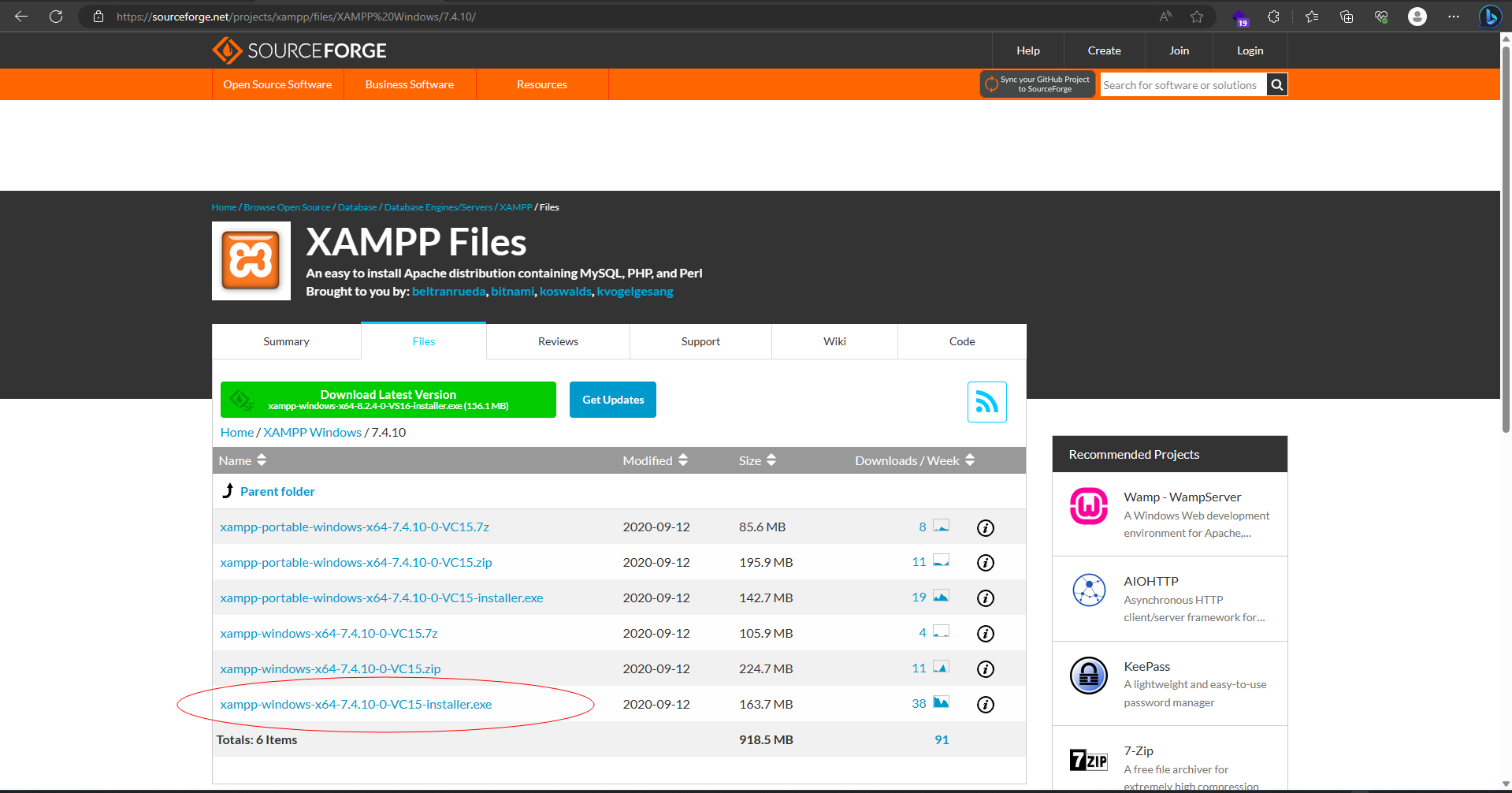
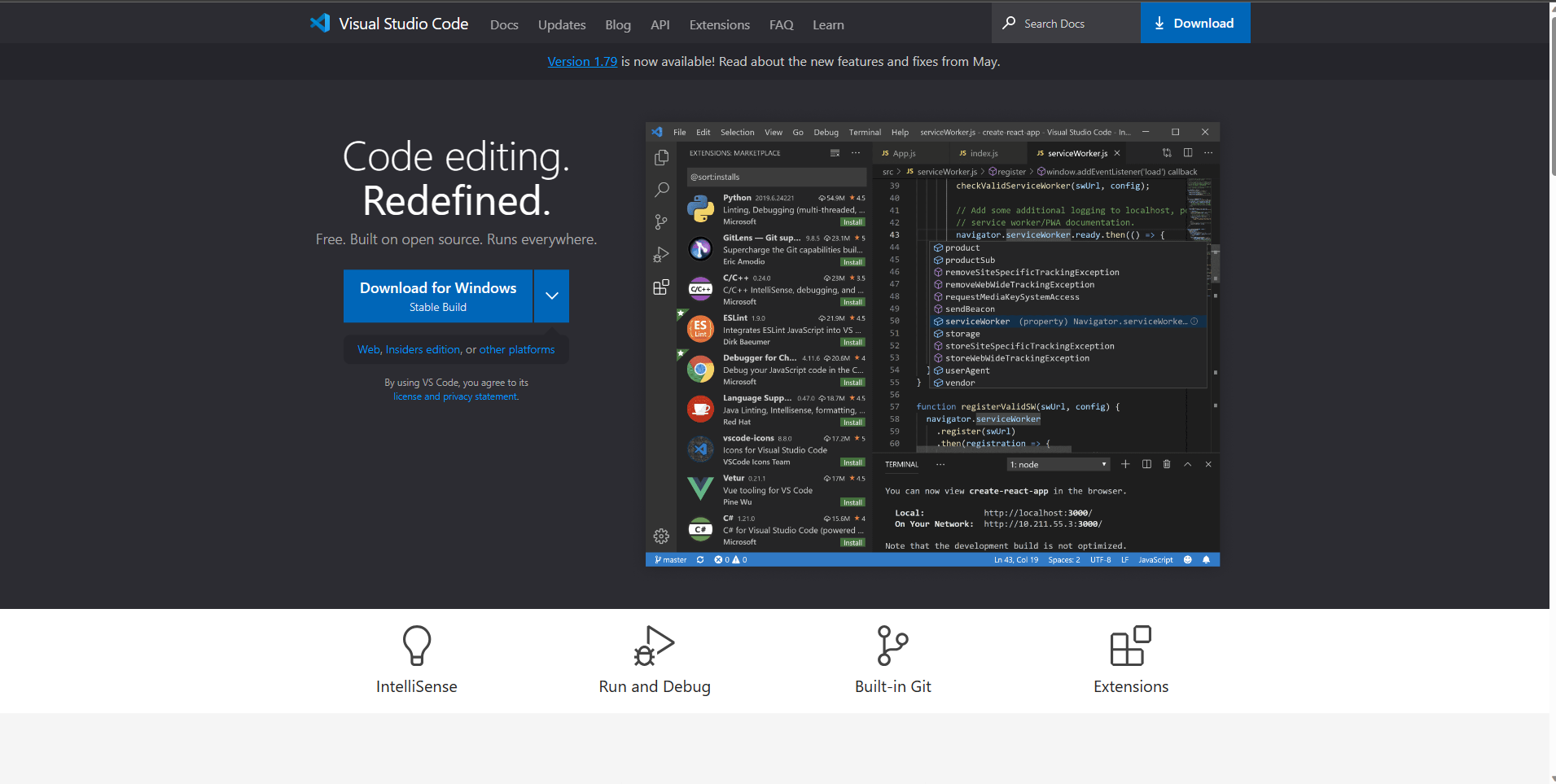
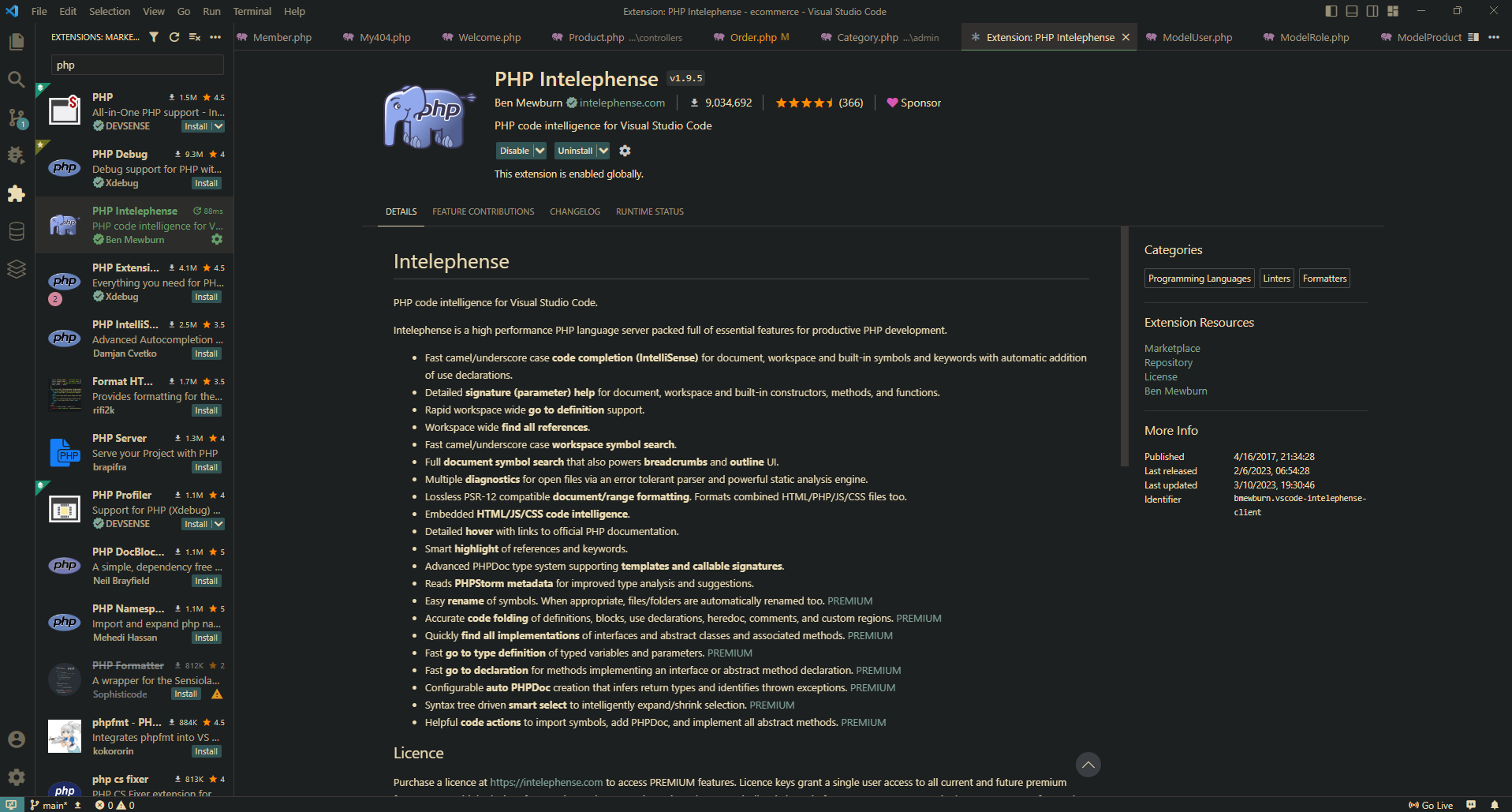
Sebelum itu download xampp 7.4.10



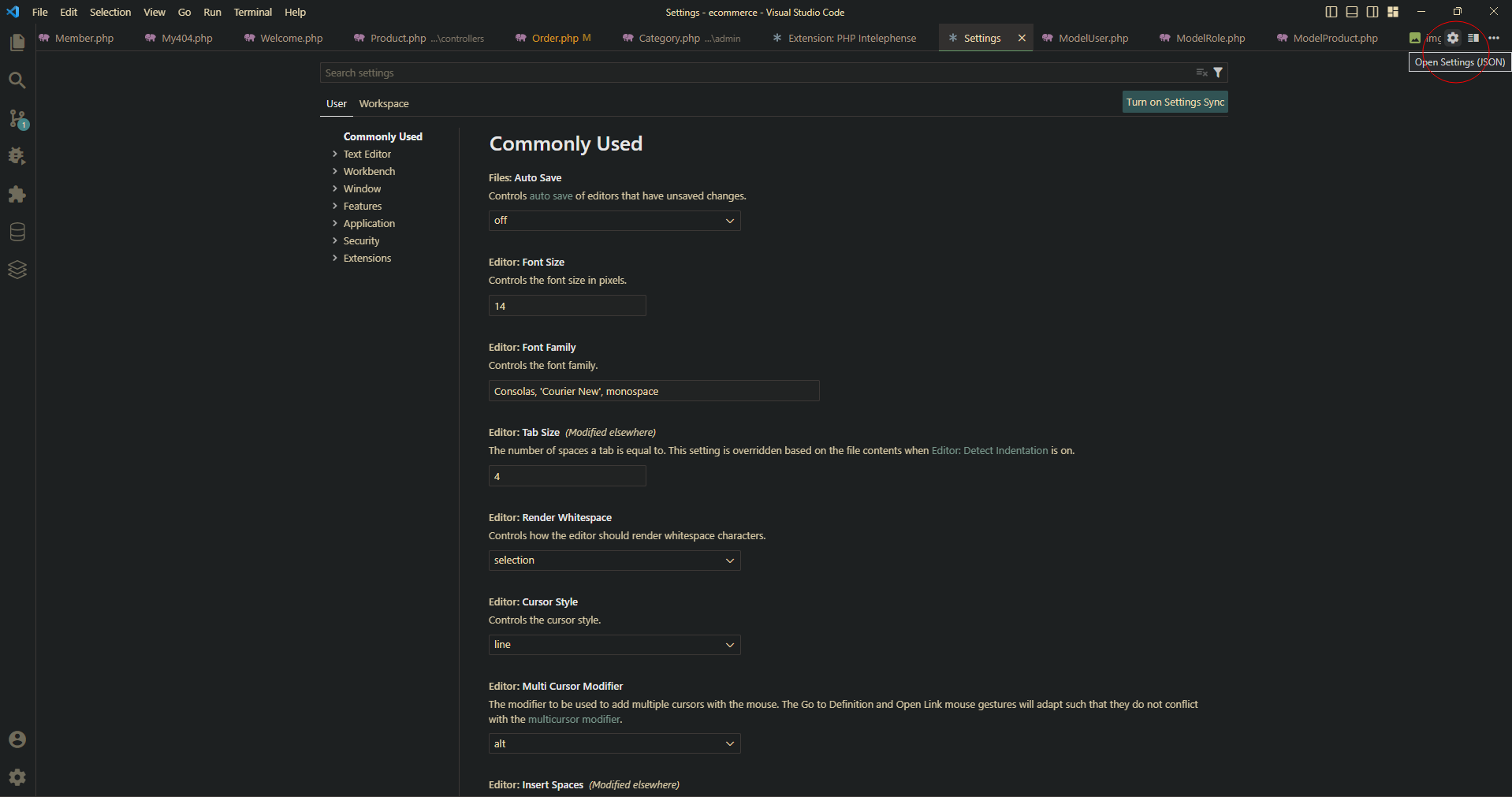
Download yang udah di tandain merah



Download vscode terlebih dahulu sebagai text editornya



Install Php Intelephense sebagai lsp biar keliatan errornya

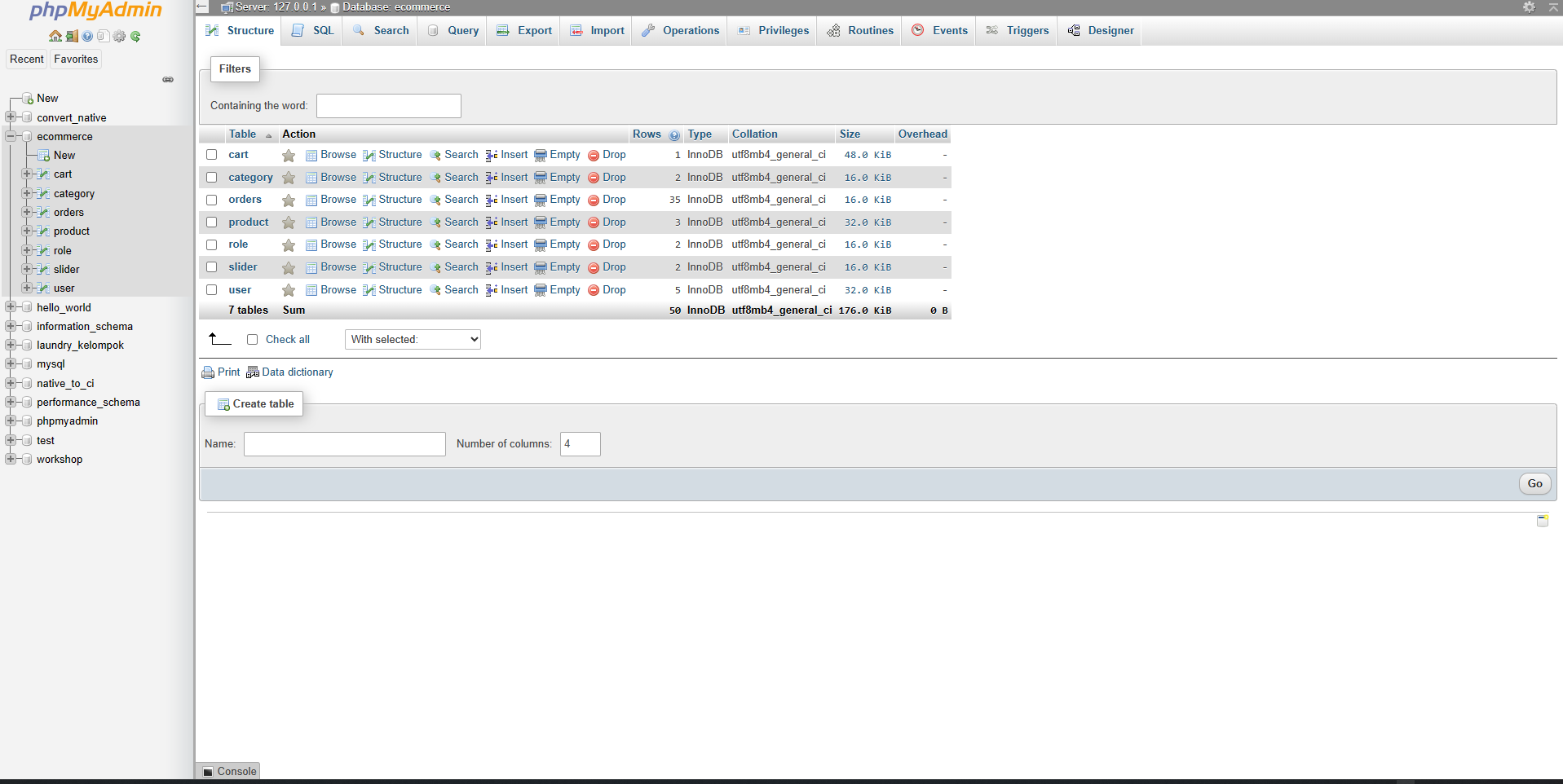


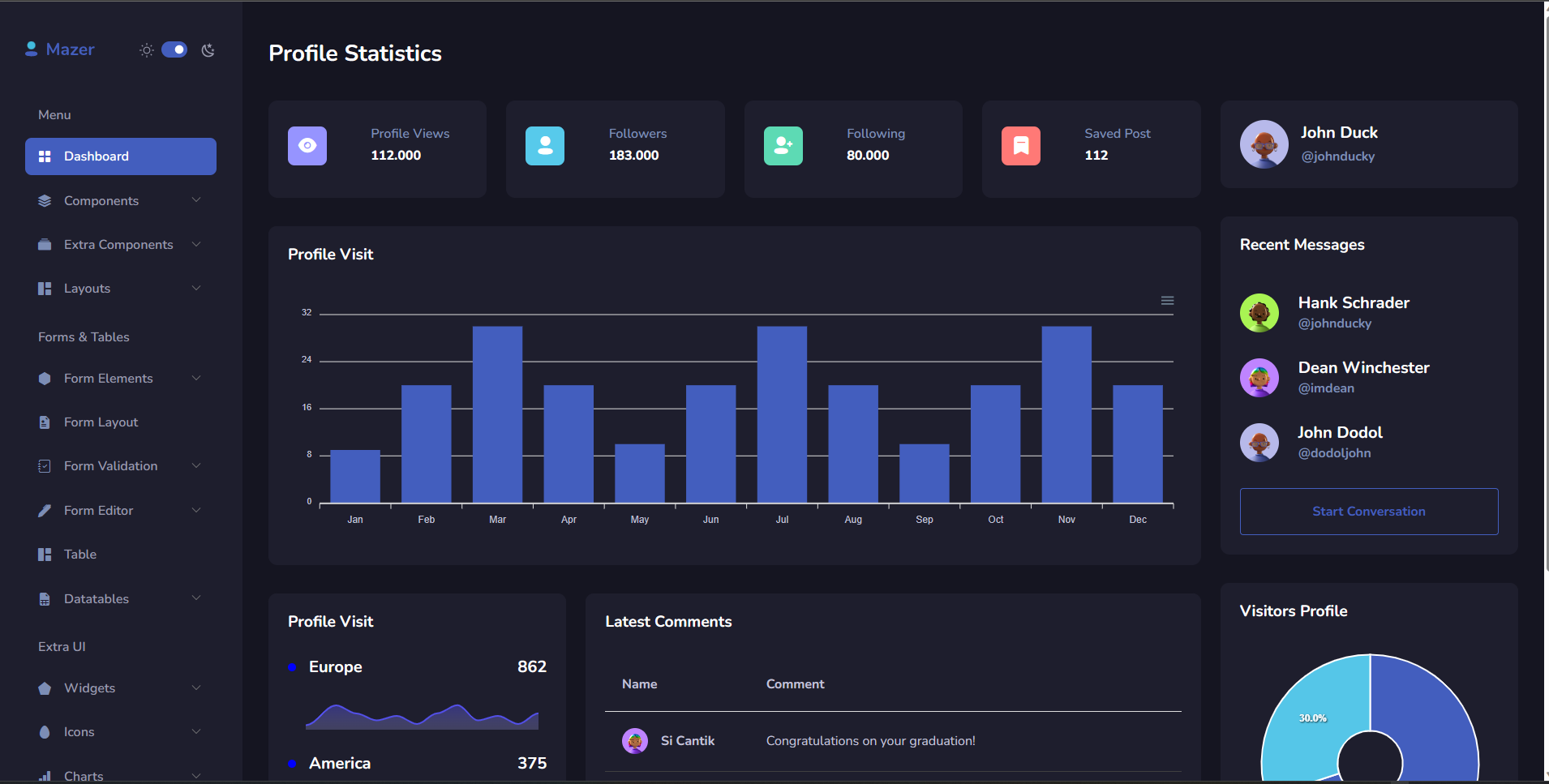
Open settings.json

"php.validate.executablePath": "C:\\xampp\\php\\php.exe",

Tambahkan path atau lokasi php

Buatkan Database ecommerce terlebih dahulu





Kita menggunakan Mazer dashboard bootstrap 5

Sebelum kita setup terlebih dahulu autoload.php

$autoload['libraries'] = array('form\_validation', 'session', 'database');

*/\**

*| -------------------------------------------------------------------*

*|  Auto-load Drivers*

*| -------------------------------------------------------------------*

*| These classes are located in system/libraries/ or in your*

*| application/libraries/ directory, but are also placed inside their*

*| own subdirectory and they extend the CI\_Driver\_Library class. They*

*| offer multiple interchangeable driver options.*

*|*

*| Prototype:*

*|*

*|   $autoload['drivers'] = array('cache');*

*|*

*| You can also supply an alternative property name to be assigned in*

*| the controller:*

*|*

*|   $autoload['drivers'] = array('cache' => 'cch');*

*|*

*\*/*

$autoload['drivers'] = array();

*/\**

*| -------------------------------------------------------------------*

*|  Auto-load Helper Files*

*| -------------------------------------------------------------------*

*| Prototype:*

*|*

*|   $autoload['helper'] = array('url', 'file');*

*\*/*

$autoload['helper'] = array('url', 'file', 'ecom');

*/\**

*| -------------------------------------------------------------------*

*|  Auto-load Config files*

*| -------------------------------------------------------------------*

*| Prototype:*

*|*

*|   $autoload['config'] = array('config1', 'config2');*

*|*

*| NOTE: This item is intended for use ONLY if you have created custom*

*| config files.  Otherwise, leave it blank.*

*|*

*\*/*

$autoload['config'] = array();

*/\**

*| -------------------------------------------------------------------*

*|  Auto-load Language files*

*| -------------------------------------------------------------------*

*| Prototype:*

*|*

*|   $autoload['language'] = array('lang1', 'lang2');*

*|*

*| NOTE: Do not include the "\_lang" part of your file.  For example*

*| "codeigniter\_lang.php" would be referenced as array('codeigniter');*

*|*

*\*/*

$autoload['language'] = array();

*/\**

*| -------------------------------------------------------------------*

*|  Auto-load Models*

*| -------------------------------------------------------------------*

*| Prototype:*

*|*

*|   $autoload['model'] = array('first\_model', 'second\_model');*

*|*

*| You can also supply an alternative model name to be assigned*

*| in the controller:*

*|*

*|   $autoload['model'] = array('first\_model' => 'first');*

*\*/*

$autoload['model'] = array('ModelAuth', 'ModelCart', 'ModelCategory', 'ModelProduct', "ModelRole", 'ModelSlider', 'ModelUser', 'ModelOrder');

**File application/config/autoload.php**

Libraries:

'form\_validation': Mengaktifkan library form\_validation untuk validasi form.

'session': Mengaktifkan library session untuk pengelolaan session.

'database': Mengaktifkan library database untuk interaksi dengan database.

Drivers:

Tidak ada driver yang dimuat secara otomatis dalam autoload. Driver-driver ini dapat dimuat secara manual di dalam controller jika diperlukan.

Helper:

'url': Memuat helper url untuk fungsi-fungsi URL.

'file': Memuat helper file untuk fungsi-fungsi file.

'ecom': Memuat helper ecom (diasumsikan sebagai helper khusus aplikasi) untuk fungsi-fungsi khusus aplikasi.

Config:

Tidak ada file konfigurasi yang dimuat secara otomatis dalam autoload. File konfigurasi kustom dapat dimuat secara manual jika diperlukan.

Language:

Tidak ada file bahasa yang dimuat secara otomatis dalam autoload. File bahasa kustom dapat dimuat secara manual jika diperlukan.

Models:

'ModelAuth': Memuat model ModelAuth.

'ModelCart': Memuat model ModelCart.

'ModelCategory': Memuat model ModelCategory.

'ModelProduct': Memuat model ModelProduct.

'ModelRole': Memuat model ModelRole.

'ModelSlider': Memuat model ModelSlider.

'ModelUser': Memuat model ModelUser.

'ModelOrder': Memuat model ModelOrder.

$db['default'] = array(

    'dsn'   => '',

    'hostname' => 'localhost',

    'username' => 'root',

    'password' => '',

    'database' => 'ecommerce',

    'dbdriver' => 'mysqli',

    'dbprefix' => '',

    'pconnect' => FALSE,

    'db\_debug' => (ENVIRONMENT !== 'production'),

    'cache\_on' => FALSE,

    'cachedir' => '',

    'char\_set' => 'utf8',

    'dbcollat' => 'utf8\_general\_ci',

    'swap\_pre' => '',

    'encrypt' => FALSE,

    'compress' => FALSE,

    'stricton' => FALSE,

    'failover' => array(),

    'save\_queries' => TRUE

);

**File application/config/database.php**

'dsn': DSN (Data Source Name) untuk koneksi database. Dalam kasus ini, tidak ada DSN yang digunakan.

'hostname': Nama host atau alamat IP server database. Dalam contoh ini, digunakan localhost.

'username': Nama pengguna (username) untuk mengakses database. Dalam contoh ini, digunakan 'root'.

'password': Kata sandi (password) untuk mengakses database. Dalam contoh ini, tidak ada kata sandi yang digunakan.

'database': Nama database yang akan digunakan. Dalam contoh ini, digunakan 'ecommerce'.

'dbdriver': Driver database yang digunakan. Dalam contoh ini, digunakan driver mysqli.

'dbprefix': Awalan tabel database. Dalam contoh ini, tidak ada awalan yang digunakan.

'pconnect': Menggunakan persistent connection atau tidak. Dalam contoh ini, pengaturannya adalah FALSE (tidak menggunakan persistent connection).

'db\_debug': Mengaktifkan atau menonaktifkan mode debug database. Nilainya bergantung pada nilai ENVIRONMENT yang didefinisikan di environment file. Jika ENVIRONMENT bukan 'production', maka db\_debug akan diaktifkan.

'cache\_on': Mengaktifkan atau menonaktifkan fitur caching query. Dalam contoh ini, pengaturannya adalah FALSE (tidak menggunakan caching).

'cachedir': Direktori untuk menyimpan file cache. Dalam contoh ini, tidak ada direktori cache yang digunakan.

'char\_set': Set karakter untuk koneksi database. Dalam contoh ini, menggunakan set karakter utf8.

'dbcollat': Kolasi karakter untuk koneksi database. Dalam contoh ini, menggunakan kolasi utf8\_general\_ci.

'swap\_pre': Awalan placeholder yang digunakan untuk swap. Dalam contoh ini, tidak ada awalan swap yang digunakan.

'encrypt': Mengenkripsi atau tidak koneksi database. Dalam contoh ini, pengaturannya adalah FALSE (tidak menggunakan enkripsi).

'compress': Mengkompres atau tidak koneksi database. Dalam contoh ini, pengaturannya adalah FALSE (tidak menggunakan kompresi).

'stricton': Mengaktifkan atau menonaktifkan mode strict SQL. Dalam contoh ini, pengaturannya adalah FALSE (mode strict SQL dinonaktifkan).

'failover': Daftar server database failover. Dalam contoh ini, tidak ada server failover yang ditentukan.

'save\_queries': Mengaktifkan atau menonaktifkan penyimpanan query yang dieksekusi. Dalam contoh ini, pengaturannya adalah TRUE (penyimpanan query diaktifkan).

$root = (isset($\_SERVER['HTTPS']) ? "https://" : "http://").$\_SERVER['HTTP\_HOST'];

$root .= str\_replace(basename($\_SERVER['SCRIPT\_NAME']), '',

$\_SERVER['SCRIPT\_NAME']);

$config['base\_url'] = $root;

*/\**

*|--------------------------------------------------------------------------*

*| Index File*

*|--------------------------------------------------------------------------*

*|*

*| Typically this will be your index.php file, unless you've renamed it to*

*| something else. If you are using mod\_rewrite to remove the page set this*

*| variable so that it is blank.*

*|*

*\*/*

$config['index\_page'] = '';

**File application/config/config.php**

Berikut adalah penjelasan mengenai setiap parameter konfigurasi pada kode tersebut:

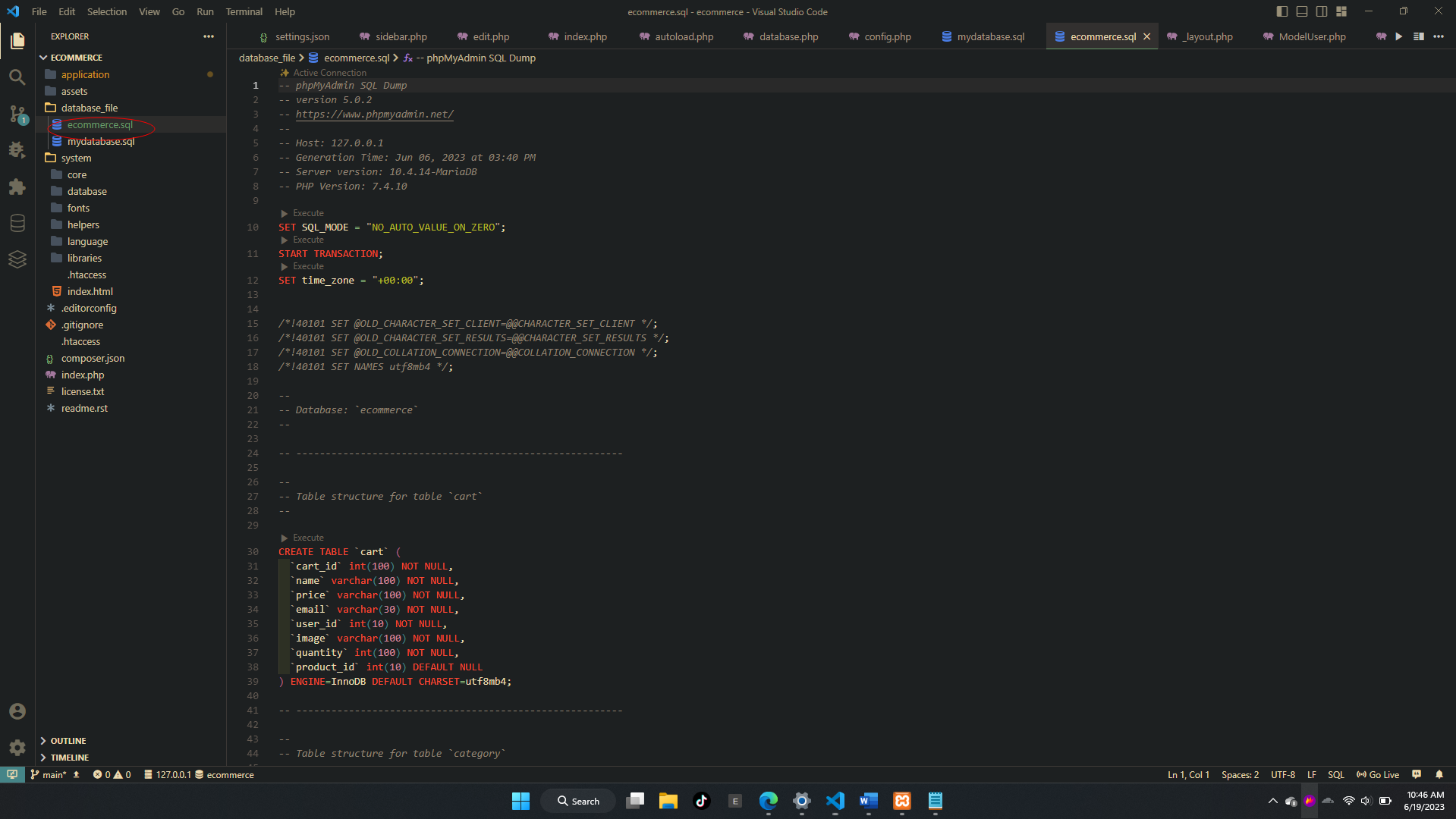
$root: Variabel yang digunakan untuk menyimpan URL dasar (base URL) aplikasi. Nilainya akan tergantung pada apakah protokol HTTPS digunakan atau tidak. Jika protokol HTTPS digunakan, maka nilai $root akan diawali dengan "https://", jika tidak, maka akan diawali dengan "http://". Selanjutnya, URL dasar akan ditambahkan dengan direktori aplikasi (berdasarkan $\_SERVER['SCRIPT\_NAME']) untuk membentuk URL dasar yang lengkap.

$config['base\_url']: Konfigurasi yang digunakan untuk menyimpan URL dasar (base URL) aplikasi. Nilainya diambil dari variabel $root yang telah dibentuk sebelumnya. URL dasar ini akan digunakan oleh CodeIgniter untuk menghasilkan URL lengkap pada aplikasi.

Misalnya, jika aplikasi berjalan di http://localhost/myapp/, maka $root akan menjadi http://localhost/myapp/ dan $config['base\_url'] akan diatur dengan nilai tersebut.

$config['index\_page']: Konfigurasi yang digunakan untuk mengatur nama file index. Secara default, nilai ini diatur kosong (''). Jika menggunakan mod\_rewrite untuk menghapus nama file index.php dari URL, maka nilai ini harus diatur sebagai kosong. Jika tidak menggunakan mod\_rewrite, maka nilai ini akan berisi nama file index.php.

Dengan konfigurasi ini, CodeIgniter akan menggunakan URL dasar yang ditentukan oleh $config['base\_url'] untuk menghasilkan URL yang lengkap dalam aplikasi, termasuk mengabaikan nama file index.php jika diperlukan (tergantung pada pengaturan $config['index\_page']).



Kalian import saja databasenya

Fungsinya

Tabel role:

Kolom role\_id: Kolom yang menyimpan ID peran (role) dengan tipe data int.

