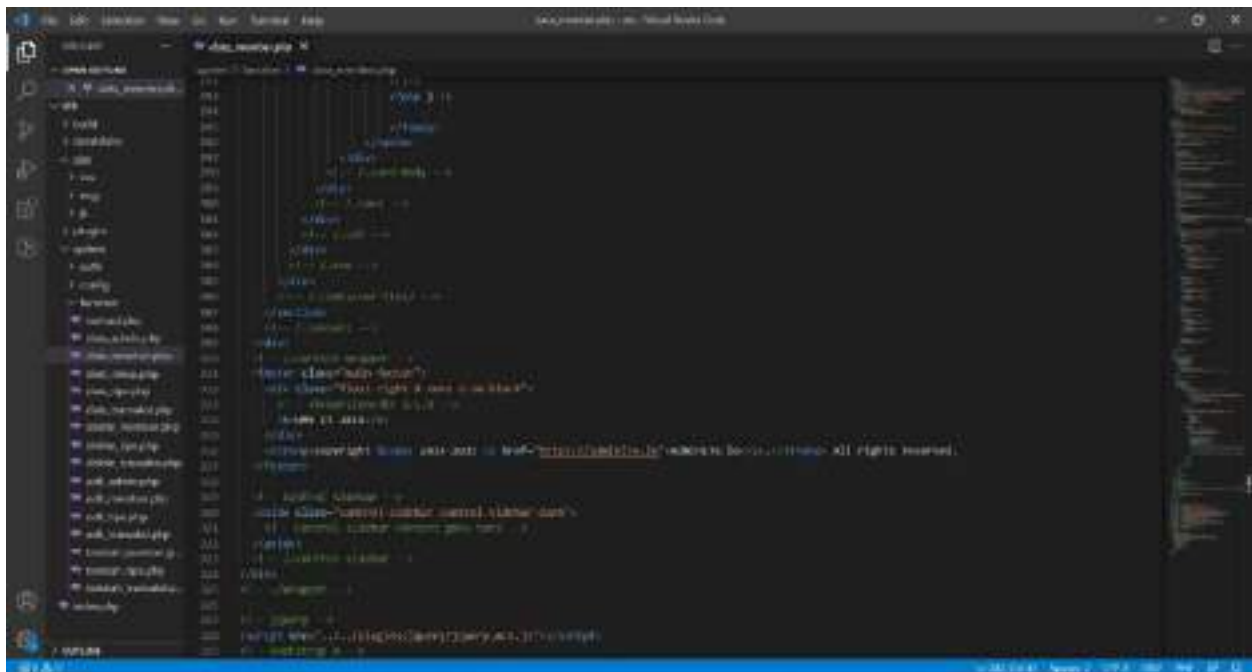
```

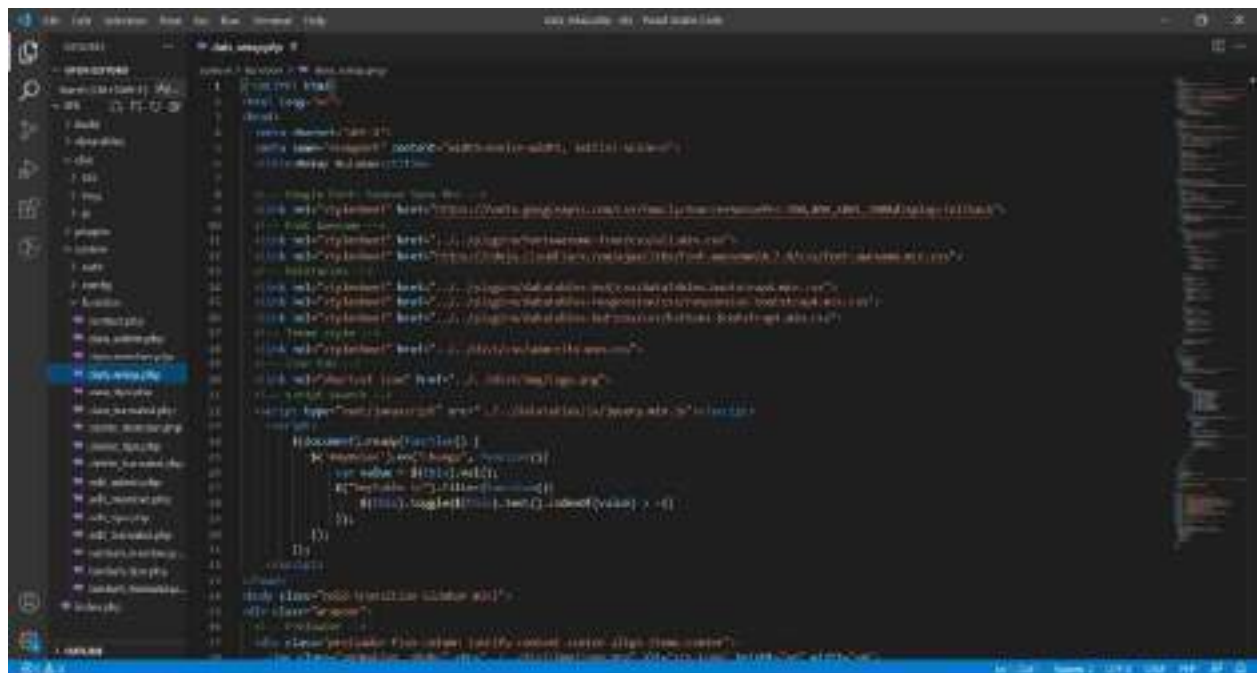
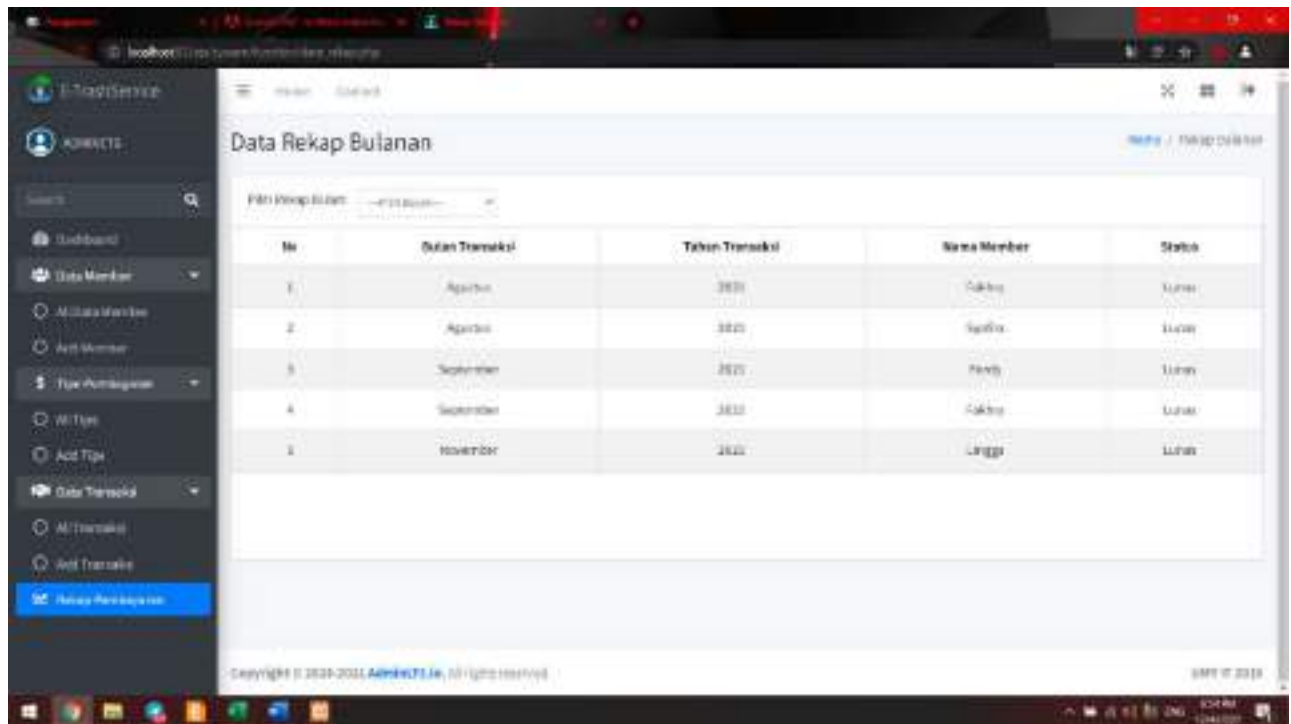


















```

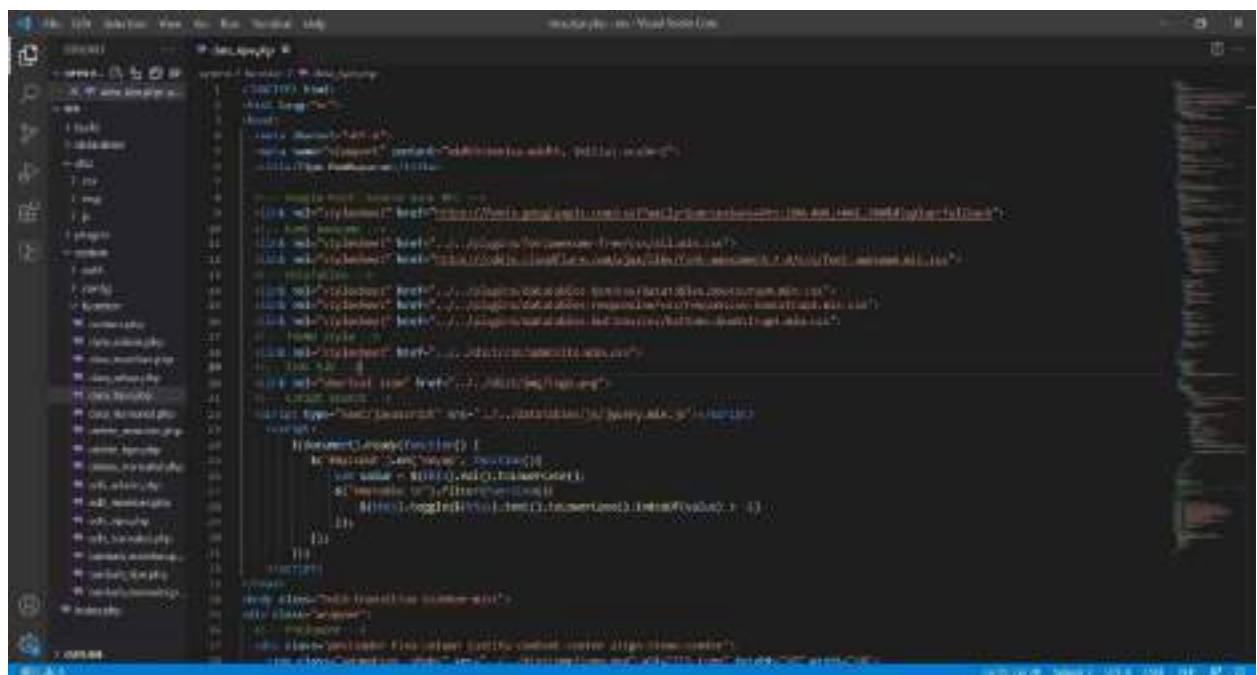
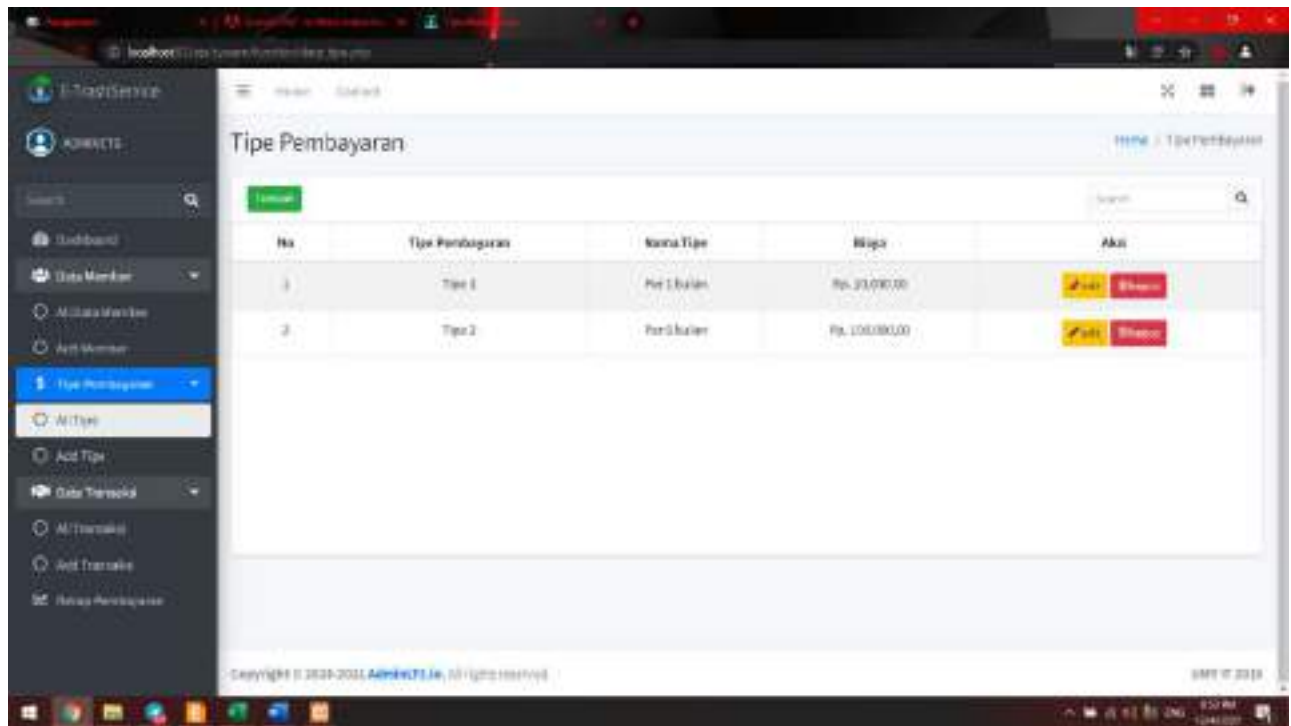
118 <div class="table">
119 <table border="1">
120 <thead>
121 <tr>
122 <th>ID</th>
123 <th>Name</th>
124 <th>Age</th>
125 <th>Gender</th>
126 <th>Address</th>
127 </tr>
128 </thead>
129 <tbody>
130 <tr>
131 <td>1</td>
132 <td>John</td>
133 <td>25</td>
134 <td>Male</td>
135 <td>123 Main St</td>
136 </tr>
137 <tr>
138 <td>2</td>
139 <td>Jane</td>
140 <td>30</td>
141 <td>Female</td>
142 <td>456 Elm St</td>
143 </tr>
144 <tr>
145 <td>3</td>
146 <td>Mike</td>
147 <td>35</td>
148 <td>Male</td>
149 <td>789 Oak St</td>
150 </tr>
151 <tr>
152 <td>4</td>
153 <td>Sarah</td>
154 <td>28</td>
155 <td>Female</td>
156 <td>101 Pine St</td>
157 </tr>
158 <tr>
159 <td>5</td>
160 <td>David</td>
161 <td>40</td>
162 <td>Male</td>
163 <td>202 Cedar St</td>
164 </tr>
165 </tbody>
166 </table>
167 </div>
168 </div>
169 </div>
170 </div>
171 </div>
172 </div>
173 </div>
174 </div>
175 </div>
176 </div>
177 </div>
178 </div>
179 </div>
180 </div>
181 </div>
182 </div>
183 </div>
184 </div>
185 </div>
186 </div>
187 </div>
188 </div>
189 </div>
190 </div>
191 </div>
192 </div>
193 </div>
194 </div>
195 </div>
196 </div>
197 </div>
198 </div>
199 </div>
200 </div>

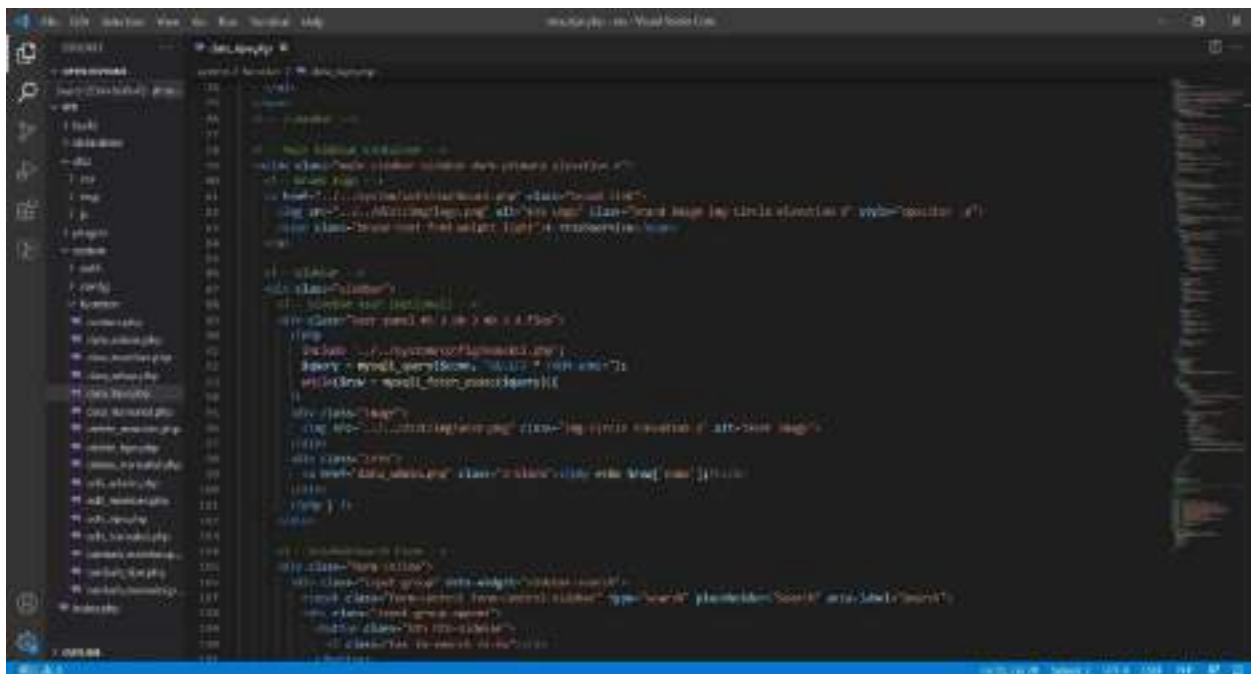
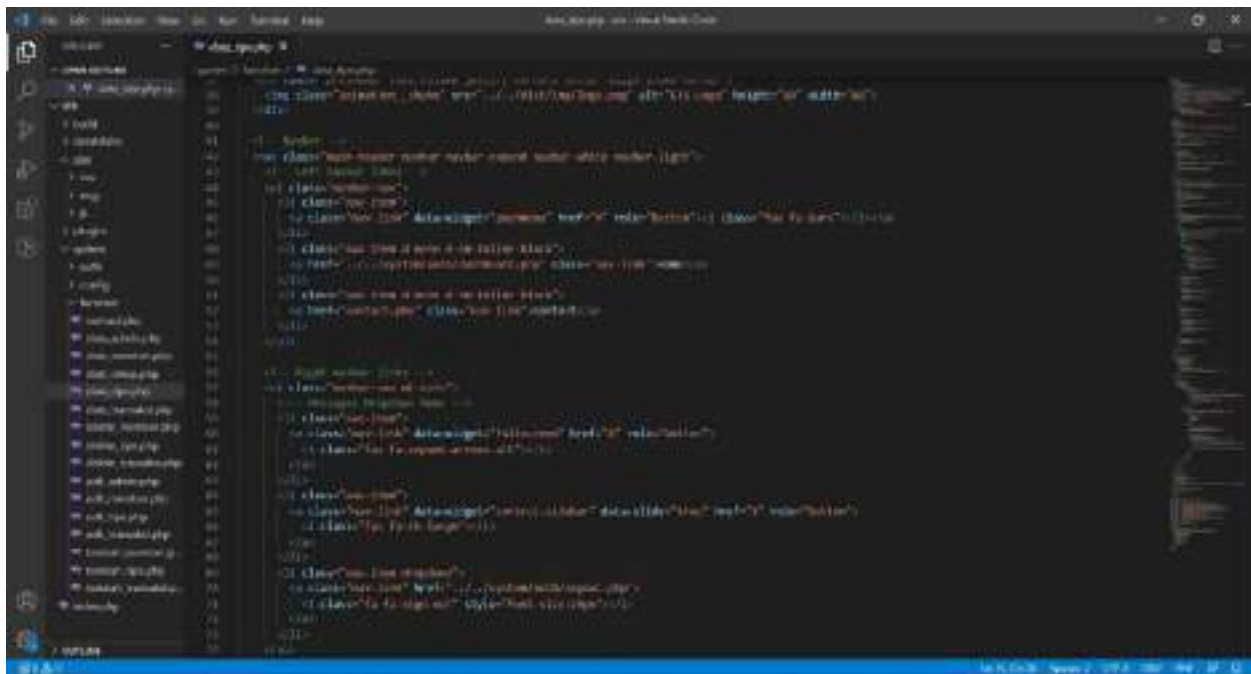
```

```

201 <div class="table">
202 <table border="1">
203 <thead>
204 <tr>
205 <th>ID</th>
206 <th>Name</th>
207 <th>Age</th>
208 <th>Gender</th>
209 <th>Address</th>
210 </tr>
211 </thead>
212 <tbody>
213 <tr>
214 <td>1</td>
215 <td>John</td>
216 <td>25</td>
217 <td>Male</td>
218 <td>123 Main St</td>
219 </tr>
220 <tr>
221 <td>2</td>
222 <td>Jane</td>
223 <td>30</td>
224 <td>Female</td>
225 <td>456 Elm St</td>
226 </tr>
227 <tr>
228 <td>3</td>
229 <td>Mike</td>
230 <td>35</td>
231 <td>Male</td>
232 <td>789 Oak St</td>
233 </tr>
234 <tr>
235 <td>4</td>
236 <td>Sarah</td>
237 <td>28</td>
238 <td>Female</td>
239 <td>101 Pine St</td>
240 </tr>
241 <tr>
242 <td>5</td>
243 <td>David</td>
244 <td>40</td>
245 <td>Male</td>
246 <td>202 Cedar St</td>
247 </tr>
248 </tbody>
249 </table>
250 </div>
251 </div>
252 </div>
253 </div>
254 </div>
255 </div>
256 </div>
257 </div>
258 </div>
259 </div>
260 </div>
261 </div>
262 </div>
263 </div>
264 </div>
265 </div>
266 </div>
267 </div>
268 </div>
269 </div>
270 </div>
271 </div>
272 </div>
273 </div>
274 </div>
275 </div>
276 </div>
277 </div>
278 </div>
279 </div>
280 </div>
281 </div>
282 </div>
283 </div>
284 </div>
285 </div>
286 </div>
287 </div>
288 </div>
289 </div>
290 </div>
291 </div>
292 </div>
293 </div>
294 </div>
295 </div>
296 </div>
297 </div>
298 </div>
299 </div>
300 </div>

```

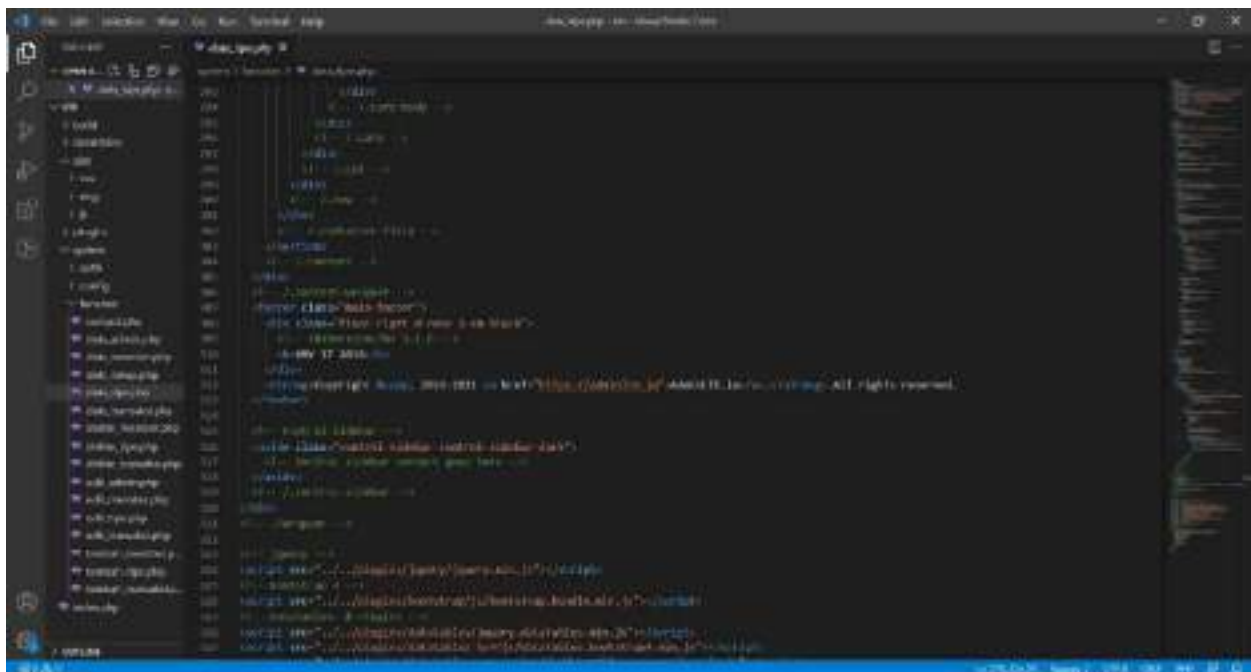
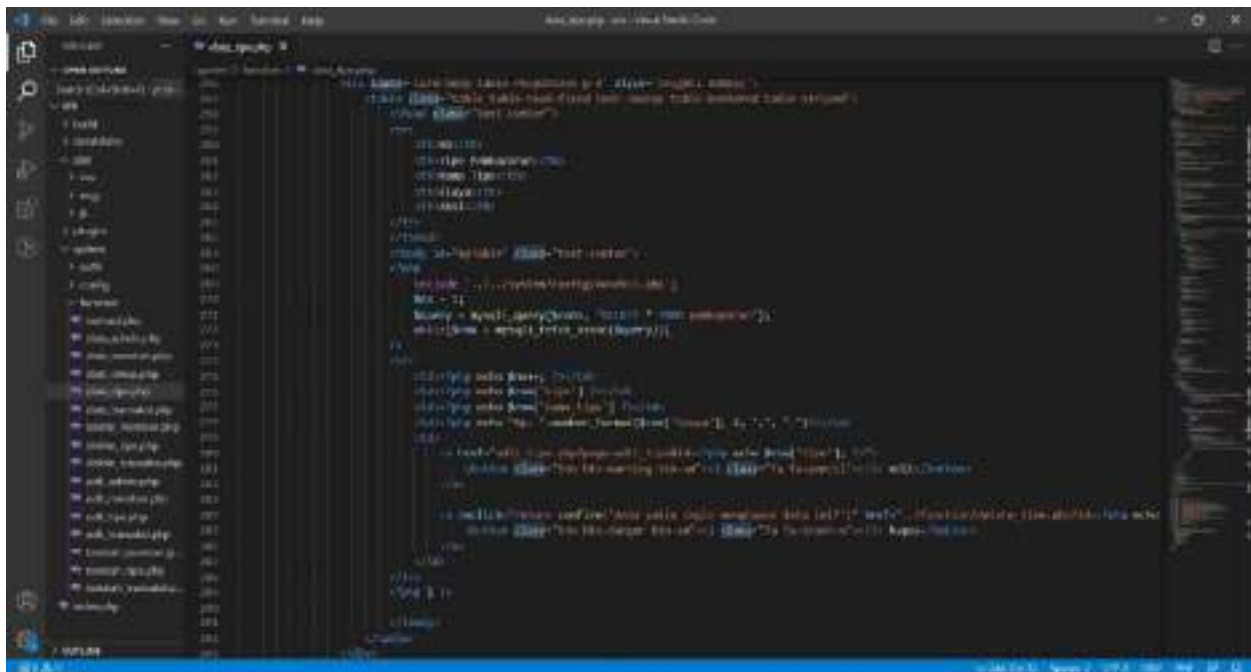


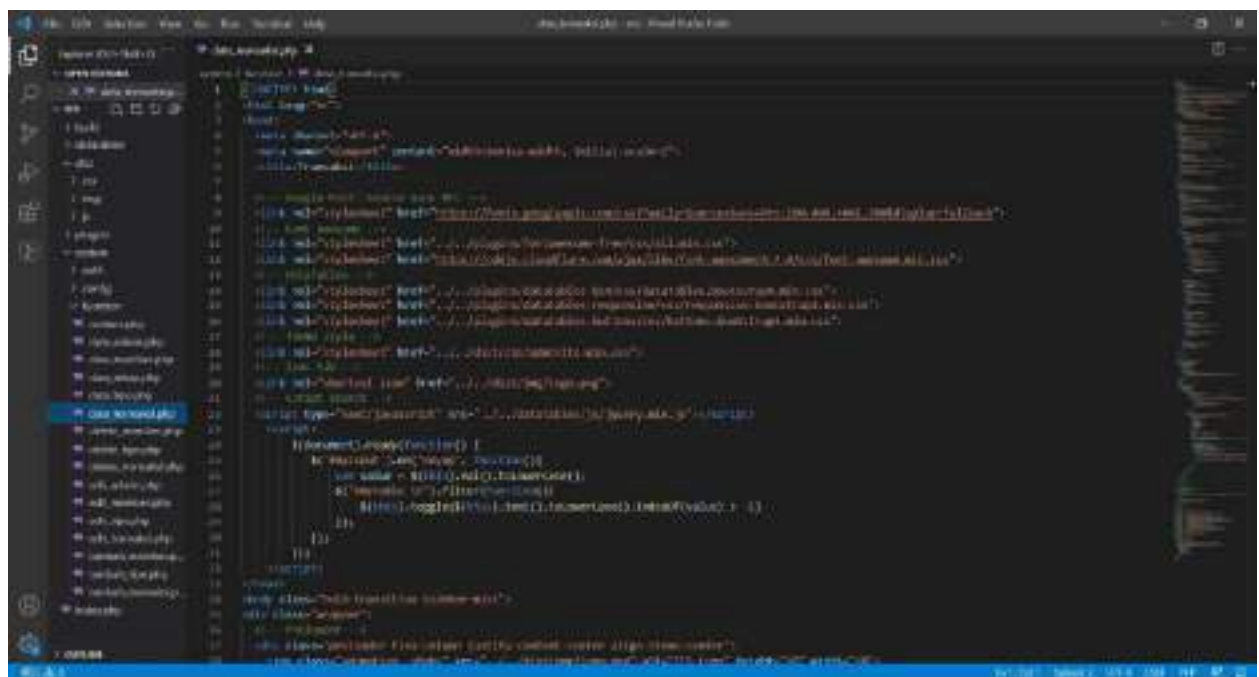
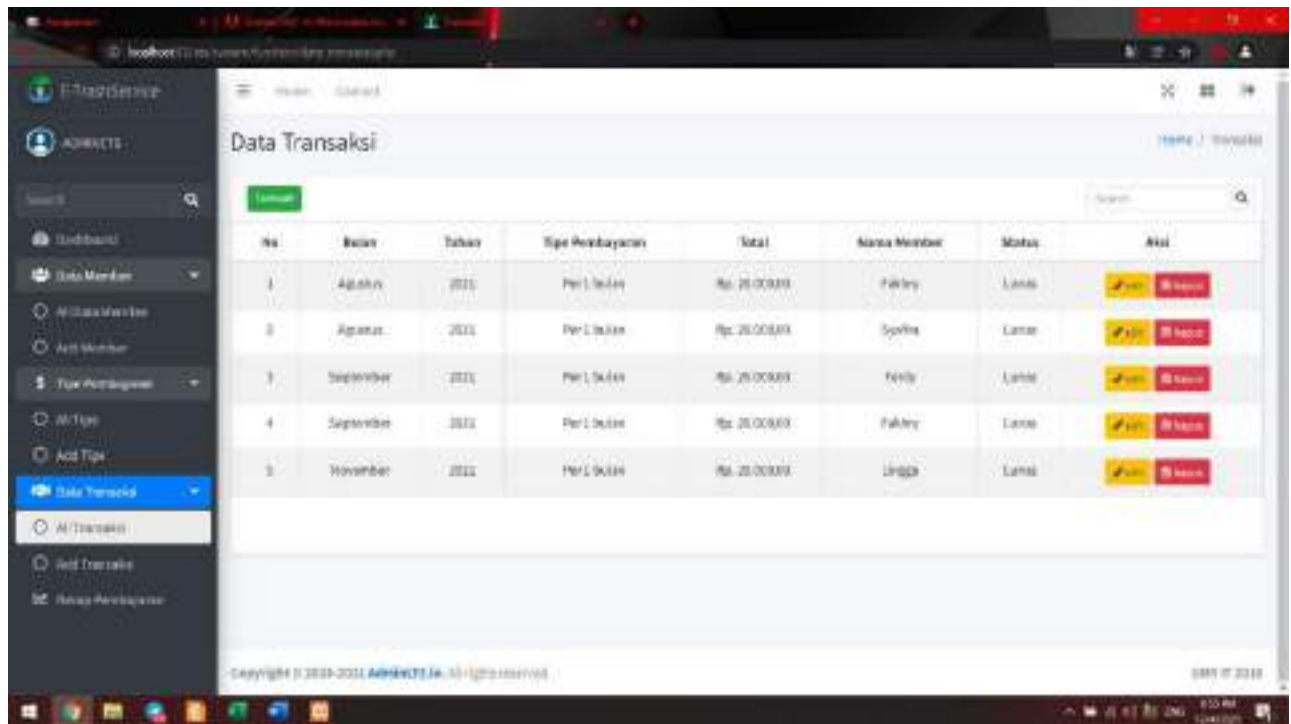




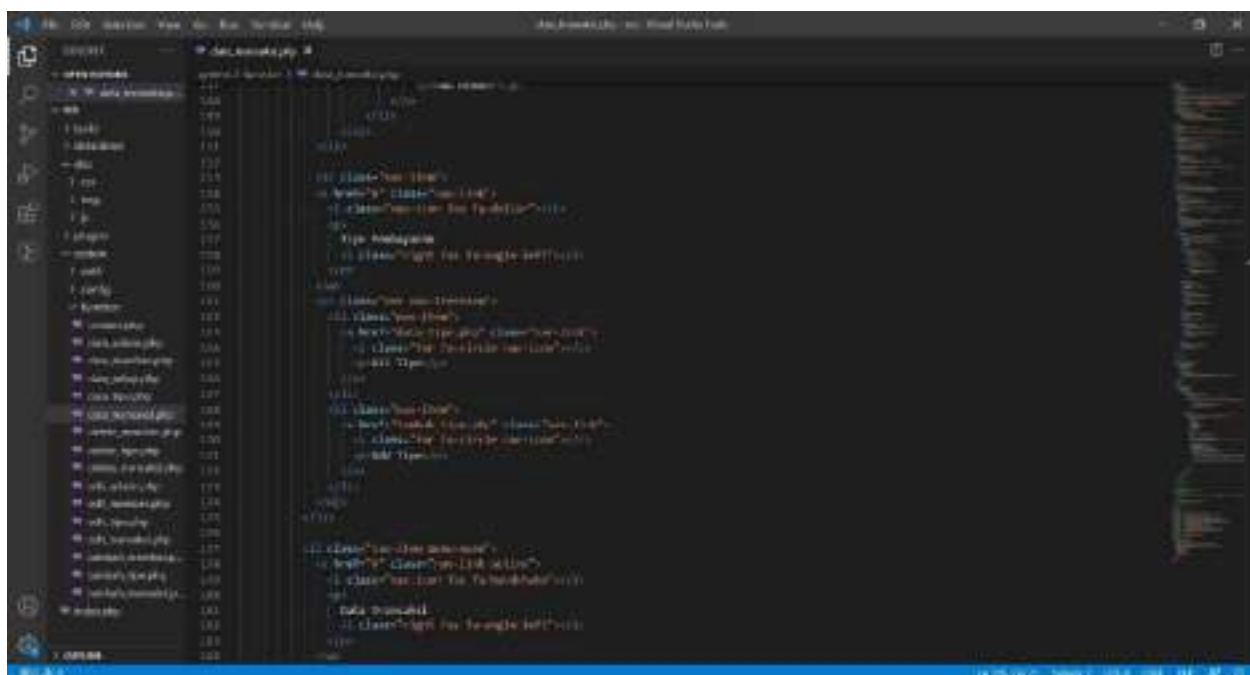
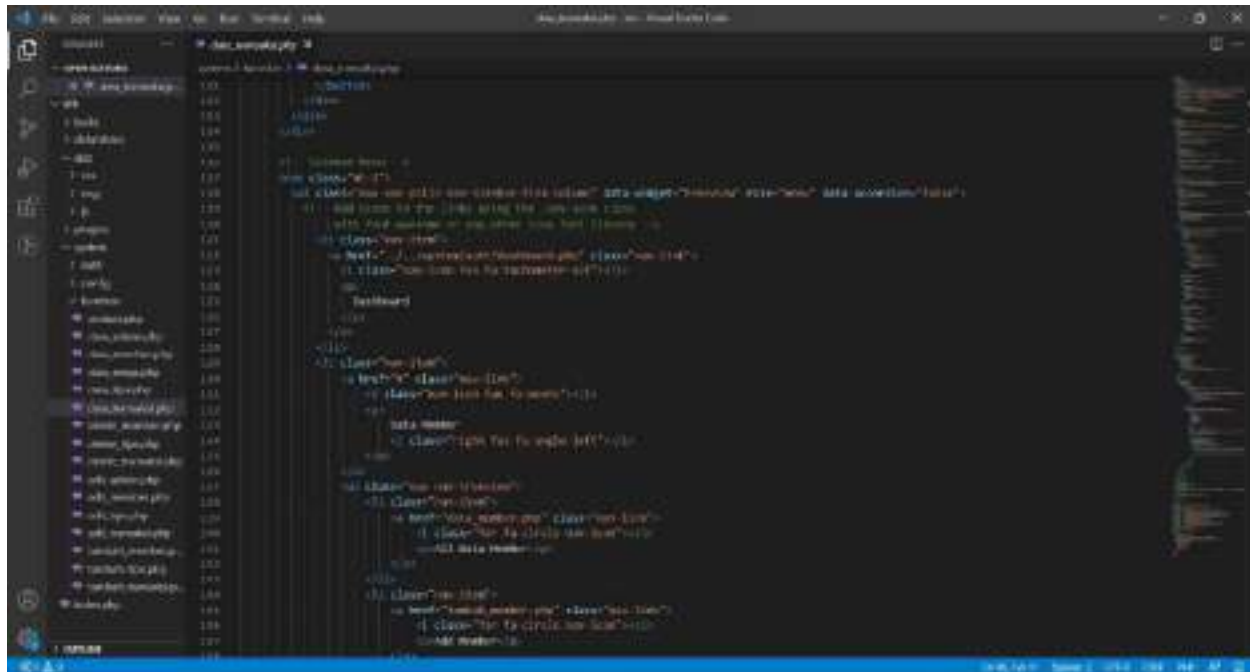