Kolom role: Kolom yang menyimpan nama peran (role) dengan tipe data varchar.

Kolom created\_at: Kolom yang menyimpan waktu pembuatan dengan tipe data timestamp.

Kolom updated\_at: Kolom yang menyimpan waktu terakhir diperbarui dengan tipe data timestamp.

PRIMARY KEY: Kolom role\_id dijadikan sebagai primary key.

Tabel user:

Kolom user\_id: Kolom yang menyimpan ID pengguna dengan tipe data int.

Kolom firstname: Kolom yang menyimpan nama depan pengguna dengan tipe data varchar.

Kolom lastname: Kolom yang menyimpan nama belakang pengguna dengan tipe data varchar.

Kolom email: Kolom yang menyimpan alamat email pengguna dengan tipe data varchar.

Kolom password: Kolom yang menyimpan kata sandi pengguna dengan tipe data varchar.

Kolom role\_id: Kolom yang menyimpan ID peran (role) pengguna dengan tipe data int.

Kolom created\_at: Kolom yang menyimpan waktu pembuatan dengan tipe data timestamp.

Kolom updated\_at: Kolom yang menyimpan waktu terakhir diperbarui dengan tipe data timestamp.

FOREIGN KEY: Kolom role\_id dijadikan sebagai foreign key yang merujuk ke kolom role\_id pada tabel role.

PRIMARY KEY: Kolom user\_id dijadikan sebagai primary key.

Tabel category:

Kolom category\_id: Kolom yang menyimpan ID kategori dengan tipe data int.

Kolom nama\_kategori: Kolom yang menyimpan nama kategori dengan tipe data varchar.

Kolom slug\_category: Kolom yang menyimpan slug kategori dengan tipe data varchar.

Kolom image\_category: Kolom yang menyimpan nama file gambar kategori dengan tipe data varchar.

Kolom created\_at: Kolom yang menyimpan waktu pembuatan dengan tipe data timestamp.

Kolom updated\_at: Kolom yang menyimpan waktu terakhir diperbarui dengan tipe data timestamp.

PRIMARY KEY: Kolom category\_id dijadikan sebagai primary key.

Tabel product:

Kolom product\_id: Kolom yang menyimpan ID produk dengan tipe data int.

Kolom name: Kolom yang menyimpan nama produk dengan tipe data varchar.

Kolom slug\_product: Kolom yang menyimpan slug produk dengan tipe data varchar.

Kolom image\_product: Kolom yang menyimpan nama file gambar produk dengan tipe data varchar.

Kolom category\_id: Kolom yang menyimpan ID kategori produk dengan tipe data int.

Kolom description: Kolom yang menyimpan deskripsi produk dengan tipe data varchar.

Kolom price: Kolom yang menyimpan harga produk dengan tipe data int.

Kolom countInStock: Kolom yang menyimpan jumlah stok produk dengan tipe data int.

Kolom created\_at: Kolom yang menyimpan waktu pembuatan dengan tipe data datetime.

Kolom updated\_at: Kolom yang menyimpan waktu terakhir diperbarui dengan tipe data datetime.

FOREIGN KEY: Kolom category\_id dijadikan sebagai foreign key yang merujuk ke kolom category\_id pada tabel category.

PRIMARY KEY: Kolom product\_id dijadikan sebagai primary key.

Tabel cart:

Kolom cart\_id: Kolom yang menyimpan ID keranjang dengan tipe data int.

Kolom name: Kolom yang menyimpan nama produk dalam keranjang dengan tipe data varchar.

Kolom price: Kolom yang menyimpan harga produk dalam keranjang dengan tipe data varchar.

Kolom email: Kolom yang menyimpan alamat email pengguna dengan tipe data varchar.

Kolom user\_id: Kolom yang menyimpan ID pengguna dengan tipe data int.

Kolom image: Kolom yang menyimpan nama file gambar produk dalam keranjang dengan tipe data varchar.

Kolom quantity: Kolom yang menyimpan jumlah produk dalam keranjang dengan tipe data int.

FOREIGN KEY: Kolom user\_id dijadikan sebagai foreign key yang merujuk ke kolom user\_id pada tabel user.

PRIMARY KEY: Kolom cart\_id dijadikan sebagai primary key.

Tabel orders:

Kolom order\_id: Kolom yang menyimpan ID pesanan dengan tipe data varchar.

Kolom email: Kolom yang menyimpan alamat email pengguna dengan tipe data varchar.

Kolom postal\_code: Kolom yang menyimpan kode pos pengguna dengan tipe data varchar.

Kolom user\_id: Kolom yang menyimpan ID pengguna dengan tipe data int.

Kolom country\_code: Kolom yang menyimpan kode negara pengguna dengan tipe data varchar.

Kolom total\_price: Kolom yang menyimpan total harga pesanan dengan tipe data varchar.

Kolom total\_product: Kolom yang menyimpan total produk dalam pesanan dengan tipe data varchar.

Kolom created\_at: Kolom yang menyimpan waktu pembuatan dengan tipe data datetime.

Kolom updated\_at: Kolom yang menyimpan waktu terakhir diperbarui dengan tipe data datetime.

FOREIGN KEY: Kolom user\_id dijadikan sebagai foreign key yang merujuk ke kolom user\_id pada tabel user.

PRIMARY KEY: Kolom order\_id dijadikan sebagai primary key.

Tabel slider:

Kolom slider\_id: Kolom yang menyimpan ID slider dengan tipe data int.

Kolom nama: Kolom yang menyimpan nama slider dengan tipe data varchar.

Kolom image: Kolom yang menyimpan nama file gambar slider dengan tipe data varchar.

Kolom created\_at: Kolom yang menyimpan waktu pembuatan dengan tipe data timestamp.

Kolom updated\_at: Kolom yang menyimpan waktu terakhir diperbarui dengan tipe data timestamp.

PRIMARY KEY: Kolom slider\_id dijadikan sebagai primary key.

<!**DOCTYPE** html>

<**html** lang="en">

<**head**>

    <**meta** charset="UTF-8" />

    <**meta** name="viewport" content="width=device-width, initial-scale=1.0" />

    <**title**><?php echo $title; ?></**title**>

    <**link** rel="shortcut icon" href="<?= base\_url("assets/compiled/svg/favicon.svg"); ?>" type="image/x-icon" />

    <**link** rel="shortcut icon" href="data:image/png;base64," type="image/png" />

    <**link** rel="stylesheet" href="<?= base\_url("assets/compiled/css/app.css"); ?>" />

    <**link** rel="stylesheet" href="<?= base\_url("assets/compiled/css/app-dark.css"); ?>" />

    <**link** rel="stylesheet" href="<?= base\_url("assets/compiled/css/iconly.css"); ?>" />

    <**link** rel="stylesheet" href="<?= base\_url("assets/extensions/@icon/dripicons/dripicons.css"); ?>" />

    <**link** rel="stylesheet" href="<?= base\_url("assets/compiled/css/ui-icons-dripicons.css"); ?>" />

    <**link** rel="stylesheet" href="<?= base\_url("assets/extensions/simple-datatables/style.css"); ?>" />

    <**link** rel="stylesheet" href="<?= base\_url("assets/extensions/simple-datatables/style.css"); ?>" />

    <**link** rel="stylesheet" href="<?= base\_url("assets/compiled/css/table-datatable.css"); ?>" />

</**head**>

<**body**>

    <**div** id="app">

        <?php $this->load->view('admin/components/sidebar'); ?>

        <**div** id="main">

            <**header** class="mb-3">

                <**a** href="#" class="burger-btn d-block d-xl-none">

                    <**i** class="bi bi-justify fs-3"></**i**>

                </**a**>

            </**header**>

            <?php $this->load->view($subview); ?>

        </**div**>

        <?php $this->load->view('admin/user/modal'); ?>

        <?php $this->load->view('admin/category/modal'); ?>

        <?php $this->load->view('admin/product/modal'); ?>

        <?php $this->load->view('admin/role/modal'); ?>

        <?php $this->load->view('admin/slider/modal'); ?>

    </**div**>

    <**script** src="<?= base\_url("assets/static/js/initTheme.js"); ?>"></**script**>

    <**script** src="<?= base\_url("assets/static/js/components/dark.js"); ?>"></**script**>

    <**script** src="<?= base\_url("assets/extensions/perfect-scrollbar/perfect-scrollbar.min.js"); ?>"></**script**>

    <**script** src="<?= base\_url("assets/extensions/apexcharts/apexcharts.min.js"); ?>"></**script**>

    <**script** src="<?= base\_url("assets/compiled/js/app.js"); ?>"></**script**>

    <**script** src="<?= base\_url("assets/static/js/pages/dashboard.js"); ?>"></**script**>

    <**script** src="<?= base\_url("assets/extensions/simple-datatables/umd/simple-datatables.js"); ?>"></**script**>

    <**script** src="<?= base\_url("assets/static/js/pages/simple-datatables.js"); ?>"></**script**>

    <**script** src="<?= base\_url("assets/extensions/chart.js/chart.umd.js") ?>"></**script**>

</**body**>

</**html**>

**File application/views/admin/\_layout.php**

1. Mendefinisikan judul halaman: <title><?php echo $title; ?></title>
2. Judul halaman akan ditampilkan pada tab browser.
3. Mengatur favicon: <link rel="shortcut icon" href="<?= base\_url("assets/compiled/svg/favicon.svg"); ?>" type="image/x-icon" />
4. Mengatur ikon yang ditampilkan di sebelah judul halaman pada tab browser.
5. Menghubungkan file CSS: <link rel="stylesheet" href="<?= base\_url("assets/compiled/css/app.css"); ?>" />
6. Menghubungkan file CSS eksternal untuk mengatur gaya halaman.
7. Memuat komponen sidebar: <?php $this->load->view('admin/components/sidebar'); ?>
8. Memuat komponen sidebar dari file terpisah.
9. Memuat konten utama: <?php $this->load->view($subview); ?>
10. Memuat konten utama dari file terpisah yang ditentukan oleh variabel $subview.
11. Memuat file JavaScript: <script src="<?= base\_url("assets/static/js/initTheme.js"); ?>"></script>
12. Memuat file JavaScript eksternal untuk mengatur perilaku halaman.
13. Memuat file JavaScript yang khusus untuk halaman dashboard: <script src="<?= base\_url("assets/static/js/pages/dashboard.js"); ?>"></script>
14. Memuat file JavaScript yang spesifik untuk halaman dashboard.
15. Memuat file JavaScript untuk menggunakan plugin Simple Datatables: <script src="<?= base\_url("assets/extensions/simple-datatables/umd/simple-datatables.js"); ?>"></script>
16. Memuat file JavaScript yang diperlukan untuk menggunakan plugin Simple Datatables.
17. Memuat file JavaScript untuk menggunakan plugin Chart.js: <script src="<?= base\_url("assets/extensions/chart.js/chart.umd.js") ?>"></script>
18. Memuat file JavaScript yang diperlukan untuk menggunakan plugin Chart.js.

<**div** id="sidebar">

    <**div** class="sidebar-wrapper active">

        <**div** class="sidebar-header position-relative">

            <**div** class="d-flex justify-content-between align-items-center">

                <**div** class="logo">

                    <**a** href="index.html"><**img** src="<?= base\_url("assets/compiled/svg/logo.svg"); ?>" alt="Logo" srcset="" /></**a**>

                </**div**>

                <**div** class="theme-toggle d-flex gap-2 align-items-center mt-2">

                    <**svg** xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" aria-hidden="true" role="img" class="iconify iconify--system-uicons" width="20" height="20" preserveAspectRatio="xMidYMid meet" viewBox="0 0 21 21">

                        <**g** fill="none" fill-rule="evenodd" stroke="currentColor" stroke-linecap="round" stroke-linejoin="round">

                            <**path** d="M10.5 14.5c2.219 0 4-1.763 4-3.982a4.003 4.003 0 0 0-4-4.018c-2.219 0-4 1.781-4 4c0 2.219 1.781 4 4 4zM4.136 4.136L5.55 5.55m9.9 9.9l1.414 1.414M1.5 10.5h2m14 0h2M4.135 16.863L5.55 15.45m9.899-9.9l1.414-1.415M10.5 19.5v-2m0-14v-2" opacity=".3"></**path**>

                            <**g** transform="translate(-210 -1)">

                                <**path** d="M220.5 2.5v2m6.5.5l-1.5 1.5"></**path**>

                                <**circle** cx="220.5" cy="11.5" r="4"></**circle**>

                                <**path** d="m214 5l1.5 1.5m5 14v-2m6.5-.5l-1.5-1.5M214 18l1.5-1.5m-4-5h2m14 0h2"></**path**>

                            </**g**>

                        </**g**>

                    </**svg**>

                    <**div** class="form-check form-switch fs-6">

                        <**input** class="form-check-input me-0" type="checkbox" id="toggle-dark" style="cursor: pointer" />

                        <**label** class="form-check-label"></**label**>

                    </**div**>

                    <**svg** xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" aria-hidden="true" role="img" class="iconify iconify--mdi" width="20" height="20" preserveAspectRatio="xMidYMid meet" viewBox="0 0 24 24">

                        <**path** fill="currentColor" d="m17.75 4.09l-2.53 1.94l.91 3.06l-2.63-1.81l-2.63 1.81l.91-3.06l-2.53-1.94L12.44 4l1.06-3l1.06 3l3.19.09m3.5 6.91l-1.64 1.25l.59 1.98l-1.7-1.17l-1.7 1.17l.59-1.98L15.75 11l2.06-.05L18.5 9l.69 1.95l2.06.05m-2.28 4.95c.83-.08 1.72 1.1 1.19 1.85c-.32.45-.66.87-1.08 1.27C15.17 23 8.84 23 4.94 19.07c-3.91-3.9-3.91-10.24 0-14.14c.4-.4.82-.76 1.27-1.08c.75-.53 1.93.36 1.85 1.19c-.27 2.86.69 5.83 2.89 8.02a9.96 9.96 0 0 0 8.02 2.89m-1.64 2.02a12.08 12.08 0 0 1-7.8-3.47c-2.17-2.19-3.33-5-3.49-7.82c-2.81 3.14-2.7 7.96.31 10.98c3.02 3.01 7.84 3.12 10.98.31Z"></**path**>

                    </**svg**>

                </**div**>

                <**div** class="sidebar-toggler x">

                    <**a** href="#" class="sidebar-hide d-xl-none d-block"><**i** class="bi bi-x bi-middle"></**i**></**a**>

                </**div**>

            </**div**>

        </**div**>

        <**div** class="sidebar-menu">

            <**ul** class="menu">

                <**li** class="sidebar-title">Menu</**li**>

                <**li** class="sidebar-item active">

                    <**a** href="<?= base\_url('admin/dashboard'); ?>" class="sidebar-link">

                        <**i** class="bi bi-grid-fill"></**i**>

                        <**span**>Dashboard</**span**>

                    </**a**>

                </**li**>

                <**li** class="sidebar-item">

                    <**a** href="<?= base\_url('admin/user'); ?>" class="sidebar-link">

                        <**i** class="bi bi-person-fill"></**i**>

                        <**span**>User</**span**>

                    </**a**>

                </**li**>

                <**li** class="sidebar-item ">

                    <**a** href="<?= base\_url('admin/role'); ?>" class="sidebar-link">

                        <**i** class="bi bi-hammer"></**i**>

                        <**span**>Role</**span**>

                    </**a**>

                </**li**>

                <**li** class="sidebar-item ">

                    <**a** href="<?= base\_url('admin/category'); ?>" class="sidebar-link">

                        <**i** class="bi bi-hammer"></**i**>

                        <**span**>Category</**span**>

                    </**a**>

                </**li**>

                <**li** class="sidebar-item ">

                    <**a** href="<?= base\_url('admin/product'); ?>" class="sidebar-link">

                        <**i** class="bi bi-hammer"></**i**>

                        <**span**>Product</**span**>

                    </**a**>

                </**li**>

                <**li** class="sidebar-item ">

                    <**a** href="<?= base\_url('admin/order'); ?>" class="sidebar-link">

                        <**i** class="bi bi-hammer"></**i**>

                        <**span**>order</**span**>

                    </**a**>

                </**li**>

                <**li** class="sidebar-item ">

                    <**a** href="<?= base\_url('admin/slider'); ?>" class="sidebar-link">

                        <**i** class="bi bi-hammer"></**i**>

                        <**span**>Slider</**span**>

                    </**a**>

                </**li**>

                <**li** class="sidebar-title">Logout</**li**>

                <**li** class="sidebar-item ">

                    <**a** href="<?= base\_url('auth/logout'); ?>" class="sidebar-link">

                        <**i** class="bi bi-hammer"></**i**>

                        <**span**>Logout</**span**>

                    </**a**>

                </**li**>

            </**ul**>

        </**div**>

    </**div**>

</**div**>

**File application/views/admin/components/sidebar.php**

Fungsinya adalah untuk menyediakan navigasi yang mudah bagi pengguna agar dapat dengan cepat mengakses berbagai fitur atau halaman yang ada dalam aplikasi atau situs web tersebut. Dalam contoh kode yang Anda berikan, terdapat beberapa menu seperti "Dashboard", "User", "Role", "Category", "Product", "Order", "Slider", dan "Logout". Setiap menu tersebut memiliki tautan yang mengarah ke halaman-halaman terkait.

**File application/models/Category.php**

class ModelCategory extends CI\_Model

{

    public function \_\_construct()

    {

        parent::\_\_construct();

    }

    public function findAll()

    {

        try {

            $sql = "SELECT \*FROM category";

            $query = $this->db->query($sql);

            $result = $query->result();

            return $result;

        } catch (Exception $e) {

            echo $e->getMessage();

        }

    }

    public function createCategory($nama, $image, $slug)

    {

        try {

            $data = [

                'nama\_kategori' => $nama,

                'image\_category' => $image,

                'slug\_category' => $slug,

            ];

            $this->db->insert('category', $data);

            return true;

        } catch (Exception $e) {

            echo $e->getMessage();

        }

    }

    public function findById($id)

    {

        try {

            $sql = "SELECT \* FROM category WHERE category\_id = ?";

            $query = $this->db->query($sql, [$id]);

            $result = $query->result();

            return $result;

        } catch (Exception $e) {

            echo $e->getMessage();

            return null;

        }

    }

    public function updateCategory($id, $nama, $image, $slug)

    {

        $data = [

            'nama\_kategori' => $nama,

            'image\_category' => $image,

            'slug\_category' => $slug

        ];

        $this->db->where('category\_id', $id);

        $this->db->update('category', $data);

        return true;

    }

    public function findByNameCategory($name)

    {

        try {

            $sql = "SELECT \* FROM category WHERE name = ?";

            $query = $this->db->query($sql, [$name]);

            $result = $query->result();

            return $result;

        } catch (Exception $e) {

            echo $e->getMessage();

        }

    }

    public function deleteById($id)

    {

        try {

            $this->db->where('category\_id', $id);

            $this->db->delete('category');

            return true;

        } catch (Exception $e) {

            echo $e->getMessage();

            return false;

        }

    }

    public function findBySlug($slug)

    {

        try {

            $this->db->select('category.\*, product.\*');

            $this->db->from('category');

            $this->db->join('product', 'category.category\_id = product.category\_id');

            $this->db->where('category.slug\_category', $slug);

            $query = $this->db->get();

            $result = $query->result();

            if ($result) {

                return $result;

            } else {

                return null;

            }

        } catch (Exception $e) {

            echo $e->getMessage();

            return null;

        }

    }

    public function getImage($category\_id)

    {

        $this->db->select('image\_category');

        $this->db->from('category');

        $this->db->where('category\_id', $category\_id);

        $query = $this->db->get();

        if ($query->num\_rows() > 0) {

            $row = $query->row();

            return $row->image\_category;

        } else {

            return null;

        }

    }

}

Fungsi findAll(): Fungsi ini digunakan untuk mendapatkan semua data kategori dari tabel "category" dalam database. SQL query dijalankan menggunakan method query() dari objek database ($this->db) dan hasilnya dikembalikan dalam bentuk array objek.

Fungsi createCategory($nama, $image, $slug): Fungsi ini digunakan untuk membuat data kategori baru dalam database. Parameter yang diterima adalah nama kategori, nama file gambar, dan slug. Data kategori baru disimpan dalam array $data dan kemudian diinsert ke tabel "category" menggunakan method insert() dari objek database.

Fungsi findById($id): Fungsi ini digunakan untuk mencari data kategori berdasarkan ID yang diberikan. SQL query dieksekusi dengan menggunakan method query() dengan menggunakan parameter binding untuk ID. Hasil query dikembalikan dalam bentuk array objek.

Fungsi updateCategory($id, $nama, $image, $slug): Fungsi ini digunakan untuk memperbarui data kategori berdasarkan ID. Data yang akan diperbarui, yaitu nama kategori, nama file gambar, dan slug, disimpan dalam array $data. Kemudian, method update() digunakan untuk memperbarui data kategori dalam tabel "category" berdasarkan ID yang diberikan.

Fungsi findByNameCategory($name): Fungsi ini digunakan untuk mencari data kategori berdasarkan nama kategori yang diberikan. SQL query dieksekusi dengan menggunakan method query() dengan menggunakan parameter binding untuk nama kategori. Hasil query dikembalikan dalam bentuk array objek.

Fungsi deleteById($id): Fungsi ini digunakan untuk menghapus data kategori berdasarkan ID yang diberikan. Method delete() digunakan untuk menghapus data kategori dalam tabel "category" berdasarkan ID yang diberikan.

Fungsi findBySlug($slug): Fungsi ini digunakan untuk mencari data kategori berdasarkan slug yang diberikan. SQL query dieksekusi menggunakan method select(), from(), join(), dan where() untuk melakukan join antara tabel "category" dan "product" berdasarkan ID kategori. Hasil query dikembalikan dalam bentuk array objek.

Fungsi getImage($category\_id): Fungsi ini digunakan untuk mendapatkan nama file gambar kategori berdasarkan ID kategori yang diberikan. SQL query dieksekusi menggunakan method select(), from(), dan where() untuk mendapatkan nama file gambar dari tabel "category" berdasarkan ID kategori. Nama file gambar dikembalikan sebagai hasil query.

**File application/views/admin/controllers/Category.php**

public function \_\_construct()

    {

        parent::\_\_construct();

        cek\_login();

        cek\_user();

    }

Konstruktor \_\_construct(): Konstruktor ini dipanggil saat objek kontroler dibuat. Pada contoh ini, konstruktor ini memanggil fungsi cek\_login() dan cek\_user() dari kontroler yang sama, yang mungkin digunakan untuk memeriksa status login dan hak akses pengguna sebelum menjalankan fungsi index().

**File application/helpers/ecom\_helper.php**

function cek\_login()

{

    $ci = get\_instance();

    if (!$ci->session->userdata('email')) {

        $ci->session->set\_flashdata('pesan', '<div id="alert-border-2" class="flex p-4 mb-4 text-red-800 border-t-4 border-red-300 bg-red-50 dark:text-red-400 dark:bg-gray-800 dark:border-red-800" role="alert">

    <svg class="flex-shrink-0 w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clip-rule="evenodd"></path></svg>

    <div class="ml-3 text-sm font-medium">

    Akses ditolak. Anda belum login!!

    </div>

    <button type="button" class="ml-auto -mx-1.5 -my-1.5 bg-red-50 text-red-500 rounded-lg focus:ring-2 focus:ring-red-400 p-1.5 hover:bg-red-200 inline-flex h-8 w-8 dark:bg-gray-800 dark:text-red-400 dark:hover:bg-gray-700" data-dismiss-target="#alert-border-2" aria-label="Close">

      <span class="sr-only">Dismiss</span>

      <svg aria-hidden="true" class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd"></path></svg>

    </button>

</div>');

        if ($ci->session->userdata('role\_id') == 1) {

            redirect('auth');

        } else {

            redirect('home');

        }

    } else {

        $role\_id = $ci->session->userdata('role\_id');

        $id\_user = $ci->session->userdata('id\_user');

    }

}

function cek\_user()

{

    $ci = get\_instance();

    $role\_id = $ci->session->userdata('role\_id');

    if ($role\_id != 1) {

        $ci->session->set\_flashdata('pesan', '<div id="alert-border-2" class="flex p-4 mb-4 text-red-800 border-t-4 border-red-300 bg-red-50 dark:text-red-400 dark:bg-gray-800 dark:border-red-800" role="alert">

        <svg class="flex-shrink-0 w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clip-rule="evenodd"></path></svg>

        <div class="ml-3 text-sm font-medium">

        Akses tidak diizinkan

        </div>

        <button type="button" class="ml-auto -mx-1.5 -my-1.5 bg-red-50 text-red-500 rounded-lg focus:ring-2 focus:ring-red-400 p-1.5 hover:bg-red-200 inline-flex h-8 w-8 dark:bg-gray-800 dark:text-red-400 dark:hover:bg-gray-700" data-dismiss-target="#alert-border-2" aria-label="Close">

          <span class="sr-only">Dismiss</span>

          <svg aria-hidden="true" class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd"></path></svg>

        </button>

    </div>');

        redirect('home');

    }

}

cek\_login(): Fungsi ini digunakan untuk memeriksa apakah pengguna sudah login atau belum. Jika pengguna belum login, maka akan ditampilkan pesan error yang memberitahu bahwa akses ditolak karena pengguna belum login. Selain itu, fungsi ini juga memeriksa peran (role) pengguna. Jika peran pengguna adalah 1, maka pengguna akan diarahkan ke halaman autentikasi. Jika bukan peran 1, pengguna akan diarahkan ke halaman utama (home).

cek\_user(): Fungsi ini digunakan untuk memeriksa peran (role) pengguna. Jika peran pengguna bukan 1, maka akan ditampilkan pesan error yang memberitahu bahwa akses tidak diizinkan. Pengguna akan diarahkan ke halaman utama (home).

public function index()

    {

        $this->form\_validation->set\_rules(

            'nama\_kategori',

            'Nama Kategori',

            'required|min\_length[6]',

            [

                'required' => "<div class='alert alert-danger' role='alert'>Nama Kategori Harus di Isi</div>",

                'min\_length' => "<div class='alert alert-danger' role='alert'>Nama Kategori Terlalu Pendek</div>"

            ]

        );

        $config['upload\_path'] = './assets/img/upload/category';

        $config['allowed\_types'] = 'jpg|png|jpeg';

        $config['max\_size'] = '3000';

        $config['max\_width'] = '3000';

        $config['max\_height'] = '3000';

        $config['file\_name'] = 'img' . time();

        $this->load->library('upload', $config);

        if ($this->form\_validation->run() != true) {

            $data['title'] = 'Category - Ecommerce Admin';

            $data['heading'] = 'Category';

            $data['subview'] = 'admin/category/index';

            $data['category'] = $this->ModelCategory->findAll();

            $this->load->view('admin/\_layout', $data);

        } else {

            if ($this->upload->do\_upload('image')) {

                $image = $this->upload->data();

                $gambar = $image['file\_name'];

            } else {

                $gambar = '';

            }

            $nama\_kategori = $this->input->post('nama\_kategori');

            $slug = url\_title($nama\_kategori, 'dash', true);

            $image = $gambar;

            if ($this->ModelCategory->createCategory($nama\_kategori, $image, $slug) == true) {

                $this->session->set\_flashdata('success\_category', 'Proses membuat Category Berhasil');

                redirect("admin/category");

            } else {

                $this->session->set\_flashdata('error\_category', 'Error pembuatan category');

                redirect("admin/category", 'refresh');

            }

        }

    }

Fungsi index(): Fungsi ini merupakan handler untuk permintaan ke URL yang sesuai dengan rute kontroler ini. Pada contoh ini, fungsi ini melakukan beberapa tugas, antara lain:

Mengatur aturan validasi form menggunakan $this->form\_validation->set\_rules(). Aturan validasi tersebut mensyaratkan bahwa input dengan nama "nama\_kategori" harus diisi dan memiliki panjang minimal 6 karakter.

Mengatur konfigurasi untuk proses upload gambar menggunakan library Upload CI. Konfigurasi tersebut mencakup path penyimpanan file, jenis file yang diizinkan, ukuran maksimum file, lebar maksimum gambar, tinggi maksimum gambar, dan nama file yang dihasilkan.

Memeriksa apakah validasi form berhasil ($this->form\_validation->run() == true). Jika tidak berhasil, maka fungsi akan menampilkan tampilan dengan pesan error validasi dan daftar kategori yang ada.

Jika validasi form berhasil, maka fungsi akan melakukan langkah-langkah berikut:

Melakukan proses upload gambar menggunakan $this->upload->do\_upload(). Jika berhasil, maka nama file gambar akan disimpan dalam variabel $gambar. Jika gagal, maka variabel $gambar akan diisi dengan string kosong.

Mengambil nilai input "nama\_kategori" dari form.

Membuat slug dari nama kategori menggunakan url\_title().

Mengambil nama file gambar dari variabel $gambar.

Memanggil fungsi createCategory() dari model ModelCategory untuk membuat kategori baru dengan parameter yang sesuai.

Jika proses pembuatan kategori berhasil, maka akan ditampilkan pesan sukses dan pengguna akan dialihkan ke halaman kategori. Jika gagal, maka akan ditampilkan pesan error dan pengguna akan dialihkan kembali ke halaman kategori.

public function edit()

    {

        $this->form\_validation->set\_rules(

            'nama\_kategori',

            'Nama Kategori',

            'required|min\_length[6]',

            [

                'required' => "<div class='alert alert-danger' role='alert'>Nama Kategori Harus di Isi</div>",

                'min\_length' => "<div class='alert alert-danger' role='alert'>Nama Kategori Terlalu Pendek</div>"

            ]

        );

        try {

            $id = $this->uri->segment(4);

            if (!is\_numeric($id)) {

                throw new Exception('Invalid category ID');

            }

            $config['upload\_path'] = './assets/img/upload/category';

            $config['allowed\_types'] = 'jpg|png|jpeg';

            $config['max\_size'] = '3000';

            $config['max\_width'] = '3000';

            $config['max\_height'] = '3000';

            $config['file\_name'] = 'img' . time();

            $this->load->library('upload', $config);

            $findById = $this->ModelCategory->findById($id);

            if ($findById == null) {

                throw new Exception('Category not found');

            }

            if ($this->form\_validation->run() != true) {

                $data['title'] = 'Category - Ecommerce Admin';

                $data['heading'] = 'Category';

                $data['subview'] = 'admin/category/edit';

                $data['category'] = $findById;

                $this->load->view('admin/\_layout', $data);

            } else {

                $id\_category = $this->input->post('category\_id');

                $nama\_kategori = $this->input->post('nama\_kategori');

                $slug = url\_title($nama\_kategori, 'dash', true);

                $old\_image = $this->ModelCategory->getImage($id\_category);

                if ($this->upload->do\_upload('image')) {

                    $imageData = $this->upload->data();

                    $image = $imageData['file\_name'];

                    if (!empty($old\_image)) {

                        unlink('./assets/img/upload/category/' . $old\_image);

                    }

                } else {

                    $image = $old\_image;

                }

                if ($this->ModelCategory->updateCategory($id\_category, $nama\_kategori, $image, $slug) == true) {

                    $this->session->set\_flashdata('success\_category', 'Proses update Category Berhasil');

                    redirect("admin/category");

                } else {

                    $this->session->set\_flashdata('error\_category', 'Error update category');

                    redirect("admin/category", 'refresh');

                }

            }

        } catch (Exception $e) {

            $this->session->set\_flashdata('error\_category', $e->getMessage());

            redirect("admin/category", 'refresh');

        }

    }

Fungsi edit(): Fungsi ini merupakan handler untuk permintaan ke URL yang sesuai dengan rute kontroler ini. Pada contoh ini, fungsi ini melakukan beberapa tugas, antara lain:

Mengatur aturan validasi form menggunakan $this->form\_validation->set\_rules(). Aturan validasi tersebut mensyaratkan bahwa input dengan nama "nama\_kategori" harus diisi dan memiliki panjang minimal 6 karakter.

Mengambil ID kategori dari URI menggunakan $this->uri->segment(4). Jika ID tersebut tidak berupa angka, maka akan dihasilkan pengecualian dengan pesan "Invalid category ID".

Mengatur konfigurasi untuk proses upload gambar menggunakan library Upload CI. Konfigurasi tersebut mencakup path penyimpanan file, jenis file yang diizinkan, ukuran maksimum file, lebar maksimum gambar, tinggi maksimum gambar, dan nama file yang dihasilkan.

Mencari kategori berdasarkan ID menggunakan fungsi findById() dari model ModelCategory. Jika kategori tidak ditemukan, maka akan dihasilkan pengecualian dengan pesan "Category not found".

Memeriksa apakah validasi form berhasil ($this->form\_validation->run() == true). Jika tidak berhasil, maka fungsi akan menampilkan tampilan edit dengan pesan error validasi dan data kategori yang sedang diedit.

Jika validasi form berhasil, maka fungsi akan melakukan langkah-langkah berikut:

Mengambil nilai input "category\_id" dan "nama\_kategori" dari form.

Membuat slug dari nama kategori menggunakan url\_title().

Mengambil nama gambar lama kategori menggunakan getImage(). Gambar lama ini akan digunakan untuk menghapus gambar lama jika ada penggantian gambar.

Jika ada file gambar yang diupload menggunakan $this->upload->do\_upload(), maka file gambar tersebut akan diambil informasinya menggunakan $this->upload->data() dan nama file akan disimpan dalam variabel $image. Jika gambar lama tidak kosong, maka gambar lama akan dihapus menggunakan unlink().

Jika tidak ada file gambar yang diupload, maka gambar lama akan tetap digunakan.

Memanggil fungsi updateCategory() dari model ModelCategory untuk memperbarui informasi kategori dengan parameter yang sesuai.

Jika proses pembaharuan kategori berhasil, maka akan ditampilkan pesan sukses dan pengguna akan dialihkan ke halaman kategori. Jika gagal, maka akan ditampilkan pesan error dan pengguna akan dialihkan kembali ke halaman kategori.

Jika terjadi pengecualian selama proses, maka akan ditangkap dan pesan error dari pengecualian akan ditampilkan dalam pesan flash. Pengguna akan dialihkan kembali ke halaman kategori.

public function delete()

    {

        try {

            $id = $this->uri->segment(4);

            if (!is\_numeric($id)) {

                throw new Exception('Invalid Category ID');

            }

            $category = $this->ModelCategory->findById($id);

            if (!$category) {

                throw new Exception('Category not found');

            }

            $image = $this->ModelCategory->getImage($id);

            if (!empty($image)) {

                unlink('./assets/img/upload/category/' . $image);

            }

            $result = $this->ModelCategory->deleteById($id);

            if($result){

                $this->session->set\_flashdata('success\_category', 'Category deleted successfully');

                redirect("admin/category");

            }else{

                $this->session->set\_flashdata('error\_category','Failed to delete category.');

                redirect("admin/category", 'refresh');

            }

        } catch (Exception $e) {

            $this->session->set\_flashdata('error\_category', $e->getMessage());

            redirect("admin/category", 'refresh');

        }

    }

Fungsi delete(): Fungsi ini merupakan handler untuk permintaan ke URL yang sesuai dengan rute kontroler ini. Pada contoh ini, fungsi ini melakukan beberapa tugas, antara lain:

Mengambil ID kategori dari URI menggunakan $this->uri->segment(4). Jika ID tersebut tidak berupa angka, maka akan dihasilkan pengecualian dengan pesan "Invalid Category ID".

Menggunakan fungsi findById() dari model ModelCategory untuk mencari kategori berdasarkan ID. Jika kategori tidak ditemukan, maka akan dihasilkan pengecualian dengan pesan "Category not found".

Menggunakan fungsi getImage() dari model ModelCategory untuk mengambil nama gambar kategori berdasarkan ID. Jika gambar tidak kosong, maka gambar tersebut akan dihapus menggunakan unlink().

Menggunakan fungsi deleteById() dari model ModelCategory untuk menghapus kategori berdasarkan ID. Hasil operasi penghapusan akan disimpan dalam variabel $result.

Jika penghapusan kategori berhasil ($result == true), maka akan ditampilkan pesan sukses dan pengguna akan dialihkan ke halaman kategori. Jika gagal, maka akan ditampilkan pesan error dan pengguna akan dialihkan kembali ke halaman kategori.

Jika terjadi pengecualian selama proses, maka akan ditangkap dan pesan error dari pengecualian akan ditampilkan dalam pesan flash. Pengguna akan dialihkan kembali ke halaman kategori.