```
class: "table", style="width: 100%; border-collapse: collapse; text-align: center;">| Table with 2 columns | |
| --- | --- |
| Header 1 | Header 2 |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |
| 6 | 6 |
| 7 | 7 |
| 8 | 8 |
| 9 | 9 |
| 10 | 10 |
| 11 | 11 |
| 12 | 12 |
| 13 | 13 |
| 14 | 14 |
| 15 | 15 |
| 16 | 16 |
| 17 | 17 |
| 18 | 18 |
| 19 | 19 |
| 20 | 20 |
| 21 | 21 |
| 22 | 22 |
| 23 | 23 |
| 24 | 24 |
| 25 | 25 |
| 26 | 26 |
| 27 | 27 |
| 28 | 28 |
| 29 | 29 |
| 30 | 30 |
| 31 | 31 |
| 32 | 32 |
| 33 | 33 |
| 34 | 34 |
| 35 | 35 |
| 36 | 36 |
| 37 | 37 |
| 38 | 38 |
| 39 | 39 |
| 40 | 40 |
| 41 | 41 |
| 42 | 42 |
| 43 | 43 |
| 44 | 44 |
| 45 | 45 |
| 46 | 46 |
| 47 | 47 |
| 48 | 48 |
| 49 | 49 |
| 50 | 50 |
| 51 | 51 |
| 52 | 52 |
| 53 | 53 |
| 54 | 54 |
| 55 | 55 |
| 56 | 56 |
| 57 | 57 |
| 58 | 58 |
| 59 | 59 |
| 60 | 60 |
| 61 | 61 |
| 62 | 62 |
| 63 | 63 |
| 64 | 64 |
| 65 | 65 |
| 66 | 66 |
| 67 | 67 |
| 68 | 68 |
| 69 | 69 |
| 70 | 70 |
| 71 | 71 |
| 72 | 72 |
| 73 | 73 |
| 74 | 74 |
| 75 | 75 |
| 76 | 76 |
| 77 | 77 |
| 78 | 78 |
| 79 | 79 |
| 80 | 80 |
| 81 | 81 |
| 82 | 82 |
| 83 | 83 |
| 84 | 84 |
| 85 | 85 |
| 86 | 86 |
| 87 | 87 |
| 88 | 88 |
| 89 | 89 |
| 90 | 90 |
| 91 | 91 |
| 92 | 92 |
| 93 | 93 |
| 94 | 94 |
| 95 | 95 |
| 96 | 96 |
| 97 | 97 |
| 98 | 98 |
| 99 | 99 |
| 100 | 100 |

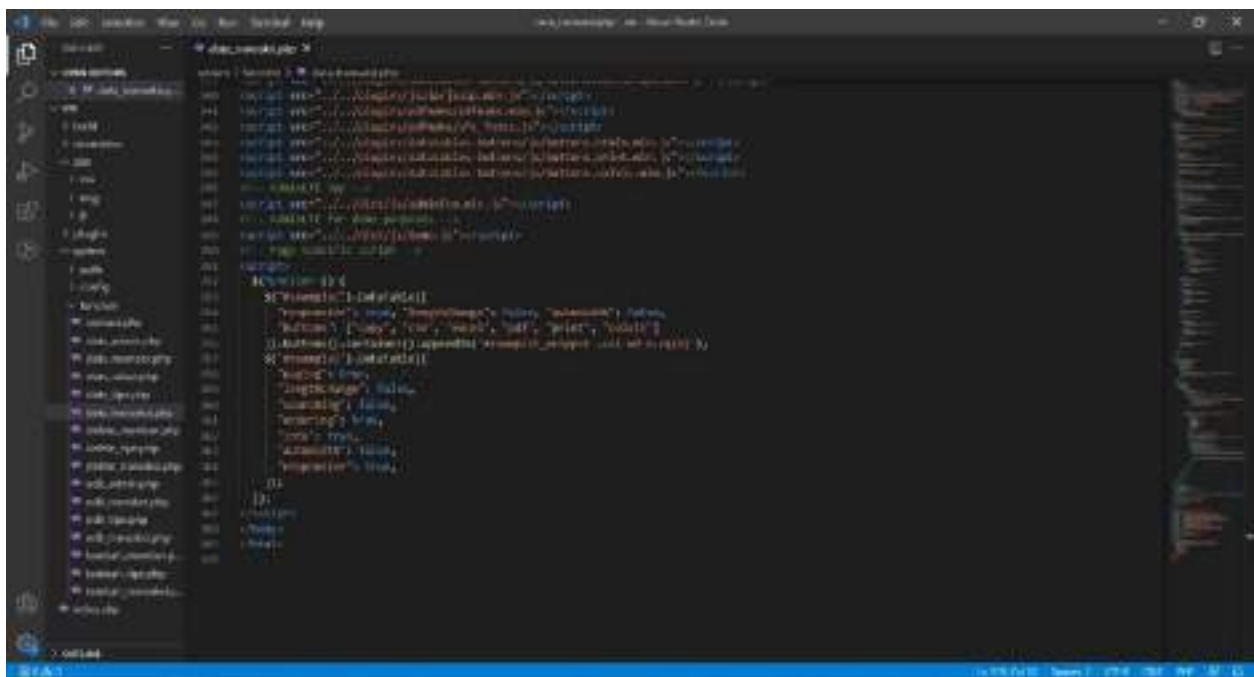
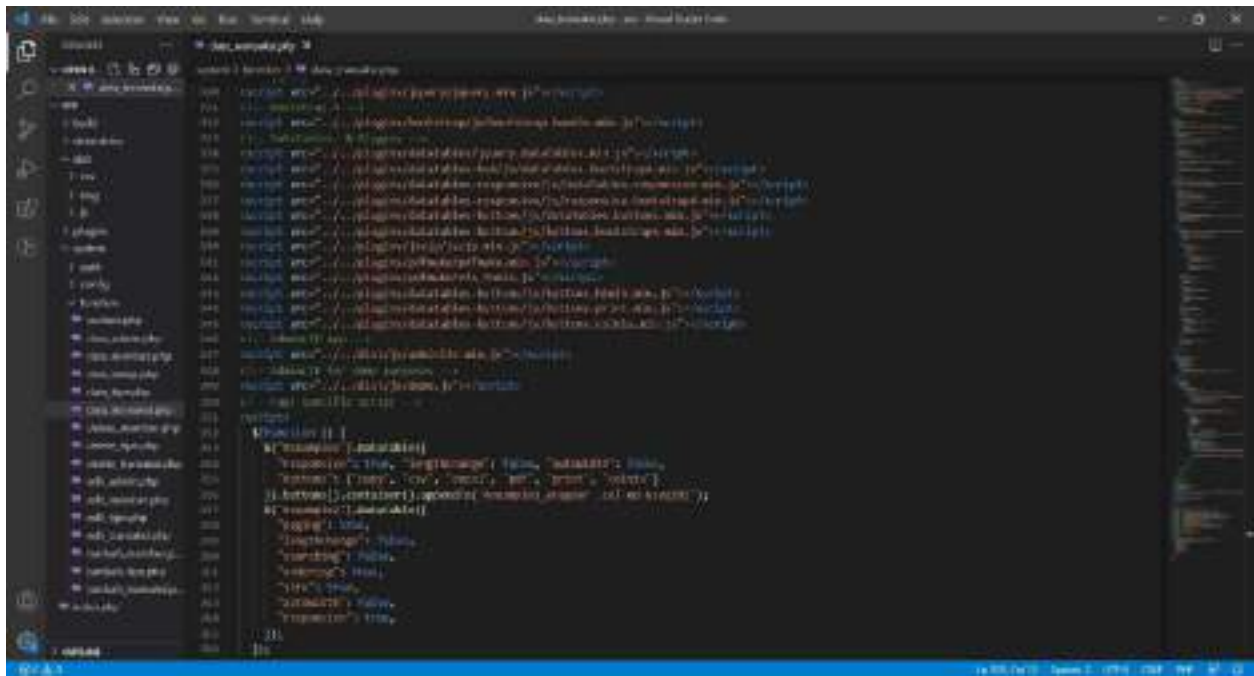
```

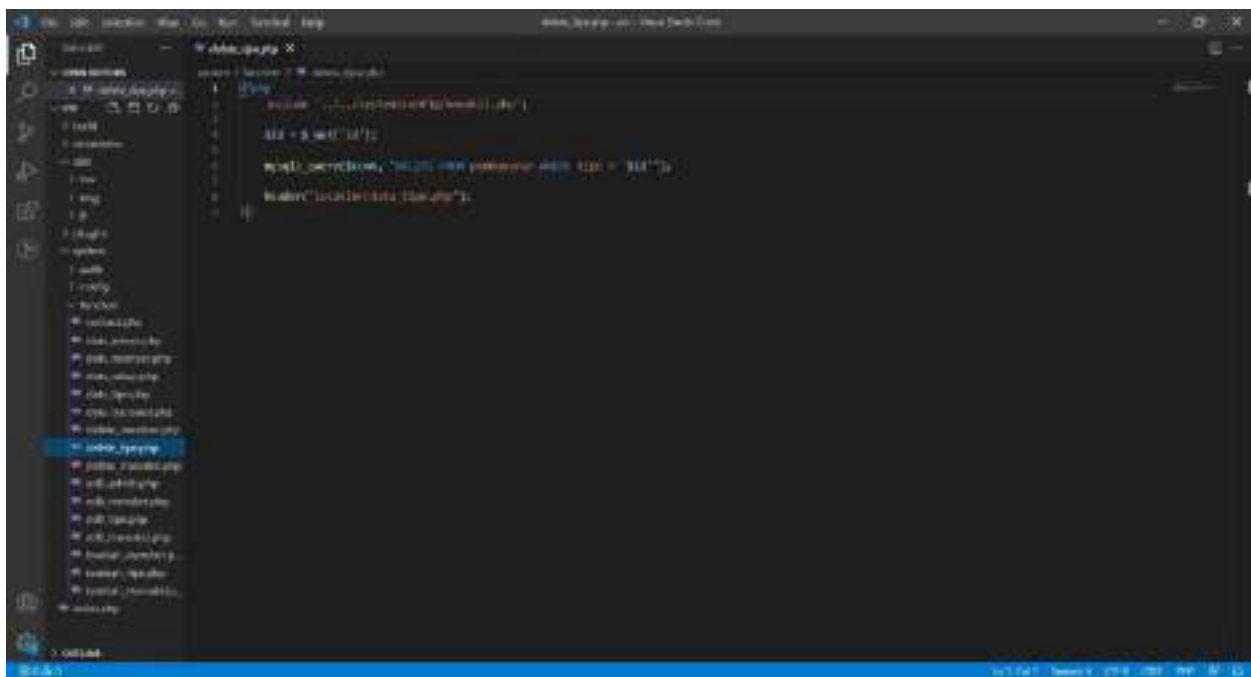
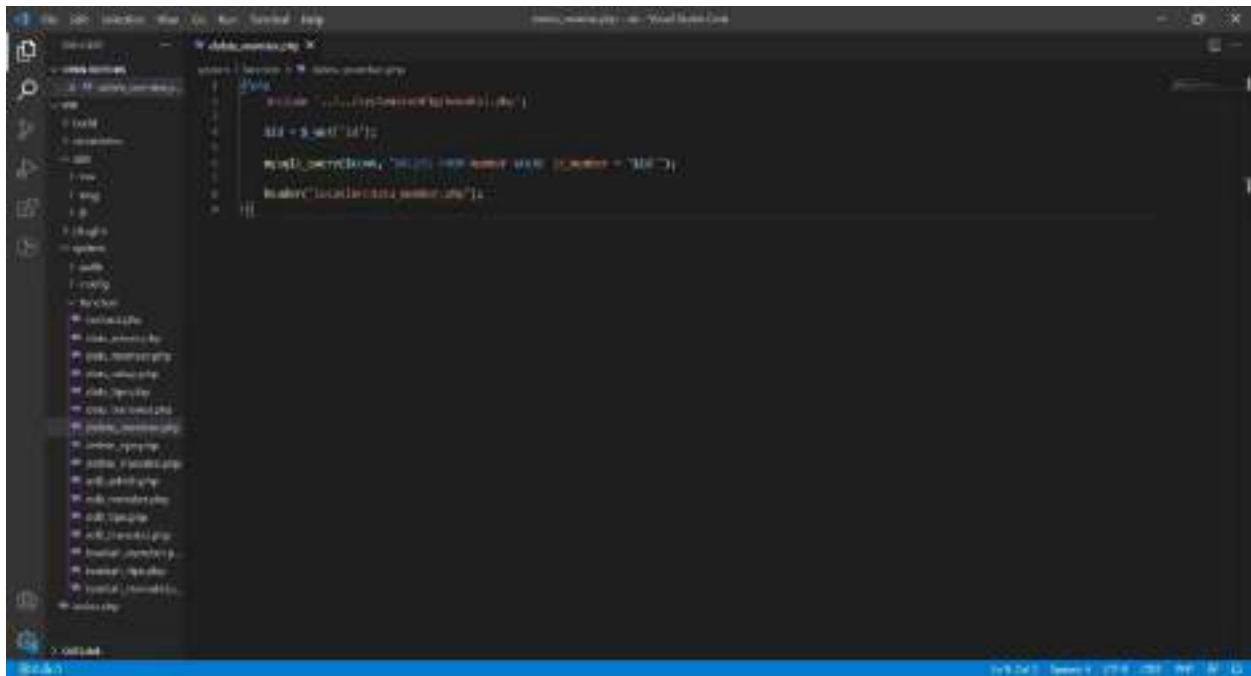
```
class: "table", style="width: 100%; border-collapse: collapse; text-align: center;">| Table with 2 columns | |
| --- | --- |
| Header 1 | Header 2 |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |
| 6 | 6 |
| 7 | 7 |
| 8 | 8 |
| 9 | 9 |
| 10 | 10 |
| 11 | 11 |
| 12 | 12 |
| 13 | 13 |
| 14 | 14 |
| 15 | 15 |
| 16 | 16 |
| 17 | 17 |
| 18 | 18 |
| 19 | 19 |
| 20 | 20 |
| 21 | 21 |
| 22 | 22 |
| 23 | 23 |
| 24 | 24 |
| 25 | 25 |
| 26 | 26 |
| 27 | 27 |
| 28 | 28 |
| 29 | 29 |
| 30 | 30 |
| 31 | 31 |
| 32 | 32 |
| 33 | 33 |
| 34 | 34 |
| 35 | 35 |
| 36 | 36 |
| 37 | 37 |
| 38 | 38 |
| 39 | 39 |
| 40 | 40 |
| 41 | 41 |
| 42 | 42 |
| 43 | 43 |
| 44 | 44 |
| 45 | 45 |
| 46 | 46 |
| 47 | 47 |
| 48 | 48 |
| 49 | 49 |
| 50 | 50 |
| 51 | 51 |
| 52 | 52 |
| 53 | 53 |
| 54 | 54 |
| 55 | 55 |
| 56 | 56 |
| 57 | 57 |
| 58 | 58 |
| 59 | 59 |
| 60 | 60 |
| 61 | 61 |
| 62 | 62 |
| 63 | 63 |
| 64 | 64 |
| 65 | 65 |
| 66 | 66 |
| 67 | 67 |
| 68 | 68 |
| 69 | 69 |
| 70 | 70 |
| 71 | 71 |
| 72 | 72 |
| 73 | 73 |
| 74 | 74 |
| 75 | 75 |
| 76 | 76 |
| 77 | 77 |
| 78 | 78 |
| 79 | 79 |
| 80 | 80 |
| 81 | 81 |
| 82 | 82 |
| 83 | 83 |
| 84 | 84 |
| 85 | 85 |
| 86 | 86 |
| 87 | 87 |
| 88 | 88 |
| 89 | 89 |
| 90 | 90 |
| 91 | 91 |
| 92 | 92 |
| 93 | 93 |
| 94 | 94 |
| 95 | 95 |
| 96 | 96 |
| 97 | 97 |
| 98 | 98 |
| 99 | 99 |
| 100 | 100 |

```









```
1 #!/usr/bin/perl
2
3 use strict;
4 use warnings;
5
6 my $url = "http://www.example.com";
7 my $method = "GET";
8 my $headers = {
9     "Host" => "www.example.com",
10    "User-Agent" => "Mozilla/5.0 (Windows NT 6.0; rv:1.9.2.1) Gecko/20100101 Firefox/3.6.10"
11 };
12
13 my $response = $curl->get($url, $headers);
14 my $content = $response->content;
15
16 print "Content: $content\n";
```

```
1 #!/usr/bin/perl
2
3 use strict;
4 use warnings;
5
6 my $url = "http://www.example.com";
7 my $method = "GET";
8 my $headers = {
9     "Host" => "www.example.com",
10    "User-Agent" => "Mozilla/5.0 (Windows NT 6.0; rv:1.9.2.1) Gecko/20100101 Firefox/3.6.10"
11 };
12
13 my $response = $curl->get($url, $headers);
14 my $content = $response->content;
15
16 # Parse the content
17 my @lines = split("\n", $content);
18
19 # Print the content
20 for my $line (@lines) {
21     print "$line\n";
22 }
23
24 # Print the status code
25 my $status_code = $response->status_code;
26 print "Status Code: $status_code\n";
```