**File application/views/ admin/category/index.php**

<**div** class="page-heading">

    <**div** class="page-title">

        <**div** class="row">

            <**div** class="col-12 col-md-6 order-md-1 order-last">

                <**h3**><?php echo $heading; ?></**h3**>

                <?php if (validation\_errors()) { ?>

                    <**div** class="alert alert-danger alert-dismissible fade show" role="alert">

                        <?= validation\_errors(); ?>

                        <**button** type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></**button**>

                    </**div**>

                <?php } ?>

                <?php

                if ($this->session->flashdata('error\_category') != '') {

                    echo '<div class="alert alert-danger alert-dismissible fade show" role="alert">';

                    echo $this->session->flashdata('error\_category');

                    echo '<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>';

                    echo '</div>';

                }

                ?>

                <?php

                if ($this->session->flashdata('success\_category') != '') {

                    echo '<div class="alert alert-success alert-dismissible fade show" role="alert">';

                    echo $this->session->flashdata('success\_category');

                    echo '<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>';

                    echo '</div>';

                }

                ?>

            </**div**>

            <**div** class="col-12 col-md-6 order-md-2 order-first">

                <**nav** aria-label="breadcrumb" class="breadcrumb-header float-start float-lg-end">

                    <**ol** class="breadcrumb">

                        <**li** class="breadcrumb-item">

                            <**a** href="index.html">Dashboard</**a**>

                        </**li**>

                        <**li** class="breadcrumb-item active" aria-current="page">

                            <?php echo $heading; ?>

                        </**li**>

                    </**ol**>

                </**nav**>

            </**div**>

        </**div**>

    </**div**>

    <**section** class="section">

        <**div** class="card">

            <**div** class="card-header">

                <**button** type="button" class="btn btn-primary btn-sm mb-3" data-bs-toggle="modal" data-bs-target="#category"><**i** class="fas fa-user"></**i**> Add Category</**button**>

            </**div**>

            <**div** class="card-body">

                <**table** class="table table-striped" id="table1">

                    <**thead**>

                        <**tr**>

                            <**th**>No</**th**>

                            <**th**>Nama Kategori</**th**>

                            <**th**>Image</**th**>

                            <**th**>Created\_at</**th**>

                            <**th**>Updated\_at</**th**>

                            <**th**>Status</**th**>

                        </**tr**>

                    </**thead**>

                    <**tbody**>

                        <?php

                        if (!empty($category)) {

                            foreach ($category as $row) {

                        ?>

                                <**tr**>

                                    <**td**><?php echo $row->category\_id ?></**td**>

                                    <**td**><?php echo $row->nama\_kategori ?></**td**>

                                    <**td**>

                                        <**img** src="<?= base\_url('assets/img/upload/category/' . $row->image\_category) ?>" alt="Current Image" style="width: 200px;">

                                    </**td**>

                                    <**td**><?php echo $row->created\_at ?></**td**>

                                    <**td**><?php echo $row->updated\_at ?></**td**>

                                    <**td** width="250">

                                        <**a** href="<?php echo site\_url('admin/category/edit/') . $row->category\_id; ?>" class="btn btn-success">Edit</**a**>

                                        <**a** onclick="return confirm('Are you sure you want to delete this record?');" href="<?php echo site\_url('admin/category/delete/' . $row->category\_id); ?>" class="btn btn-danger">Delete</**a**>

                                    </**td**>

                                </**tr**>

                        <?php

                            }

                        }

                        ?>

                    </**tbody**>

                </**table**>

            </**div**>

        </**div**>

    </**section**>

</**div**>

Bagian page-heading: Merupakan tampilan judul halaman dengan informasi dinamis. Pada bagian ini, terdapat judul halaman yang diambil dari variabel $heading yang dikirim dari kontroler.

Bagian validation\_errors(): Jika terjadi kesalahan validasi pada pengisian form, maka akan ditampilkan pesan error dalam kotak peringatan (alert box) dengan class alert-danger. Pesan error ini dihasilkan menggunakan fungsi validation\_errors() dari CodeIgniter.

Bagian error\_category: Jika terdapat pesan error dari session dengan key error\_category, maka akan ditampilkan pesan error dalam kotak peringatan (alert box) dengan class alert-danger.

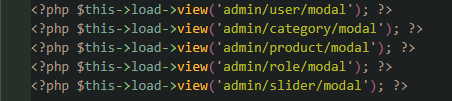
Bagian success\_category: Jika terdapat pesan sukses dari session dengan key success\_category, maka akan ditampilkan pesan sukses dalam kotak peringatan (alert box) dengan class alert-success.

Bagian breadcrumb: Menampilkan breadcrumb (navigasi hierarki) untuk memudahkan navigasi pengguna. Breadcrumb ini menunjukkan jalur navigasi dari halaman utama (Dashboard) menuju halaman kategori.

Bagian card-header: Berisi tombol "Add Category" yang membuka modal untuk menambahkan kategori baru.

Bagian card-body: Menampilkan tabel kategori dengan data yang diambil dari variabel $category. Setiap baris kategori akan ditampilkan dengan kolom-kolom seperti No, Nama Kategori, Gambar, Created\_at, Updated\_at, dan tombol Edit dan Delete.

Modal Semua sudah di load admin/\_layout.php tinggal di panggil saja



**File application/views/admin/views/admin/category/modal.php**

<**div** class="modal fade text-left" id="category" tabindex="-1" role="dialog" aria-labelledby="category" aria-hidden="true">

    <**div** class="modal-dialog modal-dialog-centered modal-dialog-scrollable modal-lg" role="document">

        <**div** class="modal-content">

            <**div** class="modal-header">

                <**h4** class="modal-title" id="category">

                    Add Kategori

                </**h4**>

                <**button** type="button" class="close" data-bs-dismiss="modal" aria-label="Close">

                    <**i** data-feather="x"></**i**>

                </**button**>

            </**div**>

            <**form** action="<?= base\_url('admin/category'); ?>" method="POST" enctype="multipart/form-data">

                <**div** class="modal-body">

                    <**label** for="name\_kategori">Nama Kategori: </**label**>

                    <**div** class="form-group">

                        <**input** id="name\_kategori" type="text" name="nama\_kategori" placeholder="Nama Kategori" class="form-control" />

                    </**div**>

                    <**label** for="image">image: </**label**>

                    <**div** class="form-group">

                        <**input** type="file" class="form-control" id="image" name="image">

                    </**div**>

                </**div**>

                <**div** class="modal-footer">

                    <**button** type="button" class="btn btn-light-secondary" data-bs-dismiss="modal">

                        <**i** class="bx bx-x d-block d-sm-none"></**i**>

                        <**span** class="d-none d-sm-block">Close</**span**>

                    </**button**>

                    <**button** type="submit" class="btn btn-primary ms-1" data-bs-dismiss="modal">

                        <**i** class="bx bx-check d-block d-sm-none"></**i**>

                        <**span** class="d-none d-sm-block">Adding</**span**>

                    </**button**>

                </**div**>

            </**form**>

        </**div**>

    </**div**>

</**div**>

Bagian modal: Merupakan modal (jendela pop-up) yang digunakan untuk menambahkan kategori baru.

Bagian modal-header: Menampilkan judul modal dengan teks "Add Kategori" dan tombol close (x) untuk menutup modal.

Bagian modal-body: Berisi form input untuk mengisi nama kategori dan memilih gambar. Form ini akan mengirimkan data ke URL <?= base\_url('admin/category'); ?> menggunakan metode POST dan tipe data multipart/form-data untuk mengunggah gambar.

Bagian modal-footer: Menampilkan tombol "Close" untuk menutup modal dan tombol "Adding" untuk mengirimkan form dan menambahkan kategori baru.

Bagian <input>: Digunakan untuk memasukkan nilai nama kategori (nama\_kategori) dan memilih gambar (image). Input gambar menggunakan tipe file dengan atribut id dan name yang sama yaitu "image".

base\_url('admin/category'): Merupakan URL yang akan dituju saat form dikirimkan. Dalam hal ini, form akan dikirimkan ke kontroler admin/category untuk melakukan aksi penambahan kategori baru.

**File application/views/admin/views/admin/category/edit.php**

<**div** class="page-heading">

    <**div** class="page-title">

        <**div** class="row">

            <**div** class="col-12 col-md-6 order-md-1 order-last">

                <**h3**><?php echo $heading; ?></**h3**>

            </**div**>

            <**div** class="col-12 col-md-6 order-md-2 order-first">

                <**nav** aria-label="breadcrumb" class="breadcrumb-header float-start float-lg-end">

                    <**ol** class="breadcrumb">

                        <**li** class="breadcrumb-item">

                            <**a** href="index.html">Dashboard</**a**>

                        </**li**>

                        <**li** class="breadcrumb-item active" aria-current="page">

                            <?php echo $heading; ?>

                        </**li**>

                    </**ol**>

                </**nav**>

            </**div**>

        </**div**>

    </**div**>

    <**section** class="section">

        <**div** class="card">

            <**div** class="card-header">

                <**h3**>Edit Category</**h3**>

            </**div**>

            <**div** class="card-body">

                <?= form\_error("nama\_kategori"); ?>

                <?php if (!empty($category)) { ?>

                    <?php foreach ($category as $row) { ?>

                        <**form** action="<?= base\_url("admin/category/edit"); ?>" method="POST" enctype="multipart/form-data">

                            <**input** type="hidden" name="category\_id" value="<?php echo $row->category\_id ?>">

                            <**div** class="mb-3">

                                <**label** for="nama\_kategori" class="form-label">Nama Kategori</**label**>

                                <**input** type="text" name="nama\_kategori" value="<?php echo $row->nama\_kategori ?>" class="form-control" id="nama\_kategori" aria-describedby="emailHelp">

                            </**div**>

                            <**div** class="mb-3">

                                <**label** for="image" class="form-label">Gambar</**label**>

                                <**input** type="file" name="image" class="form-control" id="image">

                            </**div**>

                            <**div** class="mb-3">

                                <**label** for="current\_image" class="form-label">Gambar Saat Ini</**label**>

                                <**br**>

                                <?php if ($row->image\_category) { ?>

                                    <**img** src="<?= base\_url('assets/img/upload/category/' . $row->image\_category) ?>" alt="Current Image" style="width: 200px;">

                                <?php } else { ?>

                                    <**span**>Tidak ada gambar</**span**>

                                <?php } ?>

                            </**div**>

                            <**button** type="submit" name="submit" class="btn btn-primary">Submit</**button**>

                        </**form**>

                    <?php } ?>

                <?php } ?>

            </**div**>

        </**div**>

    </**section**>

</**div**>

Bagian page-heading: Merupakan bagian atas halaman yang menampilkan judul halaman dan breadcrumb (navigasi tampilan hierarkis).

Bagian card: Merupakan kartu (card) yang digunakan untuk mengelompokkan konten terkait dengan edit kategori.

Bagian card-header: Menampilkan judul "Edit Category" di dalam kartu.

Bagian card-body: Berisi form untuk mengedit kategori. Form ini akan mengirimkan data ke URL <?= base\_url("admin/category/edit"); ?> menggunakan metode POST dan tipe data multipart/form-data untuk mengunggah gambar.

Bagian form\_error("nama\_kategori"): Menampilkan pesan kesalahan (error message) jika terdapat kesalahan validasi pada input "nama\_kategori".

Bagian foreach ($category as $row): Melakukan iterasi untuk setiap kategori dalam array $category. Setiap kategori akan ditampilkan dalam form edit yang terpisah.

Bagian <input type="hidden" name="category\_id" value="<?php echo $row->category\_id ?>">: Menyimpan ID kategori yang sedang diedit sebagai input tersembunyi (hidden input) dalam form.

Bagian <input type="text" name="nama\_kategori" value="<?php echo $row->nama\_kategori ?>" class="form-control" id="nama\_kategori" aria-describedby="emailHelp">: Menampilkan input teks untuk mengedit nama kategori dengan nilai awal yang sudah ada.

Bagian <input type="file" name="image" class="form-control" id="image">: Menampilkan input untuk memilih gambar baru untuk kategori.

Bagian <img src="<?= base\_url('assets/img/upload/category/' . $row->image\_category) ?>" alt="Current Image" style="width: 200px;">: Menampilkan gambar saat ini untuk kategori yang sedang diedit. Jika tidak ada gambar, akan ditampilkan teks "Tidak ada gambar".

Bagian <button type="submit" name="submit" class="btn btn-primary">Submit</button>: Menampilkan tombol "Submit" yang digunakan untuk mengirimkan form dan melakukan pembaruan kategori.

**File application/models/ModelProduct.php**

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class ModelProduct extends CI\_Model

{

    public function createProduct($nama, $image, $category\_id, $description, $price, $countInStock, $slug)

    {

        try {

            $data = [

                'name' => $nama,

                'image\_product' => $image,

                'category\_id' => $category\_id,

                'description' => $description,

                'price' => $price,

                'countInStock' => $countInStock,

                'slug\_product' => $slug

            ];

            $this->db->insert('product', $data);

            return true;

        } catch (Exception $e) {

            echo $e->getMessage();

            return null;

        }

    }

    public function findByName($name){

        try {

            $sql = "SELECT \* FROM product WHERE name = ?";

            $query = $this->db->query($sql, [$name]);

            $result = $query->result();

            return $result;

        } catch (Exception $e) {

            echo $e->getMessage();

            return null;

        }

    }

    public function findById($id)

    {

        try {

            $sql = "SELECT \* FROM product WHERE product\_id = ?";

            $query = $this->db->query($sql, [$id]);

            $result = $query->result();

            return $result;

        } catch (Exception $e) {

            echo $e->getMessage();

            return null;

        }

    }

    public function findByIdProduct($id)

    {

        try {

            $sql = "SELECT \* FROM product WHERE product\_id = ?";

            $query = $this->db->query($sql, [$id]);

            $result = $query->row();

            return $result;

        } catch (Exception $e) {

            echo $e->getMessage();

        }

    }

*// public function findById($id)*

*// {*

*//     return $this->db->get\_where('product', ['product\_id' => $id])->row();*

*// }*

    public function findAll()

    {

        return $this->db->get('product')->result();

    }

    public function update($id, $data){

        $this->db->where('product\_id', $id);

        $this->db->update('product', $data);

        return true;

    }

*// public function update($id, $nama, $image, $category\_id, $description, $price, $countInStock, $slug)*

*// {*

*//     $data = [*

*//         'name' => $nama,*

*//         'image\_product' => $image,*

*//         'category\_id' => $category\_id,*

*//         'description' => $description,*

*//         'price' => $price,*

*//         'countInStock' => $countInStock,*

*//         'slug\_product' => $slug*

*//     ];*

*//     $this->db->where('product\_id', $id);*

*//     $this->db->update('product', $data);*

*//     // Echoing the parameters*

*//     // echo "ID: " . $id . "<br>";*

*//     // echo "Nama: " . $nama . "<br>";*

*//     // echo "Image: " . $image . "<br>";*

*//     // echo "Category ID: " . $category\_id . "<br>";*

*//     // echo "Description: " . $description . "<br>";*

*//     // echo "Price: " . $price . "<br>";*

*//     // echo "Count In Stock: " . $countInStock . "<br>";*

*//     return true;*

*// }*

    public function findBySlug($slug)

    {

        try {

            $sql = "SELECT \* FROM product WHERE slug\_product = ?";

            $query = $this->db->query($sql, [$slug]);

            $result = $query->result();

            return $result;

        } catch (Exception $e) {

            echo $e->getMessage();

        }

    }

    public function delete($id)

    {

        $this->db->where('product\_id', $id);

        $this->db->delete('product');

        return true;

    }

    public function getImage($product\_id)

    {

        $this->db->select('image\_product');

        $this->db->from('product');

        $this->db->where('product\_id', $product\_id);

        $query = $this->db->get();

        if ($query->num\_rows() > 0) {

            $row = $query->row();

            return $row->image\_product;

        } else {

            return null;

        }

    }

    public function countProduct()

    {

        return $this->db->count\_all('product');

    }

    public function updateQuantity($product\_id, $quantity)

    {

        try {

            $data = array(

                'countInStock' => $quantity

            );

            $this->db->where('product\_id', $product\_id);

            $this->db->update('product', $data);

            return true;

        } catch (Exception $e) {

            throw new Exception('Error updating product quantity: ' . $e->getMessage());

        }

    }

}

Fungsi createProduct($nama, $image, $category\_id, $description, $price, $countInStock, $slug): Fungsi ini digunakan untuk membuat data produk baru dalam database. Parameter yang diterima adalah nama produk, nama file gambar, ID kategori, deskripsi, harga, stok, dan slug. Data produk baru disimpan dalam array $data dan kemudian diinsert ke tabel "product" menggunakan method insert() dari objek database.

Fungsi findByName($name): Fungsi ini digunakan untuk mencari data produk berdasarkan nama produk yang diberikan. SQL query dieksekusi menggunakan method query() dengan menggunakan parameter binding untuk nama produk. Hasil query dikembalikan dalam bentuk array objek.

Fungsi findById($id): Fungsi ini digunakan untuk mencari data produk berdasarkan ID yang diberikan. SQL query dieksekusi dengan menggunakan method query() dengan menggunakan parameter binding untuk ID. Hasil query dikembalikan dalam bentuk array objek.

Fungsi findByIdProduct($id): Fungsi ini juga digunakan untuk mencari data produk berdasarkan ID yang diberikan, tetapi hasilnya dikembalikan dalam bentuk satu baris objek bukan dalam bentuk array.

Fungsi findAll(): Fungsi ini digunakan untuk mendapatkan semua data produk dari tabel "product" dalam database. Method get() dari objek database ($this->db) digunakan untuk mengambil semua data produk dan hasilnya dikembalikan dalam bentuk array objek.

Fungsi update($id, $data): Fungsi ini digunakan untuk memperbarui data produk berdasarkan ID. Data yang akan diperbarui disimpan dalam parameter $data dan method update() digunakan untuk memperbarui data produk dalam tabel "product" berdasarkan ID yang diberikan.

Fungsi findBySlug($slug): Fungsi ini digunakan untuk mencari data produk berdasarkan slug yang diberikan. SQL query dieksekusi menggunakan method query() dengan menggunakan parameter binding untuk slug. Hasil query dikembalikan dalam bentuk array objek.

Fungsi delete($id): Fungsi ini digunakan untuk menghapus data produk berdasarkan ID yang diberikan. Method delete() digunakan untuk menghapus data produk dalam tabel "product" berdasarkan ID yang diberikan.

Fungsi getImage($product\_id): Fungsi ini digunakan untuk mendapatkan nama file gambar produk berdasarkan ID produk yang diberikan. SQL query dieksekusi menggunakan method select(), from(), dan where() untuk mendapatkan nama file gambar dari tabel "product" berdasarkan ID produk. Nama file gambar dikembalikan sebagai hasil query.

Fungsi countProduct(): Fungsi ini digunakan untuk menghitung jumlah total produk yang ada dalam tabel "product" menggunakan method count\_all() dari objek database.

Fungsi updateQuantity($product\_id, $quantity): Fungsi ini digunakan untuk memperbarui jumlah stok (quantity) produk berdasarkan ID produk. Data yang akan diperbarui, yaitu jumlah stok, disimpan dalam array $data. Method update() digunakan untuk memperbarui jumlah stok produk dalam tabel "product" berdasarkan ID produk yang diberikan.

**File application/controllers/admin/Product.php**

public function index()

    {

        $this->form\_validation->set\_rules(

            'name',

            'Nama Produk',

            'required|min\_length[6]',

            [

                'required' => "<div class='alert alert-danger' role='alert'>Nama Produk Harus di Isi</div>",

                'min\_length' => "<div class='alert alert-danger' role='alert'>Nama Produk Terlalu Pendek</div>"

            ]

        );

        $this->form\_validation->set\_rules(

            'category\_id',

            'ID Kategori',

            'required',

            [

                'required' => "<div class='alert alert-danger' role='alert'>ID Kategori harus diisi</div>"

            ]

        );

        $this->form\_validation->set\_rules(

            'description',

            'Deskripsi',

            'required|min\_length[10]',

            [

                'required' => "<div class='alert alert-danger' role='alert'>Deskripsi harus diisi</div>",

                'min\_length' => "<div class='alert alert-danger' role='alert'>Deskripsi minimal 10 karakter</div>"

            ]

        );

        $this->form\_validation->set\_rules(

            'price',

            'Harga',

            'required|numeric',

            [

                'required' => "<div class='alert alert-danger' role='alert'>Harga harus diisi</div>",

                'numeric' => "<div class='alert alert-danger' role='alert'>Harga harus berupa angka</div>"

            ]

        );

        $this->form\_validation->set\_rules(

            'countInStock',

            'Jumlah Stok',

            'required|integer',

            [

                'required' => "<div class='alert alert-danger' role='alert'>Jumlah Stok harus diisi</div>",

                'integer' => "<div class='alert alert-danger' role='alert'>Jumlah Stok harus berupa bilangan bulat</div>"

            ]

        );

        $config['upload\_path'] = './assets/img/upload/product';

        $config['allowed\_types'] = 'jpg|png|jpeg';

        $config['max\_size'] = '3000';

        $config['max\_width'] = '3000';

        $config['max\_height'] = '3000';

        $config['file\_name'] = 'img' . time();

        $this->load->library('upload', $config);

        if ($this->form\_validation->run() != true) {

            $data['title'] = 'Product - Ecommerce Admin';

            $data['heading'] = 'Product';

            $data['subview'] = 'admin/product/index';

            $data['category'] = $this->ModelCategory->findAll();

            $data['products'] = $this->ModelProduct->findAll();

            $this->load->view('admin/\_layout', $data);

        } else {

            if ($this->upload->do\_upload('image')) {

                $imageData = $this->upload->data();

                $image = $imageData['file\_name'];

            } else {

                $image = '';

            }

            $name = $this->input->post('name');

            $category\_id = $this->input->post('category\_id');

            $description = $this->input->post('description');

            $slug = url\_title($name, 'dash', true);

            $price = $this->input->post('price');

            $countInStock = $this->input->post('countInStock');

            if ($this->ModelProduct->createProduct($name, $image, $category\_id, $description, $price, $countInStock, $slug)) {

                $this->session->set\_flashdata('success\_product', 'Proses membuat Produk Berhasil');

                redirect("admin/product");

            } else {

                $this->session->set\_flashdata('error\_product', 'Error pembuatan produk');

                redirect("admin/product");

            }

        }

    }

Pengecekan Validasi Form: Fungsi ini menggunakan library form\_validation untuk melakukan validasi terhadap data yang dikirim melalui form. Terdapat beberapa aturan validasi yang ditetapkan, seperti validasi untuk field 'name', 'category\_id', 'description', 'price', dan 'countInStock'. Jika validasi tidak berhasil (form tidak valid), maka akan ditampilkan pesan error sesuai aturan validasi yang ditetapkan.

Konfigurasi Upload File: Fungsi ini melakukan konfigurasi untuk proses upload file gambar. Konfigurasi tersebut mencakup path penyimpanan file, tipe file yang diizinkan, ukuran maksimum file, dan nama file yang akan disimpan.

Proses Validasi dan Penyimpanan Data: Jika validasi form berhasil (form valid), maka dilakukan proses validasi dan penyimpanan data produk ke dalam database. Jika gambar berhasil diunggah, nama file akan disimpan dalam variabel $image. Jika tidak, variabel $image akan diisi dengan string kosong. Selanjutnya, nilai-nilai dari form seperti 'name', 'category\_id', 'description', 'price', dan 'countInStock' akan diambil. Kemudian, fungsi createProduct() dari model ModelProduct akan dipanggil untuk menyimpan data produk ke dalam database. Jika penyimpanan berhasil, akan ditampilkan pesan sukses dan pengguna akan diarahkan kembali ke halaman daftar produk. Jika terjadi kesalahan dalam penyimpanan, akan ditampilkan pesan error dan pengguna juga akan diarahkan kembali ke halaman daftar produk.

public function edit()

    {

        $this->form\_validation->set\_rules(

            'name',

            'Nama Produk',

            'required|min\_length[6]',

            [

                'required' => "<div class='alert alert-danger' role='alert'>Nama Produk Harus di Isi</div>",

                'min\_length' => "<div class='alert alert-danger' role='alert'>Nama Produk Terlalu Pendek</div>"

            ]

        );

        $this->form\_validation->set\_rules(

            'category\_id',

            'ID Kategori',

            'required',

            [

                'required' => "<div class='alert alert-danger' role='alert'>ID Kategori harus diisi</div>"

            ]

        );

        $this->form\_validation->set\_rules(

            'description',

            'Deskripsi',

            'required|min\_length[10]',

            [

                'required' => "<div class='alert alert-danger' role='alert'>Deskripsi harus diisi</div>",

                'min\_length' => "<div class='alert alert-danger' role='alert'>Deskripsi minimal 10 karakter</div>"

            ]

        );

        $this->form\_validation->set\_rules(

            'price',

            'Harga',

            'required|numeric',

            [

                'required' => "<div class='alert alert-danger' role='alert'>Harga harus diisi</div>",

                'numeric' => "<div class='alert alert-danger' role='alert'>Harga harus berupa angka</div>"

            ]

        );

        $this->form\_validation->set\_rules(

            'countInStock',

            'Jumlah Stok',

            'required|integer',

            [

                'required' => "<div class='alert alert-danger' role='alert'>Jumlah Stok harus diisi</div>",

                'integer' => "<div class='alert alert-danger' role='alert'>Jumlah Stok harus berupa bilangan bulat</div>"

            ]

        );

        try {

            $id\_product = $this->uri->segment(4);

            if (!is\_numeric($id\_product)) {

                throw new Exception('Invalid product ID');

            }

            $config['upload\_path'] = './assets/img/upload/product';

            $config['allowed\_types'] = 'jpg|png|jpeg';

            $config['max\_size'] = '3000';

            $config['max\_width'] = '3000';

            $config['max\_height'] = '3000';

            $config['file\_name'] = 'img' . time();

            $this->load->library('upload', $config);

            $findById = $this->ModelProduct->findById($id\_product);

            if ($findById == null) {

                throw new Exception('Product not found');

            }

            if ($this->form\_validation->run() != true) {

                $data['title'] = 'Product - Ecommerce Admin';

                $data['heading'] = 'Product';

                $data['subview'] = 'admin/product/edit';

                $data['category'] = $this->ModelCategory->findAll();

                $data['product'] = $findById;

                $this->load->view('admin/\_layout', $data);

            } else {

                $id = $this->input->post('product\_id');

                $name = $this->input->post('name');

                $category\_id = $this->input->post('category\_id');

                $description = $this->input->post('description');

                $price = $this->input->post('price');

                $countInStock = $this->input->post('countInStock');

                $slug = url\_title($name, 'dash', true);

                $old\_image = $this->ModelProduct->getImage($id);

                $image = $old\_image;

                if (!empty($\_FILES['image']['name'])) {

                    if ($this->upload->do\_upload('image')) {

                        $imageData = $this->upload->data();

                        $image = $imageData['file\_name'];

                        if (!empty($old\_image)) {

                            unlink('./assets/img/upload/product/' . $old\_image);

                        }

                    } else {

                        $this->session->set\_flashdata('error\_product', 'Error uploading image');

                        redirect("admin/product", 'refresh');

                    }

                }

                $data = [

                    'name' => $name,

                    'image\_product' => $image,

                    'category\_id' => $category\_id,

                    'description' => $description,

                    'slug\_product' => $slug,

                    'price' => $price,

                    'countInStock' => $countInStock

                ];

                if ($this->ModelProduct->update($id, $data) == true) {

                    $this->session->set\_flashdata('success\_product', 'Proses update Product Berhasil');

                    redirect("admin/product");

                } else {

                    $this->session->set\_flashdata('error\_product', 'Error update product');

                    redirect("admin/product", 'refresh');

                }

            }

        } catch (Exception $e) {

            $this->session->set\_flashdata('error\_product', $e->getMessage());

            redirect("admin/product", 'refresh');

        }

    }

Pengecekan Validasi Form: Fungsi ini menggunakan library form\_validation untuk melakukan validasi terhadap data yang dikirim melalui form. Sama seperti pada fungsi index(), terdapat aturan validasi untuk field 'name', 'category\_id', 'description', 'price', dan 'countInStock'. Jika validasi tidak berhasil (form tidak valid), maka akan ditampilkan pesan error sesuai aturan validasi yang ditetapkan.

Penanganan Exception: Fungsi ini menggunakan blok try-catch untuk menangani exception yang terjadi. Jika terjadi kesalahan dalam eksekusi kode, misalnya saat mencari produk berdasarkan ID atau saat mengunggah gambar, pesan error akan ditangkap dan disimpan sebagai flash data untuk ditampilkan kepada pengguna. Pengguna akan diarahkan kembali ke halaman daftar produk.

Konfigurasi Upload File: Seperti pada fungsi index(), fungsi ini juga melakukan konfigurasi untuk proses upload file gambar.

Pencarian Data Produk: Fungsi ini menggunakan method findById() dari model ModelProduct untuk mencari data produk berdasarkan ID yang diterima dari URL. Jika produk tidak ditemukan (nilai kembaliannya adalah null), maka akan dilemparkan exception dengan pesan "Product not found".

Tampilan Form Edit: Jika validasi form tidak berhasil, maka akan ditampilkan form edit produk beserta pesan error validasi. Data produk yang akan diedit dan daftar kategori juga akan dikirimkan ke tampilan.

Proses Update Data Produk: Jika validasi form berhasil, maka dilakukan proses update data produk ke dalam database. Data dari form seperti 'product\_id', 'name', 'category\_id', 'description', 'price', dan 'countInStock' akan diambil. Nama file gambar lama juga akan diambil menggunakan method getImage() dari model ModelProduct. Jika ada gambar baru yang diunggah, gambar lama akan dihapus dan digantikan dengan gambar baru. Data produk yang akan diupdate disimpan dalam variabel $data. Jika proses update berhasil, pesan sukses ditampilkan dan pengguna diarahkan kembali ke halaman daftar produk. Jika terjadi kesalahan dalam proses update, pesan error ditampilkan dan pengguna juga diarahkan kembali ke halaman daftar produk.

 public function delete()

    {

        try {

            $id = $this->uri->segment(4);

            if (!is\_numeric($id)) {

                throw new Exception('Invalid product ID');

            }

            $product = $this->ModelProduct->findById($id);

            if (!$product) {

                throw new Exception('Product not found');

            }

            $image = $this->ModelProduct->getImage($id);

            if (!empty($image)) {

                unlink('./assets/img/upload/product/' . $image);

            }

            $result = $this->ModelProduct->delete($id);

            if($result){

                $this->session->set\_flashdata('success\_product', 'Product deleted successfully');

                redirect("admin/product");

            }else{

                $this->session->set\_flashdata('error\_product','Failed to delete Product');

                redirect("admin/product", 'refresh');

            }

        } catch (Exception $e) {

            $this->session->set\_flashdata('error\_product', $e->getMessage());

            redirect("admin/product", 'refresh');

        }

    }

Pengecekan Exception: Fungsi ini menggunakan blok try-catch untuk menangani exception yang terjadi. Jika terjadi kesalahan dalam eksekusi kode, misalnya saat mengambil ID produk dari URL atau saat mencari produk berdasarkan ID, pesan error akan ditangkap dan disimpan sebagai flash data untuk ditampilkan kepada pengguna. Pengguna akan diarahkan kembali ke halaman daftar produk.

Pencarian Data Produk: Fungsi ini menggunakan method findById() dari model ModelProduct untuk mencari data produk berdasarkan ID yang diterima dari URL. Jika produk tidak ditemukan (nilai kembaliannya adalah null), maka akan dilemparkan exception dengan pesan "Product not found".

Penghapusan Gambar: Jika produk ditemukan, gambar produk yang terkait dengan ID tersebut akan dihapus menggunakan fungsi unlink(). Gambar dihapus dari direktori ./assets/img/upload/product/.

Penghapusan Data Produk: Setelah gambar dihapus, proses penghapusan data produk dilakukan menggunakan method delete() dari model ModelProduct. Nilai kembaliannya akan disimpan dalam variabel $result.

Pengalihan Halaman: Jika proses penghapusan berhasil, pesan sukses ditampilkan dan pengguna diarahkan kembali ke halaman daftar produk. Jika terjadi kesalahan dalam proses penghapusan, pesan error ditampilkan dan pengguna juga diarahkan kembali ke halaman daftar produk.

**File application/views/admin/product/index.php**

<**div** class="page-heading">

    <**div** class="page-title">

        <**div** class="row">

            <**div** class="col-12 col-md-6 order-md-1 order-last">

                <**h3**><?php echo $heading; ?></**h3**>

                <?php if (validation\_errors()) { ?>

                    <**div** class="alert alert-danger alert-dismissible fade show" role="alert">

                        <?= validation\_errors(); ?>

                        <**button** type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></**button**>

                    </**div**>

                <?php } ?>

                <?php

                if ($this->session->flashdata('error\_product') != '') {

                    echo '<div class="alert alert-danger alert-dismissible fade show" role="alert">';

                    echo $this->session->flashdata('error\_product');

                    echo '<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>';

                    echo '</div>';

                }

                ?>

                <?php

                if ($this->session->flashdata('success\_product') != '') {

                    echo '<div class="alert alert-success alert-dismissible fade show" role="alert">';

                    echo $this->session->flashdata('success\_product');

                    echo '<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>';

                    echo '</div>';

                }

                ?>

            </**div**>

            <**div** class="col-12 col-md-6 order-md-2 order-first">

                <**nav** aria-label="breadcrumb" class="breadcrumb-header float-start float-lg-end">

                    <**ol** class="breadcrumb">

                        <**li** class="breadcrumb-item">

                            <**a** href="index.html">Dashboard</**a**>

                        </**li**>

                        <**li** class="breadcrumb-item active" aria-current="page">

                            <?php echo $heading; ?>

                        </**li**>

                    </**ol**>

                </**nav**>

            </**div**>

        </**div**>

    </**div**>

    <**section** class="section">

        <**div** class="card">

            <**div** class="card-header">

                <**button** type="button" class="btn btn-primary btn-sm mb-3" data-bs-toggle="modal" data-bs-target="#product"><**i** class="fas fa-user"></**i**> Add Product</**button**>

            </**div**>

            <**div** class="card-body">

                <**table** class="table table-striped" id="table1">

                    <**thead**>

                        <**tr**>

                            <**th**>No</**th**>

                            <**th**>Nama</**th**>

                            <**th**>Image</**th**>

                            <**th**>Category\_id</**th**>

                            <**th**>Description</**th**>

                            <**th**>Price</**th**>

                            <**th**>CountInStock</**th**>

                            <**th**>Created\_at</**th**>

                            <**th**>Updated\_at</**th**>

                            <**th**>Status</**th**>

                        </**tr**>

                    </**thead**>

                    <**tbody**>

                        <?php

                        if (!empty($products)) {

                            foreach ($products as $row) {

                        ?>

                                <**tr**>

                                    <**td**><?php echo $row->product\_id ?></**td**>

                                    <**td**><?php echo $row->name ?></**td**>

                                    <**td**>

                                    <**img** src="<?= base\_url('assets/img/upload/product/' . $row->image\_product); ?>" alt="Current Image" style="width: 200px;">

                                    </**td**>

                                    <**td**><?php echo $row->category\_id ?></**td**>

                                    <**td**><?php echo $row->description ?></**td**>

                                    <**td**><?php echo $row->price ?></**td**>

                                    <**td**><?php echo $row->countInStock ?></**td**>

                                    <**td**><?php echo $row->created\_at ?></**td**>

                                    <**td**><?php echo $row->updated\_at ?></**td**>

                                    <**td** width="250">

                                        <**a** href="<?php echo site\_url('admin/product/edit/') . $row->product\_id; ?>" class="btn btn-success">Edit</**a**>

                                        <**a** onclick="return confirm('Are you sure you want to delete this record?');" href="<?php echo site\_url('admin/product/delete/' . $row->product\_id); ?>" class="btn btn-danger">Delete</**a**>

                                    </**td**>

                                </**tr**>

                        <?php

                            }

                        }

                        ?>

                    </**tbody**>

                </**table**>

            </**div**>

        </**div**>

    </**section**>

</**div**>

Bagian page-heading: Merupakan kontainer utama untuk header halaman.

Bagian page-title: Digunakan untuk menampilkan judul halaman.

Bagian row: Digunakan untuk membuat baris menggunakan sistem grid dari framework CSS yang digunakan.

Bagian col-12 col-md-6 order-md-1 order-last dan col-12 col-md-6 order-md-2 order-first: Kolom ini mengatur tata letak elemen pada halaman menggunakan sistem grid. Kolom pertama digunakan untuk menampilkan judul halaman dan pesan kesalahan atau sukses. Kolom kedua digunakan untuk menampilkan breadcrumb (navigasi) halaman.

Bagian nav aria-label="breadcrumb" class="breadcrumb-header float-start float-lg-end" dan ol class="breadcrumb": Digunakan untuk menampilkan breadcrumb (navigasi hierarki) halaman.