```

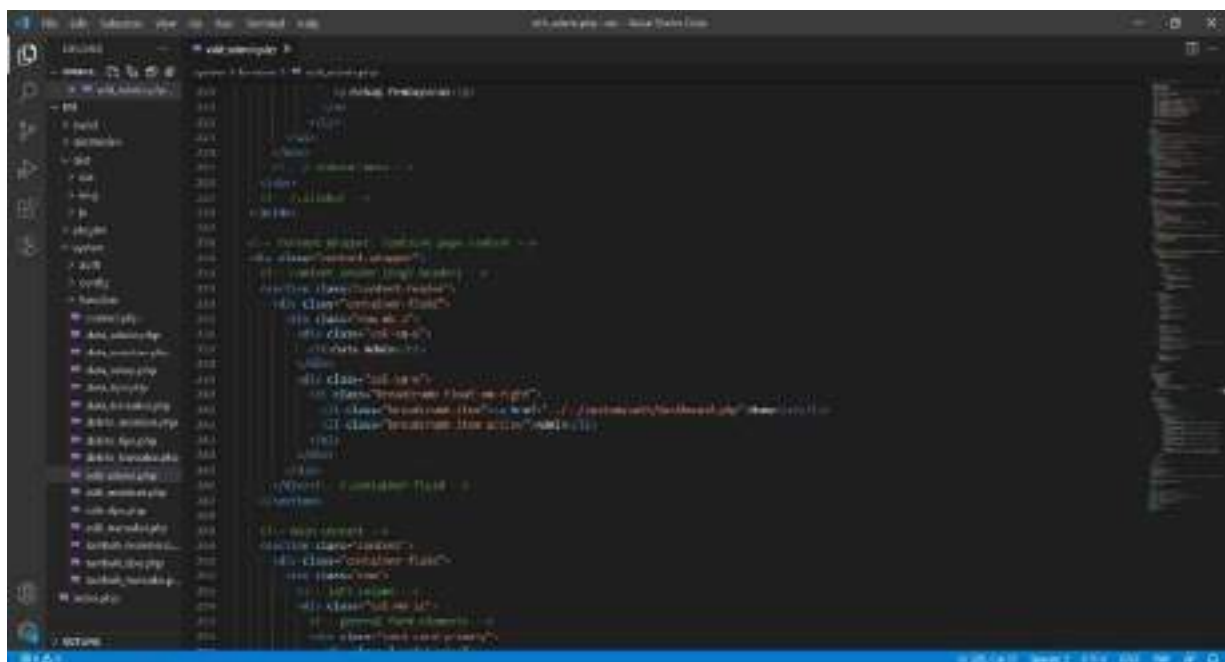
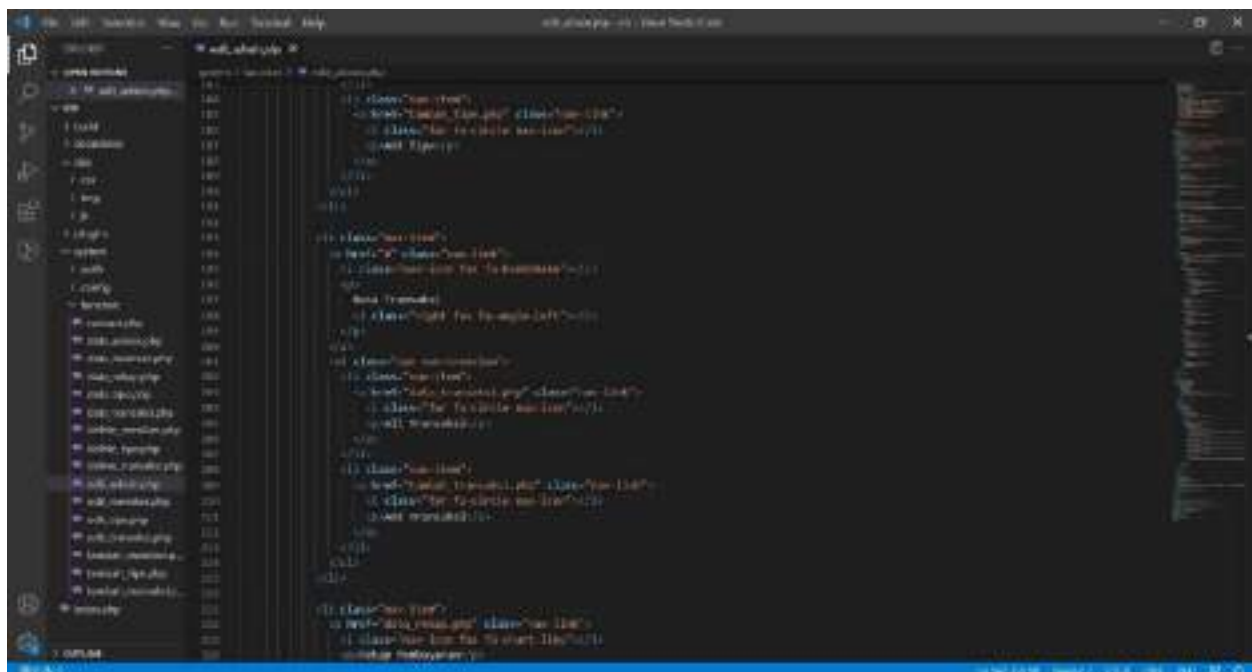
111     @objc public class AppDelegate: UIResponder, UIApplicationDelegate {
112     public var window: UIWindow?
113
114     func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
115         // Override point for customization after application launch.
116         return true
117     }
118
119     func applicationWillResignActive(_ application: UIApplication) {
120         // Sent when the application is about to move from active to inactive state. This can occur due to memory pressure, sensors not allowed for use, or short background activities like answer phone. While the user is in the foreground, this method is not called.
121     }
122
123     func applicationDidEnterBackground(_ application: UIApplication) {
124         // Use this method to alert other UI components when your app has entered the background. Apple defines when this happens: https://developer.apple.com/documentation/uikit/uiapplicationdelegate/1620701-applicationDidEnterBackground
125     }
126
127     func applicationWillEnterForeground(_ application: UIApplication) {
128         // This method is called when your app has entered the foreground.
129     }
130
131     func applicationBecomeActive(_ application: UIApplication) {
132         // Sent when the application has become active and other UI components are visible on screen.
133     }
134
135     func applicationWillTerminate(_ application: UIApplication) {
136         // Called when the application is about to terminate. Save data if appropriate. See also applicationDidEnterBackground:.
137     }
138 }
139
140 // MARK: - UISceneSession Lifecycle
141
142 @objc public class AppDelegate: UIResponder, UIApplicationDelegate, UIWindowSceneDelegate {
143     public var window: UIWindow?
144     public var sceneSession: UISceneSession?
145
146     func application(_ application: UIApplication, supportedSceneSessionClasses: UISceneSessionClasses) -> [UISceneSessionClass] {
147         // Return the supported scene session classes. This is used to decide which scene sessions are available to the application.
148         return [UISceneSession.self]
149     }
150
151     func application(_ application: UIApplication, willConnectToSession session: UISceneSession, options: UIScene.ConnectionOptions) -> Bool {
152         // Called when the application is about to connect to the session.
153         return true
154     }
155
156     func application(_ application: UIApplication, didDisconnectFromSession session: UISceneSession, reasons: UIScene.ConnectionReason) -> Bool {
157         // Called when the application has disconnected from the session.
158         return true
159     }
160
161     func application(_ application: UIApplication, supportedInterfaceOrientationsForWindow: UIWindowScene) -> [UIInterfaceOrientation] {
162         // Return the interface orientations this app supports.
163         return [UIInterfaceOrientation.landscapeLeft, UIInterfaceOrientation.landscapeRight, UIInterfaceOrientation.portrait, UIInterfaceOrientation.portraitUpsideDown]
164     }
165
166     func application(_ application: UIApplication, supportedContentSizeCategories: CGSizeRange) -> CGSizeRange {
167         // Return the supported content size categories.
168         return CGSizeRange()
169     }
170 }

```

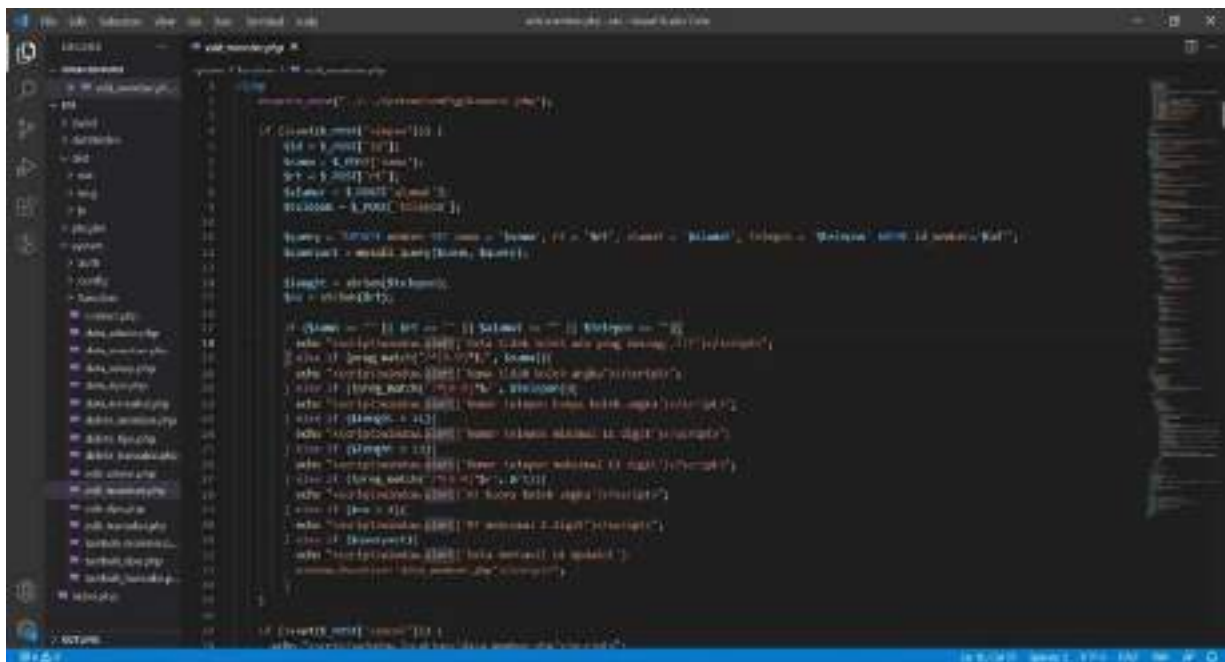
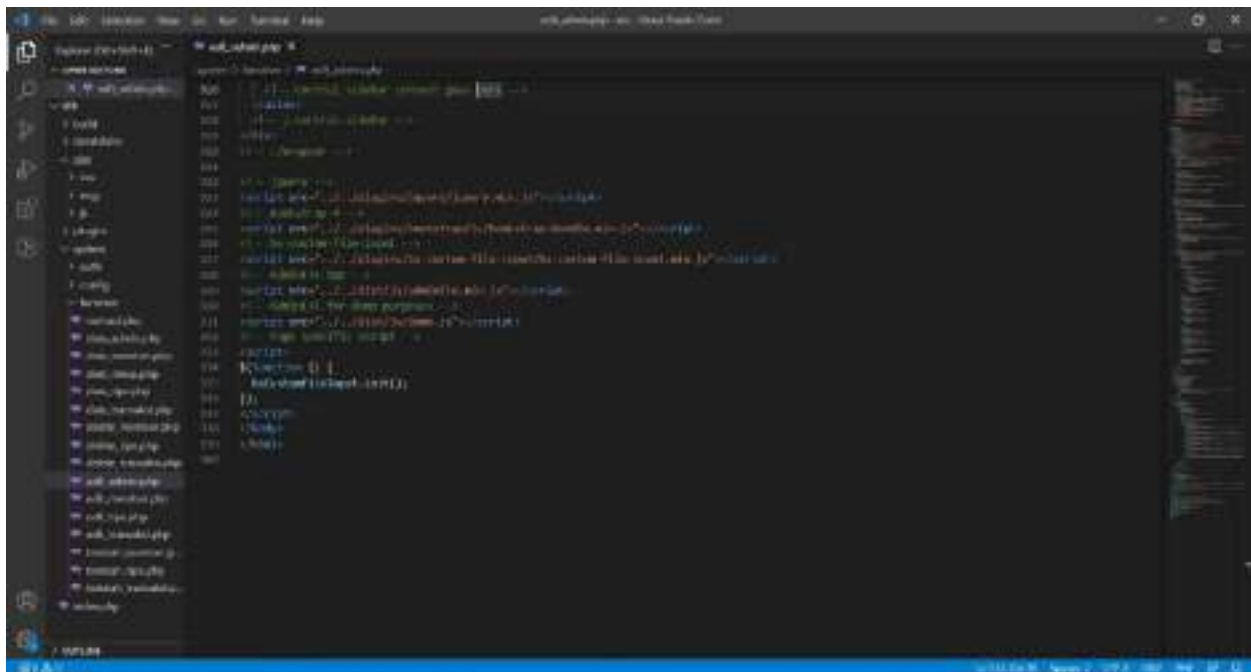
```

141     @objc public class AppDelegate: UIResponder, UIApplicationDelegate {
142     public var window: UIWindow?
143
144     func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
145         // Override point for customization after application launch.
146         return true
147     }
148
149     func applicationWillResignActive(_ application: UIApplication) {
150         // Sent when the application is about to move from active to inactive state. This can occur due to memory pressure, sensors not allowed for use, or short background activities like answer phone. While the user is in the foreground, this method is not called.
151     }
152
153     func applicationDidEnterBackground(_ application: UIApplication) {
154         // Use this method to alert other UI components when your app has entered the background. Apple defines when this happens: https://developer.apple.com/documentation/uikit/uiapplicationdelegate/1620701-applicationDidEnterBackground
155     }
156
157     func applicationWillEnterForeground(_ application: UIApplication) {
158         // This method is called when your app has entered the foreground.
159     }
160
161     func applicationBecomeActive(_ application: UIApplication) {
162         // Sent when the application has become active and other UI components are visible on screen.
163     }
164
165     func applicationWillTerminate(_ application: UIApplication) {
166         // Called when the application is about to terminate. Save data if appropriate. See also applicationDidEnterBackground:.
167     }
168 }
169
170 // MARK: - UISceneSession Lifecycle
171
172 @objc public class AppDelegate: UIResponder, UIApplicationDelegate, UIWindowSceneDelegate {
173     public var window: UIWindow?
174     public var sceneSession: UISceneSession?
175
176     func application(_ application: UIApplication, supportedSceneSessionClasses: UISceneSessionClasses) -> [UISceneSessionClass] {
177         // Return the supported scene session classes. This is used to decide which scene sessions are available to the application.
178         return [UISceneSession.self]
179     }
180
181     func application(_ application: UIApplication, willConnectToSession session: UISceneSession, options: UIScene.ConnectionOptions) -> Bool {
182         // Called when the application is about to connect to the session.
183         return true
184     }
185
186     func application(_ application: UIApplication, didDisconnectFromSession session: UISceneSession, reasons: UIScene.ConnectionReason) -> Bool {
187         // Called when the application has disconnected from the session.
188         return true
189     }
190
191     func application(_ application: UIApplication, supportedInterfaceOrientationsForWindow: UIWindowScene) -> [UIInterfaceOrientation] {
192         // Return the interface orientations this app supports.
193         return [UIInterfaceOrientation.landscapeLeft, UIInterfaceOrientation.landscapeRight, UIInterfaceOrientation.portrait, UIInterfaceOrientation.portraitUpsideDown]
194     }
195
196     func application(_ application: UIApplication, supportedContentSizeCategories: CGSizeRange) -> CGSizeRange {
197         // Return the supported content size categories.
198         return CGSizeRange()
199     }
200 }

```









```
class Base {
public:
    Base() {}
    Base(int i) : m(i) {}
    Base(int i, string s) : m(i), s(s) {}
    ~Base() {}
    virtual void print() const {
        cout << "Base: m=" << m << ", s=" << s << endl;
    }
};

class Derived : public Base {
public:
    Derived(int i) : Base(i) {}
    Derived(int i, string s) : Base(i, s) {}
    ~Derived() {}
    void print() const {
        cout << "Derived: m=" << m << ", s=" << s << endl;
    }
};

int main() {
    Base b;
    Base b1(10);
    Base b2(10, "10");
    Derived d1(20);
    Derived d2(20, "20");
    b.print();
    b1.print();
    b2.print();
    d1.print();
    d2.print();
    return 0;
}
```

```
class Base {
public:
    Base() {}
    Base(int i) : m(i) {}
    Base(int i, string s) : m(i), s(s) {}
    ~Base() {}
    virtual void print() const {
        cout << "Base: m=" << m << ", s=" << s << endl;
    }
};

class Derived : public Base {
public:
    Derived(int i) : Base(i) {}
    Derived(int i, string s) : Base(i, s) {}
    ~Derived() {}
    void print() const {
        cout << "Derived: m=" << m << ", s=" << s << endl;
    }
};

int main() {
    Base b;
    Base b1(10);
    Base b2(10, "10");
    Derived d1(20);
    Derived d2(20, "20");
    b.print();
    b1.print();
    b2.print();
    d1.print();
    d2.print();
    return 0;
}
```











```

class Solution {
public:
    int numDecodings(string s) {
        if(s.empty()) return 0;
        int n = s.length();
        int dp[n+1];
        dp[0] = 1;
        dp[1] = s[0] != '0' ? 1 : 0;
        for(int i = 2; i <= n; i++) {
            int oneDigit = s[i-1] != '0' ? dp[i-1] : 0;
            int twoDigits = s[i-2] != '0' && (s[i-2] < '1' || s[i-2] == '1' && s[i-1] < '3') ? dp[i-2] : 0;
            dp[i] = oneDigit + twoDigits;
        }
        return dp[n];
    }
};

```

```

class Solution {
public:
    int numDecodings(string s) {
        if(s.empty()) return 0;
        int n = s.length();
        int dp[n+1];
        dp[0] = 1;
        dp[1] = s[0] != '0' ? 1 : 0;
        for(int i = 2; i <= n; i++) {
            int oneDigit = s[i-1] != '0' ? dp[i-1] : 0;
            int twoDigits = s[i-2] != '0' && (s[i-2] < '1' || s[i-2] == '1' && s[i-1] < '3') ? dp[i-2] : 0;
            dp[i] = oneDigit + twoDigits;
        }
        return dp[n];
    }
};

```

```

class DataProvider:
    def __init__(self, data_dir):
        self.data_dir = data_dir

    def load_data(self):
        # Load data from the data directory
        data = {}
        for file in os.listdir(self.data_dir):
            if file.endswith('.json'):
                with open(os.path.join(self.data_dir, file)) as f:
                    data[file[:-5]] = json.load(f)
        return data

    def save_data(self, data):
        # Save data to the data directory
        for key, value in data.items():
            with open(os.path.join(self.data_dir, f'{key}.json'), 'w') as f:
                json.dump(value, f)

```

```

class DataProvider:
    def __init__(self, data_dir):
        self.data_dir = data_dir

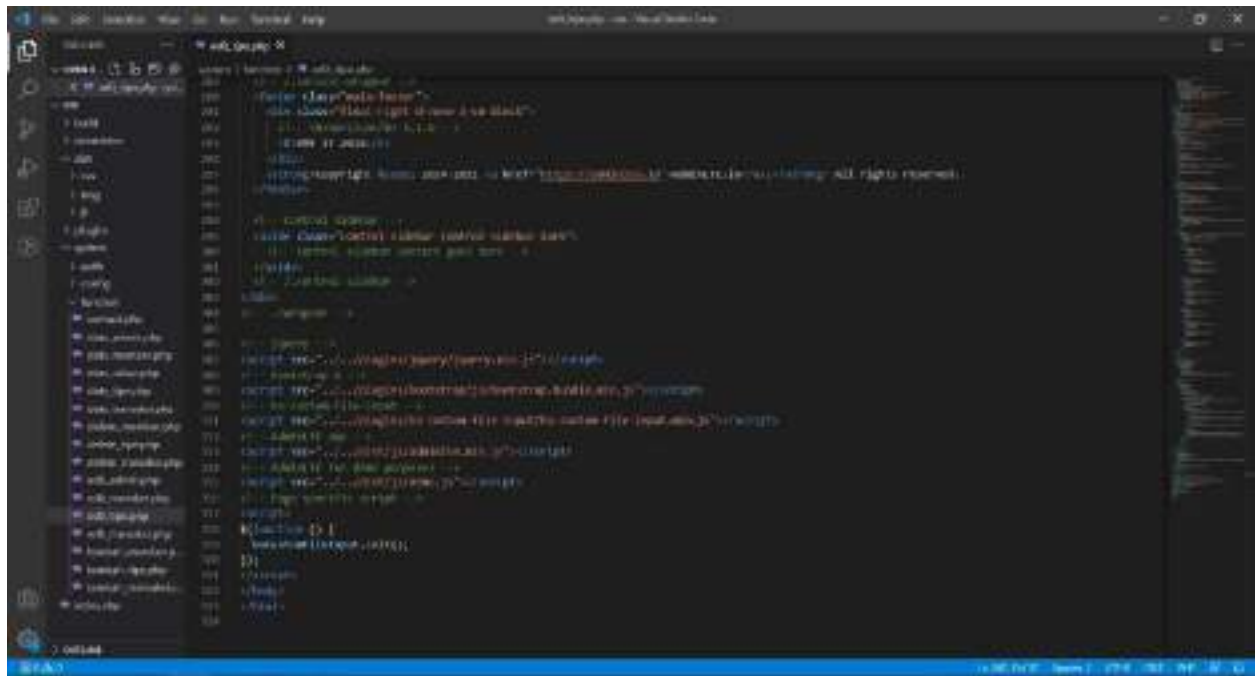
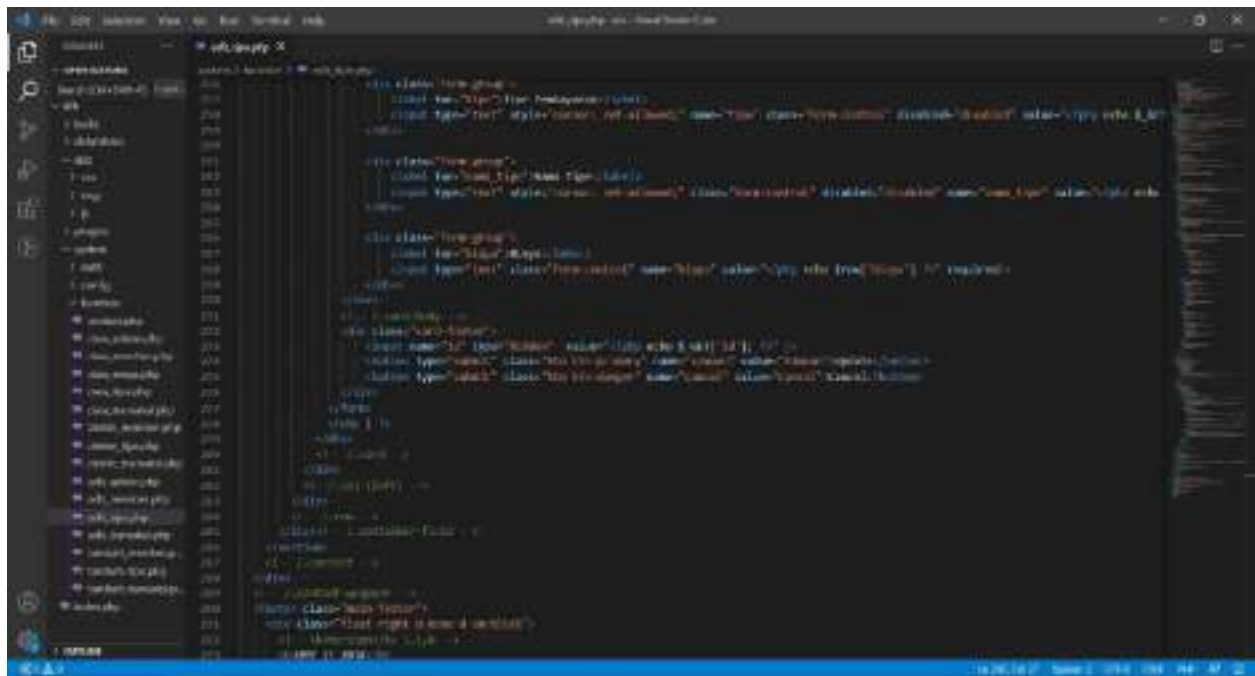
    def load_data(self):
        # Load data from the data directory
        data = {}
        for file in os.listdir(self.data_dir):
            if file.endswith('.json'):
                with open(os.path.join(self.data_dir, file)) as f:
                    data[file[:-5]] = json.load(f)
        return data

    def save_data(self, data):
        # Save data to the data directory
        for key, value in data.items():
            with open(os.path.join(self.data_dir, f'{key}.json'), 'w') as f:
                json.dump(value, f)

    def load_data(self):
        # Load data from the data directory
        data = {}
        for file in os.listdir(self.data_dir):
            if file.endswith('.json'):
                with open(os.path.join(self.data_dir, file)) as f:
                    data[file[:-5]] = json.load(f)
        return data

```





```

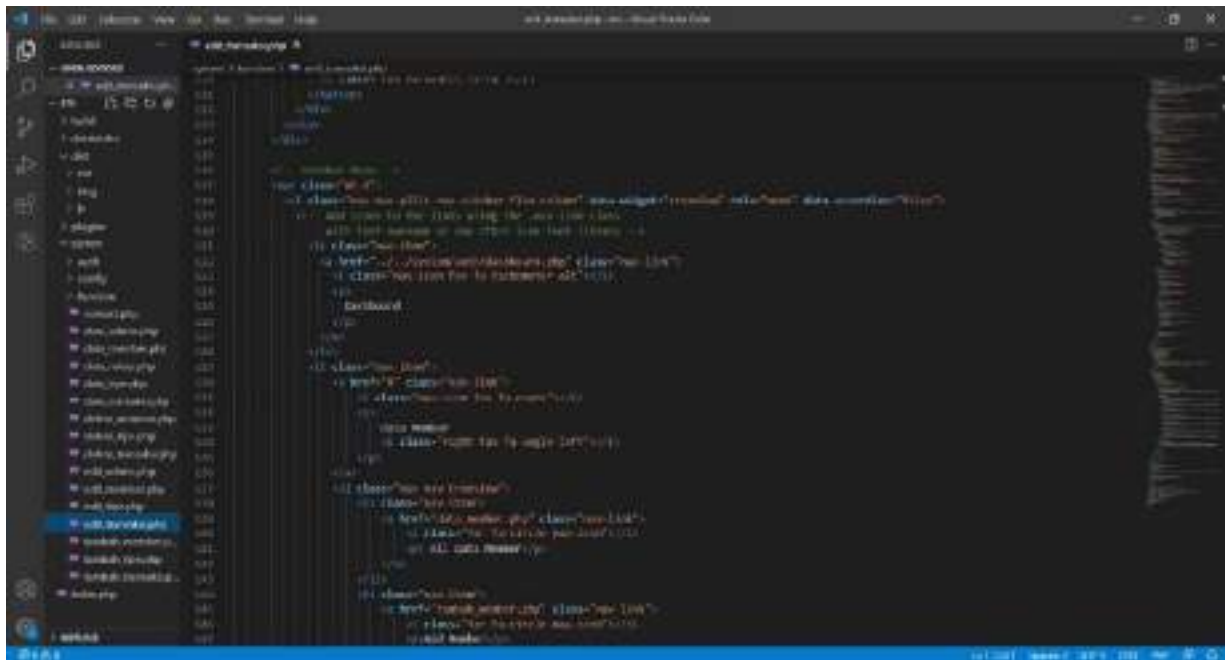
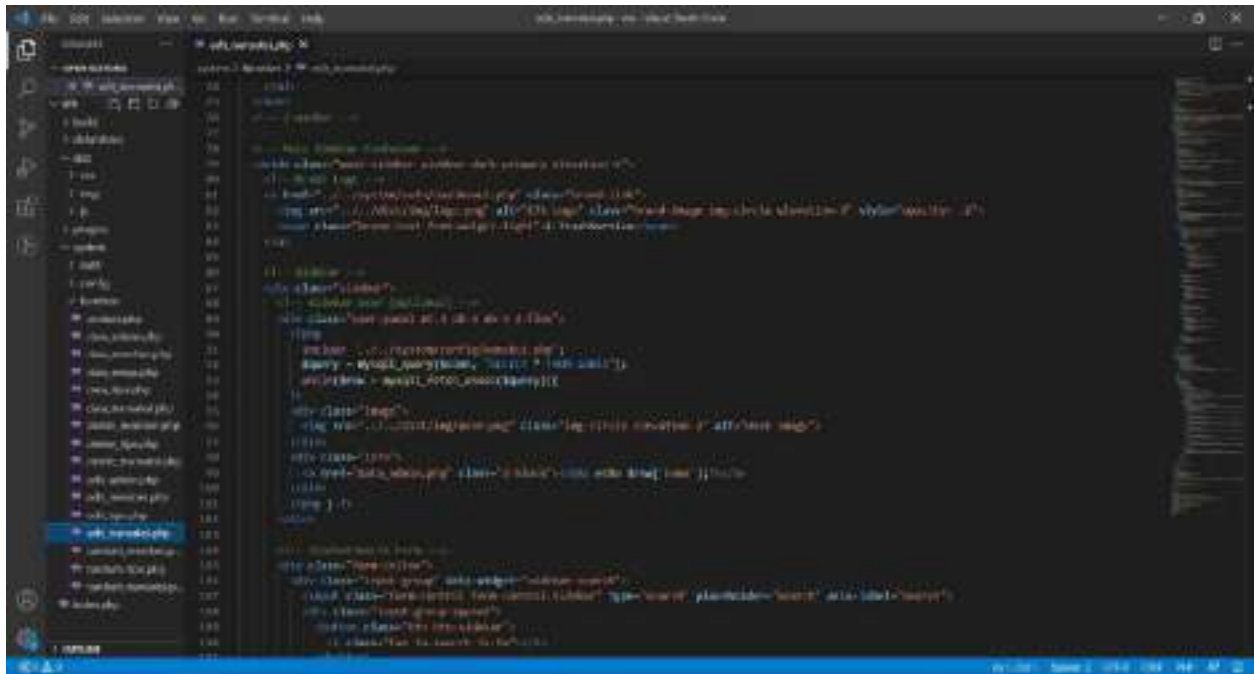
1 // ...
2 // ...
3 // ...
4 // ...
5 // ...
6 // ...
7 // ...
8 // ...
9 // ...
10 // ...
11 // ...
12 // ...
13 // ...
14 // ...
15 // ...
16 // ...
17 // ...
18 // ...
19 // ...
20 // ...
21 // ...
22 // ...
23 // ...
24 // ...
25 // ...
26 // ...
27 // ...
28 // ...
29 // ...
30 // ...
31 // ...
32 // ...
33 // ...
34 // ...
35 // ...
36 // ...
37 // ...
38 // ...
39 // ...
40 // ...
41 // ...
42 // ...
43 // ...
44 // ...
45 // ...
46 // ...
47 // ...
48 // ...
49 // ...
50 // ...
51 // ...
52 // ...
53 // ...
54 // ...
55 // ...
56 // ...
57 // ...
58 // ...
59 // ...
60 // ...
61 // ...
62 // ...
63 // ...
64 // ...
65 // ...
66 // ...
67 // ...
68 // ...
69 // ...
70 // ...
71 // ...
72 // ...
73 // ...
74 // ...
75 // ...
76 // ...
77 // ...
78 // ...
79 // ...
80 // ...
81 // ...
82 // ...
83 // ...
84 // ...
85 // ...
86 // ...
87 // ...
88 // ...
89 // ...
90 // ...
91 // ...
92 // ...
93 // ...
94 // ...
95 // ...
96 // ...
97 // ...
98 // ...
99 // ...
100 // ...

```

```

1 // ...
2 // ...
3 // ...
4 // ...
5 // ...
6 // ...
7 // ...
8 // ...
9 // ...
10 // ...
11 // ...
12 // ...
13 // ...
14 // ...
15 // ...
16 // ...
17 // ...
18 // ...
19 // ...
20 // ...
21 // ...
22 // ...
23 // ...
24 // ...
25 // ...
26 // ...
27 // ...
28 // ...
29 // ...
30 // ...
31 // ...
32 // ...
33 // ...
34 // ...
35 // ...
36 // ...
37 // ...
38 // ...
39 // ...
40 // ...
41 // ...
42 // ...
43 // ...
44 // ...
45 // ...
46 // ...
47 // ...
48 // ...
49 // ...
50 // ...
51 // ...
52 // ...
53 // ...
54 // ...
55 // ...
56 // ...
57 // ...
58 // ...
59 // ...
60 // ...
61 // ...
62 // ...
63 // ...
64 // ...
65 // ...
66 // ...
67 // ...
68 // ...
69 // ...
70 // ...
71 // ...
72 // ...
73 // ...
74 // ...
75 // ...
76 // ...
77 // ...
78 // ...
79 // ...
80 // ...
81 // ...
82 // ...
83 // ...
84 // ...
85 // ...
86 // ...
87 // ...
88 // ...
89 // ...
90 // ...
91 // ...
92 // ...
93 // ...
94 // ...
95 // ...
96 // ...
97 // ...
98 // ...
99 // ...
100 // ...

```



```

class Node {
public:
    int data;
    Node* next;
};

int main() {
    Node* head = new Node(1);
    Node* temp = head;
    while (temp != NULL) {
        temp->next = new Node(2);
        temp = temp->next;
    }
    return 0;
}

```

```

class Node {
public:
    int data;
    Node* next;
};

int main() {
    Node* head = new Node(1);
    Node* temp = head;
    while (temp != NULL) {
        temp->next = new Node(2);
        temp = temp->next;
    }
    return 0;
}

```



Visual Studio Code editor showing the source code of `all_test.py`. The file contains a `unittest` class named `AllTest` with various test methods. The `test_email` method is currently selected, showing the following code:

```

def test_email(self):
    """test email"""
    email = "test@gmail.com"
    self.assertEqual(email, "test@gmail.com")

```

The editor interface includes a sidebar on the left with a file explorer, a central editor area with a scroll bar, and a right-hand pane showing a spectrum graph. The status bar at the bottom displays "19/260" and other system information.

Visual Studio Code editor showing the source code of `all_test.py`. The `test_email` method is now highlighted in blue, indicating it is the active selection. The code remains the same as in the previous image:

```

def test_email(self):
    """test email"""
    email = "test@gmail.com"
    self.assertEqual(email, "test@gmail.com")

```

The editor interface is consistent with the previous image, showing the file explorer, editor area, spectrum graph, and status bar.