Bagian card: Digunakan untuk membuat kartu (card) yang berisi daftar produk.

Bagian card-header: Bagian header dari kartu.

Bagian <button type="button" class="btn btn-primary btn-sm mb-3" data-bs-toggle="modal" data-bs-target="#product"><i class="fas fa-user"></i> Add Product</button>: Tombol "Add Product" yang akan menampilkan modal untuk menambahkan produk baru.

Bagian card-body: Bagian isi (body) dari kartu.

Bagian table class="table table-striped" id="table1": Digunakan untuk membuat tabel yang menampilkan daftar produk.

Bagian thead: Bagian kepala tabel (header) yang berisi judul kolom.

Bagian tbody: Bagian isi tabel yang menampilkan data produk.

<?php if (!empty($products)) { foreach ($products as $row) { ?>: Ini adalah kode PHP yang digunakan untuk mengiterasi melalui array $products yang berisi data produk. Setiap baris data produk akan ditampilkan dalam satu baris tabel.

<td><?php echo $row->product\_id ?></td> hingga <td><?php echo $row->updated\_at ?></td>: Digunakan untuk menampilkan nilai atribut dari setiap objek produk dalam kolom tabel.

<td width="250">: Digunakan untuk menampilkan tombol "Edit" dan "Delete" untuk setiap produk dalam kolom tabel.

**File application/views/admin/product/modal.php**

<**div** class="modal fade text-left" id="product" tabindex="-1" role="dialog" aria-labelledby="product" aria-hidden="true">

    <**div** class="modal-dialog modal-dialog-centered modal-dialog-scrollable modal-lg" role="document">

        <**div** class="modal-content">

            <**div** class="modal-header">

                <**h4** class="modal-title" id="product">

                    Add Kategori

                </**h4**>

                <**button** type="button" class="close" data-bs-dismiss="modal" aria-label="Close">

                    <**i** data-feather="x"></**i**>

                </**button**>

            </**div**>

            <**form** action="<?= base\_url('admin/product'); ?>" method="POST" enctype="multipart/form-data">

                <**div** class="modal-body">

                    <**label** for="name\_produk">Nama Produk: </**label**>

                    <**div** class="form-group">

                        <**input** id="name\_produk" type="text" name="name" placeholder="Nama Produk" class="form-control" />

                    </**div**>

                    <**label** for="image">Gambar: </**label**>

                    <**div** class="form-group">

                        <**input** type="file" class="form-control" id="image" name="image">

                    </**div**>

                    <**label** for="category\_id">ID Kategori: </**label**>

                    <**div** class="form-group">

                        <**select** name="category\_id" class="form-control">

                            <**option** value="">Pilih Kategori</**option**>

                            <?php foreach ($category as $k) { ?>

                            <**option** value="<?= $k->category\_id;?>"><?= $k->nama\_kategori;?></**option**>

                            <?php } ?>

                        </**select**>

                    </**div**>

                    <**label** for="description">Deskripsi: </**label**>

                    <**div** class="form-group">

                        <**textarea** id="description" name="description" placeholder="Deskripsi" class="form-control"></**textarea**>

                    </**div**>

                    <**label** for="price">Harga: </**label**>

                    <**div** class="form-group">

                        <**input** id="price" type="text" name="price" placeholder="Harga" class="form-control" />

                    </**div**>

                    <**label** for="countInStock">Jumlah Stok: </**label**>

                    <**div** class="form-group">

                        <**input** id="countInStock" type="text" name="countInStock" placeholder="Jumlah Stok" class="form-control" />

                    </**div**>

                </**div**>

                <**div** class="modal-footer">

                    <**button** type="button" class="btn btn-light-secondary" data-bs-dismiss="modal">

                        <**i** class="bx bx-x d-block d-sm-none"></**i**>

                        <**span** class="d-none d-sm-block">Close</**span**>

                    </**button**>

                    <**button** type="submit" class="btn btn-primary ms-1" data-bs-dismiss="modal">

                        <**i** class="bx bx-check d-block d-sm-none"></**i**>

                        <**span** class="d-none d-sm-block">Adding</**span**>

                    </**button**>

                </**div**>

            </**form**>

        </**div**>

    </**div**>

</**div**>

<div class="modal fade text-left" id="product" tabindex="-1" role="dialog" aria-labelledby="product" aria-hidden="true">: Ini adalah kontainer utama untuk modal. id="product" digunakan untuk mengidentifikasi modal secara unik.

<div class="modal-dialog modal-dialog-centered modal-dialog-scrollable modal-lg" role="document">: Ini adalah elemen yang mengatur tata letak dan ukuran modal. modal-dialog-centered digunakan untuk membuat modal berada di tengah, modal-dialog-scrollable memungkinkan kontennya dapat di-scroll, dan modal-lg memberikan ukuran modal yang lebih besar.

<div class="modal-content">: Ini adalah elemen yang berisi semua konten modal.

<div class="modal-header">: Bagian header modal yang berisi judul dan tombol close.

<h4 class="modal-title" id="product">: Judul modal dengan teks "Add Kategori".

<button type="button" class="close" data-bs-dismiss="modal" aria-label="Close">: Tombol close (tutup) modal.

<form action="<?= base\_url('admin/product'); ?>" method="POST" enctype="multipart/form-data">: Formulir untuk menambahkan produk baru dengan action yang menunjukkan URL tujuan pengiriman data form.

<div class="modal-body">: Bagian isi modal yang berisi elemen-elemen form untuk menginputkan informasi produk seperti nama, gambar, kategori, deskripsi, harga, dan jumlah stok.

<div class="modal-footer">: Bagian footer modal yang berisi tombol close dan tombol untuk menambahkan produk.

<button type="button" class="btn btn-light-secondary" data-bs-dismiss="modal">: Tombol close dengan teks "Close".

<button type="submit" class="btn btn-primary ms-1" data-bs-dismiss="modal">: Tombol untuk menambahkan produk dengan teks "Adding". Tombol ini akan mengirimkan form ke URL yang ditentukan dalam atribut action pada saat tombol diklik.

**File application/views/admin/product/edit.php**

<**div** class="page-heading">

    <**div** class="page-title">

        <**div** class="row">

            <**div** class="col-12 col-md-6 order-md-1 order-last">

                <**h3**><?php echo $heading; ?></**h3**>

            </**div**>

            <**div** class="col-12 col-md-6 order-md-2 order-first">

                <**nav** aria-label="breadcrumb" class="breadcrumb-header float-start float-lg-end">

                    <**ol** class="breadcrumb">

                        <**li** class="breadcrumb-item">

                            <**a** href="index.html">Dashboard</**a**>

                        </**li**>

                        <**li** class="breadcrumb-item active" aria-current="page">

                            <?php echo $heading; ?>

                        </**li**>

                    </**ol**>

                </**nav**>

            </**div**>

        </**div**>

    </**div**>

    <**section** class="section">

        <**div** class="card">

            <**div** class="card-header">

                <**h3**>Edit Product</**h3**>

            </**div**>

            <**div** class="card-body">

                <?= form\_error("name"); ?>

                <?= form\_error("category\_id"); ?>

                <?= form\_error("description"); ?>

                <?= form\_error("price"); ?>

                <?= form\_error("countInStock"); ?>

                <?= form\_error("image"); ?>

                <?php if (!empty($product)) { ?>

                    <?php foreach ($product as $row) { ?>

                        <**form** action="<?= base\_url("admin/product/edit"); ?>" method="POST" enctype="multipart/form-data">

                            <**input** type="hidden" name="product\_id" value="<?php echo $row->product\_id ?>">

                            <**label** for="name\_produk">Nama Produk: </**label**>

                            <**div** class="form-group">

                                <**input** id="name\_produk" type="text" name="name" value="<?php echo $row->name ?>" placeholder="Nama Produk" class="form-control" />

                            </**div**>

                            <**div** class="mb-3">

                                <**label** for="image" class="form-label">Gambar</**label**>

                                <**input** type="file" name="image" class="form-control" id="image">

                            </**div**>

                            <**div** class="form-group">

                                <**label** for="current\_image" class="form-label">Gambar Saat Ini</**label**>

                                <**br**>

                                <?php if ($row->image\_product) { ?>

                                    <**img** src="<?= base\_url('assets/img/upload/product/' . $row->image\_product) ?>" alt="Current Image" style="width: 200px;">

                                <?php } else { ?>

                                    <**span**>Tidak ada gambar</**span**>

                                <?php } ?>

                            </**div**>

                            <**label** for="category\_id">ID Kategori: </**label**>

                            <**div** class="form-group">

                                <**select** name="category\_id" class="form-control">

                                    <**option** value="">Pilih Kategori</**option**>

                                    <?php foreach ($category as $k) { ?>

                                        <**option** value="<?= $k->category\_id; ?>"><?= $k->nama\_kategori; ?></**option**>

                                    <?php } ?>

                                </**select**>

                            </**div**>

                            <**label** for="description">Deskripsi: </**label**>

                            <**div** class="form-group">

                                <**textarea** id="description" name="description" placeholder="Deskripsi" class="form-control"><?php echo $row->description; ?></**textarea**>

                            </**div**>

                            <**label** for="price">Harga: </**label**>

                            <**div** class="form-group">

                                <**input** id="price" type="text" name="price" value="<?php echo $row->price ?>" placeholder="Harga" class="form-control" />

                            </**div**>

                            <**label** for="countInStock">Jumlah Stok: </**label**>

                            <**div** class="form-group">

                                <**input** id="countInStock" type="text" name="countInStock" value="<?php echo $row->countInStock; ?>" placeholder="Jumlah Stok" class="form-control" />

                            </**div**>

                            <**button** type="submit" name="submit" class="btn btn-primary">Submit</**button**>

                        </**form**>

                    <?php } ?>

                <?php } ?>

            </**div**>

        </**div**>

    </**section**>

</**div**>

<div class="page-heading">: Kontainer untuk tampilan judul halaman dan breadcrumb.

<div class="page-title">: Bagian judul halaman.

<div class="row">: Baris untuk mengatur tata letak elemen di dalam judul halaman.

<div class="col-12 col-md-6 order-md-1 order-last">: Kolom untuk menampilkan judul halaman. Dalam contoh ini, judul halaman diperoleh dari variabel $heading.

<h3><?php echo $heading; ?></h3>: Menampilkan judul halaman yang diambil dari variabel $heading.

<div class="col-12 col-md-6 order-md-2 order-first">: Kolom untuk menampilkan breadcrumb (navigasi hierarki).

<nav aria-label="breadcrumb" class="breadcrumb-header float-start float-lg-end">: Navigasi breadcrumb.

<ol class="breadcrumb">: Daftar breadcrumb.

<li class="breadcrumb-item">: Item breadcrumb.

<a href="index.html">Dashboard</a>: Tautan ke halaman dashboard.

<li class="breadcrumb-item active" aria-current="page">: Item aktif breadcrumb yang menampilkan judul halaman.

<div class="section">: Kontainer untuk bagian edit produk.

<div class="card">: Kartu untuk mengatur konten edit produk.

<div class="card-header">: Bagian header kartu dengan judul "Edit Product".

<div class="card-body">: Bagian isi kartu yang berisi form untuk mengedit produk.

<?= form\_error("name"); ?>: Menampilkan pesan error jika terdapat kesalahan validasi pada input "name".

<?= form\_error("category\_id"); ?>: Menampilkan pesan error jika terdapat kesalahan validasi pada input "category\_id".

<?= form\_error("description"); ?>: Menampilkan pesan error jika terdapat kesalahan validasi pada input "description".

<?= form\_error("price"); ?>: Menampilkan pesan error jika terdapat kesalahan validasi pada input "price".

<?= form\_error("countInStock"); ?>: Menampilkan pesan error jika terdapat kesalahan validasi pada input "countInStock".

<?= form\_error("image"); ?>: Menampilkan pesan error jika terdapat kesalahan validasi pada input "image".

<?php if (!empty($product)) { ?>: Mengecek apakah data produk tidak kosong sebelum menampilkan form edit.

<?php foreach ($product as $row) { ?>: Melakukan iterasi untuk setiap produk yang akan diedit.

<form action="<?= base\_url("admin/product/edit"); ?>" method="POST" enctype="multipart/form-data">: Formulir untuk mengirimkan data edit produk ke URL "admin/product/edit".

<input type="hidden" name="product\_id" value="<?php echo $row->product\_id ?>">: Input tersembunyi untuk menyimpan ID produk yang diedit.

<input id="name\_produk" type="text" name="name" value="<?php echo $row->name ?>" placeholder="Nama Produk" class="form-control" />: Input teks untuk mengedit nama produk.

<input type="file" name="image" class="form-control" id="image">: Input file untuk mengedit gambar produk.

<select name="category\_id" class="form-control">: Dropdown untuk memilih kategori produk yang akan diedit.

<textarea id="description" name="description" placeholder="Deskripsi" class="form-control"><?php echo $row->description; ?></textarea>: Input teks area untuk mengedit deskripsi produk.

<input id="price" type="text" name="price" value="<?php echo $row->price ?>" placeholder="Harga" class="form-control" />: Input teks untuk mengedit harga produk.

<input id="countInStock" type="text" name="countInStock" value="<?php echo $row->countInStock; ?>" placeholder="Jumlah Stok" class="form-control" />: Input teks untuk mengedit jumlah stok produk.

<button type="submit" name="submit" class="btn btn-primary">Submit</button>: Tombol untuk mengirimkan form edit produk.

**File application/models/ModelUser.php**

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class ModelUser extends CI\_Model

{

    public function findAll()

    {

        try {

            $query = $this->db->get('user');

            $result = $query->result();

            return $result;

        } catch (Exception $e) {

            echo $e->getMessage();

        }

    }

    public function findById($user\_id)

    {

        try {

            $this->db->where('user\_id', $user\_id);

            $query = $this->db->get('user');

            $result = $query->result();

*// Debugging statement*

            return $result;

        } catch (Exception $e) {

            echo $e->getMessage();

        }

    }

    public function update($user\_id, $firstname, $lastname, $email, $password)

    {

        $this->db->where('user\_id', $user\_id);

        $data = [

            'firstname' => $firstname,

            'lastname' => $lastname,

            'email' => $email,

            'password' => password\_hash($password, PASSWORD\_BCRYPT)

        ];

        $this->db->update('user', $data);

        return true;

    }

    public function deleteUser($user\_id)

    {

        $this->db->where('user\_id', $user\_id);

        $this->db->delete('user');

        return true;

    }

    public function countUser()

    {

        return $this->db->count\_all('user');

    }

    public function findByEmail($email)

    {

        $this->db->where('email', $email);

        $query = $this->db->get('user');

        if ($query->num\_rows() > 0) {

            return $query->row();

        } else {

            return false;

        }

    }

    public function getImage($user\_id){

        $this->db->select('image');

        $this->db->from('user');

        $this->db->where('user\_id', $user\_id);

        $query = $this->db->get();

        if ($query->num\_rows() > 0) {

            $row = $query->row();

            return $row->image;

        } else {

            return null;

        }

    }

    public function updateProfile($user\_id, $firstname, $lastname, $image, $email){

        $this->db->where('user\_id', $user\_id);

        $data = [

            'email' => $email,

            'firstname' => $firstname,

            'lastname' => $lastname,

            'image' => $image

        ];

        $this->db->update('user', $data);

        return true;

    }

}

findAll(): Mengambil semua data user dari tabel "user" dan mengembalikan hasilnya dalam bentuk array.

findById($user\_id): Mengambil data user berdasarkan ID user yang diberikan dan mengembalikan hasilnya dalam bentuk array.

update($user\_id, $firstname, $lastname, $email, $password): Memperbarui data user berdasarkan ID user yang diberikan. Data yang diperbarui termasuk firstname, lastname, email, dan password yang di-hash menggunakan password\_hash(). Fungsi ini mengembalikan true jika pembaruan berhasil.

deleteUser($user\_id): Menghapus data user berdasarkan ID user yang diberikan. Fungsi ini mengembalikan true jika penghapusan berhasil.

countUser(): Menghitung jumlah total user yang ada dalam tabel "user" dan mengembalikan hasilnya.

findByEmail($email): Mencari data user berdasarkan alamat email yang diberikan. Jika ditemukan, fungsi ini mengembalikan objek hasil query, jika tidak ditemukan, mengembalikan false.

getImage($user\_id): Mengambil nama file gambar user berdasarkan ID user yang diberikan. Fungsi ini mengembalikan nama file gambar atau null jika tidak ada gambar yang terkait dengan user tersebut.

updateProfile($user\_id, $firstname, $lastname, $image, $email): Memperbarui profil user berdasarkan ID user yang diberikan. Data yang diperbarui termasuk email, firstname, lastname, dan image. Fungsi ini mengembalikan true jika pembaruan berhasil.

**File application/controllers/admin/User.php**

public function index()

    {

        $this->form\_validation->set\_rules(

            'firstname',

            'Firstname',

            'required|min\_length[6]',

            [

                'required' => "<div class='alert alert-danger' role='alert'>Firstname Harus di Isi</div>",

                'min\_length' => "<div class='alert alert-danger' role='alert'>Firstname Terlalu Pendek</div>"

            ]

        );

        $this->form\_validation->set\_rules(

            'lastname',

            'Lastname',

            'required|min\_length[6]',

            [

                'required' => "<div class='alert alert-danger' role='alert'>Lastname Harus di Isi</div>",

                'min\_length' => "<div class='alert alert-danger' role='alert'>Lastname Terlalu Pendek</div>"

            ]

        );

        $this->form\_validation->set\_rules(

            'email',

            'Email',

            'required|min\_length[6]',

            [

                'required' => "<div class='alert alert-danger' role='alert'>Email Harus di Isi</div>",

                'min\_length' => "<div class='alert alert-danger' role='alert'>Email Terlalu Pendek</div>"

            ]

        );

        $this->form\_validation->set\_rules(

            'password',

            'Password',

            'required|min\_length[6]',

            [

                'required' => "<div class='alert alert-danger' role='alert'>Password Harus di Isi</div>",

                'min\_length' => "<div class='alert alert-danger' role='alert'>Password Terlalu Pendek</div>"

            ]

        );

        if ($this->form\_validation->run() != true) {

            $data['title'] = 'User - Ecommerce Admin';

            $data['heading'] = 'User';

            $data['subview'] = 'admin/user/index';

            $data['user'] = $this->ModelUser->findAll();

            $this->session->set\_flashdata('pesan', 'Proses Login User Berhasil');

            $this->load->view('admin/\_layout', $data);

        } else {

            $firstname = $this->input->post('firstname');

            $lastname = $this->input->post('lastname');

            $email = $this->input->post('email');

            $password = $this->input->post('password');

            if ($this->ModelAuth->register($firstname, $lastname, $email, $password) == true) {

                $this->session->set\_flashdata('success\_user', 'Proses membuat User Berhasil');

                redirect("admin/user");

            } else {

                $this->session->set\_flashdata('error\_users', 'Error pembuatan users');

                redirect("admin/user", 'refresh');

            }

        }

    }

Bagian pertama (dalam blok if) adalah validasi formulir menggunakan pustaka form\_validation dalam CodeIgniter. Beberapa aturan validasi diterapkan untuk setiap field formulir seperti firstname, lastname, email, dan password. Jika validasi tidak berhasil (method run() mengembalikan false), maka halaman akan dirender dengan pesan kesalahan validasi dan data user yang ada.

Bagian kedua (dalam blok else) dijalankan ketika validasi formulir berhasil (method run() mengembalikan true). Data dari formulir (firstname, lastname, email, password) akan diambil menggunakan $this->input->post() dan kemudian diproses. Fungsi $this->ModelAuth->register() dipanggil untuk membuat user baru dengan menggunakan data yang diinputkan. Jika pembuatan user berhasil, pesan sukses akan disimpan dalam flash data session dan pengguna akan diarahkan kembali ke halaman daftar user. Jika ada kesalahan dalam proses pembuatan user, pesan error akan disimpan dalam flash data session dan pengguna akan diarahkan kembali ke halaman daftar user.

 public function edit()

    {

        $this->form\_validation->set\_rules(

            'firstname',

            'Firstname',

            'required|min\_length[6]',

            [

                'required' => "<div class='alert alert-danger' role='alert'>Firstname Harus di Isi</div>",

                'min\_length' => "<div class='alert alert-danger' role='alert'>Firstname Terlalu Pendek</div>"

            ]

        );

        $this->form\_validation->set\_rules(

            'lastname',

            'Lastname',

            'required|min\_length[6]',

            [

                'required' => "<div class='alert alert-danger' role='alert'>Lastname Harus di Isi</div>",

                'min\_length' => "<div class='alert alert-danger' role='alert'>Lastname Terlalu Pendek</div>"

            ]

        );

        $this->form\_validation->set\_rules(

            'email',

            'Email',

            'required|min\_length[6]',

            [

                'required' => "<div class='alert alert-danger' role='alert'>Email Harus di Isi</div>",

                'min\_length' => "<div class='alert alert-danger' role='alert'>Email Terlalu Pendek</div>"

            ]

        );

        $this->form\_validation->set\_rules(

            'password',

            'Password',

            'required|min\_length[6]',

            [

                'required' => "<div class='alert alert-danger' role='alert'>Password Harus di Isi</div>",

                'min\_length' => "<div class='alert alert-danger' role='alert'>Password Terlalu Pendek</div>"

            ]

        );

        $id = $this->uri->segment(4);

        if ($this->form\_validation->run() != true) {

            $data['heading'] = 'User';

            $data['user'] = $this->ModelUser->findById($id);

            $data['subview'] = 'admin/user/edit';

            $this->load->view('admin/\_layout', $data);

        } else {

            $id = $this->input->post('id');

            $firstname = $this->input->post('firstname');

            $lastname = $this->input->post('lastname');

            $email = $this->input->post('email');

            $password = $this->input->post('password');

            if ($this->ModelUser->update($id, $firstname, $lastname, $email, $password) == true) {

                $this->session->set\_flashdata('success\_user', 'Proses update User Berhasil');

                redirect("admin/user");

            } else {

                $this->session->set\_flashdata('error\_users', 'Error update users');

                redirect("admin/user", 'refresh');

            }

        }

    }

Bagian pertama (dalam blok if) adalah validasi formulir menggunakan pustaka form\_validation dalam CodeIgniter. Aturan validasi yang sama seperti di fungsi index() diterapkan untuk setiap field formulir. Jika validasi tidak berhasil (method run() mengembalikan false), maka halaman edit user dengan pesan validasi akan ditampilkan. Data user dengan ID yang diterima dari URI ($this->uri->segment(4)) akan diambil menggunakan $this->ModelUser->findById($id) dan kemudian ditampilkan pada halaman edit.

Bagian kedua (dalam blok else) dijalankan ketika validasi formulir berhasil (method run() mengembalikan true). Data dari formulir (id, firstname, lastname, email, password) akan diambil menggunakan $this->input->post() dan kemudian diproses. Fungsi $this->ModelUser->update() dipanggil untuk memperbarui data user dengan menggunakan data yang diinputkan. Jika pembaruan berhasil, pesan sukses akan disimpan dalam flash data session dan pengguna akan diarahkan kembali ke halaman daftar user. Jika ada kesalahan dalam proses pembaruan, pesan error akan disimpan dalam flash data session dan pengguna akan diarahkan kembali ke halaman daftar user.

 public function laporan\_user\_pdf()

    {

        $this->load->library('Dompdf\_gen');

        $data['user'] = $this->ModelUser->findAll();

        $this->load->view('admin/user/export\_pdf', $data);

        $paper = 'A4';

        $orien = 'landscape';

        $html = $this->output->get\_output();

        $this->dompdf->set\_paper($paper, $orien);

        $this->dompdf->load\_html($html);

        $this->dompdf->render();

        $this->dompdf->stream('laporan\_data\_user.pdf');

    }

Pertama, pustaka Dompdf di-load menggunakan $this->load->library('Dompdf\_gen'). Pustaka ini digunakan untuk menghasilkan file PDF.

Data pengguna ($data['user']) diambil menggunakan metode $this->ModelUser->findAll(). Data ini akan digunakan untuk mengisi laporan PDF.

Kemudian, view admin/user/export\_pdf di-load dengan mengirimkan data pengguna sebagai parameter ($data). View ini berisi struktur HTML yang akan dikonversi menjadi file PDF.

Selanjutnya, ukuran kertas ($paper) dan orientasi ($orien) untuk laporan PDF diatur. Dalam contoh ini, ukuran kertas diatur sebagai A4 dan orientasinya adalah landscape.

HTML yang dihasilkan dari view sebelumnya ($html) dimuat oleh pustaka Dompdf menggunakan $this->dompdf->load\_html($html).

Setelah itu, pustaka Dompdf merender HTML menjadi file PDF dengan menggunakan $this->dompdf->render().

Terakhir, laporan PDF ditampilkan kepada pengguna menggunakan $this->dompdf->stream('laporan\_data\_user.pdf'). Pengguna akan melihat dialog untuk mengunduh file PDF dengan nama laporan\_data\_user.pdf.

public function export\_excel(){

        $data = array(

            'title' => 'Laporan User',

            'user' =>  $this->ModelUser->findAll()

        );

        $this->load->view('admin/user/export\_excel\_user', $data);

    }

Pertama, data yang akan digunakan untuk laporan Excel disiapkan dalam bentuk array. Array ini memiliki dua elemen, yaitu 'title' yang berisi judul laporan dan 'user' yang berisi data pengguna yang akan diekspor. Data pengguna tersebut diambil menggunakan $this->ModelUser->findAll().

Selanjutnya, view admin/user/export\_excel\_user di-load dengan mengirimkan data sebagai parameter ($data). View ini berisi logika untuk menghasilkan file Excel dengan menggunakan library atau metode yang sesuai.

Pada view tersebut, Anda akan menemukan kode yang bertanggung jawab untuk menghasilkan file Excel berdasarkan data yang diterima.

 public function delete()

    {

        try {

            $id = $this->uri->segment(4);

            if (!is\_numeric($id)) {

                throw new Exception('Invalid user ID');

            }

            $user = $this->ModelUser->findById($id);

            if (!$user) {

                throw new Exception('User not found');

            }

            $result = $this->ModelUser->deleteUser($id);

            if ($result) {

                $this->session->set\_flashdata('success\_user', 'User deleted successfully');

                redirect("admin/user");

            } else {

                $this->session->set\_flashdata('error\_user', 'Failed to delete User');

                redirect("admin/user", 'refresh');

            }

        } catch (Exception $e) {

            $this->session->set\_flashdata('error\_user', $e->getMessage());

            redirect("admin/user", 'refresh');

        }

    }

Mengambil ID user yang akan dihapus menggunakan $this->uri->segment(4). Jika ID tersebut bukan angka, maka akan dilemparkan exception "Invalid user ID".

Memeriksa apakah user dengan ID tersebut ada dalam database menggunakan $this->ModelUser->findById($id). Jika user tidak ditemukan, maka akan dilemparkan exception "User not found".

Memanggil metode deleteUser($id) pada model ModelUser untuk menghapus user dari database.

Jika penghapusan berhasil, maka akan diset flash data 'success\_user' dengan pesan "User deleted successfully" dan redirect ke halaman admin/user.

Jika terjadi kesalahan saat penghapusan, maka akan diset flash data 'error\_user' dengan pesan "Failed to delete User" dan redirect ke halaman admin/user.

Jika terjadi exception, misalnya saat validasi ID atau user tidak ditemukan, maka akan ditangkap oleh blok catch dan flash data 'error\_user' akan diset dengan pesan exception yang terjadi. Kemudian, akan redirect ke halaman admin/user

**File application/views/ admin/user/index.php**

<**div** class="page-heading">

    <**div** class="page-title">

        <**div** class="row">

            <**div** class="col-12 col-md-6 order-md-1 order-last">

                <**h3**><?php echo $heading; ?></**h3**>

                <?php if (validation\_errors()) { ?>

                    <**div** class="alert alert-danger alert-dismissible fade show" role="alert">

                        <?= validation\_errors(); ?>

                        <**button** type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></**button**>

                    </**div**>

                <?php } ?>

                <?php

                if ($this->session->flashdata('error\_users') != '') {

                    echo '<div class="alert alert-danger alert-dismissible fade show" role="alert">';

                    echo $this->session->flashdata('error\_users');

                    echo '<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>';

                    echo '</div>';

                }

                ?>

                <?php

                if ($this->session->flashdata('success\_user') != '') {

                    echo '<div class="alert alert-success alert-dismissible fade show" role="alert">';

                    echo $this->session->flashdata('success\_user');

                    echo '<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>';

                    echo '</div>';

                }

                ?>

            </**div**>

            <**div** class="col-12 col-md-6 order-md-2 order-first">

                <**nav** aria-label="breadcrumb" class="breadcrumb-header float-start float-lg-end">

                    <**ol** class="breadcrumb">

                        <**li** class="breadcrumb-item">

                            <**a** href="index.html">Dashboard</**a**>

                        </**li**>

                        <**li** class="breadcrumb-item active" aria-current="page">

                            <?php echo $heading; ?>

                        </**li**>

                    </**ol**>

                </**nav**>

            </**div**>

        </**div**>

    </**div**>

    <**section** class="section">

        <**div** class="card">

            <**div** class="card-header">

                <**button** type="button" class="btn btn-primary btn-sm mb-3" data-bs-toggle="modal" data-bs-target="#user"><**i** class="fas fa-user"></**i**> Add User</**button**>

                <**a** href="<?= base\_url('admin/user/laporan\_user\_pdf'); ?>" class="btn btn-warning btn-sm mb-3"><**i** class="far fa-file-pdf"></**i**> Download Pdf</**a**>

                <**a** href="<?= base\_url('admin/user/export\_excel'); ?>" class="btn btn-success btn-sm mb-3"><**i** class="far fa-file-excel"></**i**> Export ke Excel</**a**>

            </**div**>

            <**div** class="card-body">

                <**table** class="table table-striped" id="table1">

                    <**thead**>

                        <**tr**>

                            <**th**>No</**th**>

                            <**th**>Firstname</**th**>

                            <**th**>Lastname</**th**>

                            <**th**>Email</**th**>

                            <**th**>Created\_at</**th**>

                            <**th**>Updated\_at</**th**>

                            <**th**>Status</**th**>

                        </**tr**>

                    </**thead**>

                    <**tbody**>

                        <?php

                        if (!empty($user)) {

                            foreach ($user as $row) {

                        ?>

                                <**tr**>

                                    <**td**><?php echo $row->user\_id ?></**td**>

                                    <**td**><?php echo $row->firstname ?></**td**>

                                    <**td**><?php echo $row->lastname ?></**td**>

                                    <**td**><?php echo $row->email ?></**td**>

                                    <**td**><?php echo $row->created\_at ?></**td**>

                                    <**td**><?php echo $row->updated\_at ?></**td**>

                                    <**td** width="250">

                                        <**a** href="<?php echo site\_url('admin/user/edit/') . $row->user\_id; ?>" class="btn btn-success">Edit</**a**>

                                        <**a** onclick="return confirm('Are you sure you want to delete this record?');" href="<?php echo site\_url('admin/user/delete/' . $row->user\_id); ?>" class="btn btn-danger">Delete</**a**>

                                    </**td**>

                                </**tr**>

                        <?php

                            }

                        }

                        ?>

                    </**tbody**>

                </**table**>

            </**div**>

        </**div**>

    </**section**>

</**div**>

Pada bagian atas halaman, terdapat judul halaman yang diberikan oleh variabel $heading.

Jika terdapat kesalahan validasi (validation errors), akan ditampilkan pesan kesalahan dalam bentuk div dengan class alert alert-danger.

Jika terdapat flash data dengan key 'error\_users', akan ditampilkan pesan kesalahan dalam bentuk div dengan class alert alert-danger.

Jika terdapat flash data dengan key 'success\_user', akan ditampilkan pesan sukses dalam bentuk div dengan class alert alert-success.

Bagian breadcrumb menampilkan breadcrumb (navigasi hierarki) yang terdiri dari "Dashboard" dan judul halaman.

Di bagian card header, terdapat tiga tombol aksi:

Tombol "Add User" untuk menambahkan pengguna baru. Tombol ini membuka modal dengan form tambah pengguna.

Tombol "Download Pdf" untuk mengunduh laporan user dalam format PDF.

Tombol "Export ke Excel" untuk mengexport data user ke dalam format Excel.

Di bagian card body, terdapat tabel yang menampilkan data pengguna dalam bentuk tabel. Setiap baris menampilkan informasi pengguna seperti firstname, lastname, email, created\_at, updated\_at, dan status. Setiap baris juga memiliki tombol "Edit" dan "Delete" untuk mengedit atau menghapus pengguna tersebut.

**File application/views/ admin/user/modal.php**

<**div** class="modal fade text-left" id="user" tabindex="-1" role="dialog" aria-labelledby="user" aria-hidden="true">

    <**div** class="modal-dialog modal-dialog-centered modal-dialog-scrollable modal-lg" role="document">

        <**div** class="modal-content">

            <**div** class="modal-header">

                <**h4** class="modal-title" id="user">

                    Add User

                </**h4**>

                <**button** type="button" class="close" data-bs-dismiss="modal" aria-label="Close">

                    <**i** data-feather="x"></**i**>

                </**button**>

            </**div**>

            <**form** action="<?= base\_url('admin/user'); ?>" method="POST">

                <**div** class="modal-body">

                    <**label** for="firstname">Firstname: </**label**>

                    <**div** class="form-group">

                        <**input** id="firstname" type="text" name="firstname" placeholder="Firstname" class="form-control" />

                    </**div**>

                    <**label** for="lastname">Lastname: </**label**>

                    <**div** class="form-group">

                        <**input** id="lastname" type="text" name="lastname" placeholder="Lastname" class="form-control" />

                    </**div**>

                    <**label** for="email">Email: </**label**>

                    <**div** class="form-group">

                        <**input** id="email" type="text" name="email" placeholder="Email Address" class="form-control" />

                    </**div**>

                    <**label** for="password">Password: </**label**>

                    <**div** class="form-group">

                        <**input** id="password" type="password" name="password" placeholder="Password" class="form-control" />

                    </**div**>

                </**div**>

                <**div** class="modal-footer">

                    <**button** type="button" class="btn btn-light-secondary" data-bs-dismiss="modal">

                        <**i** class="bx bx-x d-block d-sm-none"></**i**>

                        <**span** class="d-none d-sm-block">Close</**span**>

                    </**button**>

                    <**button** type="submit" class="btn btn-primary ms-1" data-bs-dismiss="modal">

                        <**i** class="bx bx-check d-block d-sm-none"></**i**>

                        <**span** class="d-none d-sm-block">Adding</**span**>

                    </**button**>

                </**div**>

            </**form**>

        </**div**>

    </**div**>

</**div**>

Modal ini memiliki ID "user" yang akan digunakan untuk mengidentifikasi modal tersebut.

Modal ini memiliki judul "Add User" yang ditampilkan di bagian header modal.

Terdapat tombol "Close" yang digunakan untuk menutup modal. Tombol ini memiliki atribut data-bs-dismiss="modal" untuk mengaktifkan fungsi penutupan modal.

Form di dalam modal menggunakan metode POST dan action URL-nya adalah base\_url('admin/user'). Artinya, saat form ini disubmit, data akan dikirim ke URL tersebut.

Di dalam form terdapat beberapa input field:

Input field dengan ID "firstname" digunakan untuk memasukkan firstname pengguna.

Input field dengan ID "lastname" digunakan untuk memasukkan lastname pengguna.

Input field dengan ID "email" digunakan untuk memasukkan alamat email pengguna.

Input field dengan ID "password" digunakan untuk memasukkan password pengguna. Perhatikan bahwa tipe input field ini adalah "password" sehingga karakter yang dimasukkan akan disembunyikan.

Di bagian footer modal terdapat dua tombol:

Tombol "Close" digunakan untuk menutup modal. Tombol ini memiliki atribut data-bs-dismiss="modal" untuk mengaktifkan fungsi penutupan modal.

Tombol "Adding" digunakan untuk menyimpan data pengguna baru. Tombol ini memiliki atribut type="submit" yang akan memicu submit form saat ditekan.