```

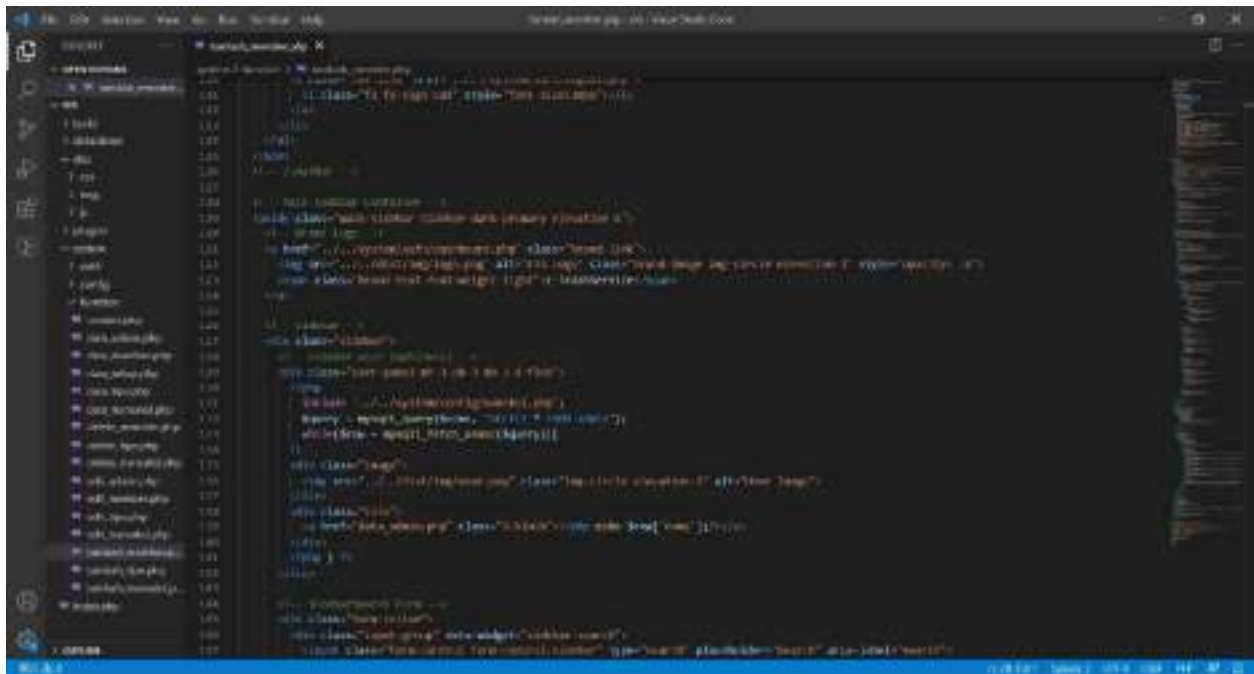
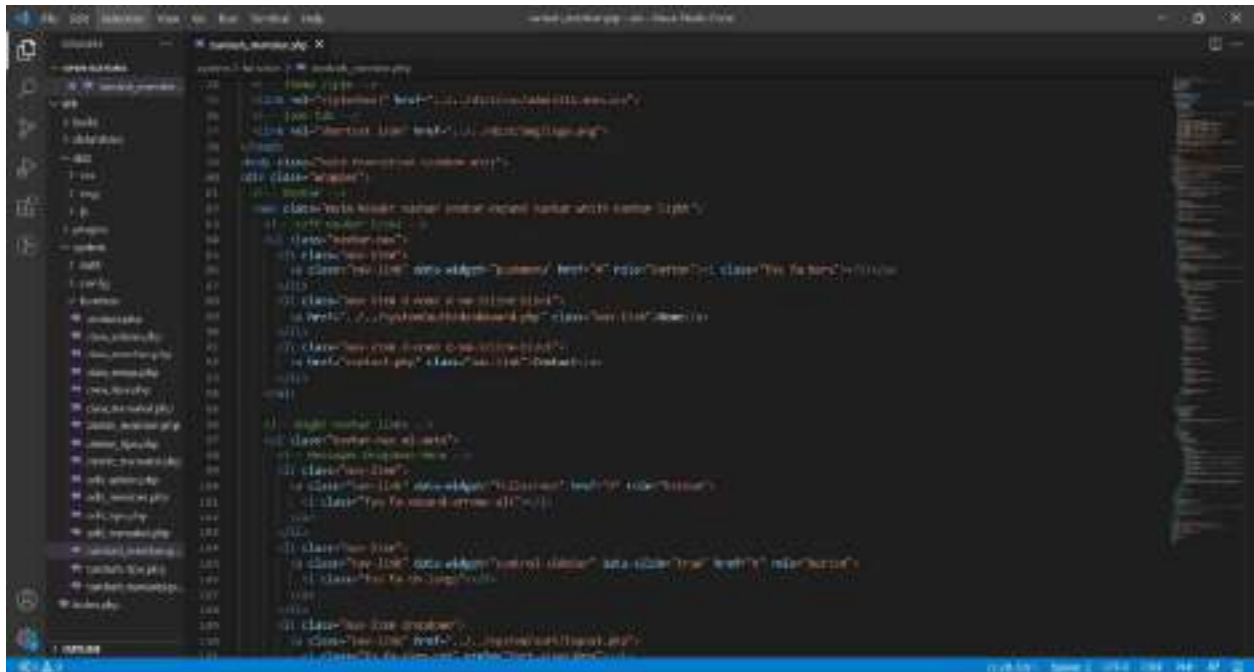
1  #!/usr/bin/perl
2  use strict;
3  use warnings;
4
5  my $url = "http://www.example.com";
6  my $method = "GET";
7  my $headers = {
8      "Host" => "www.example.com",
9      "User-Agent" => "Mozilla/5.0 (Windows NT 6.0; rv:2.0) Gecko/20100101 Firefox/4.0",
10     "Accept" => "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8",
11     "Accept-Language" => "en-us,en;q=0.5",
12     "Accept-Encoding" => "gzip, deflate",
13     "Connection" => "close",
14     "Cookie" => ""
15 };
16
17 my $response = $curl->perform($url, $method, $headers);
18 my $status = $curl->getInfo('http_code');
19 my $content = $curl->getInfo('content');
20
21 print "Status: $status\n";
22 print "Content: $content\n";
23
24 # Parse the content
25 my $html = $content;
26 my $doc = HTML::Dom->new($html);
27 my $title = $doc->title();
28 my $body = $doc->body();
29 my $p = $body->find('p');
30 my $text = $p->text();
31
32 print "Title: $title\n";
33 print "Text: $text\n";
34
35 # Close the curl object
36 $curl->close();
37
38 # Exit
39 exit 0;
40
41 # vim: set syntax=perl:
42

```

```

1  #!/usr/bin/perl
2  use strict;
3  use warnings;
4
5  my $url = "http://www.example.com";
6  my $method = "GET";
7  my $headers = {
8      "Host" => "www.example.com",
9      "User-Agent" => "Mozilla/5.0 (Windows NT 6.0; rv:2.0) Gecko/20100101 Firefox/4.0",
10     "Accept" => "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8",
11     "Accept-Language" => "en-us,en;q=0.5",
12     "Accept-Encoding" => "gzip, deflate",
13     "Connection" => "close",
14     "Cookie" => ""
15 };
16
17 my $response = $curl->perform($url, $method, $headers);
18 my $status = $curl->getInfo('http_code');
19 my $content = $curl->getInfo('content');
20
21 print "Status: $status\n";
22 print "Content: $content\n";
23
24 # Parse the content
25 my $html = $content;
26 my $doc = HTML::Dom->new($html);
27 my $title = $doc->title();
28 my $body = $doc->body();
29 my $p = $body->find('p');
30 my $text = $p->text();
31
32 print "Title: $title\n";
33 print "Text: $text\n";
34
35 # Close the curl object
36 $curl->close();
37
38 # Exit
39 exit 0;
40
41 # vim: set syntax=perl:
42

```





```

class NeuralNetwork:
    def __init__(self, input_nodes, hidden_nodes, output_nodes):
        self.input_nodes = input_nodes
        self.hidden_nodes = hidden_nodes
        self.output_nodes = output_nodes
        self.weights = []
        self.biases = []

        # Initialize weights and biases
        for i in range(1, hidden_nodes + 1):
            self.weights.append([0.1] * input_nodes)
            self.biases.append([0.1] * input_nodes)

        for i in range(1, output_nodes + 1):
            self.weights.append([0.1] * hidden_nodes)
            self.biases.append([0.1] * hidden_nodes)

        self.sigmoid = sigmoid
        self.forward_pass = forward_pass

    def forward_pass(self, input_data):
        # Input layer
        hidden_layer = []
        for i in range(1, hidden_nodes + 1):
            node_output = 0
            for j in range(1, input_nodes + 1):
                node_output += input_data[j-1] * self.weights[i][j-1] + self.biases[i]
            hidden_layer.append(self.sigmoid(node_output))

        # Output layer
        output = 0
        for i in range(1, output_nodes + 1):
            node_output = 0
            for j in range(1, hidden_nodes + 1):
                node_output += hidden_layer[j-1] * self.weights[i][j-1] + self.biases[i]
            output += self.sigmoid(node_output)

        return output
  
```

```

    def backward_pass(self, target_output, input_data):
        # Output layer error
        output_error = target_output - self.forward_pass(input_data)

        # Hidden layer error
        hidden_errors = []
        for i in range(1, hidden_nodes + 1):
            node_error = 0
            for j in range(1, output_nodes + 1):
                node_error += output_error * self.weights[j][i-1]
            hidden_errors.append(self.sigmoid'(self.forward_pass(input_data)[i-1]) * node_error)

        # Backpropagation
        for i in range(1, hidden_nodes + 1):
            for j in range(1, input_nodes + 1):
                self.weights[i][j-1] -= learning_rate * input_data[j-1] * hidden_errors[i-1]
                self.biases[i] -= learning_rate * hidden_errors[i-1]

        for i in range(1, output_nodes + 1):
            for j in range(1, hidden_nodes + 1):
                self.weights[i][j-1] -= learning_rate * hidden_layer[j-1] * output_error
                self.biases[i] -= learning_rate * output_error

        return self.weights, self.biases
  
```

```
1 // main.cpp
2 #include <iostream>
3 using namespace std;
4
5 class Person {
6     string name;
7     int age;
8     string gender;
9
10    void display() {
11        cout << "Name: " << name << endl;
12        cout << "Age: " << age << endl;
13        cout << "Gender: " << gender << endl;
14    }
15
16    int getAge() {
17        return age;
18    }
19
20    void setAge(int age) {
21        this->age = age;
22    }
23 };
24
25 int main() {
26    Person p;
27    p.name = "John";
28    p.age = 30;
29    p.gender = "Male";
30    p.display();
31    cout << "Age: " << p.getAge() << endl;
32    p.setAge(35);
33    p.display();
34    return 0;
35 }
```

```
1 // main.cpp
2 #include <iostream>
3 using namespace std;
4
5 class Person {
6     string name;
7     int age;
8     string gender;
9
10    void display() {
11        cout << "Name: " << name << endl;
12        cout << "Age: " << age << endl;
13        cout << "Gender: " << gender << endl;
14    }
15
16    int getAge() {
17        return age;
18    }
19
20    void setAge(int age) {
21        this->age = age;
22    }
23 };
24
25 int main() {
26    Person p;
27    p.name = "John";
28    p.age = 30;
29    p.gender = "Male";
30    p.display();
31    cout << "Age: " << p.getAge() << endl;
32    p.setAge(35);
33    p.display();
34    return 0;
35 }
```



```

1  #!/usr/bin/perl

2  use strict;
3  use warnings;

4  my $url = "http://www.example.com/";
5  my $method = "GET";
6  my $headers = {
7      "Host" => "www.example.com",
8      "User-Agent" => "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.164 Safari/537.36"
9  };

10 my $response = HTTP::Request::Common($url, $method, $headers);
11 my $client = HTTP::Tiny->new;
12 my $res = $client->request($response);

13 if ($res->{success}) {
14     my $content = $res->{content};
15     my $status = $res->{status};
16     my $headers = $res->{headers};

17     # Print the content
18     print "Content: $content\n";

19     # Print the status
20     print "Status: $status\n";

21     # Print the headers
22     foreach my $header (keys %$headers) {
23         print "$header: $headers->{$header}\n";
24     }
25 } else {
26     my $error = $res->{error};
27     print "Error: $error\n";
28 }

```

```

1  #!/usr/bin/perl

2  use strict;
3  use warnings;

4  my $url = "http://www.example.com/";
5  my $method = "GET";
6  my $headers = {
7      "Host" => "www.example.com",
8      "User-Agent" => "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.164 Safari/537.36"
9  };

10 my $response = HTTP::Request::Common($url, $method, $headers);
11 my $client = HTTP::Tiny->new;
12 my $res = $client->request($response);

13 if ($res->{success}) {
14     my $content = $res->{content};
15     my $status = $res->{status};
16     my $headers = $res->{headers};

17     # Print the content
18     print "Content: $content\n";

19     # Print the status
20     print "Status: $status\n";

21     # Print the headers
22     foreach my $header (keys %$headers) {
23         print "$header: $headers->{$header}\n";
24     }
25 } else {
26     my $error = $res->{error};
27     print "Error: $error\n";
28 }

```



```

class Node:
    def __init__(self, data):
        self.data = data
        self.next = None

class LinkedList:
    def __init__(self):
        self.head = None

    def insert_at_beginning(self, data):
        new_node = Node(data)
        new_node.next = self.head
        self.head = new_node

    def insert_at_end(self, data):
        new_node = Node(data)
        if self.head is None:
            self.head = new_node
        else:
            current = self.head
            while current.next is not None:
                current = current.next
            current.next = new_node

    def insert_after(self, target_data, data):
        new_node = Node(data)
        if self.head is None:
            return
        current = self.head
        while current is not None:
            if current.data == target_data:
                new_node.next = current.next
                current.next = new_node
                return
            current = current.next
        print("Target not found")

    def delete_node(self, target_data):
        if self.head is None:
            return
        if self.head.data == target_data:
            self.head = self.head.next
            return
        current = self.head
        while current is not None:
            if current.next.data == target_data:
                current.next = current.next.next
                return
            current = current.next
        print("Target not found")

    def display(self):
        current = self.head
        while current is not None:
            print(current.data, end=" ")
            current = current.next
        print()

# Example usage
ll = LinkedList()
ll.insert_at_beginning(10)
ll.insert_at_end(20)
ll.insert_after(10, 15)
ll.delete_node(10)
ll.display()

```

```

class Node:
    def __init__(self, data):
        self.data = data
        self.next = None

class LinkedList:
    def __init__(self):
        self.head = None

    def insert_at_beginning(self, data):
        new_node = Node(data)
        new_node.next = self.head
        self.head = new_node

    def insert_at_end(self, data):
        new_node = Node(data)
        if self.head is None:
            self.head = new_node
        else:
            current = self.head
            while current.next is not None:
                current = current.next
            current.next = new_node

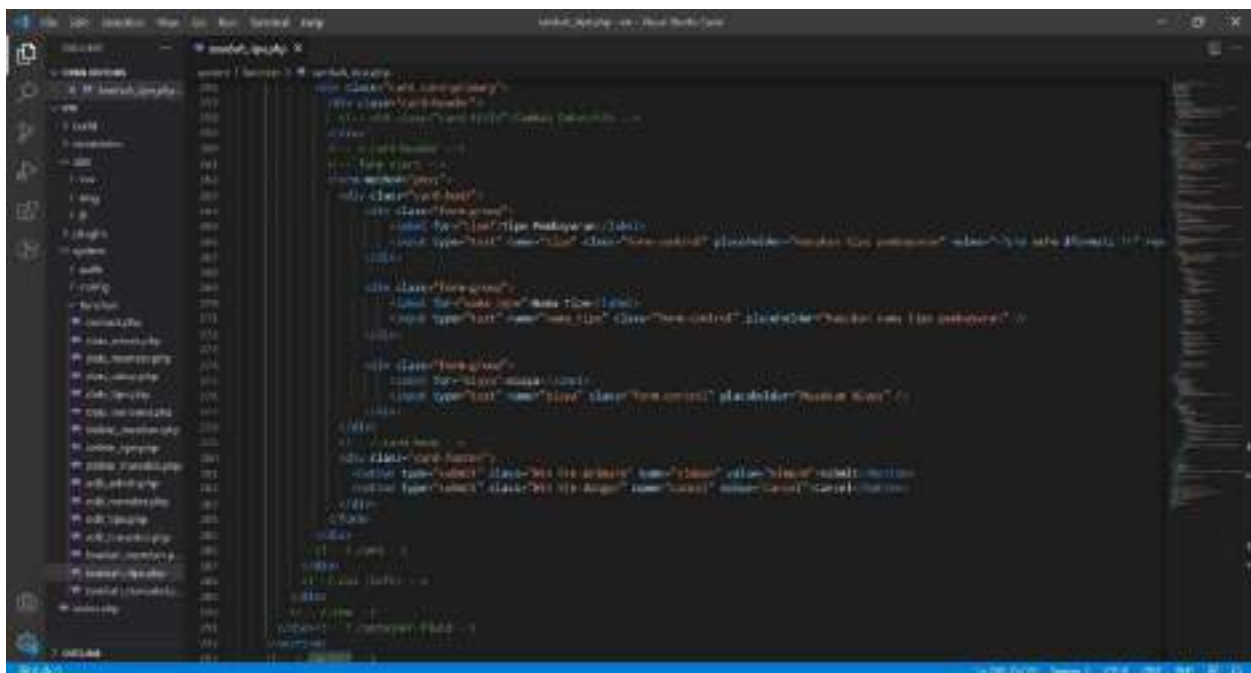
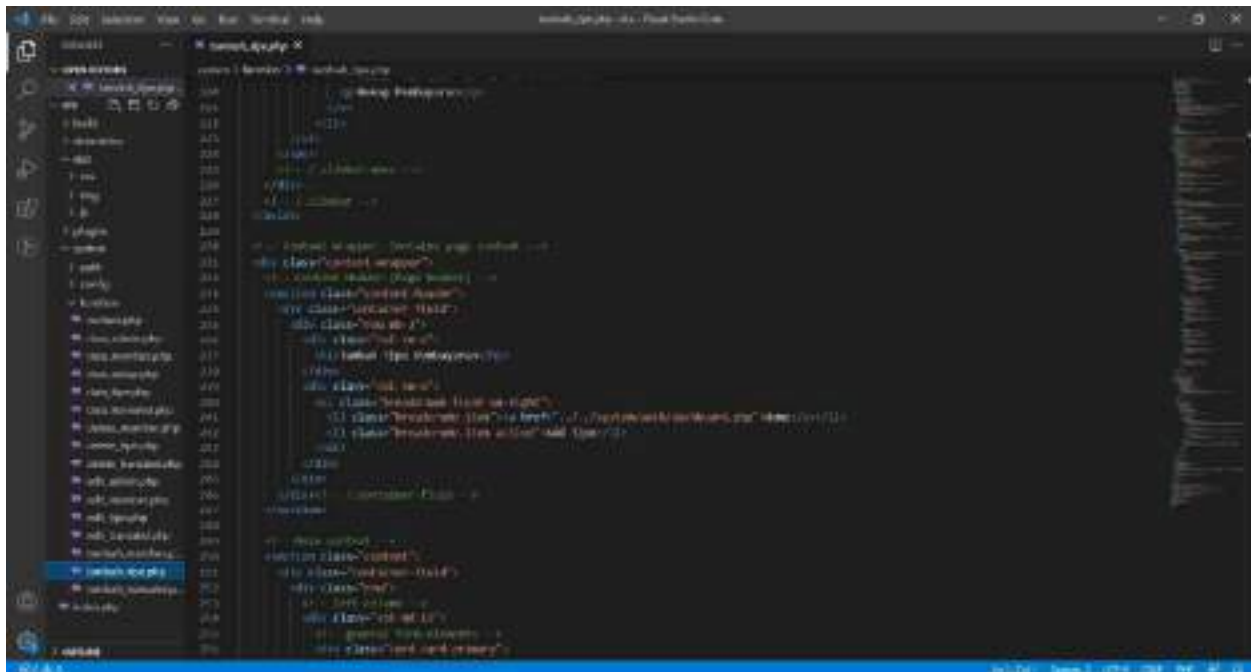
    def insert_after(self, target_data, data):
        new_node = Node(data)
        if self.head is None:
            return
        current = self.head
        while current is not None:
            if current.data == target_data:
                new_node.next = current.next
                current.next = new_node
                return
            current = current.next
        print("Target not found")

    def delete_node(self, target_data):
        if self.head is None:
            return
        if self.head.data == target_data:
            self.head = self.head.next
            return
        current = self.head
        while current is not None:
            if current.next.data == target_data:
                current.next = current.next.next
                return
            current = current.next
        print("Target not found")

    def display(self):
        current = self.head
        while current is not None:
            print(current.data, end=" ")
            current = current.next
        print()