**File application/views/ admin/user/edit.php**

<**div** class="page-heading">

    <**div** class="page-title">

        <**div** class="row">

            <**div** class="col-12 col-md-6 order-md-1 order-last">

                <**h3**><?php echo $heading; ?></**h3**>

            </**div**>

            <**div** class="col-12 col-md-6 order-md-2 order-first">

                <**nav** aria-label="breadcrumb" class="breadcrumb-header float-start float-lg-end">

                    <**ol** class="breadcrumb">

                        <**li** class="breadcrumb-item">

                            <**a** href="index.html">Dashboard</**a**>

                        </**li**>

                        <**li** class="breadcrumb-item active" aria-current="page">

                            <?php echo $heading; ?>

                        </**li**>

                    </**ol**>

                </**nav**>

            </**div**>

        </**div**>

    </**div**>

    <**section** class="section">

        <**div** class="card">

            <**div** class="card-header">

                <**h3**>Edit User</**h3**>

            </**div**>

            <**div** class="card-body">

                <?= form\_error("firstname"); ?>

                <?= form\_error("lastname"); ?>

                <?= form\_error("email"); ?>

                <?= form\_error("password"); ?>

                <?php if (!empty($user)) { ?>

                    <?php foreach ($user as $row) { ?>

                        <**form** action="<?= base\_url(); ?>admin/user/edit" method="POST">

                            <**input** type="hidden" name="id" value="<?php echo $row->user\_id ?>">

                            <**div** class="mb-3">

                                <**label** for="firstname" class="form-label">Firstname</**label**>

                                <**input** type="text" name="firstname" value="<?php echo $row->firstname ?>" class="form-control" id="nama" aria-describedby="emailHelp">

                            </**div**>

                            <**div** class="mb-3">

                                <**label** for="lastname" class="form-label">Lastname</**label**>

                                <**input** type="text" name="lastname" value="<?php echo $row->lastname ?>" class="form-control" id="nama" aria-describedby="emailHelp">

                            </**div**>

                            <**div** class="mb-3">

                                <**label** for="email" class="form-label">Email</**label**>

                                <**input** type="email" name="email" value="<?php echo $row->email ?>" class="form-control" id="nama" aria-describedby="emailHelp">

                            </**div**>

                            <**div** class="mb-3">

                                <**label** for="password" class="form-label">Password</**label**>

                                <**input** type="password" name="password" value="<?php echo $row->password ?>" class="form-control" id="nama" aria-describedby="emailHelp">

                            </**div**>

                            <**button** type="submit" name="submit" class="btn btn-primary">Submit</**button**>

                        </**form**>

                    <?php } ?>

                <?php } ?>

            </**div**>

        </**div**>

    </**section**>

</**div**>

Tampilan ini menampilkan judul halaman "Edit User" di bagian header.

Terdapat breadcrumb navigation yang menunjukkan jalur navigasi dari halaman utama (Dashboard) ke halaman saat ini (Edit User).

Bagian card-header menampilkan judul "Edit User".

Bagian card-body berisi form untuk mengedit data pengguna.

Jika terdapat error dalam validasi form, pesan error akan ditampilkan menggunakan form\_error().

Jika terdapat data pengguna ($user tidak kosong), maka form edit akan ditampilkan. Setiap input field akan diisi dengan nilai yang sesuai dari data pengguna.

Form tersebut menggunakan metode POST dan action URL-nya adalah base\_url(); ?>admin/user/edit. Artinya, saat form ini disubmit, data akan dikirim ke URL tersebut.

Di dalam form terdapat beberapa input field:

Input field untuk memasukkan firstname pengguna, dengan nilai awal diisi sesuai data pengguna.

Input field untuk memasukkan lastname pengguna, dengan nilai awal diisi sesuai data pengguna.

Input field untuk memasukkan alamat email pengguna, dengan nilai awal diisi sesuai data pengguna.

Input field untuk memasukkan password pengguna, dengan nilai awal diisi sesuai data pengguna. Perhatikan bahwa tipe input field ini adalah "password" sehingga karakter yang dimasukkan akan disembunyikan.

Terdapat tombol "Submit" yang akan mengirimkan data yang diubah saat ditekan.

**File application/views/ admin/user/export\_pdf.php**

<**style**>

    .container-fluid {

        margin-top: 20px;

    }

    .btn {

        margin-right: 10px;

    }

    table {

        width: 100%;

        border-collapse: collapse;

    }

    th, td {

        padding: 8px;

        text-align: left;

        border-bottom: 1px solid #ddd;

    }

    th {

        background-color: #f2f2f2;

    }

</**style**>

*<!-- Begin Page Content -->*

<**div** class="container-fluid">

    <**div** class="row">

        <**div** class="col-lg-12">

            <**a** href="<?= base\_url('admin/user/laporan\_user\_pdf'); ?>" class="btn btn-warning btn-sm mb-3"><**i** class="far fa-file-pdf"></**i**> Download Pdf</**a**>

            <**a** href="<?= base\_url('admin/user/export\_excel'); ?>" class="btn btn-success btn-sm mb-3"><**i** class="far fa-file-excel"></**i**> Export ke Excel</**a**>

            <**table** class="table table-hover">

                <**thead**>

                    <**tr**>

                        <**th**>No</**th**>

                        <**th**>Firstname</**th**>

                        <**th**>Lastname</**th**>

                        <**th**>Email</**th**>

                    </**tr**>

                </**thead**>

                <**tbody**>

                    <?php

                    if (!empty($user)) {

                        foreach ($user as $row) {

                    ?>

                            <**tr**>

                                <**td**><?php echo $row->user\_id ?></**td**>

                                <**td**><?php echo $row->firstname ?></**td**>

                                <**td**><?php echo $row->lastname ?></**td**>

                                <**td**><?php echo $row->email ?></**td**>

                            </**tr**>

                    <?php

                        }

                    }

                    ?>

                </**tbody**>

            </**table**>

        </**div**>

    </**div**>

</**div**>

Di dalam tag <style>, terdapat beberapa aturan CSS yang diterapkan pada elemen-elemen dalam halaman ini. Misalnya, aturan untuk memberi margin pada elemen dengan kelas container-fluid, memberi margin kanan pada elemen dengan kelas btn, dan menentukan tata letak dan gaya tabel.

Tampilan ini menggunakan Bootstrap CSS framework dengan beberapa kelas yang ditambahkan pada elemen-elemen HTML untuk menerapkan gaya Bootstrap.

Di dalam tag <div class="container-fluid">, terdapat beberapa elemen HTML untuk menampilkan tombol "Download Pdf" dan "Export ke Excel" yang akan mengarahkan pengguna ke URL yang ditentukan ketika tombol tersebut diklik.

Di dalam tag <table>, terdapat elemen <thead> dan <tbody> yang berisi baris-baris tabel.

Di dalam elemen <thead>, terdapat baris yang berisi judul kolom-kolom dalam tabel, yaitu "No", "Firstname", "Lastname", dan "Email".

Di dalam elemen <tbody>, terdapat baris-baris data pengguna yang diambil dari variabel $user menggunakan perulangan foreach. Setiap data pengguna akan ditampilkan dalam baris terpisah dengan kolom-kolom sesuai dengan judul yang telah ditentukan sebelumnya.

**File application/views/ admin/user/export\_excel\_user.php**

<?php

header('Content-type: application/vnd-ms-excel');

header('Content-Disposition: attachment; filename=' . $title . '.xls');

header('Pragma: no-cache');

header('Expires: 0');

?>

<**h3**>

    <**center**>Laporan Data User Toko Online</**center**>

</**h3**>

<**table** class="table-data">

    <**thead**>

        <**tr**>

            <**th**>No</**th**>

            <**th**>Firstname</**th**>

            <**th**>Lastname</**th**>

            <**th**>Email</**th**>

        </**tr**>

    </**thead**>

    <**tbody**>

        <?php

        if (!empty($user)) {

            foreach ($user as $row) {

        ?>

                <**tr**>

                    <**td**><?php echo $row->user\_id ?></**td**>

                    <**td**><?php echo $row->firstname ?></**td**>

                    <**td**><?php echo $row->lastname ?></**td**>

                    <**td**><?php echo $row->email ?></**td**>

            <?php

            }

        }

            ?>

    </**tbody**>

</**table**>

Baris pertama menggunakan header header() untuk mengatur header respons HTTP agar browser mengenali file ini sebagai file Excel yang akan diunduh.

Di dalam tag <h3>, terdapat judul laporan yang ditampilkan di tengah halaman.

Di dalam tag <table class="table-data">, terdapat elemen-elemen tabel yang mirip dengan kode sebelumnya. Hanya saja, di sini tidak ada kelas Bootstrap yang ditambahkan. Tabel ini akan digunakan untuk menampilkan data pengguna yang akan diunduh dalam file Excel.

Di dalam elemen <thead>, terdapat baris yang berisi judul kolom-kolom dalam tabel, yaitu "No", "Firstname", "Lastname", dan "Email".

Di dalam elemen <tbody>, terdapat baris-baris data pengguna yang diambil dari variabel $user menggunakan perulangan foreach. Setiap data pengguna akan ditampilkan dalam baris terpisah dengan kolom-kolom sesuai dengan judul yang telah ditentukan sebelumnya.

**File application/models/ModelRole.php**

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class ModelRole extends CI\_Model

{

    public function \_\_construct()

    {

        parent::\_\_construct();

    }

    public function findAll()

    {

        try {

            $sql = "SELECT \*FROM role";

            $query = $this->db->query($sql);

            $result = $query->result();

            return $result;

        } catch (Exception $e) {

            echo $e->getMessage();

        }

    }

    public function createRole($nama)

    {

        try {

            $data = [

                'role' => $nama,

            ];

            $this->db->insert('role', $data);

            return true;

        } catch (Exception $e) {

            echo $e->getMessage();

        }

    }

    public function findById($id)

    {

        try {

            $sql = "SELECT \* FROM role WHERE role\_id = ?";

            $query = $this->db->query($sql, [$id]);

            $result = $query->result();

            return $result;

        } catch (Exception $e) {

            echo $e->getMessage();

            return null;

        }

    }

    public function updateRole($id, $nama)

    {

        $data = [

            'role' => $nama,

        ];

        $this->db->where('role\_id', $id);

        $this->db->update('role', $data);

        return true;

    }

    public function deleteById($id)

    {

        try {

            $this->db->where('role\_id', $id);

            $this->db->delete('role');

            return true;

        } catch (Exception $e) {

            echo $e->getMessage();

            return false;

        }

    }

}

Di dalam konstruktor kelas ModelRole, pemanggilan parent::\_\_construct(); dilakukan untuk memanggil konstruktor kelas induk (CI\_Model).

Metode findAll() digunakan untuk mengambil semua data dari tabel "role". Query SQL dilakukan dengan menggunakan $this->db->query() yang mengeksekusi SQL SELECT statement, dan hasilnya diambil menggunakan $query->result(). Hasil query kemudian dikembalikan sebagai array objek.

Metode createRole($nama) digunakan untuk membuat entitas "role" baru dengan menggunakan $this->db->insert(). Data yang akan dimasukkan ke dalam tabel "role" diberikan dalam bentuk array asosiatif, di mana kunci array merupakan nama kolom dalam tabel dan nilainya berasal dari argumen $nama. Jika proses insert berhasil, maka metode ini mengembalikan nilai true.

Metode findById($id) digunakan untuk mencari entitas "role" berdasarkan ID. Query SQL dilakukan dengan menggunakan $this->db->query() dan nilai ID diberikan sebagai parameter pada query. Hasil query kemudian dikembalikan sebagai array objek.

Metode updateRole($id, $nama) digunakan untuk mengupdate entitas "role" berdasarkan ID. Data yang akan diupdate diberikan dalam bentuk array asosiatif, di mana kunci array merupakan nama kolom dalam tabel dan nilainya berasal dari argumen $id dan $nama. Proses update dilakukan dengan menggunakan $this->db->update() dengan kondisi WHERE berdasarkan ID. Jika proses update berhasil, maka metode ini mengembalikan nilai true.

Metode deleteById($id) digunakan untuk menghapus entitas "role" berdasarkan ID. Proses delete dilakukan dengan menggunakan $this->db->delete() dengan kondisi WHERE berdasarkan ID. Jika proses delete berhasil, maka metode ini mengembalikan nilai true.

**File application/controllers/admin/Role.php**

 public function index()

    {

        $this->form\_validation->set\_rules(

            'role',

            'Nama Role',

            'required|min\_length[6]',

            [

                'required' => "<div class='alert alert-danger' role='alert'>Nama Role Harus di Isi</div>",

                'min\_length' => "<div class='alert alert-danger' role='alert'>Nama Role Terlalu Pendek</div>"

            ]

        );

        if ($this->form\_validation->run() == false) {

            $data['title'] = 'Role - Ecommerce Admin';

            $data['heading'] = 'Role';

            $data['subview'] = 'admin/role/index';

            $data['role'] = $this->ModelRole->findAll();

            $this->load->view('admin/\_layout', $data);

        } else {

            $role = $this->input->post('role');

            $result = $this->ModelRole->createRole($role);

            if ($result) {

                $this->session->set\_flashdata('success\_role', 'Role created successfully.');

            } else {

                $this->session->set\_flashdata('error\_role', 'Failed to create role.');

            }

            redirect('admin/role');

        }

    }

Pada bagian awal, terdapat penggunaan $this->form\_validation->set\_rules() untuk mengatur aturan validasi form. Dalam contoh ini, aturan validasi diterapkan pada field 'role' dengan aturan required (wajib diisi) dan min\_length (panjang minimal 6 karakter). Jika aturan validasi tidak terpenuhi, maka pesan error akan ditampilkan.

Jika validasi form tidak berhasil ($this->form\_validation->run() == false), maka kode di dalam blok if akan dieksekusi.

Variabel $data diisi dengan beberapa data yang akan digunakan untuk rendering halaman, seperti judul, judul utama ($heading), subview yang menunjukkan file view yang akan digunakan, dan data 'role' yang diambil dari model ModelRole menggunakan metode findAll().

Kemudian, halaman view 'admin/\_layout' dimuat menggunakan $this->load->view() dengan mengirimkan data sebagai parameter.

Jika validasi form berhasil ($this->form\_validation->run() == true), maka kode di dalam blok else akan dieksekusi.

Nilai dari field 'role' diambil menggunakan $this->input->post('role') dan disimpan dalam variabel $role.

Metode createRole($role) dari model ModelRole dipanggil untuk membuat entitas "role" baru dengan nilai $role.

Jika pembuatan role berhasil ($result bernilai true), maka pesan sukses disimpan dalam flash data menggunakan $this->session->set\_flashdata().

Jika pembuatan role gagal ($result bernilai false), maka pesan error disimpan dalam flash data menggunakan $this->session->set\_flashdata().

Pengalihan (redirect) dilakukan ke halaman "Role" menggunakan redirect('admin/role').

 public function edit()

    {

        $this->form\_validation->set\_rules(

            'role',

            'Nama Role',

            'required|min\_length[6]',

            [

                'required' => "<div class='alert alert-danger' role='alert'>Nama Role Harus di Isi</div>",

                'min\_length' => "<div class='alert alert-danger' role='alert'>Nama Role Terlalu Pendek</div>"

            ]

        );

        try {

            $id = $this->uri->segment(4);

            if (!is\_numeric($id)) {

                throw new Exception('Invalid role ID');

            }

            $findById = $this->ModelRole->findById($id);

            if ($this->form\_validation->run() != true) {

                $data['title'] = 'Role - Ecommerce Admin';

                $data['heading'] = 'Role';

                $data['subview'] = 'admin/role/edit';

                $data['role'] = $findById;

                $this->load->view('admin/\_layout', $data);

            } else {

                $id\_role = $this->input->post('role\_id');

                $role = $this->input->post('role');

                $result = $this->ModelRole->updateRole($id\_role, $role);

                if ($result) {

                    $this->session->set\_flashdata('success\_role', 'Role updated successfully.');

                    redirect("admin/role");

                } else {

                    $this->session->set\_flashdata('error\_role', 'Failed to update role.');

                    redirect("admin/role", 'refresh');

                }

                redirect('admin/role');

            }

        } catch (Exception $e) {

            $this->session->set\_flashdata('error\_role', $e->getMessage());

            redirect("admin/category", 'refresh');

        }

    }

Metode ini dipanggil saat halaman "Edit Role" diakses.

Pada bagian awal, terdapat penggunaan $this->form\_validation->set\_rules() untuk mengatur aturan validasi form. Dalam contoh ini, aturan validasi diterapkan pada field 'role' dengan aturan required (wajib diisi) dan min\_length (panjang minimal 6 karakter). Jika aturan validasi tidak terpenuhi, maka pesan error akan ditampilkan.

Setelah aturan validasi, terdapat blok try-catch untuk menangani exception yang mungkin terjadi.

Pada baris berikutnya, nilai id diambil dari URI segment 4 menggunakan $this->uri->segment(4).

Dilakukan pengecekan apakah id yang didapatkan adalah angka. Jika bukan angka, maka exception akan dilempar dengan pesan 'Invalid role ID'.

Selanjutnya, menggunakan findById($id) dari model ModelRole, data role dengan id tersebut diambil.

Jika validasi form tidak berhasil ($this->form\_validation->run() != true), maka kode di dalam blok if akan dieksekusi.

Variabel $data diisi dengan beberapa data yang akan digunakan untuk rendering halaman, seperti judul, judul utama ($heading), subview yang menunjukkan file view yang akan digunakan, dan data 'role' yang diambil dari model ModelRole menggunakan metode findById($id).

Kemudian, halaman view 'admin/\_layout' dimuat menggunakan $this->load->view() dengan mengirimkan data sebagai parameter.

Jika validasi form berhasil ($this->form\_validation->run() == true), maka kode di dalam blok else akan dieksekusi.

Nilai dari field 'role\_id' dan 'role' diambil menggunakan $this->input->post() dan disimpan dalam variabel $id\_role dan $role.

Metode updateRole($id\_role, $role) dari model ModelRole dipanggil untuk mengupdate entitas "role" dengan nilai baru.

Jika pembaruan role berhasil ($result bernilai true), maka pesan sukses disimpan dalam flash data menggunakan $this->session->set\_flashdata(), dan dilakukan pengalihan (redirect) ke halaman "Role" menggunakan redirect("admin/role").

Jika pembaruan role gagal ($result bernilai false), maka pesan error disimpan dalam flash data menggunakan $this->session->set\_flashdata(), dan dilakukan pengalihan (redirect) ke halaman "Role" menggunakan redirect("admin/role", 'refresh').

Jika terjadi exception di dalam blok try, maka pesan error disimpan dalam flash data menggunakan $this->session->set\_flashdata(), dan dilakukan pengalihan (redirect) ke halaman "Category" menggunakan redirect("admin/category", 'refresh').

 public function delete()

    {

        try{

            $id = $this->uri->segment(4);

            if (!is\_numeric($id)) {

                throw new Exception('Invalid Role ID');

            }

            $role = $this->ModelRole->findById($id);

            if(!$role){

                throw new Exception('Role not found');

            }

            $result = $this->ModelRole->deleteById($id);

            if($result){

                $this->session->set\_flashdata('success\_role', 'Role deleted successfully');

                redirect("admin/role");

            }else{

                $this->session->set\_flashdata('error\_role','Failed to delete role.');

                redirect("admin/role", 'refresh');

            }

        }catch(Exception $e){

            $this->session->set\_flashdata('error\_role', $e->getMessage());

            redirect("admin/role", 'refresh');

        }

    }

Metode ini dipanggil saat pengguna mengakses halaman "Delete Role" atau saat mengirimkan permintaan penghapusan role.

Pada bagian awal, terdapat blok try-catch untuk menangani exception yang mungkin terjadi.

Pada baris berikutnya, nilai id diambil dari URI segment 4 menggunakan $this->uri->segment(4).

Dilakukan pengecekan apakah id yang didapatkan adalah angka. Jika bukan angka, maka exception akan dilempar dengan pesan 'Invalid Role ID'.

Selanjutnya, menggunakan findById($id) dari model ModelRole, data role dengan id tersebut diambil.

Jika data role tidak ditemukan (!$role), maka exception akan dilempar dengan pesan 'Role not found'.

Metode deleteById($id) dari model ModelRole dipanggil untuk menghapus entitas "role" berdasarkan id yang diberikan.

Jika penghapusan berhasil ($result bernilai true), maka pesan sukses disimpan dalam flash data menggunakan $this->session->set\_flashdata(), dan dilakukan pengalihan (redirect) ke halaman "Role" menggunakan redirect("admin/role").

Jika penghapusan gagal ($result bernilai false), maka pesan error disimpan dalam flash data menggunakan $this->session->set\_flashdata(), dan dilakukan pengalihan (redirect) ke halaman "Role" menggunakan redirect("admin/role", 'refresh').

Jika terjadi exception di dalam blok try, maka pesan error disimpan dalam flash data menggunakan $this->session->set\_flashdata(), dan dilakukan pengalihan (redirect) ke halaman "Role" menggunakan redirect("admin/role", 'refresh').

**File application/views/ admin/role/index.php**

<**div** class="page-heading">

    <**div** class="page-title">

        <**div** class="row">

            <**div** class="col-12 col-md-6 order-md-1 order-last">

                <**h3**><?php echo $heading; ?></**h3**>

                <?php if (validation\_errors()) { ?>

                    <**div** class="alert alert-danger alert-dismissible fade show" role="alert">

                        <?= validation\_errors(); ?>

                        <**button** type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></**button**>

                    </**div**>

                <?php } ?>

                <?php

                if ($this->session->flashdata('error\_role') != '') {

                    echo '<div class="alert alert-danger alert-dismissible fade show" role="alert">';

                    echo $this->session->flashdata('error\_role');

                    echo '<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>';

                    echo '</div>';

                }

                ?>

                <?php

                if ($this->session->flashdata('success\_role') != '') {

                    echo '<div class="alert alert-success alert-dismissible fade show" role="alert">';

                    echo $this->session->flashdata('success\_role');

                    echo '<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>';

                    echo '</div>';

                }

                ?>

            </**div**>

            <**div** class="col-12 col-md-6 order-md-2 order-first">

                <**nav** aria-label="breadcrumb" class="breadcrumb-header float-start float-lg-end">

                    <**ol** class="breadcrumb">

                        <**li** class="breadcrumb-item">

                            <**a** href="index.html">Dashboard</**a**>

                        </**li**>

                        <**li** class="breadcrumb-item active" aria-current="page">

                            <?php echo $heading; ?>

                        </**li**>

                    </**ol**>

                </**nav**>

            </**div**>

        </**div**>

    </**div**>

    <**section** class="section">

        <**div** class="card">

            <**div** class="card-header">

                <**button** type="button" class="btn btn-primary btn-sm mb-3" data-bs-toggle="modal" data-bs-target="#role"><**i** class="fas fa-user"></**i**> Add Role</**button**>

            </**div**>

            <**div** class="card-body">

                <**table** class="table table-striped" id="table1">

                    <**thead**>

                        <**tr**>

                            <**th**>No</**th**>

                            <**th**>Role</**th**>

                            <**th**>Created\_at</**th**>

                            <**th**>Updated\_at</**th**>

                            <**th**>Status</**th**>

                        </**tr**>

                    </**thead**>

                    <**tbody**>

                        <?php

                        if (!empty($role)) {

                            foreach ($role as $row) {

                        ?>

                                <**tr**>

                                    <**td**><?php echo $row->role\_id ?></**td**>

                                    <**td**><?php echo $row->role ?></**td**>

                                    <**td**><?php echo $row->created\_at ?></**td**>

                                    <**td**><?php echo $row->updated\_at ?></**td**>

                                    <**td** width="250">

                                        <**a** href="<?php echo site\_url('admin/role/edit/') . $row->role\_id; ?>" class="btn btn-success">Edit</**a**>

                                        <**a** onclick="return confirm('Are you sure you want to delete this record?');" href="<?php echo site\_url('admin/role/delete/' . $row->role\_id); ?>" class="btn btn-danger">Delete</**a**>

                                    </**td**>

                                </**tr**>

                        <?php

                            }

                        }

                        ?>

                    </**tbody**>

                </**table**>

            </**div**>

        </**div**>

    </**section**>

</**div**>

Tampilan halaman "Role" terdiri dari judul halaman, tampilan pesan error dan sukses (jika ada), serta tabel yang menampilkan data role.

Pada bagian judul halaman, nilai $heading digunakan untuk menampilkan judul yang sesuai.

Jika terdapat error validasi (validation\_errors()), pesan error akan ditampilkan dalam kotak peringatan (alert danger).

Jika terdapat pesan error dari flash data ($this->session->flashdata('error\_role')), pesan error tersebut akan ditampilkan dalam kotak peringatan (alert danger).

Jika terdapat pesan sukses dari flash data ($this->session->flashdata('success\_role')), pesan sukses tersebut akan ditampilkan dalam kotak peringatan (alert success).

Terdapat tombol "Add Role" yang menggunakan modal untuk menambahkan role baru.

Pada tabel, data role ditampilkan dalam baris-baris. Setiap baris menampilkan informasi seperti nomor, nama role, tanggal dibuat (created\_at), tanggal diperbarui (updated\_at), dan tombol "Edit" dan "Delete" untuk mengedit atau menghapus role tersebut.

Tombol "Edit" mengarahkan pengguna ke halaman edit role (site\_url('admin/role/edit/') . $row->role\_id) menggunakan URL segment.

Tombol "Delete" akan menampilkan konfirmasi dialog untuk memastikan pengguna ingin menghapus role tersebut. Jika pengguna menyetujui, halaman akan mengarahkan ke metode delete role (site\_url('admin/role/delete/' . $row->role\_id)).

**File application/views/ admin/role/modal.php**

<**div** class="modal fade text-left" id="role" tabindex="-1" role="dialog" aria-labelledby="role" aria-hidden="true">

    <**div** class="modal-dialog modal-dialog-centered modal-dialog-scrollable modal-lg" role="document">

        <**div** class="modal-content">

            <**div** class="modal-header">

                <**h4** class="modal-title" id="role">

                    Add Role

                </**h4**>

                <**button** type="button" class="close" data-bs-dismiss="modal" aria-label="Close">

                    <**i** data-feather="x"></**i**>

                </**button**>

            </**div**>

            <**form** action="<?= base\_url('admin/role'); ?>" method="POST">

                <**div** class="modal-body">

                    <**label** for="role">Nama Role: </**label**>

                    <**div** class="form-group">

                        <**input** id="role" type="text" name="role" placeholder="Nama Role" class="form-control" />

                    </**div**>

                </**div**>

                <**div** class="modal-footer">

                    <**button** type="button" class="btn btn-light-secondary" data-bs-dismiss="modal">

                        <**i** class="bx bx-x d-block d-sm-none"></**i**>

                        <**span** class="d-none d-sm-block">Close</**span**>

                    </**button**>

                    <**button** type="submit" class="btn btn-primary ms-1" data-bs-dismiss="modal">

                        <**i** class="bx bx-check d-block d-sm-none"></**i**>

                        <**span** class="d-none d-sm-block">Adding</**span**>

                    </**button**>

                </**div**>

            </**form**>

        </**div**>

    </**div**>

</**div**>

Modal ini akan ditampilkan saat tombol "Add Role" pada halaman "Role" ditekan.

Modal memiliki ID "role" dan diatur sebagai modal tersembunyi (aria-hidden="true").

Modal memiliki ukuran besar (modal-lg) dan diatur agar dapat digulir (modal-dialog-scrollable).

Bagian kepala modal (modal-header) berisi judul modal "Add Role" dan tombol close untuk menutup modal (data-bs-dismiss="modal").

Isi modal (modal-body) berisi form untuk menambahkan role baru.

Form mengarahkan ke URL base\_url('admin/role') dengan metode POST untuk mengirim data ke server.

Di dalam form, terdapat input teks dengan ID dan nama "role" yang digunakan untuk memasukkan nama role.

Bagian kaki modal (modal-footer) berisi tombol "Close" untuk menutup modal (data-bs-dismiss="modal") dan tombol "Adding" yang digunakan untuk menambahkan role baru (data-bs-dismiss="modal").

**File application/views/ admin/role/edit.php**

<**div** class="page-heading">

    <**div** class="page-title">

        <**div** class="row">

            <**div** class="col-12 col-md-6 order-md-1 order-last">

                <**h3**><?php echo $heading; ?></**h3**>

            </**div**>

            <**div** class="col-12 col-md-6 order-md-2 order-first">

                <**nav** aria-label="breadcrumb" class="breadcrumb-header float-start float-lg-end">

                    <**ol** class="breadcrumb">

                        <**li** class="breadcrumb-item">

                            <**a** href="index.html">Dashboard</**a**>

                        </**li**>

                        <**li** class="breadcrumb-item active" aria-current="page">

                            <?php echo $heading; ?>

                        </**li**>

                    </**ol**>

                </**nav**>

            </**div**>

        </**div**>

    </**div**>

    <**section** class="section">

        <**div** class="card">

            <**div** class="card-header">

                <**h3**>Edit Role</**h3**>

            </**div**>

            <**div** class="card-body">

                <?= form\_error("role"); ?>

                <?php if (!empty($role)) { ?>

                    <?php foreach ($role as $row) { ?>

                        <**form** action="<?= base\_url("admin/role/edit"); ?>" method="POST" enctype="multipart/form-data">

                            <**input** type="hidden" name="role\_id" value="<?php echo $row->role\_id ?>">

                            <**div** class="mb-3">

                                <**label** for="role" class="form-label">Nama Role</**label**>

                                <**input** type="text" name="role" value="<?php echo $row->role ?>" class="form-control" id="role" aria-describedby="emailHelp">

                            </**div**>

                            <**button** type="submit" name="submit" class="btn btn-primary">Submit</**button**>

                        </**form**>

                    <?php } ?>

                <?php } ?>

            </**div**>

        </**div**>

    </**section**>

</**div**>

Halaman ini akan menampilkan judul "Edit Role" di bagian atas.

Di bagian atas kanan halaman, terdapat navigasi breadcrumb yang menunjukkan jalur halaman saat ini.

Di dalam kartu (card), terdapat bagian kepala (card-header) yang berisi judul "Edit Role".

Di dalam kartu (card), terdapat bagian isi (card-body) yang berisi form untuk mengedit role.

Jika variabel $role tidak kosong, maka form akan ditampilkan untuk setiap elemen dalam variabel $role.

Form mengarahkan ke URL base\_url("admin/role/edit") dengan metode POST untuk mengirim data ke server.

Di dalam form, terdapat input teks untuk memasukkan nama role dengan nilai awal yang diambil dari variabel $row->role.

Terdapat tombol "Submit" untuk mengirim perubahan role ke server.

**File application/models/ModelSlider.php**

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class ModelSlider extends CI\_Model

{

    public function \_\_construct()

    {

        parent::\_\_construct();

    }

    public function findAll()

    {

        try {

            $sql = "SELECT \*FROM slider";

            $query = $this->db->query($sql);

            $result = $query->result();

            return $result;

        } catch (Exception $e) {

            echo $e->getMessage();

        }

    }

    public function createSlider($nama, $image)

    {

        try {

            $data = [

                'nama' => $nama,

                'image' => $image,

            ];

            $this->db->insert('slider', $data);

            return true;

        } catch (Exception $e) {

            echo $e->getMessage();

        }

    }

    public function findById($id)

    {

        try {

            $sql = "SELECT \* FROM slider WHERE slider\_id = ?";

            $query = $this->db->query($sql, [$id]);

            $result = $query->result();

            return $result;

        } catch (Exception $e) {

            echo $e->getMessage();

        }

    }

    public function updateSlider($id, $nama, $image)

    {

        $data = [

            'nama' => $nama,

            'image' => $image,

        ];

        $this->db->where('slider\_id', $id);

        $this->db->update('slider', $data);

        return true;

    }

    public function deleteById($id)

    {

        try {

            $this->db->where('slider\_id', $id);

            $this->db->delete('slider');

            return true;

        } catch (Exception $e) {

            echo $e->getMessage();

            return false;

        }

    }

    public function getImage($slider\_id)

    {

        $this->db->select('image');

        $this->db->from('slider');

        $this->db->where('slider\_id', $slider\_id);

        $query = $this->db->get();

        if ($query->num\_rows() > 0) {

            $row = $query->row();

            return $row->image;

        } else {

            return null;

        }

    }

}

Di dalam metode konstruktor (\_\_construct()), tidak ada implementasi khusus.

Metode findAll() digunakan untuk mengambil semua data slider dari tabel "slider". Query SELECT digunakan untuk mengambil semua data, dan hasilnya dikembalikan sebagai array objek.

Metode createSlider($nama, $image) digunakan untuk membuat data slider baru di tabel "slider". Data berupa nama dan gambar slider yang disimpan dalam array $data, kemudian dilakukan operasi insert menggunakan $this->db->insert().

Metode findById($id) digunakan untuk mencari data slider berdasarkan ID yang diberikan. Query SELECT digunakan dengan kondisi WHERE pada kolom slider\_id, dan hasilnya dikembalikan sebagai array objek.

Metode updateSlider($id, $nama, $image) digunakan untuk memperbarui data slider yang telah ada di tabel "slider". Data baru berupa nama dan gambar slider disimpan dalam array $data, kemudian dilakukan operasi update menggunakan $this->db->update().

Metode deleteById($id) digunakan untuk menghapus data slider berdasarkan ID yang diberikan. Operasi delete dilakukan dengan kondisi WHERE pada kolom slider\_id.

Metode getImage($slider\_id) digunakan untuk mengambil nama file gambar dari sebuah slider berdasarkan ID slider yang diberikan. Query SELECT digunakan dengan kondisi WHERE pada kolom slider\_id, dan hasilnya hanya mengambil kolom image.

**File application/controllers/admin/Slider.php**

public function index()

    {

        $this->form\_validation->set\_rules(

            'nama',

            'Nama',

            'required|min\_length[6]',

            [

                'required' => "<div class='alert alert-danger' role='alert'>Nama Slider Harus di Isi</div>",

                'min\_length' => "<div class='alert alert-danger' role='alert'>Nama Slider Terlalu Pendek</div>"

            ]

        );

        $config['upload\_path'] = './assets/img/upload/slider';

        $config['allowed\_types'] = 'jpg|png|jpeg';

        $config['max\_size'] = '3000';

        $config['max\_width'] = '3000';

        $config['max\_height'] = '3000';

        $config['file\_name'] = 'img' . time();

        $this->load->library('upload', $config);

        if ($this->form\_validation->run() != true) {

            $data['title'] = 'Slider - Ecommerce Admin';

            $data['heading'] = 'Slider';

            $data['subview'] = 'admin/slider/index';

            $data['slider'] = $this->ModelSlider->findAll();

            $this->load->view('admin/\_layout', $data);

        } else {

            if ($this->upload->do\_upload('image')) {

                $image = $this->upload->data();

                $gambar = $image['file\_name'];

            } else {

                $gambar = '';

            }

            $nama = $this->input->post('nama');

            $image = $gambar;

            if ($this->ModelSlider->createSlider($nama, $image) == true) {

                $this->session->set\_flashdata('success\_slider', 'Proses membuat Slider Berhasil');

                redirect("admin/slider");

            } else {

                $this->session->set\_flashdata('error\_slider', 'Error pembuatan slider');

                redirect("admin/slider", 'refresh');

            }

        }

    }

Di dalam fungsi ini, pertama-tama dilakukan pengaturan aturan validasi menggunakan $this->form\_validation->set\_rules(). Aturan validasi tersebut menyatakan bahwa field "nama" harus diisi dan memiliki panjang minimal 6 karakter. Jika aturan validasi tidak terpenuhi, pesan error akan ditampilkan.

Selanjutnya, dilakukan konfigurasi untuk proses upload gambar menggunakan library upload dari CodeIgniter. Konfigurasi tersebut mencakup path penyimpanan file, jenis file yang diizinkan, ukuran maksimum file, lebar maksimum gambar, tinggi maksimum gambar, dan nama file yang digunakan.

Jika aturan validasi terpenuhi, maka akan dilanjutkan dengan proses upload gambar. Jika upload berhasil ($this->upload->do\_upload()), maka nama file gambar disimpan dalam variabel $gambar. Jika upload gagal, nilai $gambar akan diisi dengan string kosong.

Selanjutnya, nilai dari field "nama" dan $gambar akan diambil dari inputan form.

Jika proses upload dan pengambilan data berhasil, maka dilakukan pemanggilan fungsi createSlider() dari model ModelSlider untuk membuat slider baru dengan menggunakan data yang telah dikumpulkan.

Jika pembuatan slider berhasil, pesan sukses akan ditampilkan melalui flash data ($this->session->set\_flashdata()) dan pengguna akan diarahkan kembali ke halaman daftar slider.

Jika terjadi error dalam pembuatan slider, pesan error akan ditampilkan melalui flash data dan pengguna akan diarahkan kembali ke halaman daftar slider.

 public function edit()

    {

        $this->form\_validation->set\_rules(

            'nama',

            'Nama',

            'required|min\_length[6]',

            [

                'required' => "<div class='alert alert-danger' role='alert'>Nama Slider Harus di Isi</div>",

                'min\_length' => "<div class='alert alert-danger' role='alert'>Nama Slider Terlalu Pendek</div>"

            ]

        );

        try {

            $id = $this->uri->segment(4);

            if (!is\_numeric($id)) {

                throw new Exception('Invalid slider ID');

            }

            $config['upload\_path'] = './assets/img/upload/slider';

            $config['allowed\_types'] = 'jpg|png|jpeg';

            $config['max\_size'] = '3000';

            $config['max\_width'] = '3000';

            $config['max\_height'] = '3000';

            $config['file\_name'] = 'img' . time();

            $this->load->library('upload', $config);

            $findById =  $this->ModelSlider->findById($id);

            if ($findById == null) {

                throw new Exception('Slider not found');

            }

            if ($this->form\_validation->run() != true) {

                $data['title'] = 'Slider - Ecommerce Admin';

                $data['heading'] = 'Slider';

                $data['subview'] = 'admin/slider/edit';

                $data['slider'] = $findById;

                $this->load->view('admin/\_layout', $data);

            } else {

                $id\_slider = $this->input->post('slider\_id');

                $nama = $this->input->post('nama');

                $old\_image = $this->ModelSlider->getImage($id\_slider);

                if (!empty($\_FILES['image']['name'])) {

                    if ($this->upload->do\_upload('image')) {

                        $imageData = $this->upload->data();

                        $image = $imageData['file\_name'];

                        if (!empty($old\_image)) {

                            unlink('./assets/img/upload/slider/' . $old\_image);

                        }

                    } else {

                        $this->session->set\_flashdata('error\_slider', 'Error uploading image');

                        redirect("admin/slider", 'refresh');

                    }

                } else {

                    $image = $old\_image;

                }

                if ($this->ModelSlider->updateSlider($id\_slider, $nama, $image) == true) {

                    $this->session->set\_flashdata('success\_slider', 'Proses update Slider Berhasil');

                    redirect("admin/slider");

                } else {

                    $this->session->set\_flashdata('error\_slider', 'Error update Slider');

                    redirect("admin/slider", 'refresh');

                }

            }

        } catch (Exception $e) {

            $this->session->set\_flashdata('error\_slider', $e->getMessage());

            redirect("admin/slider", 'refresh');

        }

    }

Di dalam fungsi ini, pertama-tama dilakukan pengaturan aturan validasi menggunakan $this->form\_validation->set\_rules(). Aturan validasi tersebut menyatakan bahwa field "nama" harus diisi dan memiliki panjang minimal 6 karakter. Jika aturan validasi tidak terpenuhi, pesan error akan ditampilkan.

Selanjutnya, dilakukan pengecekan dan pengambilan ID slider yang akan diedit dari URI segment menggunakan $this->uri->segment(). Jika ID tidak valid atau bukan angka, akan dilemparkan Exception dengan pesan "Invalid slider ID".

Kemudian, dilakukan konfigurasi untuk proses upload gambar yang sama dengan konfigurasi pada fungsi index().

Selanjutnya, dilakukan pemanggilan fungsi findById() dari model ModelSlider untuk mencari slider berdasarkan ID yang telah diambil.

Jika slider dengan ID tersebut tidak ditemukan, akan dilemparkan Exception dengan pesan "Slider not found".

Jika aturan validasi terpenuhi, maka akan dilanjutkan dengan proses pengambilan nilai dari inputan form seperti ID slider dan nama.

Selanjutnya, dilakukan pengambilan nama file gambar lama menggunakan fungsi getImage() dari model ModelSlider.

Jika ada gambar yang diunggah melalui form ($\_FILES['image']['name']), maka dilakukan proses upload gambar menggunakan $this->upload->do\_upload(). Jika upload berhasil, nama file gambar baru disimpan dalam variabel $image. Jika upload gagal, pesan error akan ditampilkan melalui flash data dan pengguna akan diarahkan kembali ke halaman daftar slider.

Jika tidak ada gambar yang diunggah melalui form, maka nilai $image akan menggunakan nama file gambar lama.

Selanjutnya, dilakukan pemanggilan fungsi updateSlider() dari model ModelSlider untuk melakukan update data slider berdasarkan ID, nama, dan gambar baru (jika ada).

Jika proses update slider berhasil, pesan sukses akan ditampilkan melalui flash data dan pengguna akan diarahkan kembali ke halaman daftar slider.

Jika terjadi error dalam proses update slider, pesan error akan ditampilkan melalui

 public function delete()

    {

        try {

            $id = $this->uri->segment(4);

            if (!is\_numeric($id)) {

                throw new Exception('Invalid Slider ID');

            }

            $slider = $this->ModelSlider->findById($id);

            if (!$slider) {

                throw new Exception('Slider not found');

            }

            $image = $this->ModelSlider->getImage($id);

            if (!empty($image)) {

                unlink('./assets/img/upload/slider/' . $image);

            }

            $result = $this->ModelCategory->deleteById($id);

            if ($result) {

                $this->session->set\_flashdata('success\_slider', 'Slider deleted successfully');

                redirect("admin/slider");

            } else {

                $this->session->set\_flashdata('error\_slider', 'Failed to delete Slider');

                redirect("admin/slider", 'refresh');

            }

        } catch (Exception $e) {

            $this->session->set\_flashdata('error\_slider', $e->getMessage());

            redirect("admin/slider", 'refresh');

        }

    }

Di dalam fungsi ini, pertama-tama dilakukan pengecekan dan pengambilan ID slider yang akan dihapus dari URI segment menggunakan $this->uri->segment(). Jika ID tidak valid atau bukan angka, akan dilemparkan Exception dengan pesan "Invalid Slider ID".

Selanjutnya, dilakukan pemanggilan fungsi findById() dari model ModelSlider untuk mencari slider berdasarkan ID yang telah diambil.

Jika slider dengan ID tersebut tidak ditemukan, akan dilemparkan Exception dengan pesan "Slider not found".

Kemudian, dilakukan pengambilan nama file gambar menggunakan fungsi getImage() dari model ModelSlider.

Jika nama file gambar tidak kosong, maka dilakukan penghapusan file gambar dari direktori menggunakan unlink().

Selanjutnya, dilakukan pemanggilan fungsi deleteById() dari model ModelCategory untuk menghapus data slider berdasarkan ID.

Jika proses penghapusan berhasil, pesan sukses akan ditampilkan melalui flash data dan pengguna akan diarahkan kembali ke halaman daftar slider.

Jika terjadi error dalam proses penghapusan slider, pesan error akan ditampilkan melalui flash data dan pengguna akan diarahkan kembali ke halaman daftar slider

**File application/views/ admin/slider/index.php**

<**div** class="page-heading">

    <**div** class="page-title">

        <**div** class="row">

            <**div** class="col-12 col-md-6 order-md-1 order-last">

                <**h3**><?php echo $heading; ?></**h3**>

                <?php if (validation\_errors()) { ?>

                    <**div** class="alert alert-danger alert-dismissible fade show" role="alert">

                        <?= validation\_errors(); ?>

                        <**button** type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></**button**>

                    </**div**>

                <?php } ?>

                <?php

                if ($this->session->flashdata('error\_slider') != '') {

                    echo '<div class="alert alert-danger alert-dismissible fade show" role="alert">';

                    echo $this->session->flashdata('error\_slider');

                    echo '<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>';

                    echo '</div>';

                }

                ?>

                <?php

                if ($this->session->flashdata('success\_slider') != '') {

                    echo '<div class="alert alert-success alert-dismissible fade show" role="alert">';

                    echo $this->session->flashdata('success\_slider');

                    echo '<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>';

                    echo '</div>';

                }

                ?>

            </**div**>

            <**div** class="col-12 col-md-6 order-md-2 order-first">

                <**nav** aria-label="breadcrumb" class="breadcrumb-header float-start float-lg-end">

                    <**ol** class="breadcrumb">

                        <**li** class="breadcrumb-item">

                            <**a** href="index.html">Dashboard</**a**>

                        </**li**>

                        <**li** class="breadcrumb-item active" aria-current="page">

                            <?php echo $heading; ?>

                        </**li**>

                    </**ol**>

                </**nav**>

            </**div**>

        </**div**>

    </**div**>

    <**section** class="section">

        <**div** class="card">

            <**div** class="card-header">

                <**button** type="button" class="btn btn-primary btn-sm mb-3" data-bs-toggle="modal" data-bs-target="#slider"><**i** class="fas fa-user"></**i**> Add Slider</**button**>

            </**div**>

            <**div** class="card-body">

                <**table** class="table table-striped" id="table1">

                    <**thead**>

                        <**tr**>

                            <**th**>No</**th**>

                            <**th**>Nama</**th**>

                            <**th**>Image</**th**>

                            <**th**>Created\_at</**th**>

                            <**th**>Updated\_at</**th**>

                            <**th**>Status</**th**>

                        </**tr**>

                    </**thead**>

                    <**tbody**>

                        <?php

                        if (!empty($slider)) {

                            foreach ($slider as $row) {

                        ?>

                                <**tr**>

                                    <**td**><?php echo $row->slider\_id ?></**td**>

                                    <**td**><?php echo $row->nama ?></**td**>

                                    <**td**>

                                        <**img** src="<?= base\_url('assets/img/upload/slider/' . $row->image) ?>" alt="Current Image" style="width: 200px;">

                                    </**td**>

                                    <**td**><?php echo $row->created\_at ?></**td**>

                                    <**td**><?php echo $row->updated\_at ?></**td**>

                                    <**td** width="250">

                                        <**a** href="<?php echo site\_url('admin/slider/edit/') . $row->slider\_id; ?>" class="btn btn-success">Edit</**a**>

                                        <**a** onclick="return confirm('Are you sure you want to delete this record?');" href="<?php echo site\_url('admin/slider/delete/' . $row->slider\_id); ?>" class="btn btn-danger">Delete</**a**>

                                    </**td**>

                                </**tr**>

                        <?php

                            }

                        }

                        ?>

                    </**tbody**>

                </**table**>

            </**div**>

        </**div**>

    </**section**>

</**div**>

Pada bagian atas, terdapat elemen <h3> yang menampilkan judul halaman ($heading) yang diterima dari controller.

Setelah itu, terdapat pengecekan apakah ada kesalahan validasi (validation errors) dari form. Jika ada, pesan kesalahan validasi akan ditampilkan dalam sebuah elemen <div> dengan kelas alert alert-danger.

Kemudian, dilakukan pengecekan apakah terdapat flash data dengan kunci 'error\_slider'. Jika ada, maka pesan kesalahan tersebut akan ditampilkan dalam elemen <div> dengan kelas alert alert-danger.

Selanjutnya, dilakukan pengecekan apakah terdapat flash data dengan kunci 'success\_slider'. Jika ada, maka pesan keberhasilan tersebut akan ditampilkan dalam elemen <div> dengan kelas alert alert-success.

Di dalam elemen <div> dengan kelas card-header, terdapat sebuah tombol "Add Slider" yang ketika diklik akan memunculkan modal untuk menambahkan slider baru.

Di dalam elemen <div> dengan kelas card-body, terdapat sebuah tabel (<table>) yang menampilkan daftar slider.

Pada bagian <thead> tabel, ditampilkan header kolom-kolom yang ada, seperti "No", "Nama", "Image", "Created\_at", "Updated\_at", dan "Status".

Pada bagian <tbody> tabel, dilakukan perulangan untuk setiap data slider yang diterima dari controller. Setiap data slider ditampilkan dalam sebuah baris tabel (<tr>) dengan kolom-kolom yang sesuai.

Pada kolom "Image", ditampilkan gambar slider dengan menggunakan tag <img>, dengan sumber gambar diambil dari direktori yang sesuai dengan base\_url() dan nama file gambar yang tersimpan dalam data slider.

Pada kolom "Status", terdapat dua tombol yaitu "Edit" dan "Delete" yang mengarahkan pengguna ke halaman edit slider atau melakukan penghapusan slider. Tombol "Delete" juga memunculkan konfirmasi penghapusan dengan menggunakan confirm().

**File application/views/ admin/slider/modal.php**

<**div** class="modal fade text-left" id="slider" tabindex="-1" role="dialog" aria-labelledby="slider" aria-hidden="true">

    <**div** class="modal-dialog modal-dialog-centered modal-dialog-scrollable modal-lg" role="document">

        <**div** class="modal-content">

            <**div** class="modal-header">

                <**h4** class="modal-title" id="slider">

                    Add Slider

                </**h4**>

                <**button** type="button" class="close" data-bs-dismiss="modal" aria-label="Close">

                    <**i** data-feather="x"></**i**>

                </**button**>

            </**div**>

            <**form** action="<?= base\_url('admin/slider'); ?>" method="POST" enctype="multipart/form-data">

                <**div** class="modal-body">

                    <**label** for="nama">Nama: </**label**>

                    <**div** class="form-group">

                        <**input** id="nama" type="text" name="nama" placeholder="Nama" class="form-control" />

                    </**div**>

                    <**label** for="image">image: </**label**>

                    <**div** class="form-group">

                        <**input** type="file" class="form-control" id="image" name="image">

                    </**div**>

                </**div**>

                <**div** class="modal-footer">

                    <**button** type="button" class="btn btn-light-secondary" data-bs-dismiss="modal">

                        <**i** class="bx bx-x d-block d-sm-none"></**i**>

                        <**span** class="d-none d-sm-block">Close</**span**>

                    </**button**>

                    <**button** type="submit" class="btn btn-primary ms-1" data-bs-dismiss="modal">

                        <**i** class="bx bx-check d-block d-sm-none"></**i**>

                        <**span** class="d-none d-sm-block">Adding</**span**>

                    </**button**>

                </**div**>

            </**form**>

        </**div**>

    </**div**>

</**div**>

Modal ini digunakan untuk menambahkan slider baru.

Modal ini ditampilkan saat tombol "Add Slider" pada halaman daftar slider ditekan.

Pada bagian header modal, terdapat judul modal yang menampilkan teks "Add Slider".

Terdapat tombol close (x) di pojok kanan atas modal untuk menutup modal.

Di dalam elemen <form>, ditentukan aksi form (action) yang mengarahkan ke URL <?= base\_url('admin/slider'); ?> dengan metode POST.

Pada bagian body modal, terdapat beberapa elemen <input> yang memungkinkan pengguna memasukkan informasi slider yang ingin ditambahkan.

Elemen input dengan id nama digunakan untuk memasukkan nama slider.

Elemen input dengan id image digunakan untuk memilih file gambar slider yang akan diunggah.

Pada bagian footer modal, terdapat tombol "Close" yang berfungsi untuk menutup modal.

Terdapat tombol "Adding" yang berfungsi untuk mengirimkan form dan menambahkan slider baru. Tombol ini memiliki atribut type="submit" yang akan mengirimkan form saat ditekan.

Setelah form ditutup, tidak ada elemen lain dalam modal.

**File application/views/ admin/slider/edit.php**

<**div** class="page-heading">

    <**div** class="page-title">

        <**div** class="row">

            <**div** class="col-12 col-md-6 order-md-1 order-last">

                <**h3**><?php echo $heading; ?></**h3**>

            </**div**>

            <**div** class="col-12 col-md-6 order-md-2 order-first">

                <**nav** aria-label="breadcrumb" class="breadcrumb-header float-start float-lg-end">

                    <**ol** class="breadcrumb">

                        <**li** class="breadcrumb-item">

                            <**a** href="index.html">Dashboard</**a**>

                        </**li**>

                        <**li** class="breadcrumb-item active" aria-current="page">

                            <?php echo $heading; ?>

                        </**li**>

                    </**ol**>

                </**nav**>

            </**div**>

        </**div**>

    </**div**>

    <**section** class="section">

        <**div** class="card">

            <**div** class="card-header">

                <**h3**>Edit Slider</**h3**>

            </**div**>

            <**div** class="card-body">

                <?= form\_error("nama"); ?>

                <?php if (!empty($slider)) { ?>

                    <?php foreach ($slider as $row) { ?>

                        <**form** action="<?= base\_url("admin/slider/edit"); ?>" method="POST" enctype="multipart/form-data">

                            <**input** type="hidden" name="slider\_id" value="<?php echo $row->slider\_id ?>">

                            <**div** class="mb-3">

                                <**label** for="nama" class="form-label">Nama</**label**>

                                <**input** type="text" name="nama" value="<?php echo $row->nama ?>" class="form-control" id="nama" aria-describedby="emailHelp">

                            </**div**>

                            <**div** class="mb-3">

                                <**label** for="image" class="form-label">Gambar</**label**>

                                <**input** type="file" name="image" class="form-control" id="image">

                            </**div**>

                            <**div** class="mb-3">

                                <**label** for="current\_image" class="form-label">Gambar Saat Ini</**label**>

                                <**br**>

                                <?php if ($row->image) { ?>

                                    <**img** src="<?= base\_url('assets/img/upload/slider/' . $row->image) ?>" alt="Current Image" style="width: 200px;">

                                <?php } else { ?>

                                    <**span**>Tidak ada gambar</**span**>

                                <?php } ?>

                            </**div**>

                            <**button** type="submit" name="submit" class="btn btn-primary">Submit</**button**>

                        </**form**>

                    <?php } ?>

                <?php } ?>

            </**div**>

        </**div**>

    </**section**>

</**div**>

Halaman ini digunakan untuk mengedit slider yang ada.

Pada bagian header halaman, terdapat judul halaman yang ditampilkan menggunakan variabel $heading.

Terdapat juga breadcrumb yang menunjukkan navigasi halaman, dengan tautan ke halaman dashboard dan halaman edit slider.

Di dalam elemen <div class="card-body">, ditampilkan pesan error jika ada validasi form yang tidak valid dengan menggunakan fungsi form\_error("nama").

Jika ada data slider yang diterima (variabel $slider tidak kosong), maka dilakukan iterasi menggunakan foreach untuk menampilkan form edit untuk setiap slider.

Setiap form edit memiliki atribut action="<?= base\_url("admin/slider/edit"); ?>" yang akan mengarahkan form ke URL admin/slider/edit saat form disubmit.

Terdapat elemen input hidden dengan nama slider\_id yang menyimpan ID slider yang akan diubah.

Terdapat elemen input dengan nama nama yang menampilkan nilai nama slider saat ini, sehingga pengguna dapat mengeditnya.

Terdapat elemen input file dengan nama image yang memungkinkan pengguna memilih gambar slider baru.

Terdapat elemen <img> yang menampilkan gambar slider saat ini (jika ada) menggunakan atribut src="<?= base\_url('assets/img/upload/slider/' . $row->image) ?>".

Terdapat tombol "Submit" yang akan mengirimkan form saat ditekan.

Setiap form ditutup setelah iterasi selesai.

**application/controllers/admin/Order.php**

 public function index(){

        $data['title'] = 'Order - Ecommerce Admin';

        $data['heading'] = 'Order';

        $data['subview'] = 'admin/order/index';

        $data['order'] = $this->ModelOrder->findAll();

        $this->load->view('admin/\_layout', $data);

    }

Fungsi ini akan memuat halaman order pada admin e-commerce.

Variabel $data['title'] digunakan untuk mengatur judul halaman.

Variabel $data['heading'] digunakan untuk menentukan judul yang akan ditampilkan di halaman.

Variabel $data['subview'] berisi path ke file view yang akan dimuat sebagai bagian dari layout halaman admin.

Variabel $data['order'] berisi data order yang akan ditampilkan pada halaman order.

Kemudian, fungsi load->view('admin/\_layout', $data) digunakan untuk memuat view dengan menggunakan layout admin yang ditentukan. Variabel $data akan diteruskan ke view tersebut.

  public function delete(){

        $id = $this->uri->segment(4);

        $this->ModelOrder->delete($id);

        return redirect("admin/order");

    }

Fungsi ini akan menghapus data order berdasarkan ID yang diperoleh dari segment URI.

Variabel $id akan memuat nilai dari segment URI ke-4, yang diasumsikan sebagai ID order yang akan dihapus.

Selanjutnya, fungsi $this->ModelOrder->delete($id) akan dipanggil untuk menghapus data order dengan menggunakan model yang sesuai.

Setelah penghapusan berhasil, fungsi akan melakukan redirect ke halaman admin order dengan menggunakan redirect("admin/order").

 public function export\_excel(){

        $data = array(

            'title' => 'Laporan Order',

            'order' =>  $this->ModelOrder->findAll()

        );

        $this->load->view('admin/order/export\_excel\_order', $data);

    }

Fungsi ini akan membuat array $data yang berisi informasi yang akan digunakan untuk menghasilkan laporan order.

Array $data terdiri dari dua elemen, yaitu 'title' yang berisi judul laporan order dan 'order' yang berisi data order yang diperoleh dari pemanggilan $this->ModelOrder->findAll().

Selanjutnya, fungsi $this->load->view('admin/order/export\_excel\_order', $data) akan dipanggil untuk memuat view export\_excel\_order.php yang berada dalam direktori admin/order dan akan mengirimkan data $data ke dalam view tersebut.

Dalam view export\_excel\_order.php, Anda dapat mengatur kode untuk menghasilkan file Excel menggunakan library atau package yang sesuai untuk membaca dan menulis file Excel.

Dengan menggunakan kode

 public function laporan\_order\_pdf()

    {

        $this->load->library('Dompdf\_gen');

        $data['order'] = $this->ModelOrder->findAll();

        $this->load->view('admin/order/export\_pdf', $data);

        $paper = 'A4';

        $orien = 'landscape';

        $html = $this->output->get\_output();

        $this->dompdf->set\_paper($paper, $orien);

        $this->dompdf->load\_html($html);

        $this->dompdf->render();

        $this->dompdf->stream('laporan\_data\_order.pdf');

    }

Fungsi ini menggunakan library Dompdf untuk menghasilkan file PDF. Pastikan Anda telah menginstal dan memuat library Dompdf sebelum menggunakan fungsi ini.

Pertama, library Dompdf dimuat menggunakan $this->load->library('Dompdf\_gen').

Selanjutnya, data order diperoleh dengan memanggil $this->ModelOrder->findAll() dan disimpan dalam array $data.

Kemudian, fungsi $this->load->view('admin/order/export\_pdf', $data) dipanggil untuk memuat view export\_pdf.php yang berada dalam direktori admin/order dan mengirimkan data $data ke dalam view tersebut.

Setelah itu, variabel $paper dan $orien digunakan untuk mengatur ukuran kertas dan orientasi PDF. Dalam contoh ini, ukuran kertas diatur sebagai A4 dan orientasi diatur sebagai landscape.

Selanjutnya, output HTML yang dihasilkan dari view dimuat ke dalam Dompdf menggunakan $this->dompdf->load\_html($html).

Dompdf kemudian merender HTML menjadi file PDF menggunakan $this->dompdf->render().

Terakhir, file PDF di-generate dan di-stream ke browser menggunakan $this->dompdf->stream('laporan\_data\_order.pdf').

**File application/views/ admin/order/index.php**

<**div** class="page-heading">

    <**div** class="page-title">

        <**div** class="row">

            <**div** class="col-12 col-md-6 order-md-1 order-last">

                <**h3**><?php echo $heading; ?></**h3**>

                <?php if (validation\_errors()) { ?>

                    <**div** class="alert alert-danger alert-dismissible fade show" role="alert">

                        <?= validation\_errors(); ?>

                        <**button** type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></**button**>

                    </**div**>

                <?php } ?>

                <?php

                if ($this->session->flashdata('error\_order') != '') {

                    echo '<div class="alert alert-danger alert-dismissible fade show" role="alert">';

                    echo $this->session->flashdata('error\_order');

                    echo '<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>';

                    echo '</div>';

                }

                ?>

                <?php

                if ($this->session->flashdata('success\_order') != '') {

                    echo '<div class="alert alert-success alert-dismissible fade show" role="alert">';

                    echo $this->session->flashdata('success\_order');

                    echo '<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>';

                    echo '</div>';

                }

                ?>

            </**div**>

            <**div** class="col-12 col-md-6 order-md-2 order-first">

                <**nav** aria-label="breadcrumb" class="breadcrumb-header float-start float-lg-end">

                    <**ol** class="breadcrumb">

                        <**li** class="breadcrumb-item">

                            <**a** href="index.html">Dashboard</**a**>

                        </**li**>

                        <**li** class="breadcrumb-item active" aria-current="page">

                            <?php echo $heading; ?>

                        </**li**>

                    </**ol**>

                </**nav**>

            </**div**>

        </**div**>

    </**div**>

    <**section** class="section">

        <**div** class="card">

            <**div** class="card-header">

                <**a** href="<?= base\_url('admin/order/laporan\_order\_pdf'); ?>" class="btn btn-warning btn-sm mb-3"><**i** class="far fa-file-pdf"></**i**> Download Pdf</**a**>

                <**a** href="<?= base\_url('admin/order/export\_excel'); ?>" class="btn btn-success btn-sm mb-3"><**i** class="far fa-file-excel"></**i**> Export ke Excel</**a**>

            </**div**>

            <**div** class="card-body">

                <**table** class="table table-striped" id="table1">

                    <**thead**>

                        <**tr**>

                            <**th**>Order ID</**th**>

                            <**th**>Email</**th**>

                            <**th**>Postal Code</**th**>

                            <**th**>Country Code</**th**>

                            <**th**>Total Product</**th**>

                            <**th**>Total Price</**th**>

                            <**th**>Delete</**th**>

                        </**tr**>

                    </**thead**>

                    <**tbody**>

                        <?php

                        if (!empty($order)) {

                            foreach ($order as $row) {

                        ?>

                                <**tr**>

                                    <**td**><?php echo $row->order\_id ?></**td**>

                                    <**td**><?php echo $row->email ?></**td**>

                                    <**td**><?php echo $row->postal\_code ?></**td**>

                                    <**td**><?php echo $row->country\_code ?></**td**>

                                    <**td**><?php echo $row->total\_product ?></**td**>

                                    <**td**><?php echo $row->total\_price ?></**td**>

                                    <**td**><?php echo $row->created\_at ?></**td**>

                                    <**td**><?php echo $row->updated\_at ?></**td**>

                                    <**td** width="250">

                                        <**a** onclick="return confirm('Are you sure you want to delete this record?');" href="<?php echo site\_url('admin/order/delete/' . $row->order\_id); ?>" class="btn btn-danger">Delete</**a**>

                                    </**td**>

                                </**tr**>

                        <?php

                            }

                        }

                        ?>

                    </**tbody**>

                </**table**>

            </**div**>

        </**div**>

    </**section**>

</**div**>

Di dalam blok PHP, terdapat penggunaan if untuk menampilkan pesan error dan pesan sukses berdasarkan flashdata yang disetel sebelumnya.

Bagian <div class="card-header"> berisi dua tombol, yaitu "Download PDF" dan "Export ke Excel". Tombol "Download PDF" mengarah ke URL admin/order/laporan\_order\_pdf menggunakan base\_url(), sedangkan tombol "Export ke Excel" mengarah ke URL admin/order/export\_excel.

Di dalam <table> terdapat struktur tabel dengan beberapa kolom seperti "Order ID", "Email", "Postal Code", "Country Code", "Total Product", "Total Price", "Created At", "Updated At", dan kolom "Delete".

Di dalam blok PHP, terdapat penggunaan perulangan foreach untuk menampilkan data order dari variabel $order.

Setiap baris data order ditampilkan dalam elemen <tr> dan nilai-nilainya ditampilkan dalam elemen <td>.

Terdapat juga tombol "Delete" dengan tautan admin/order/delete/{order\_id} yang akan mengarah ke fungsi delete() di controller Order dengan mengirimkan order\_id sebagai parameter. Tombol ini memunculkan konfirmasi dialog sebelum menghapus rekaman order.

**File application/views/ admin/order/export\_excel\_order.php**

<?php

header('Content-type: application/vnd-ms-excel');

header('Content-Disposition: attachment; filename=' . $title . '.xls');

header('Pragma: no-cache');

header('Expires: 0');

?>

<**style** type="text/css">

    .table-data {

        width: 100%;

        border-collapse: collapse;

    }

    .table-data tr th,

    .table-data tr td {

        border: 1px solid black;

        font-size: 11pt;

        font-family: Verdana;

        padding: 10px 10px 10px 10px;

    }

    .table-data th {

        background-color: grey;

    }

    h3 {

        font-family: Verdana;

    }

</**style**>

<**h3**>

    <**center**>Laporan Data User Toko Online</**center**>

</**h3**>

<**table** class="table-data">

    <**thead**>

        <**tr**>

            <**th** class="border border-gray-800 p-2">Order ID</**th**>

            <**th** class="border border-gray-800 p-2">Email</**th**>

            <**th** class="border border-gray-800 p-2">Postal Code</**th**>

            <**th** class="border border-gray-800 p-2">Country Code</**th**>

            <**th** class="border border-gray-800 p-2">Total Product</**th**>

            <**th** class="border border-gray-800 p-2">Total Price</**th**>

        </**tr**>

    </**thead**>

    <**tbody**>

        <?php

        if (!empty($user)) {

            foreach ($user as $row) {

        ?>

                <**tr**>

                    <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->order\_id ?></**td**>

                    <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->email ?></**td**>

                    <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->postal\_code ?></**td**>

                    <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->country\_code ?></**td**>

                    <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->total\_product ?></**td**>

                    <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->total\_price ?></**td**>

                </**tr**>

        <?php

            }

        }

        ?>

    </**tbody**>

</**table**>

Baris-baris kode header di awal mengatur header respons HTTP untuk menentukan tipe konten sebagai application/vnd-ms-excel dan header Content-Disposition untuk menentukan nama file dan disposisi (attachment).

Baris-baris kode di dalam tag <style> mengatur tampilan tabel data dengan menggunakan CSS. Tabel memiliki batas sel dan menggunakan font Verdana dengan ukuran 11pt. Kolom header memiliki latar belakang abu-abu.

Di dalam tag <h3>, terdapat judul laporan yang ditampilkan di tengah.

Di dalam tag <table> terdapat struktur tabel dengan beberapa kolom seperti "Order ID", "Email", "Postal Code", "Country Code", "Total Product", dan "Total Price".

Di dalam blok PHP, terdapat penggunaan perulangan foreach untuk menampilkan data order dari variabel $user.

Setiap baris data order ditampilkan dalam elemen <tr> dan nilai-nilainya ditampilkan dalam elemen <td>.

Setiap sel memiliki atribut class untuk mengatur batas sel dan padding menggunakan CSS.

**File application/views/ admin/order/export\_pdf.php**

<!**DOCTYPE** html>

<**html**>

<**head**>

    <**title**></**title**>

</**head**>

<**body**>

    <**style** type="text/css">

        .table-data {

            width: 100%;

            border-collapse: collapse;

        }

        .table-data tr th,

        .table-data tr td {

            border: 1px solid black;

            font-size: 11pt;

            font-family: Verdana;

            padding: 10px 10px 10px 10px;

        }

        .table-data th {

            background-color: grey;

        }

        h3 {

            font-family: Verdana;

        }

    </**style**>

    <**h3**>

        <**center**>LAPORAN DATA Order</**center**>

    </**h3**>

    <**br** />

    <**table** class="table-data">

        <**thead**>

            <**tr**>

                <**th** class="border border-gray-800 p-2">Order ID</**th**>

                <**th** class="border border-gray-800 p-2">Email</**th**>

                <**th** class="border border-gray-800 p-2">Postal Code</**th**>

                <**th** class="border border-gray-800 p-2">Country Code</**th**>

                <**th** class="border border-gray-800 p-2">Total Product</**th**>

                <**th** class="border border-gray-800 p-2">Total Price</**th**>

            </**tr**>

        </**thead**>

        <**tbody**>

            <?php

            $no = 1;

            foreach ($order as $row) {

            ?>

                <**tr**>

                    <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->order\_id ?></**td**>

                    <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->email ?></**td**>

                    <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->postal\_code ?></**td**>

                    <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->country\_code ?></**td**>

                    <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->total\_product ?></**td**>

                    <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->total\_price ?></**td**>

                </**tr**>

            <?php

            }

            ?>

        </**tbody**>

    </**table**>

</**body**>

</**html**>

Di dalam tag <style>, terdapat pengaturan tampilan tabel dengan menggunakan CSS. Tabel memiliki batas sel dan menggunakan font Verdana dengan ukuran 11pt. Kolom header memiliki latar belakang abu-abu.

Di dalam tag <h3>, terdapat judul laporan yang ditampilkan di tengah.

Di dalam tag <table>, terdapat struktur tabel dengan beberapa kolom seperti "Order ID", "Email", "Postal Code", "Country Code", "Total Product", dan "Total Price".

Di dalam blok PHP, terdapat penggunaan perulangan foreach untuk menampilkan data order dari variabel $order.

Setiap baris data order ditampilkan dalam elemen <tr> dan nilai-nilainya ditampilkan dalam elemen <td>.

Setiap sel memiliki atribut class untuk mengatur batas sel dan padding menggunakan CSS.

**File application/modals/ModelAuth.php**

<?php

defined('BASEPATH') or exit('No direct script access allowed');

class ModelAuth extends CI\_Model

{

    public function \_\_construct()

    {

        parent::\_\_construct();

    }

    public function login($email, $password)

    {

        $query = $this->db->get\_where('user', array('email' => $email));

        if ($query->num\_rows() > 0) { *// menghitung row*

            $data\_user = $query->row();

            if (password\_verify($password, $data\_user->password)) {

                $data = [

                    'user\_id' => $data\_user->user\_id,

                    'email' => $data\_user->email,

                    'firstname' => $data\_user->firstname,

                    'lastname' => $data\_user->lastname,

                    'loggedin' => TRUE,

                    'role\_id' => $data\_user->role\_id,

                ];

                $this->session->set\_userdata($data);

                return True;

            }

        } else {

            return false;

        }

    }

    public function register($firstname, $lastname,$email, $password)

    {

        $data\_user = array(

            'firstname' => $firstname,

            'lastname' => $lastname,

            'email' => $email,

            'role\_id' => 1,

            'password' => password\_hash($password, PASSWORD\_BCRYPT),

        );

        $this->db->insert('user', $data\_user);

        return true;

    }

}

login($email, $password): Fungsi ini digunakan untuk melakukan proses login. Parameter yang diterima adalah $email dan $password yang dikirim dari form login. Fungsi ini melakukan query ke tabel "user" berdasarkan email yang diberikan. Jika email ditemukan, maka dilakukan verifikasi password menggunakan password\_verify(). Jika verifikasi berhasil, maka data pengguna (user) akan disimpan ke dalam session menggunakan set\_userdata(), dan fungsi ini mengembalikan nilai true. Jika email tidak ditemukan atau verifikasi password gagal, maka fungsi ini mengembalikan nilai false.

register($firstname, $lastname, $email, $password): Fungsi ini digunakan untuk melakukan proses registrasi pengguna baru. Parameter yang diterima adalah $firstname, $lastname, $email, dan $password yang dikirim dari form registrasi. Fungsi ini membuat array $data\_user yang berisi informasi pengguna baru, termasuk hash dari password menggunakan password\_hash(). Selanjutnya, data pengguna baru disimpan ke dalam tabel "user" menggunakan insert(). Fungsi ini mengembalikan nilai true jika proses registrasi berhasil.