# Example usage
ll = LinkedList()
ll.insert_at_beginning(10)
ll.insert_at_end(20)
ll.insert_after(10, 15)
ll.delete_node(10)
ll.display()

```







```
class Contact {
public:
    Contact() {}
    Contact(int id, string name, string phone) {
        this->id = id;
        this->name = name;
        this->phone = phone;
    }
    Contact(int id, string name, string phone, string email) {
        this->id = id;
        this->name = name;
        this->phone = phone;
        this->email = email;
    }
};

class ContactList {
public:
    ContactList() {}
    ContactList(int id, string name, string phone, string email) {
        this->id = id;
        this->name = name;
        this->phone = phone;
        this->email = email;
    }
};

int main() {
    Contact c1(1, "John", "123-4567");
    Contact c2(2, "Jane", "987-6543");
    ContactList c3(3, "John", "123-4567", "john@example.com");
    ContactList c4(4, "Jane", "987-6543", "jane@example.com");

    return 0;
}
```

```
class Contact {
public:
    Contact() {}
    Contact(int id, string name, string phone) {
        this->id = id;
        this->name = name;
        this->phone = phone;
    }
};

class ContactList {
public:
    ContactList() {}
    ContactList(int id, string name, string phone, string email) {
        this->id = id;
        this->name = name;
        this->phone = phone;
        this->email = email;
    }
};

int main() {
    Contact c1(1, "John", "123-4567");
    Contact c2(2, "Jane", "987-6543");
    ContactList c3(3, "John", "123-4567", "john@example.com");
    ContactList c4(4, "Jane", "987-6543", "jane@example.com");

    return 0;
}
```



```

...
class Transaksi {
    private String no_transaksi;
    private String nama_barang;
    private double harga;
    private double jumlah;
    private double total;

    public Transaksi(String no_transaksi, String nama_barang, double harga, double jumlah, double total) {
        this.no_transaksi = no_transaksi;
        this.nama_barang = nama_barang;
        this.harga = harga;
        this.jumlah = jumlah;
        this.total = total;
    }

    public void tampil() {
        System.out.println("No Transaksi: " + no_transaksi);
        System.out.println("Nama Barang: " + nama_barang);
        System.out.println("Harga: " + harga);
        System.out.println("Jumlah: " + jumlah);
        System.out.println("Total: " + total);
    }
}

// Main Method
public class Main {
    public static void main(String[] args) {
        Transaksi t1 = new Transaksi("001", "Beras", 15000, 2, 30000);
        Transaksi t2 = new Transaksi("002", "Gula", 12000, 3, 36000);
        Transaksi t3 = new Transaksi("003", "Telur", 10000, 5, 50000);
        Transaksi t4 = new Transaksi("004", "Mie", 8000, 4, 32000);
        Transaksi t5 = new Transaksi("005", "Susu", 12000, 3, 36000);

        t1.tampil();
        t2.tampil();
        t3.tampil();
        t4.tampil();
        t5.tampil();
    }
}

```

```

...
class Transaksi {
    private String no_transaksi;
    private String nama_barang;
    private double harga;
    private double jumlah;
    private double total;

    public Transaksi(String no_transaksi, String nama_barang, double harga, double jumlah, double total) {
        this.no_transaksi = no_transaksi;
        this.nama_barang = nama_barang;
        this.harga = harga;
        this.jumlah = jumlah;
        this.total = total;
    }

    public void tampil() {
        System.out.println("No Transaksi: " + no_transaksi);
        System.out.println("Nama Barang: " + nama_barang);
        System.out.println("Harga: " + harga);
        System.out.println("Jumlah: " + jumlah);
        System.out.println("Total: " + total);
    }
}

// Main Method
public class Main {
    public static void main(String[] args) {
        Transaksi t1 = new Transaksi("001", "Beras", 15000, 2, 30000);
        Transaksi t2 = new Transaksi("002", "Gula", 12000, 3, 36000);
        Transaksi t3 = new Transaksi("003", "Telur", 10000, 5, 50000);
        Transaksi t4 = new Transaksi("004", "Mie", 8000, 4, 32000);
        Transaksi t5 = new Transaksi("005", "Susu", 12000, 3, 36000);

        t1.tampil();
        t2.tampil();
        t3.tampil();
        t4.tampil();
        t5.tampil();
    }
}

```



```
class {
  constructor() {
    this.name = "kawaii";
  }
  say() {
    console.log(`kawaii says: meow`);
  }
  meow() {
    console.log(`kawaii meows`);
  }
}

const kawaii = new class {
  constructor() {
    this.name = "kawaii";
  }
  say() {
    console.log(`kawaii says: meow`);
  }
  meow() {
    console.log(`kawaii meows`);
  }
};

kawaii.say();
kawaii.meow();
```

```
class {
  constructor() {
    this.name = "kawaii";
  }
  say() {
    console.log(`kawaii says: meow`);
  }
  meow() {
    console.log(`kawaii meows`);
  }
}

const kawaii = new class {
  constructor() {
    this.name = "kawaii";
  }
  say() {
    console.log(`kawaii says: meow`);
  }
  meow() {
    console.log(`kawaii meows`);
  }
};

kawaii.say();
kawaii.meow();
```

```

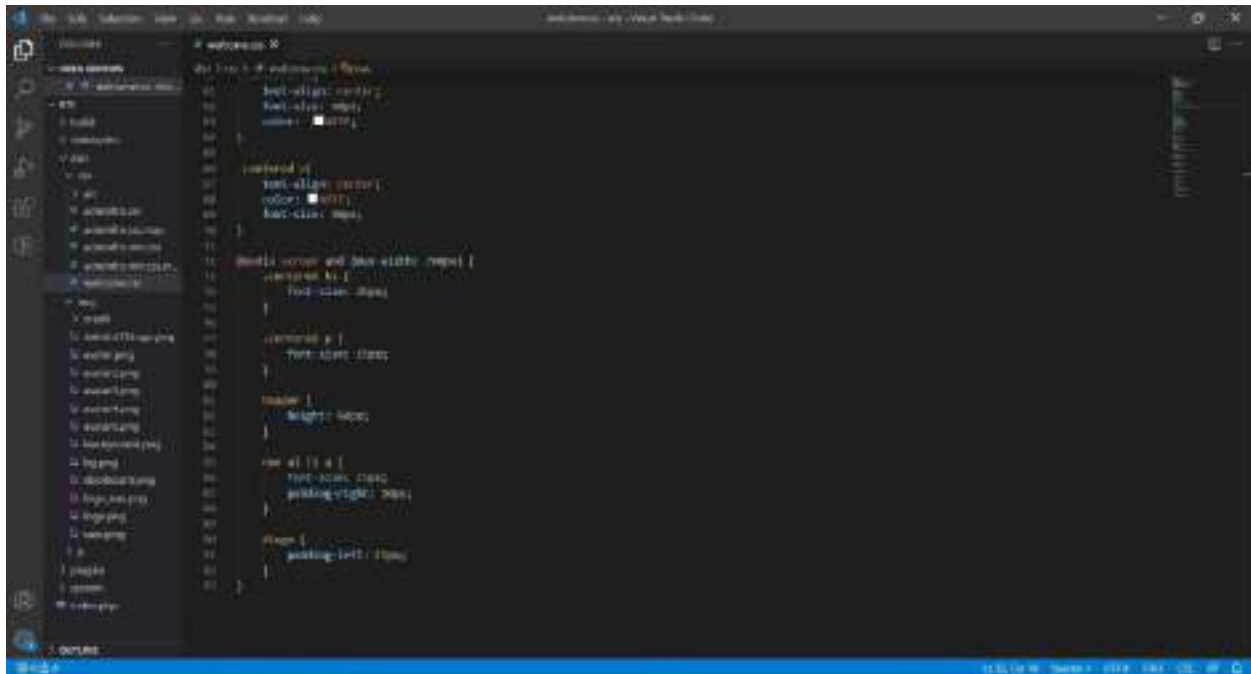
1 | body {
2 |   padding: 0;
3 |   background: url("../img/background.png");
4 |   background-size: cover;
5 |   background-position: center;
6 |   background-repeat: no-repeat;
7 |   background-attachment: fixed;
8 | }
9 |
10 | header {
11 |   background: #0070c0;
12 |   color: white;
13 |   padding: 5px;
14 |   text-align: center;
15 | }
16 |
17 | main {
18 |   padding: 10px;
19 |   min-height: 200px;
20 | }
21 |
22 | footer {
23 |   padding: 5px;
24 |   text-align: center;
25 | }
26 |
27 | .container {
28 |   width: 100%;
29 |   height: 100%;
30 | }
31 |
32 | .text {
33 |   color: white;
34 |   font-size: 24px;
35 |   font-weight: bold;
36 |   text-align: center;
37 | }
38 |
39 | .button {
40 |   padding: 10px 20px;
41 |   background-color: white;
42 |   color: #0070c0;
43 |   text-decoration: none;
44 |   border-radius: 5px;
45 |   display: inline-block;
46 | }
47 |
48 | .button:hover {
49 |   background-color: #e0e0e0;
50 | }
51 |
52 | .button:active {
53 |   background-color: #c0c0c0;
54 | }
55 |
56 | .button:focus {
57 |   outline: 2px solid #0070c0;
58 | }
59 |
60 | .button:disabled {
61 |   background-color: #a0a0a0;
62 |   color: #505050;
63 |   cursor: not-allowed;
64 | }
65 |
66 | .button:disabled:hover {
67 |   background-color: #a0a0a0;
68 | }
69 |
70 | .button:disabled:active {
71 |   background-color: #a0a0a0;
72 | }
73 |
74 | .button:disabled:focus {
75 |   outline: none;
76 | }
77 |
78 | .button:disabled:disabled {
79 |   background-color: #a0a0a0;
80 |   color: #505050;
81 |   cursor: not-allowed;
82 | }
83 |
84 | .button:disabled:disabled:hover {
85 |   background-color: #a0a0a0;
86 | }
87 |
88 | .button:disabled:disabled:active {
89 |   background-color: #a0a0a0;
90 | }
91 |
92 | .button:disabled:disabled:focus {
93 |   outline: none;
94 | }
95 |
96 | .button:disabled:disabled:disabled {
97 |   background-color: #a0a0a0;
98 |   color: #505050;
99 |   cursor: not-allowed;
100 | }

```

```

1 | body {
2 |   padding: 0;
3 |   background: url("../img/background.png");
4 |   background-size: cover;
5 |   background-position: center;
6 |   background-repeat: no-repeat;
7 |   background-attachment: fixed;
8 | }
9 |
10 | header {
11 |   background: #0070c0;
12 |   color: white;
13 |   padding: 5px;
14 |   text-align: center;
15 | }
16 |
17 | main {
18 |   padding: 10px;
19 |   min-height: 200px;
20 | }
21 |
22 | footer {
23 |   padding: 5px;
24 |   text-align: center;
25 | }
26 |
27 | .container {
28 |   width: 100%;
29 |   height: 100%;
30 | }
31 |
32 | .text {
33 |   color: white;
34 |   font-size: 24px;
35 |   font-weight: bold;
36 |   text-align: center;
37 | }
38 |
39 | .button {
40 |   padding: 10px 20px;
41 |   background-color: white;
42 |   color: #0070c0;
43 |   text-decoration: none;
44 |   border-radius: 5px;
45 |   display: inline-block;
46 | }
47 |
48 | .button:hover {
49 |   background-color: #e0e0e0;
50 | }
51 |
52 | .button:active {
53 |   background-color: #c0c0c0;
54 | }
55 |
56 | .button:focus {
57 |   outline: 2px solid #0070c0;
58 | }
59 |
60 | .button:disabled {
61 |   background-color: #a0a0a0;
62 |   color: #505050;
63 |   cursor: not-allowed;
64 | }
65 |
66 | .button:disabled:hover {
67 |   background-color: #a0a0a0;
68 | }
69 |
70 | .button:disabled:active {
71 |   background-color: #a0a0a0;
72 | }
73 |
74 | .button:disabled:focus {
75 |   outline: none;
76 | }
77 |
78 | .button:disabled:disabled {
79 |   background-color: #a0a0a0;
80 |   color: #505050;
81 |   cursor: not-allowed;
82 | }
83 |
84 | .button:disabled:disabled:hover {
85 |   background-color: #a0a0a0;
86 | }
87 |
88 | .button:disabled:disabled:active {
89 |   background-color: #a0a0a0;
90 | }
91 |
92 | .button:disabled:disabled:focus {
93 |   outline: none;
94 | }
95 |
96 | .button:disabled:disabled:disabled {
97 |   background-color: #a0a0a0;
98 |   color: #505050;
99 |   cursor: not-allowed;
100 | }

```



## DAFTAR PUSTAKA

[www.php.net](http://www.php.net). (2020-02-12)

[w3.org](http://w3.org) *World Wide Web Consortium – 1997*

McBreen, Sean (2015). *Announcing Visual Studio Code*

Flanagan, David (2011). *JavaScript The Definitive Guide (edisi ke-6<sup>th</sup>)*

[www.phpMyAdmin.net](http://www.phpMyAdmin.net). (2010-09-05)

Bryan J. Hong, *Building a Server with FreeBSD 7: A Modular Approach*. halaman 157. NO STARCH PRESS.

Arief Ramadhan (2006), *SGS: Pemr. Web Database PHP&MySQL*, halaman 70. Elex Media Komputindo.

Delisle, Marc (2010). *Mastering phpMyAdmin 3.3.x for Effective MySQL Management*.

Anderson, Wade (2020). *Evolution of Visual Studio Code Insiders*.

Krill, Paul (2018). *Whats new in Mucrosoft Visual Studio Code*.