**File application/controllers/Auth.php**

public function login()

    {

        $this->form\_validation->set\_rules(

            'email',

            'Email',

            'required|min\_length[6]|valid\_email',

            [

                'required' => '<div id="alert-1" class="flex p-4 mb-4 text-red-800 rounded-lg bg-red-50 dark:bg-gray-800 dark:text-red-400" role="alert">

                                <svg aria-hidden="true" class="flex-shrink-0 w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clip-rule="evenodd"></path></svg>

                                <span class="sr-only">Info</span>

                                <div class="ml-3 text-sm font-medium">

                                    Email harus di isi

                                </div>

                                <button type="button" class="ml-auto -mx-1.5 -my-1.5 bg-red-50 text-red-500 rounded-lg focus:ring-2 focus:ring-red-400 p-1.5 hover:bg-red-200 inline-flex h-8 w-8 dark:bg-gray-800 dark:text-red-400 dark:hover:bg-gray-700" data-dismiss-target="#alert-1" aria-label="Close">

                                <span class="sr-only">Close</span>

                                <svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd"></path></svg>

                                </button>

                            </div>',

                'min\_length' => '<div id="alert-1" class="flex p-4 mb-4 text-red-800 rounded-lg bg-red-50 dark:bg-gray-800 dark:text-red-400" role="alert">

                                    <svg aria-hidden="true" class="flex-shrink-0 w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clip-rule="evenodd"></path></svg>

                                    <span class="sr-only">Info</span>

                                    <div class="ml-3 text-sm font-medium">

                                        Email terlalu pendek

                                    </div>

                                    <button type="button" class="ml-auto -mx-1.5 -my-1.5 bg-red-50 text-red-500 rounded-lg focus:ring-2 focus:ring-red-400 p-1.5 hover:bg-red-200 inline-flex h-8 w-8 dark:bg-gray-800 dark:text-red-400 dark:hover:bg-gray-700" data-dismiss-target="#alert-1" aria-label="Close">

                                    <span class="sr-only">Close</span>

                                    <svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd"></path></svg>

                                    </button>

                                </div>',

                'valid\_email' => '<div id="alert-1" class="flex p-4 mb-4 text-red-800 rounded-lg bg-red-50 dark:bg-gray-800 dark:text-red-400" role="alert">

                                    <svg aria-hidden="true" class="flex-shrink-0 w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clip-rule="evenodd"></path></svg>

                                    <span class="sr-only">Info</span>

                                    <div class="ml-3 text-sm font-medium">

                                        Format email tidak benar

                                    </div>

                                    <button type="button" class="ml-auto -mx-1.5 -my-1.5 bg-red-50 text-red-500 rounded-lg focus:ring-2 focus:ring-red-400 p-1.5 hover:bg-red-200 inline-flex h-8 w-8 dark:bg-gray-800 dark:text-red-400 dark:hover:bg-gray-700" data-dismiss-target="#alert-1" aria-label="Close">

                                    <span class="sr-only">Close</span>

                                    <svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd"></path></svg>

                                    </button>

                                </div>'

            ]

        );

        $this->form\_validation->set\_rules(

            'password',

            'Password',

            'required|min\_length[6]',

            [

                'required' => '<div id="alert-2" class="flex p-4 mb-4 text-red-800 rounded-lg bg-red-50 dark:bg-gray-800 dark:text-red-400" role="alert">

                                <svg aria-hidden="true" class="flex-shrink-0 w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clip-rule="evenodd"></path></svg>

                                <span class="sr-only">Info</span>

                                <div class="ml-3 text-sm font-medium">

                                    Password harus di isi

                                </div>

                                <button type="button" class="ml-auto -mx-1.5 -my-1.5 bg-red-50 text-red-500 rounded-lg focus:ring-2 focus:ring-red-400 p-1.5 hover:bg-red-200 inline-flex h-8 w-8 dark:bg-gray-800 dark:text-red-400 dark:hover:bg-gray-700" data-dismiss-target="#alert-2" aria-label="Close">

                                <span class="sr-only">Close</span>

                                <svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd"></path></svg>

                                </button>

                            </div>',

                'min\_length' => '<div id="alert-2" class="flex p-4 mb-4 text-red-800 rounded-lg bg-red-50 dark:bg-gray-800 dark:text-red-400" role="alert">

                                    <svg aria-hidden="true" class="flex-shrink-0 w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clip-rule="evenodd"></path></svg>

                                    <span class="sr-only">Info</span>

                                    <div class="ml-3 text-sm font-medium">

                                        Password terlalu pendek

                                    </div>

                                    <button type="button" class="ml-auto -mx-1.5 -my-1.5 bg-red-50 text-red-500 rounded-lg focus:ring-2 focus:ring-red-400 p-1.5 hover:bg-red-200 inline-flex h-8 w-8 dark:bg-gray-800 dark:text-red-400 dark:hover:bg-gray-700" data-dismiss-target="#alert-2" aria-label="Close">

                                    <span class="sr-only">Close</span>

                                    <svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd"></path></svg>

                                    </button>

                                </div>',

            ]

        );

        if ($this->form\_validation->run() != true) {

            $data['subview'] = 'auth/login';

            $this->load->view('\_layout', $data);

        } else {

            $email = $this->input->post('email');

            $password = $this->input->post('password');

            if($this->ModelAuth->login($email,$password) == true){

                $this->session->set\_flashdata('success\_login','Proses Login User Berhasil');

                redirect("home");

            }else{

                $this->session->set\_flashdata('error','Write your email or password correctly');

                redirect('auth/login', 'refresh');

            }

        }

    }

Pertama, fungsi ini menggunakan library form\_validation dari CodeIgniter untuk melakukan validasi data yang dikirimkan oleh pengguna melalui form login.

Pada blok kode berikut, terdapat aturan validasi untuk field email. Aturan validasi yang diterapkan adalah required (wajib diisi), min\_length (panjang minimal 6 karakter), dan valid\_email (format email yang benar).

Jika aturan validasi tidak terpenuhi, maka akan ditampilkan pesan error yang sesuai dengan aturan validasi yang dilanggar. Pesan error ditampilkan dalam bentuk elemen HTML dengan menggunakan CSS dan SVG untuk tampilan yang lebih menarik.

Setelah aturan validasi untuk field email selesai, blok kode yang sama juga diterapkan untuk validasi field password.

Setelah semua aturan validasi selesai, dilakukan pengecekan apakah validasi berhasil atau tidak dengan menggunakan $this->form\_validation->run(). Jika validasi gagal, maka akan ditampilkan halaman login kembali beserta pesan error. Jika validasi berhasil, maka dilanjutkan ke langkah berikutnya.

Dalam langkah ini, nilai dari field email dan password diambil menggunakan $this->input->post().

Nilai email dan password kemudian digunakan sebagai parameter dalam pemanggilan fungsi login() yang ada pada model ModelAuth. Jika fungsi login() mengembalikan nilai true, artinya proses login berhasil. Pesan sukses disimpan dalam flash data dan pengguna diarahkan ke halaman "home".

Jika fungsi login() mengembalikan nilai false, artinya proses login gagal. Pesan error disimpan dalam flash data dan pengguna diarahkan kembali ke halaman login

public function register()

    {

        $this->form\_validation->set\_rules(

            'firstname',

            'Firstname',

            'required|min\_length[6]',

            [

                'required' => '<div id="alert-1" class="flex p-4 mb-4 text-red-800 rounded-lg bg-red-50 dark:bg-gray-800 dark:text-red-400" role="alert">

                                <svg aria-hidden="true" class="flex-shrink-0 w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clip-rule="evenodd"></path></svg>

                                <span class="sr-only">Info</span>

                                <div class="ml-3 text-sm font-medium">

                                    Firstname Harus di Isi

                                </div>

                                <button type="button" class="ml-auto -mx-1.5 -my-1.5 bg-red-50 text-red-500 rounded-lg focus:ring-2 focus:ring-red-400 p-1.5 hover:bg-red-200 inline-flex h-8 w-8 dark:bg-gray-800 dark:text-red-400 dark:hover:bg-gray-700" data-dismiss-target="#alert-1" aria-label="Close">

                                <span class="sr-only">Close</span>

                                <svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd"></path></svg>

                                </button>

                            </div>',

                'min\_length' => '<div id="alert-1" class="flex p-4 mb-4 text-red-800 rounded-lg bg-red-50 dark:bg-gray-800 dark:text-red-400" role="alert">

                                    <svg aria-hidden="true" class="flex-shrink-0 w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clip-rule="evenodd"></path></svg>

                                    <span class="sr-only">Info</span>

                                    <div class="ml-3 text-sm font-medium">

                                        Firstname Terlalu Pendek

                                    </div>

                                    <button type="button" class="ml-auto -mx-1.5 -my-1.5 bg-red-50 text-red-500 rounded-lg focus:ring-2 focus:ring-red-400 p-1.5 hover:bg-red-200 inline-flex h-8 w-8 dark:bg-gray-800 dark:text-red-400 dark:hover:bg-gray-700" data-dismiss-target="#alert-1" aria-label="Close">

                                    <span class="sr-only">Close</span>

                                    <svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd"></path></svg>

                                    </button>

                                </div>',

            ]

        );

        $this->form\_validation->set\_rules(

            'lastname',

            'Lastname',

            'required|min\_length[6]',

            [

                'required' => '<div id="alert-2" class="flex p-4 mb-4 text-red-800 rounded-lg bg-red-50 dark:bg-gray-800 dark:text-red-400" role="alert">

                                <svg aria-hidden="true" class="flex-shrink-0 w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clip-rule="evenodd"></path></svg>

                                <span class="sr-only">Info</span>

                                <div class="ml-3 text-sm font-medium">

                                    Lastname Harus di Is

                                </div>

                                <button type="button" class="ml-auto -mx-1.5 -my-1.5 bg-red-50 text-red-500 rounded-lg focus:ring-2 focus:ring-red-400 p-1.5 hover:bg-red-200 inline-flex h-8 w-8 dark:bg-gray-800 dark:text-red-400 dark:hover:bg-gray-700" data-dismiss-target="#alert-2 aria-label="Close">

                                <span class="sr-only">Close</span>

                                <svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd"></path></svg>

                                </button>

                            </div>',

                'min\_length' => '<div id="alert-2" class="flex p-4 mb-4 text-red-800 rounded-lg bg-red-50 dark:bg-gray-800 dark:text-red-400" role="alert">

                                    <svg aria-hidden="true" class="flex-shrink-0 w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clip-rule="evenodd"></path></svg>

                                    <span class="sr-only">Info</span>

                                    <div class="ml-3 text-sm font-medium">

                                        Lastname Terlalu Pendek

                                    </div>

                                    <button type="button" class="ml-auto -mx-1.5 -my-1.5 bg-red-50 text-red-500 rounded-lg focus:ring-2 focus:ring-red-400 p-1.5 hover:bg-red-200 inline-flex h-8 w-8 dark:bg-gray-800 dark:text-red-400 dark:hover:bg-gray-700" data-dismiss-target="#alert-2" aria-label="Close">

                                    <span class="sr-only">Close</span>

                                    <svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd"></path></svg>

                                    </button>

                                </div>',

            ]

        );

        $this->form\_validation->set\_rules(

            'email',

            'Email',

            'required|min\_length[6]|valid\_email',

            [

                'required' => '<div id="alert-3" class="flex p-4 mb-4 text-red-800 rounded-lg bg-red-50 dark:bg-gray-800 dark:text-red-400" role="alert">

                                <svg aria-hidden="true" class="flex-shrink-0 w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clip-rule="evenodd"></path></svg>

                                <span class="sr-only">Info</span>

                                <div class="ml-3 text-sm font-medium">

                                    Email harus di isi

                                </div>

                                <button type="button" class="ml-auto -mx-1.5 -my-1.5 bg-red-50 text-red-500 rounded-lg focus:ring-2 focus:ring-red-400 p-1.5 hover:bg-red-200 inline-flex h-8 w-8 dark:bg-gray-800 dark:text-red-400 dark:hover:bg-gray-700" data-dismiss-target="#alert-3" aria-label="Close">

                                <span class="sr-only">Close</span>

                                <svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd"></path></svg>

                                </button>

                            </div>',

                'min\_length' => '<div id="alert-3" class="flex p-4 mb-4 text-red-800 rounded-lg bg-red-50 dark:bg-gray-800 dark:text-red-400" role="alert">

                                    <svg aria-hidden="true" class="flex-shrink-0 w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clip-rule="evenodd"></path></svg>

                                    <span class="sr-only">Info</span>

                                    <div class="ml-3 text-sm font-medium">

                                        Email terlalu pendek

                                    </div>

                                    <button type="button" class="ml-auto -mx-1.5 -my-1.5 bg-red-50 text-red-500 rounded-lg focus:ring-2 focus:ring-red-400 p-1.5 hover:bg-red-200 inline-flex h-8 w-8 dark:bg-gray-800 dark:text-red-400 dark:hover:bg-gray-700" data-dismiss-target="#alert-3" aria-label="Close">

                                    <span class="sr-only">Close</span>

                                    <svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd"></path></svg>

                                    </button>

                                </div>',

                'valid\_email' => '<div id="alert-3" class="flex p-4 mb-4 text-red-800 rounded-lg bg-red-50 dark:bg-gray-800 dark:text-red-400" role="alert">

                                    <svg aria-hidden="true" class="flex-shrink-0 w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clip-rule="evenodd"></path></svg>

                                    <span class="sr-only">Info</span>

                                    <div class="ml-3 text-sm font-medium">

                                        Format email tidak benar

                                    </div>

                                    <button type="button" class="ml-auto -mx-1.5 -my-1.5 bg-red-50 text-red-500 rounded-lg focus:ring-2 focus:ring-red-400 p-1.5 hover:bg-red-200 inline-flex h-8 w-8 dark:bg-gray-800 dark:text-red-400 dark:hover:bg-gray-700" data-dismiss-target="#alert-3" aria-label="Close">

                                    <span class="sr-only">Close</span>

                                    <svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd"></path></svg>

                                    </button>

                                </div>'

            ]

        );

        $this->form\_validation->set\_rules(

            'password',

            'Password',

            'required|min\_length[6]',

            [

                'required' => '<div id="alert-4" class="flex p-4 mb-4 text-red-800 rounded-lg bg-red-50 dark:bg-gray-800 dark:text-red-400" role="alert">

                                <svg aria-hidden="true" class="flex-shrink-0 w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clip-rule="evenodd"></path></svg>

                                <span class="sr-only">Info</span>

                                <div class="ml-3 text-sm font-medium">

                                    Password harus di isi

                                </div>

                                <button type="button" class="ml-auto -mx-1.5 -my-1.5 bg-red-50 text-red-500 rounded-lg focus:ring-2 focus:ring-red-400 p-1.5 hover:bg-red-200 inline-flex h-8 w-8 dark:bg-gray-800 dark:text-red-400 dark:hover:bg-gray-700" data-dismiss-target="#alert-4" aria-label="Close">

                                <span class="sr-only">Close</span>

                                <svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd"></path></svg>

                                </button>

                            </div>',

                'min\_length' => '<div id="alert-4" class="flex p-4 mb-4 text-red-800 rounded-lg bg-red-50 dark:bg-gray-800 dark:text-red-400" role="alert">

                                    <svg aria-hidden="true" class="flex-shrink-0 w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clip-rule="evenodd"></path></svg>

                                    <span class="sr-only">Info</span>

                                    <div class="ml-3 text-sm font-medium">

                                        Password terlalu pendek

                                    </div>

                                    <button type="button" class="ml-auto -mx-1.5 -my-1.5 bg-red-50 text-red-500 rounded-lg focus:ring-2 focus:ring-red-400 p-1.5 hover:bg-red-200 inline-flex h-8 w-8 dark:bg-gray-800 dark:text-red-400 dark:hover:bg-gray-700" data-dismiss-target="#alert-4" aria-label="Close">

                                    <span class="sr-only">Close</span>

                                    <svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd"></path></svg>

                                    </button>

                                </div>',

            ]

        );

        $this->form\_validation->set\_rules(

            'confirm\_password',

            'Confirm Password',

            'required|matches[password]',

            [

                'required' => '<div id="alert-5" class="flex p-4 mb-4 text-red-800 rounded-lg bg-red-50 dark:bg-gray-800 dark:text-red-400" role="alert">

                                <svg aria-hidden="true" class="flex-shrink-0 w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clip-rule="evenodd"></path></svg>

                                <span class="sr-only">Info</span>

                                <div class="ml-3 text-sm font-medium">

                                Konfirmasi Password Harus di Isi

                                </div>

                                <button type="button" class="ml-auto -mx-1.5 -my-1.5 bg-red-50 text-red-500 rounded-lg focus:ring-2 focus:ring-red-400 p-1.5 hover:bg-red-200 inline-flex h-8 w-8 dark:bg-gray-800 dark:text-red-400 dark:hover:bg-gray-700" data-dismiss-target="#alert-5" aria-label="Close">

                                <span class="sr-only">Close</span>

                                <svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd"></path></svg>

                                </button>

                            </div>',

                'matches' => '<div id="alert-5" class="flex p-4 mb-4 text-red-800 rounded-lg bg-red-50 dark:bg-gray-800 dark:text-red-400" role="alert">

                                    <svg aria-hidden="true" class="flex-shrink-0 w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clip-rule="evenodd"></path></svg>

                                    <span class="sr-only">Info</span>

                                    <div class="ml-3 text-sm font-medium">

                                    Konfirmasi Password tidak sesuai

                                    </div>

                                    <button type="button" class="ml-auto -mx-1.5 -my-1.5 bg-red-50 text-red-500 rounded-lg focus:ring-2 focus:ring-red-400 p-1.5 hover:bg-red-200 inline-flex h-8 w-8 dark:bg-gray-800 dark:text-red-400 dark:hover:bg-gray-700" data-dismiss-target="#alert-5" aria-label="Close">

                                    <span class="sr-only">Close</span>

                                    <svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd"></path></svg>

                                    </button>

                                </div>',

                'required' => "<div class='alert alert-danger' role='alert'>Konfirmasi Password Harus di Isi</div>",

                'matches' => "<div class='alert alert-danger' role='alert'>Konfirmasi Password tidak sesuai</div>"

            ]

        );

        if ($this->form\_validation->run() != true) {

            $data['subview'] = 'auth/register';

            $this->load->view("\_layout", $data);

        } else {

            $firstname = $this->input->post('firstname');

            $lastname = $this->input->post('lastname');

            $email = $this->input->post('email');

            $password = $this->input->post('password');

            $this->ModelAuth->register($firstname, $lastname, $email, $password);

            $this->session->set\_flashdata('success\_register','Proses Pendaftaran User Berhasil');

            redirect("auth/login");

        }

    }

Fungsi register() adalah fungsi yang digunakan untuk melakukan validasi pada data yang dikirimkan dalam formulir pendaftaran.

set\_rules() digunakan untuk mengatur aturan validasi untuk setiap field pada formulir. Pada contoh ini, ada tiga field yang divalidasi, yaitu firstname, lastname, dan email.

Untuk setiap field, kita menentukan aturan validasi yang harus dipenuhi. Misalnya, required menunjukkan bahwa field harus diisi, min\_length[6] menunjukkan bahwa nilai field harus memiliki panjang minimal 6 karakter, dan valid\_email menunjukkan bahwa field harus berisi alamat email yang valid.

Setelah menentukan aturan validasi, kita juga dapat menentukan pesan kesalahan khusus untuk setiap aturan yang tidak terpenuhi. Pesan kesalahan ditentukan dalam bentuk array asosiatif di parameter ketiga set\_rules(). Pada contoh ini, pesan kesalahan ditampilkan dalam bentuk HTML dengan menggunakan kelas-kelas CSS tertentu untuk tampilan yang lebih menarik.

Pesan kesalahan tersebut terdiri dari dua bagian: pesan untuk aturan required dan pesan untuk aturan min\_length. Pesan kesalahan untuk aturan required akan muncul jika field tidak diisi, sedangkan pesan kesalahan untuk aturan min\_length akan muncul jika nilai field terlalu pendek.

Pada setiap pesan kesalahan, terdapat tombol 'Close' yang digunakan untuk menutup pesan kesalahan. Tombol ini dikaitkan dengan atribut data-dismiss-target yang merujuk ke ID elemen yang akan ditutup.

Setelah mengatur aturan validasi, selanjutnya Anda dapat menjalankan validasi dengan menggunakan metode $this->form\_validation->run(). Metode ini akan mengembalikan nilai true jika semua aturan validasi terpenuhi, dan false jika ada setidaknya satu aturan yang tidak terpenuhi.

 public function logout(){

        $item = array('email','role\_id', 'user\_id','firstname', 'lastname', 'loggedin');

        $this->session->unset\_userdata($item); *// delete Session*

        $this->session->set\_flashdata('logout', 'Berhasil logout dengan baik');

        redirect("home");

    }

Pada baris pertama, kita mendefinisikan fungsi logout() yang akan digunakan untuk proses logout.

Pada baris kedua, kita membuat sebuah array dengan nama $item yang berisi daftar variabel sesi yang ingin dihapus. Dalam contoh ini, variabel sesi yang ingin dihapus adalah 'email', 'role\_id', 'user\_id', 'firstname', 'lastname', dan 'loggedin'.

Baris ketiga menjalankan fungsi unset\_userdata() pada objek $this->session. Fungsi ini digunakan untuk menghapus data sesi yang ditentukan dari variabel sesi. Dalam contoh ini, data sesi yang terkait dengan variabel yang terdapat dalam array $item akan dihapus.

Baris keempat menggunakan fungsi set\_flashdata() pada objek $this->session untuk mengatur pesan flash yang akan ditampilkan setelah logout berhasil. Pesan flash digunakan untuk menyimpan data yang hanya dapat ditampilkan satu kali pada request berikutnya. Dalam contoh ini, pesan flash dengan kunci 'logout' dan nilai 'Berhasil logout dengan baik' akan disimpan.

Pada baris kelima, kita menggunakan fungsi redirect() untuk mengarahkan pengguna ke halaman 'home'. Setelah logout berhasil, pengguna akan dialihkan ke halaman utama atau halaman yang telah ditentukan.

**File application/views/auth/register.php**

<**div** class="max-w-screen-xl mx-auto mt-10">

    <**div**>

        <?php

        if ($this->session->flashdata('error') != '') {

            echo '<div id="alert-2" class="flex p-4 mb-4 text-red-800 rounded-lg bg-red-50 dark:bg-gray-800 dark:text-red-400" role="alert">

        <svg aria-hidden="true" class="flex-shrink-0 w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg">

            <path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clip-rule="evenodd"></path>

        </svg>

        <span class="sr-only">Info</span>

        <div class="ml-3 text-sm font-medium">

           ' . $this->session->flashdata('error') . '

        </div>

        <button type="button" class="ml-auto -mx-1.5 -my-1.5 bg-red-50 text-red-500 rounded-lg focus:ring-2 focus:ring-red-400 p-1.5 hover:bg-red-200 inline-flex h-8 w-8 dark:bg-gray-800 dark:text-red-400 dark:hover:bg-gray-700" data-dismiss-target="#alert-2" aria-label="Close">

            <span class="sr-only">Close</span>

            <svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg">

                <path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd"></path>

            </svg>

        </button>

        </div>';

        }

        ?>

        <?php

        if ($this->session->flashdata('success\_register') != '') {

            echo '

    <div id="alert-3" class="flex p-4 mb-4 text-green-800 rounded-lg bg-green-50 dark:bg-gray-800 dark:text-green-400" role="alert">

        <svg aria-hidden="true" class="flex-shrink-0 w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clip-rule="evenodd"></path></svg>

        <span class="sr-only">Info</span>

        <div class="ml-3 text-sm font-medium">

            ' . $this->session->flashdata('success\_register');

            '

        </div>

        <button type="button" class="ml-auto -mx-1.5 -my-1.5 bg-green-50 text-green-500 rounded-lg focus:ring-2 focus:ring-green-400 p-1.5 hover:bg-green-200 inline-flex h-8 w-8 dark:bg-gray-800 dark:text-green-400 dark:hover:bg-gray-700" data-dismiss-target="#alert-3" aria-label="Close">

        <span class="sr-only">Close</span>

        <svg aria-hidden="true" class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd"></path></svg>

        </button>

    </div>

    ';

        }

        ?>

        <?= form\_error("firstname"); ?>

        <?= form\_error("lastname"); ?>

        <?= form\_error("email"); ?>

        <?= form\_error("password"); ?>

    </**div**>

    <**div** class="flex justify-center items-center">

        <**div** class="w-full max-w-xl">

            <**div** class="bg-slate-300 shadow-xl p-6 rounded-lg">

                <**h2** class="text-center text-2xl font-semibold mb-6">Register</**h2**>

                <**form** action="<?= base\_url("auth/register") ?>" method="POST">

                    <**input** type="text" name="firstname" placeholder="Firstname" class="w-full p-2 mb-4 border border-gray-300 rounded" required />

                    <**input** type="text" name="lastname" placeholder="Lastname" class="w-full p-2 mb-4 border border-gray-300 rounded" required />

                    <**input** type="email" name="email" placeholder="Email" class="w-full p-2 mb-4 border border-gray-300 rounded" />

                    <**input** type="password" name="password" placeholder="Password" class="w-full p-2 mb-4 border border-gray-300 rounded" />

                    <**input** type="password" name="confirm\_password" placeholder="Confirm Password" class="w-full p-2 mb-4 border border-gray-300 rounded" />

                    <**div** class="text-right">

                        <**button** type="submit" class="bg-blue-500 text-white px-4 py-2 rounded">

                            REGISTER

                        </**button**>

                    </**div**>

                </**form**>

                <**a** href="<?= base\_url('auth/login') ?>" class="block mt-4 text-center">

                    Click Here To Login

                </**a**>

            </**div**>

        </**div**>

    </**div**>

</**div**>

Bagian pertama (<?php if ($this->session->flashdata('error') != '') { ... } ?>) digunakan untuk menampilkan pesan kesalahan jika ada. Jika ada pesan kesalahan yang disimpan dalam sesi flash dengan kunci 'error', maka akan ditampilkan sebuah kotak peringatan dengan latar belakang merah. Pesan kesalahan akan ditampilkan dalam elemen dengan class 'text-sm font-medium'. Kotak peringatan ini juga memiliki tombol 'Close' untuk menutup kotak peringatan tersebut.

Bagian kedua (<?php if ($this->session->flashdata('success\_register') != '') { ... } ?>) digunakan untuk menampilkan pesan sukses jika ada. Jika ada pesan sukses yang disimpan dalam sesi flash dengan kunci 'success\_register', maka akan ditampilkan sebuah kotak peringatan dengan latar belakang hijau. Pesan sukses akan ditampilkan dalam elemen dengan class 'text-sm font-medium'. Kotak peringatan ini juga memiliki tombol 'Close' untuk menutup kotak peringatan tersebut.

Bagian selanjutnya adalah pemanggilan fungsi form\_error() untuk menampilkan pesan kesalahan validasi form yang dihasilkan oleh library CodeIgniter. Jika ada kesalahan validasi untuk field "firstname", "lastname", "email", atau "password", pesan kesalahan akan ditampilkan.

Pada bagian berikutnya, terdapat form registrasi dengan beberapa field seperti "firstname", "lastname", "email", "password", dan "confirm\_password". Field-field ini memiliki atribut seperti placeholder (teks yang muncul sebagai petunjuk di dalam field), class untuk styling, dan atribut lain yang sesuai dengan kebutuhan.

Terakhir, terdapat tombol "REGISTER" yang digunakan untuk mengirimkan data registrasi yang diisi pengguna ke URL 'auth/register' melalui metode POST.

Setelah form registrasi, terdapat sebuah tautan yang mengarahkan pengguna ke halaman login dengan URL 'auth/login'.

**File application/views/auth/login.php**

<**div** class="max-w-screen-xl mx-auto mt-10">

    <**div** class="flex justify-center items-center">

        <?php

        if ($this->session->flashdata('error') != '') {

            echo '<div id="alert-2" class="flex p-4 mb-4 text-red-800 rounded-lg bg-red-50 dark:bg-gray-800 dark:text-red-400" role="alert">

                    <svg aria-hidden="true" class="flex-shrink-0 w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg">

                        <path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clip-rule="evenodd"></path>

                    </svg>

                    <span class="sr-only">Info</span>

                    <div class="ml-3 text-sm font-medium">

                        A simple info alert with an ' . $this->session->flashdata('error') . '

                    </div>

                    <button type="button" class="ml-auto -mx-1.5 -my-1.5 bg-red-50 text-red-500 rounded-lg focus:ring-2 focus:ring-red-400 p-1.5 hover:bg-red-200 inline-flex h-8 w-8 dark:bg-gray-800 dark:text-red-400 dark:hover:bg-gray-700" data-dismiss-target="#alert-2" aria-label="Close">

                        <span class="sr-only">Close</span>

                        <svg class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg">

                            <path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd"></path>

                        </svg>

                    </button>

                    </div>';

        }

        ?>

        <?php

        if ($this->session->flashdata('success\_login') != '') {

            echo '

                <div id="alert-3" class="flex p-4 mb-4 text-green-800 rounded-lg bg-green-50 dark:bg-gray-800 dark:text-green-400" role="alert">

                    <svg aria-hidden="true" class="flex-shrink-0 w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clip-rule="evenodd"></path></svg>

                    <span class="sr-only">Info</span>

                    <div class="ml-3 text-sm font-medium">

                        ' . $this->session->flashdata('success\_login').

            '

                    </div>

                    <button type="button" class="ml-auto -mx-1.5 -my-1.5 bg-green-50 text-green-500 rounded-lg focus:ring-2 focus:ring-green-400 p-1.5 hover:bg-green-200 inline-flex h-8 w-8 dark:bg-gray-800 dark:text-green-400 dark:hover:bg-gray-700" data-dismiss-target="#alert-3" aria-label="Close">

                    <span class="sr-only">Close</span>

                    <svg aria-hidden="true" class="w-5 h-5" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 111.414 1.414L11.414 10l4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd"></path></svg>

                    </button>

                </div>

                ';

        }

        ?>

    </**div**>

    <?= form\_error("email") ?>

    <?= form\_error("password") ?>

    <**div** class="flex justify-center items-center">

        <**div** class="w-full max-w-md ">

            <**div** class="bg-white shadow-lg p-6 rounded-lg">

                <**h2** class="text-center text-2xl font-semibold mb-6">LOGIN</**h2**>

                <**form** action="<?= base\_url('auth/login'); ?>" method="POST">

                    <**input** name="email" type="text" placeholder="Email" class="w-full p-2 mb-4 border border-gray-300 rounded" required />

                    <**input** name="password" type="password" placeholder="Password" class="w-full p-2 mb-4 border border-gray-300 rounded" />

                    <**div** class="text-right">

                        <**button** type="submit" class="bg-blue-500 text-white px-4 py-2 rounded">

                            LOGIN

                        </**button**>

                    </**div**>

                </**form**>

                <**a** href="<?= base\_url('auth/register') ?>" class="block mt-4 text-center">

                    Click Here To Register

                </**a**>

            </**div**>

        </**div**>

    </**div**>

</**div**>

Alert Error: Jika terdapat flash data dengan nama 'error' pada sesi, maka akan ditampilkan sebuah alert dengan background merah dan pesan error yang sesuai.

Alert Success: Jika terdapat flash data dengan nama 'success\_login' pada sesi, maka akan ditampilkan sebuah alert dengan background hijau dan pesan success\_login yang sesuai.

Form Registrasi: Form ini digunakan untuk memasukkan informasi pengguna seperti nama depan, nama belakang, email, password, dan konfirmasi password. Setelah diisi, pengguna dapat menekan tombol "REGISTER" untuk mengirimkan data ke URL "auth/register".

Form Login: Form ini digunakan untuk memasukkan email dan password pengguna yang sudah terdaftar. Setelah diisi, pengguna dapat menekan tombol "LOGIN" untuk mengirimkan data ke URL "auth/login".

Link Registrasi: Tautan ini mengarahkan pengguna ke halaman registrasi jika mereka belum memiliki akun. URL yang dituju adalah "auth/register".

Link Login: Tautan ini mengarahkan pengguna ke halaman login jika mereka sudah memiliki akun. URL yang dituju adalah "auth/login".

**File application/controllers/Home.php**

<?php

defined("BASEPATH") or exit("No direct script access allowed");

class Home extends CI\_Controller

{

    public function index()

    {

        $data['subview'] = 'index';

        $data['category'] = $this->ModelCategory->findAll();

        $data['product'] = $this->ModelProduct->findAll();

        $data['slider'] = $this->ModelSlider->findAll();

        $this->load->view('\_layout', $data);

    }

}

Membuat class "Home" yang merupakan controller untuk halaman utama.

Membuat method "index" yang akan dipanggil ketika mengakses URL utama. Method ini akan menampilkan halaman utama dengan menggunakan view "\_layout".

Variabel "$data" digunakan untuk menyimpan data yang akan dikirimkan ke view. Dalam contoh ini, variabel "$data" memiliki beberapa elemen:

"subview": Menyimpan nama file view yang akan dimuat di dalam "\_layout". Di contoh ini, nilai "index" menunjukkan bahwa view yang akan dimuat adalah "index.php".

"category": Menyimpan hasil dari pemanggilan method "findAll()" pada model "ModelCategory". Data ini kemungkinan digunakan untuk menampilkan daftar kategori produk di halaman utama.

"product": Menyimpan hasil dari pemanggilan method "findAll()" pada model "ModelProduct". Data ini kemungkinan digunakan untuk menampilkan daftar produk di halaman utama.

"slider": Menyimpan hasil dari pemanggilan method "findAll()" pada model "ModelSlider". Data ini kemungkinan digunakan untuk menampilkan slider gambar di halaman utama.

Terakhir, method "index" memanggil view "\_layout" dan mengirimkan data "$data" ke view tersebut.

**File application/views/index.php**

<**div** class="mx-auto flex justify-center mt-10">

    <**div** class="swiper-container relative h-56 sm:h-72 md:h-96 w-full sm:w-10/12 md:w-11/12 lg:w-8/12 xl:w-7/12 2xl:w-6/12 overflow-hidden rounded-lg" style="border-radius: 10px;">

        <**div** class="swiper-wrapper">

            <?php

            if (!empty($slider)) {

                foreach ($slider as $row) {

            ?>

                    <**div** class="swiper-slide">

                        <**img** src="<?= base\_url('assets/img/upload/slider/' . $row->image) ?>" alt="<?= $row->nama ?>" />

                    </**div**>

            <?php

                }

            }

            ?>

        </**div**>

        <**div** class="swiper-button-next"></**div**>

        <**div** class="swiper-button-prev"></**div**>

    </**div**>

</**div**>

<**div** class="max-w-screen-xl mx-auto mt-10">

    <**div** class="mt-10">

        <**h1** class="text-2xl font-semibold p-3 ">Category</**h1**>

    </**div**>

    <**div** class="grid grid-cols-1 gap-6 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 mt-10">

        <?php

        if (!empty($category)) {

            foreach ($category as $row) {

        ?>

                <**a** href="<?= base\_url('category/detail/' . $row->slug\_category); ?>" class="p-2 border rounded-lg shadow">

                    <**div** class="flex items-center justify-center h-24">

                        <**img** src="<?= base\_url('assets/img/upload/category/' . $row->image\_category) ?>" alt="Category 1" class="h-16" />

                    </**div**>

                    <**div** class="mt-2 text-center">

                        <**h4** class="text-lg font-semibold"><?= $row->nama\_kategori; ?></**h4**>

                    </**div**>

                </**a**>

        <?php

            }

        }

        ?>

    </**div**>

</**div**>

<**div** class="max-w-screen-xl mx-auto mt-10">

    <**div** class="mt-10">

        <**h1** class="text-2xl font-semibold p-3 ">Product</**h1**>

    </**div**>

    <**div** class="grid grid-cols-1 gap-4 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4">

        <?php if (!empty($product)) {

            foreach ($product as $row) { ?>

                <**a** href="<?= base\_url('product/detail/' . $row->slug\_product); ?>">

                    <**div** class="p-4 border rounded-lg shadow-md bg-white">

                        <**div** class="relative overflow-hidden">

                            <**img** src="<?= base\_url('assets/img/upload/product/' . $row->image\_product) ?>" alt="<?= $row->name; ?>" class="w-full h-32 sm:h-40 object-cover rounded-lg" />

                        </**div**>

                        <**div** class="mt-4">

                            <**h3** class="text-lg font-semibold text-gray-800">

                                <?= $row->name; ?>

                            </**h3**>

                            <**p** class="mt-1 text-gray-600"><?= $row->description; ?></**p**>

                            <**p** class="mt-2 text-gray-800 font-semibold">

                                Rp. <?= number\_format($row->price, 0) ?>

                            </**p**>

                        </**div**>

                    </**div**>

                </**a**>

        <?php }

        } ?>

    </**div**>

</**div**>

<**script**>

    document.addEventListener('DOMContentLoaded', function() {

        var swiper = new Swiper('.swiper-container', {

            loop: true,

            spaceBetween: 0,

            navigation: {

                nextEl: '.swiper-button-next',

                prevEl: '.swiper-button-prev',

            },

        });

    });

</**script**>

<**script** src="https://cdn.jsdelivr.net/npm/swiper@9/swiper-bundle.min.js"></**script**>

Membuat sebuah div dengan class "swiper-container" untuk menampilkan gambar-gambar dalam bentuk slider. Gambar-gambar tersebut diambil dari variabel "$slider" yang merupakan hasil dari query database. Setiap gambar ditampilkan dalam elemen div dengan class "swiper-slide".

Setelah slider, div dengan class "max-w-screen-xl mx-auto mt-10" digunakan untuk menampilkan daftar kategori dan daftar produk.

Bagian pertama menampilkan daftar kategori dengan menggunakan grid. Setiap kategori ditampilkan dalam elemen a dengan class "p-2 border rounded-lg shadow". Pada bagian ini juga terdapat gambar yang merupakan thumbnail dari setiap kategori.

Bagian kedua menampilkan daftar produk dengan menggunakan grid. Setiap produk ditampilkan dalam elemen a dengan class "p-4 border rounded-lg shadow-md bg-white". Pada bagian ini juga terdapat gambar, nama produk, deskripsi, dan harga produk.

Terakhir, terdapat script JavaScript yang digunakan untuk menginisialisasi Swiper (library slider) pada elemen dengan class "swiper-container". Slider ini memiliki fitur loop (berputar secara terus-menerus) dan tombol navigasi untuk berpindah slide (next dan prev).

Script JavaScript juga memuat library Swiper dengan sumber dari "https://cdn.jsdelivr.net/npm/swiper@9/swiper-bundle.min.js".

**File application/views/my404.php**

<**section** class="bg-white mt-10">

    <**div** class="py-8 px-4 mx-auto max-w-screen-xl lg:py-16 lg:px-6">

        <**div** class="mx-auto max-w-screen-sm text-center">

            <**h1** class="mb-4 text-7xl tracking-tight font-extrabold lg:text-9xl text-primary-600">404</**h1**>

            <**p** class="mb-4 text-3xl tracking-tight font-bold text-gray-900 md:text-4xl">Something's missing.</**p**>

            <**p** class="mb-4 text-lg font-light text-gray-500">Sorry, we can't find that page. You'll find lots to explore on the home page. </**p**>

            <**a** href="<?= base\_url(); ?>" class="inline-flex text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:ring-blue-300 font-medium rounded-lg text-sm px-5 py-2.5 mr-2 mb-2">Back to Homepage</**a**>

        </**div**>

    </**div**>

</**section**>

Membuat sebuah section dengan class "bg-white" untuk menampilkan konten halaman 404.

Di dalam section tersebut, terdapat div dengan class "py-8 px-4 mx-auto max-w-screen-xl lg:py-16 lg:px-6" yang digunakan untuk mengatur padding dan lebar konten.

Di dalam div tersebut, terdapat div dengan class "mx-auto max-w-screen-sm text-center" yang digunakan untuk mengatur posisi dan lebar konten teks.

Pada bagian ini, terdapat elemen h1 dengan class "mb-4 text-7xl tracking-tight font-extrabold lg:text-9xl text-primary-600" yang menampilkan teks "404". Teks ini memiliki ukuran dan gaya tipografi yang menonjol.

Kemudian, terdapat elemen p dengan class "mb-4 text-3xl tracking-tight font-bold text-gray-900 md:text-4xl" yang menampilkan teks "Something's missing." Teks ini adalah judul yang mengindikasikan bahwa halaman tidak ditemukan.

Selanjutnya, terdapat elemen p dengan class "mb-4 text-lg font-light text-gray-500" yang menampilkan pesan penjelasan bahwa halaman yang diminta tidak dapat ditemukan. Pengguna disarankan untuk kembali ke halaman utama.

Terakhir, terdapat elemen a dengan href yang mengarah ke halaman utama menggunakan "<?= base\_url(); ?>". Elemen ini memiliki class "inline-flex text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:ring-blue-300 font-medium rounded-lg text-sm px-5 py-2.5 mr-2 mb-2" yang menampilkan tombol "Back to Homepage". Tombol ini memiliki tampilan yang menarik dengan latar belakang biru dan efek hover.

**File application/controllers/Category.php**

<?php

defined("BASEPATH") or exit("No direct script access allowed");

class Category extends CI\_Controller{

    public function \_\_construct()

    {

        parent::\_\_construct();

    }

    public function detail($slug){

        $category = $this->ModelCategory->findBySlug($slug);

        if ($category) {

            $data['slug'] = $slug;

            $data['subview'] = 'category/index';

            $data['category'] = $category;

            $this->load->view('\_layout', $data);

        }else{

            $data['subview'] = 'category/notfound';

            $this->load->view('\_layout', $data);

        }

    }

}

Mendefinisikan class Category yang merupakan sebuah Controller dalam CodeIgniter.

Di dalam class Category, terdapat method \_\_construct() yang merupakan constructor dari class tersebut. Method ini dipanggil saat class Category diinisialisasi.

Terdapat method detail($slug) yang akan dipanggil ketika mengakses URL dengan format /category/detail/$slug, dimana $slug adalah parameter yang mewakili slug dari kategori yang akan ditampilkan.

Pada method detail(), terdapat pemanggilan $this->ModelCategory->findBySlug($slug) yang digunakan untuk mencari data kategori berdasarkan slug yang diberikan. Model ModelCategory akan mengakses database untuk melakukan pencarian berdasarkan slug.

Jika kategori ditemukan (nilai dari $category tidak kosong), maka akan dilakukan pengaturan data yang akan dikirim ke view, yaitu $slug, $subview, dan $category. Selanjutnya, view yang akan ditampilkan adalah \_layout, yang akan memuat konten dari view category/index.

Jika kategori tidak ditemukan (nilai dari $category kosong), maka akan ditampilkan view category/notfound dengan pengaturan $subview sebagai category/notfound.

**File application/views/category/index.php**

<**div** class="max-w-screen-xl mx-auto mt-10">

  <?php if (!empty($slug)): ?>

    <**h1** class="text-2xl font-semibold p-3"><?= $slug ?></**h1**>

  <?php endif; ?>

  <**div** class="grid grid-cols-1 gap-4 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4">

    <?php if (!empty($category)): ?>

      <?php foreach ($category as $row): ?>

        <**a** href="<?= base\_url('product/detail/' . $row->slug\_product) ?>">

          <**div** class="p-4 border rounded-lg shadow-md bg-white">

            <**div** class="relative overflow-hidden">

              <**img** src="<?= base\_url('assets/img/upload/product/' . $row->image\_product) ?>" class="w-full h-40 object-cover rounded-lg" />

            </**div**>

            <**div** class="mt-4">

              <**h3** class="text-lg font-semibold text-gray-800">

                <?= $row->name ?>

              </**h3**>

              <**p** class="mt-1 text-gray-600"><?= $row->description ?></**p**>

              <**p** class="mt-2 text-gray-800 font-semibold">

                Rp. <?= number\_format($row->price, 2) ?>

              </**p**>

            </**div**>

          </**div**>

        </**a**>

      <?php endforeach; ?>

      <?php endif; ?>

  </**div**>

</**div**>

Terdapat sebuah <div> dengan class max-w-screen-xl mx-auto mt-10, yang digunakan untuk mengatur lebar, posisi, dan margin pada konten.

Pada baris kode <?php if (!empty($slug)): ?>, dilakukan pengecekan apakah variabel $slug tidak kosong. Jika tidak kosong, maka akan ditampilkan sebuah <h1> dengan class text-2xl font-semibold p-3 yang berisi nilai dari variabel $slug.

Di dalam <div> dengan class grid grid-cols-1 gap-4 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4, ditampilkan daftar produk yang terkait dengan kategori tersebut.

Pada baris kode <?php if (!empty($category)): ?>, dilakukan pengecekan apakah variabel $category tidak kosong. Jika tidak kosong, maka akan dilakukan perulangan menggunakan foreach untuk setiap elemen dalam $category.

Pada setiap iterasi perulangan, ditampilkan sebuah <a> yang mengarah ke halaman detail produk dengan URL berdasarkan slug produk.

Di dalam <a>, terdapat sebuah <div> dengan class p-4 border rounded-lg shadow-md bg-white yang menampilkan informasi produk.

Di dalam <div>, terdapat sebuah <img> dengan atribut src yang mengarah ke gambar produk.

Di dalam <div>, terdapat sebuah <div> dengan class mt-4 yang menampilkan informasi nama, deskripsi, dan harga produk.

Pada baris kode Rp. <?= number\_format($row->price, 2) ?>, harga produk diformat dengan menggunakan number\_format() untuk menampilkan dua angka desimal.

**File application/views/category/notfound.php**

<**div** class="max-w-screen-xl mx-auto mt-10">

    <**h1** class="text-2xl font-semibold p-3">Not found</**h1**>

</**div**>

Untuk menghandle category

**File application/controllers/ Product.php**

 public function detail($slug)

    {

        $product = $this->ModelProduct->findBySlug($slug);

        if ($product) {

            $data['slug'] = $slug;

            $data['subview'] = 'product/index';

            $data['product'] = $product;

            $this->load->view('\_layout', $data);

        } else {

            $data['subview'] = 'category/notfound';

            $this->load->view('\_layout', $data);

        }

    }

Menerima parameter $slug, yang digunakan untuk mencari produk berdasarkan slug tersebut.

Menggunakan method findBySlug($slug) pada model ModelProduct untuk mencari produk dengan slug yang sesuai.

Jika produk ditemukan (kondisi if ($product)), maka dilakukan pengaturan data yang akan ditampilkan pada halaman, yaitu $slug (slug kategori), $subview (view yang akan ditampilkan, dalam hal ini 'product/index'), dan $product (data produk yang ditemukan). Selanjutnya, view tersebut akan dimuat menggunakan method load->view('\_layout', $data).

Jika produk tidak ditemukan (kondisi else), maka halaman 'category/notfound' akan ditampilkan. Pada halaman tersebut, akan menampilkan pesan "Not found" yang telah dijelaskan sebelumnya.

 public function updateQuantity()

    {

        $cart = $this->input->post('cart'); *// Mendapatkan data cart dari input*

        try {

            if (!empty($cart)) {

                $cartData = json\_decode($cart, true); *// Mengubah string JSON menjadi array asosiatif*

                foreach ($cartData as $item) {

                    $product\_id = $item['product\_id'];

                    $quantity = $item['quantity'];

*// Mendapatkan informasi produk dari database*

                    $product = $this->ModelProduct->findById($product\_id);

                    $currentStock = $product[0]->countInStock;

*// Mengurangi stok berdasarkan jumlah produk di cart*

                    $newStock = $currentStock - $quantity;

*// Memperbarui nilai countInStock di database*

                    $result = $this->ModelProduct->updateQuantity($product\_id, $newStock);

                    if ($result == true) {

                        $response = "Product quantity updated: " . $quantity . " for product ID: " . $product\_id;

                    } else {

                        $response = "Error updating quantity for product ID: " . $product\_id;

                    }

                    echo json\_encode($response); *// Mengirimkan respon sebagai JSON*

                }

            } else {

                echo "No cart data received";

            }

        } catch (Exception $e) {

            $response = $e->getMessage();

            echo json\_encode($response); *// Mengirimkan respon sebagai JSON*

        }

    }

Mendapatkan data cart dari input menggunakan $this->input->post('cart').

Mengubah string JSON yang diperoleh dari data cart menjadi array asosiatif menggunakan json\_decode($cart, true).

Melakukan iterasi terhadap setiap item dalam cart menggunakan foreach loop.

Untuk setiap item, mendapatkan product\_id dan quantity.

Mengambil informasi produk dari database menggunakan ModelProduct->findById($product\_id).

Mengambil nilai countInStock (jumlah stok saat ini) dari produk tersebut.

Mengurangi stok dengan jumlah produk dalam cart menggunakan operasi $newStock = $currentStock - $quantity.

Memperbarui nilai countInStock di database menggunakan ModelProduct->updateQuantity($product\_id, $newStock).

Berdasarkan hasil update, mengatur respon yang akan dikirimkan. Jika update berhasil, respon akan berisi pesan sukses. Jika tidak, respon akan berisi pesan error.

Mengirimkan respon sebagai JSON menggunakan json\_encode($response).

Jika tidak ada data cart yang diterima, akan mengirimkan pesan "No cart data received".

Jika terjadi exception atau error selama proses, akan mengirimkan pesan error sebagai respon JSON.

<**script**>

    function updateQuantity() {

        var quantitySelect = document.getElementById('quantitySelect');

        var selectedQuantity = quantitySelect.options[quantitySelect.selectedIndex].value;

        document.getElementById('quantity').value = selectedQuantity;

    }

    function addToCart() {

        var selectedQuantity = parseInt(document.getElementById('quantity').value);

        if (isNaN(selectedQuantity) || selectedQuantity <= 0) {

            alert('Please select a valid quantity.');

            return;

        }

        var form = document.getElementById('cartForm');

        var formData = new FormData(form);

        fetch(form.action, {

                method: 'POST',

                body: formData

            })

            .then(response => {

                if (response.ok) {

                    alert('Product added to cart successfully.');

                    window.location.href = '<?= base\_url(); ?>';

                } else {

                    throw new Error('An error occurred while adding the product to cart.');

                }

            })

            .catch(error => {

                console.error('Error adding product to cart: ', error);

                alert('An error occurred while adding the product to cart.');

            });

    }

</**script**>

<**div** class="max-w-screen-xl mx-auto mt-10">

    <**div** class="flex flex-col md:flex-row">

        <?php if (!empty($product)) {

            foreach ($product as $row) { ?>

                <**div** class="md:w-1/2 mb-4 md:mb-0">

                    <**img** src="<?= base\_url('assets/img/upload/product/' . $row->image\_product) ?>" alt="<?= $row->name; ?>" class="w-full h-auto" />

                </**div**>

                <**div** class="md:w-1/2 md:pl-8">

                    <**div** class="bg-white shadow p-4 rounded-md">

                        <**h1** class="text-xl xl:text-2xl font-medium mb-4">

                            <?= $row->name; ?>

                        </**h1**>

                        <**p** class="mt-2 text-gray-600"><?= $row->description; ?></**p**>

                        <**div** class="mt-6">

                            <**h1**>Selected Quantity</**h1**>

                            <**select** id="quantitySelect" name="quantity" onchange="updateQuantity();" class="border border-gray-300 rounded-md py-2 px-4">

                                <**option** value="pilih quantity">Pilih quantity</**option**>

                                <?php

                                for ($i = 1; $i <= $row->countInStock; $i++) {

                                    echo "<option value='$i'>$i</option>";

                                }

                                ?>

                            </**select**>

                        </**div**>

                        <**p** class="mt-6 text-gray-800 font-semibold">

                            Rp.<?= $row->price; ?>

                        </**p**>

                        <**div** class="mt-6">

                            <?php if ($row->countInStock > 0) { ?>

                                <**form** id="cartForm" action="<?= base\_url('cart/create') ?>" method="POST">

                                    <**input** type='hidden' name='name' value='<?= $row->name; ?>'>

                                    <**input** type='hidden' name='price' value='<?= $row->price; ?>'>

                                    <**input** type='hidden' name='product\_id' value='<?= $row->product\_id; ?>'>

                                    <**input** type='hidden' name='image\_product' value='<?= $row->image\_product; ?>'>

                                    <**input** type="hidden" id="quantity" name="quantity" />

                                    <?php if ($this->session->userdata('email')) { ?>

                                        <**button** class="bg-gray-800 text-white py-2 px-4 rounded-md mt-6" type="button" onclick="addToCart();">

                                            Add to Cart

                                        </**button**>

                                    <?php } else { ?>

                                        <**button** class="bg-gray-300 text-gray-500 py-2 px-4 rounded-md cursor-not-allowed" disabled>

                                            Add to Cart

                                        </**button**>

                                    <?php } ?>

                                </**form**>

                            <?php } else { ?>

                                <**div**>

                                    <**h1**>Out Of Stock</**h1**>

                                    <**button** class="bg-gray-300 text-gray-500 py-2 px-4 rounded-md cursor-not-allowed" disabled>

                                        Add to Cart

                                    </**button**>

                                </**div**>

                            <?php } ?>

                        </**div**>

                    </**div**>

                </**div**>

        <?php }

        } ?>

    </**div**>

</**div**>

**File application/views/product/index.php**

<**script**>

    function updateQuantity() {

        var quantitySelect = document.getElementById('quantitySelect');

        var selectedQuantity = quantitySelect.options[quantitySelect.selectedIndex].value;

        document.getElementById('quantity').value = selectedQuantity;

    }

    function addToCart() {

        var selectedQuantity = parseInt(document.getElementById('quantity').value);

        if (isNaN(selectedQuantity) || selectedQuantity <= 0) {

            alert('Please select a valid quantity.');

            return;

        }

        var form = document.getElementById('cartForm');

        var formData = new FormData(form);

        fetch(form.action, {

                method: 'POST',

                body: formData

            })

            .then(response => {

                if (response.ok) {

                    alert('Product added to cart successfully.');

                    window.location.href = '<?= base\_url(); ?>';

                } else {

                    throw new Error('An error occurred while adding the product to cart.');

                }

            })

            .catch(error => {

                console.error('Error adding product to cart: ', error);

                alert('An error occurred while adding the product to cart.');

            });

    }

</**script**>

<**div** class="max-w-screen-xl mx-auto mt-10">

    <**div** class="flex flex-col md:flex-row">

        <?php if (!empty($product)) {

            foreach ($product as $row) { ?>

                <**div** class="md:w-1/2 mb-4 md:mb-0">

                    <**img** src="<?= base\_url('assets/img/upload/product/' . $row->image\_product) ?>" alt="<?= $row->name; ?>" class="w-full h-auto" />

                </**div**>

                <**div** class="md:w-1/2 md:pl-8">

                    <**div** class="bg-white shadow p-4 rounded-md">

                        <**h1** class="text-xl xl:text-2xl font-medium mb-4">

                            <?= $row->name; ?>

                        </**h1**>

                        <**p** class="mt-2 text-gray-600"><?= $row->description; ?></**p**>

                        <**div** class="mt-6">

                            <**h1**>Selected Quantity</**h1**>

                            <**select** id="quantitySelect" name="quantity" onchange="updateQuantity();" class="border border-gray-300 rounded-md py-2 px-4">

                                <**option** value="pilih quantity">Pilih quantity</**option**>

                                <?php

                                for ($i = 1; $i <= $row->countInStock; $i++) {

                                    echo "<option value='$i'>$i</option>";

                                }

                                ?>

                            </**select**>

                        </**div**>

                        <**p** class="mt-6 text-gray-800 font-semibold">

                            Rp.<?= $row->price; ?>

                        </**p**>

                        <**div** class="mt-6">

                            <?php if ($row->countInStock > 0) { ?>

                                <**form** id="cartForm" action="<?= base\_url('cart/create') ?>" method="POST">

                                    <**input** type='hidden' name='name' value='<?= $row->name; ?>'>

                                    <**input** type='hidden' name='price' value='<?= $row->price; ?>'>

                                    <**input** type='hidden' name='product\_id' value='<?= $row->product\_id; ?>'>

                                    <**input** type='hidden' name='image\_product' value='<?= $row->image\_product; ?>'>

                                    <**input** type="hidden" id="quantity" name="quantity" />

                                    <?php if ($this->session->userdata('email')) { ?>

                                        <**button** class="bg-gray-800 text-white py-2 px-4 rounded-md mt-6" type="button" onclick="addToCart();">

                                            Add to Cart

                                        </**button**>

                                    <?php } else { ?>

                                        <**button** class="bg-gray-300 text-gray-500 py-2 px-4 rounded-md cursor-not-allowed" disabled>

                                            Add to Cart

                                        </**button**>

                                    <?php } ?>

                                </**form**>

                            <?php } else { ?>

                                <**div**>

                                    <**h1**>Out Of Stock</**h1**>

                                    <**button** class="bg-gray-300 text-gray-500 py-2 px-4 rounded-md cursor-not-allowed" disabled>

                                        Add to Cart

                                    </**button**>

                                </**div**>

                            <?php } ?>

                        </**div**>

                    </**div**>

                </**div**>

        <?php }

        } ?>

    </**div**>

</**div**>

updateQuantity():

Fungsi ini dipanggil saat ada perubahan pada elemen quantitySelect (dropdown untuk memilih jumlah produk).

Fungsi ini mengambil nilai jumlah yang dipilih dari quantitySelect dan mengatur nilainya pada elemen dengan id quantity.

Tujuannya adalah untuk menyimpan jumlah yang dipilih dalam elemen tersembunyi sehingga dapat dikirimkan dalam form saat menambahkan produk ke keranjang.

addToCart():

Fungsi ini dipanggil saat tombol "Add to Cart" diklik.

Pertama, fungsi ini mengambil nilai jumlah produk yang dipilih dari elemen dengan id quantity.

Kemudian, dilakukan pengecekan apakah nilai jumlah tersebut valid (angka positif).

Jika jumlah valid, fungsi ini membuat objek FormData yang berisi data form dengan id cartForm.

Selanjutnya, menggunakan fetch API, fungsi ini melakukan permintaan POST ke URL yang ditentukan dalam atribut action pada form.

Data form dalam bentuk formData dikirimkan sebagai body permintaan.

Jika permintaan berhasil (status response response.ok), maka akan muncul alert "Product added to cart successfully." dan akan diarahkan ke halaman beranda (window.location.href = '<?= base\_url(); ?>';).

Jika terjadi kesalahan dalam permintaan, akan muncul alert "An error occurred while adding the product to cart.".

**File application/controllers/Cart.php**

 public function index(){

        $data['subview'] = 'cart/index';

        $data['cart'] = $this->ModelCart->findAll();

        $this->load->view('\_layout', $data);

    }

Pertama, $data['subview'] = 'cart/index'; mengatur nilai $data['subview'] menjadi 'cart/index'. Nilai ini akan digunakan untuk memuat view halaman keranjang belanja.

Selanjutnya, $data['cart'] = $this->ModelCart->findAll(); mengambil data semua item keranjang belanja dengan menggunakan model ModelCart dan method findAll(). Data ini akan disimpan dalam variabel $data['cart'].

Terakhir, $this->load->view('\_layout', $data); memuat view dengan nama \_layout dan mengirimkan data $data ke view tersebut. View \_layout akan memuat view 'cart/index' sebagai subview yang akan ditampilkan kepada pengguna.

 public function create()

    {

        $nama = $this->input->post('name');

        $price = $this->input->post('price');

        $image\_product = $this->input->post('image\_product');

        $quantity = $this->input->post('quantity');

        $product\_id = $this->input->post('product\_id');

        $user\_id = $this->session->userdata('user\_id');

        $email = $this->session->userdata('email');

        $result = $this->ModelCart->createCart($nama, $price, $image\_product, $quantity, $user\_id, $email, $product\_id);

        if ($result != true) {

            $this->session->set\_flashdata('pesan', 'Proses membuat Cart Berhasil');

            redirect("home");

        } else {

            $this->session->set\_flashdata('error', 'Error membuat cart');

            redirect("home");

        }

    }

Pertama, data yang dibutuhkan untuk membuat item baru diambil dari input form menggunakan $this->input->post(). Variabel seperti $nama, $price, $image\_product, $quantity, dan $product\_id mewakili nilai-nilai yang diinputkan oleh pengguna.

Selanjutnya, $user\_id = $this->session->userdata('user\_id'); dan $email = $this->session->userdata('email'); digunakan untuk mendapatkan ID pengguna dan email dari data sesi yang aktif.

$result = $this->ModelCart->createCart($nama, $price, $image\_product, $quantity, $user\_id, $email, $product\_id); memanggil method createCart() dari model ModelCart untuk membuat item baru dalam keranjang belanja dengan menggunakan data yang telah dikumpulkan sebelumnya.

Setelah itu, dilakukan pengecekan hasil dari pembuatan item baru. Jika $result tidak bernilai true, maka artinya proses pembuatan item berhasil. Pada kondisi tersebut, digunakan $this->session->set\_flashdata() untuk mengatur pesan sukses, dan pengguna akan dialihkan ke halaman "home" menggunakan redirect("home").

Jika $result bernilai true, maka artinya terjadi error saat membuat item baru dalam keranjang belanja. Pada kondisi tersebut, digunakan $this->session->set\_flashdata() untuk mengatur pesan error, dan pengguna akan dialihkan ke halaman "home" menggunakan redirect("home").

public function delete(){

        $id = $this->uri->segment(3);

        $result = $this->ModelCart->deleteCart($id);

        if($result){

            redirect("cart", 'refresh');

        }else{

            redirect("cart", 'refresh');

        }

    }

Pertama, $id = $this->uri->segment(3); digunakan untuk mengambil segment ketiga dari URI, yang merupakan ID dari item yang akan dihapus. Ini memungkinkan untuk mengidentifikasi item yang tepat yang akan dihapus dari keranjang belanja.

Selanjutnya, $result = $this->ModelCart->deleteCart($id); memanggil method deleteCart() dari model ModelCart untuk menghapus item dari keranjang belanja berdasarkan ID yang diterima.

Setelah itu, dilakukan pengecekan hasil penghapusan item. Jika $result bernilai true, artinya penghapusan item berhasil. Pada kondisi tersebut, pengguna akan diarahkan ulang ke halaman "cart" menggunakan redirect("cart", 'refresh') agar dapat melihat keranjang belanja yang telah diperbarui.

Jika $result tidak bernilai true, artinya terjadi error saat menghapus item dari keranjang belanja. Pada kondisi tersebut, pengguna tetap akan diarahkan ulang ke halaman "cart" menggunakan redirect("cart", 'refresh'), namun tanpa perubahan apapun pada keranjang belanja.

**File application/views/cart/index.php**

<**div** class="relative overflow-x-auto">

    <**div** class="flex justify-center">

        <**div** class="w-full md:w-2/3 card text-center shadow-lg p-3 mb-5 bg-white rounded">

            <**div** class="text-center m-5">My Cart</**div**>

            <**div** class="overflow-x-auto mt-10">

                <**table** class="table-auto w-full text-sm text-left">

                    <**thead**>

                        <**tr**>

                            <**th** class="border border-gray-800 p-2">Name</**th**>

                            <**th** class="border border-gray-800 p-2">Quantity</**th**>

                            <**th** class="border border-gray-800 p-2">Price</**th**>

                            <**th** class="border border-gray-800 p-2">Total Price</**th**>

                            <**th** class="border border-gray-800 p-2">Delete</**th**>

                        </**tr**>

                    </**thead**>

                    <**tbody**>

                        <?php

                        $subtotal = 0; *// Inisialisasi subtotal*

                        $totalProducts = 0; *// Inisialisasi total produk*

                        foreach ($cart as $row) {

                            $totalProducts += $row->quantity; *// Tambahkan jumlah produk ke total produk*

                        ?>

                            <**tr**>

                                <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->name ?></**td**>

                                <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->quantity ?></**td**>

                                <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->price ?></**td**>

                                <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->quantity \* $row->price; ?></**td**>

                                <**td** class="border border-gray-800 px-4 py-2">

                                    <**a** onclick="return confirm('Are you sure you want to delete this record?');" href="<?= base\_url('cart/delete/' . $row->cart\_id); ?>">

                                        <**i** style="color: red" class="far fa-trash-alt"></**i**>

                                    </**a**>

                                </**td**>

                            </**tr**>

                        <?php

                            $subtotal += $row->quantity \* $row->price; *// Tambahkan subtotal dengan total harga untuk setiap produk*

                        }

                        ?>

                    </**tbody**>

                </**table**>

            </**div**>

            <**hr** />

            <**h2** class="text-center">SubTotal: Rp:<?php echo $subtotal; ?></**h2**>

            <**p** class="text-center">Total Products: <?php echo $totalProducts; ?></**p**>

            <**hr** />

            <**div** class="flex justify-center">

                <**div** id="paypal-button-container"></**div**>

            </**div**>

        </**div**>

    </**div**>

</**div**>

<**style**>

    #paypal-button-container .paypal-button-container {

        transform: scale(0.8);

        transform-origin: center;

    }

</**style**>

<**script** src="https://www.paypal.com/sdk/js?client-id=AWB0KnJcTsTgLbSZbr9VJTJx3Llrwy6e9DurXdi5Ir1VN2zIFvRq62hoVo9W54zi8Ghpob-D8VHDx-dz" data-sdk-integration-source="button-factory"></**script**>

<**script**>

    function generateOrderID() {

        var characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';

        var length = 10;

        var orderID = '';

        for (var i = 0; i < length; i++) {

            var randomIndex = Math.floor(Math.random() \* characters.length);

            orderID += characters.charAt(randomIndex);

        }

        return orderID;

    }

    var baseUrl = '<?php echo base\_url(); ?>';

    var orderID = generateOrderID();

    function updateProductQuantities() {

        var products = <?php echo json\_encode($cart); ?>;

        var formData = new FormData();

        console.log("products", products);

        formData.append('cart', JSON.stringify(products));

        fetch(`${baseUrl}product/updateQuantity`, {

                method: 'POST',

                body: formData

            })

            .then(response => response.text())

            .then(data => {

                console.log('Product quantity updated: ', data);

            })

            .catch(error => {

                console.error('Error updating product quantity: ', error);

            });

    }

    function createOrder(postData) {

        fetch(`${baseUrl}order/create`, {

                method: 'POST',

                headers: {

                    'Content-Type': 'application/x-www-form-urlencoded'

                },

                body: Object.keys(postData)

                    .map((key) => encodeURIComponent(key) + '=' + encodeURIComponent(postData[key]))

                    .join('&')

            })

            .then(response => response.text())

            .then(data => {

                window.location.reload();

                console.log(data);

            })

            .catch(error => {

                alert('Error: ' + error);

                console.error(error);

            });

    }

    paypal.Buttons({

        createOrder: function(data, actions) {

            return actions.order.create({

                purchase\_units: [{

                    amount: {

                        value: '<?php echo $subtotal; ?>'

                    },

                    custom\_id: orderID

                }]

            });

        },

        onApprove: function(data, actions) {

            return actions.order.capture().then(function(details) {

                var shippingAddress = details.purchase\_units[0].shipping.address;

                var orderID = details.purchase\_units[0].custom\_id;

                var email = details.payer.email\_address;

                var postalCode = shippingAddress.postal\_code;

                var countryCode = shippingAddress.country\_code;

                var totalProduct = '<?php echo $totalProducts; ?>';

                var postData = {

                    order\_id: orderID,

                    email: email,

                    postal\_code: postalCode,

                    country\_code: countryCode,

                    total\_product: totalProduct,

                    total\_price: '<?php echo $subtotal; ?>'

                };

                updateProductQuantities();

                createOrder(postData);

*// alert('Payment successful' +*

*//     '\nShipping Address: ' + JSON.stringify(shippingAddress) +*

*//     '\nOrder ID: ' + orderID +*

*//     '\nEmail: ' + email +*

*//     '\nPostal Code: ' + postalCode +*

*//     '\nCountry Code: ' + countryCode*

*// );*

            });

        }

    }).render('#paypal-button-container');

</**script**>

Script pertama adalah tampilan HTML dan CSS untuk menampilkan daftar item dalam keranjang belanja. Ini termasuk pembuatan tabel yang menampilkan nama, jumlah, harga, dan total harga untuk setiap item dalam keranjang belanja. Ada juga kolom "Delete" yang memungkinkan pengguna untuk menghapus item.

Setelah tabel, ditampilkan subtotal dan total produk dalam keranjang belanja.

Kemudian, ada script JavaScript untuk mengatur logika pembayaran menggunakan PayPal. Script ini menggunakan SDK PayPal untuk membuat tombol pembayaran PayPal dan menangani logika pembayaran.

Fungsi generateOrderID() digunakan untuk menghasilkan ID pesanan acak yang akan digunakan untuk mengidentifikasi pesanan saat melakukan pembayaran dengan PayPal.

Fungsi updateProductQuantities() digunakan untuk mengupdate kuantitas produk dalam database setelah pembayaran berhasil dilakukan. Fungsi ini mengirimkan permintaan HTTP POST ke URL /product/updateQuantity dengan menggunakan Fetch API untuk memperbarui kuantitas produk.

Fungsi createOrder() digunakan untuk membuat pesanan baru setelah pembayaran berhasil dilakukan. Fungsi ini mengirimkan permintaan HTTP POST ke URL /order/create untuk membuat pesanan dengan menggunakan data yang diperoleh dari pembayaran PayPal.

Bagian terakhir adalah penggunaan objek paypal.Buttons() untuk membuat tombol pembayaran PayPal. Metode createOrder digunakan untuk membuat pesanan saat tombol pembayaran diklik. Metode onApprove digunakan untuk menangani tindakan setelah pembayaran berhasil dilakukan. Setelah pembayaran berhasil, fungsi updateProductQuantities() dan createOrder() dipanggil untuk mengupdate kuantitas produk dan membuat pesanan.

**File application/controllers/Order.php**

    public function index()

    {

        $data['subview'] = 'order/index';

        $data['order'] = $this->ModelOrder->findByUserId($this->session->userdata('user\_id'));

        $this->load->view('\_layout', $data);

    }

Fungsi index() menerima permintaan HTTP GET dan tidak mengambil argumen tambahan.

Variabel $data digunakan untuk menyimpan data yang akan dikirimkan ke tampilan. Anda mengatur nilai $data['subview'] sebagai 'order/index' yang menunjukkan bahwa tampilan yang akan digunakan adalah 'order/index'.

Selanjutnya, Anda menggunakan Model ModelOrder untuk mengambil daftar pesanan menggunakan metode findByUserId(). Metode ini menerima user\_id dari pengguna yang sedang aktif (diperoleh melalui $this->session->userdata('user\_id')) dan mengembalikan daftar pesanan yang terkait dengan pengguna tersebut.

Hasil dari pengambilan daftar pesanan dimasukkan ke dalam variabel $data['order'].

Terakhir, Anda memuat tampilan \_layout dengan mengirimkan data $data ke tampilan. Dalam tampilan \_layout, variabel $subview akan digunakan untuk memuat tampilan 'order/index'.

 public function create()

    {

        $orderID = $this->input->post('order\_id');

        $email = $this->input->post('email');

        $postalCode = $this->input->post('postal\_code');

        $countryCode = $this->input->post('country\_code');

        $totalProduct = $this->input->post('total\_product');

        $totalPrice = $this->input->post('total\_price');

        $user\_id = $this->session->userdata('user\_id');

        if ($this->ModelOrder->createOrder($orderID, $email, $postalCode, $countryCode, $totalProduct, $totalPrice, $user\_id) == true) {

            $orderData = array(

                'order\_id' => $orderID,

                'email' => $email,

                'postal\_code' => $postalCode,

                'country\_code' => $countryCode,

                'total\_product' => $totalProduct,

                'total\_price' => $totalPrice

            );

            $this->ModelCart->deleteAll($this->session->userdata('email'));

            $response = array(

                'status' => 'success',

                'message' => 'Order created successfully',

                'mydat' => $orderData

            );

            $this->output->set\_content\_type('application/json')->set\_output(json\_encode($response));

        } else {

            $response = array(

                'status' => 'failed',

                'message' => 'Order creation failed'

            );

            $this->output->set\_content\_type('application/json')->set\_output(json\_encode($response));

        }

    }

Fungsi create() menerima permintaan HTTP POST dan tidak mengambil argumen tambahan.

Anda menggunakan $this->input->post() untuk mengambil nilai dari inputan form yang dikirimkan dengan metode POST.

Variabel $orderID, $email, $postalCode, $countryCode, $totalProduct, $totalPrice dan $user\_id diinisialisasi dengan nilai yang diperoleh dari inputan form.

Anda menggunakan model ModelOrder untuk membuat pesanan baru dengan memanggil metode createOrder(). Metode ini menerima parameter-order seperti $orderID, $email, $postalCode, $countryCode, $totalProduct, $totalPrice, dan $user\_id. Jika pesanan berhasil dibuat, metode ini akan mengembalikan true.

Jika pesanan berhasil dibuat, Anda membuat array $orderData yang berisi informasi pesanan yang berhasil dibuat.

Selanjutnya, Anda menggunakan model ModelCart untuk menghapus semua item di keranjang pengguna yang sedang aktif menggunakan metode deleteAll(). Metode ini menerima email pengguna yang diperoleh melalui $this->session->userdata('email').

Anda mengatur respon yang akan dikirimkan sebagai JSON dengan mengatur tipe konten menjadi application/json menggunakan $this->output->set\_content\_type('application/json').

Jika pesanan berhasil dibuat, Anda mengirimkan respon dengan status success, pesan sukses, dan data pesanan yang dibuat.

Jika pesanan gagal dibuat, Anda mengirimkan respon dengan status failed dan pesan kegagalan.

  public function generatePdf(){

        $user\_id = $this->session->userdata('user\_id');

        $data['order'] = $this->ModelOrder->findByUserId($user\_id);

        $this->load->library('dompdf\_gen');

        $this->load->view('order/pdf', $data);

        $paper\_size = 'A4'; *// ukuran kertas*

        $orientation = 'landscape'; *//tipe format kertas potrait atau landscape*

        $html = $this->output->get\_output();

        $this->dompdf->set\_paper($paper\_size, $orientation);

*//Convert to PDF*

        $this->dompdf->load\_html($html);

        $this->dompdf->render();

        $this->dompdf->stream("bukti-order-$user\_id.pdf", array('Attachment' => 0));

    }

Fungsi generatePdf() pertama-tama mengambil user\_id dari sesi pengguna menggunakan $this->session->userdata('user\_id').

Kemudian, fungsi ini menggunakan model ModelOrder untuk mencari pesanan berdasarkan user\_id yang diperoleh. Data pesanan tersebut kemudian disimpan dalam array $data.

Fungsi ini memuat library dompdf menggunakan $this->load->library('dompdf\_gen'). Library ini digunakan untuk menghasilkan file PDF.

Selanjutnya, fungsi ini memuat view order/pdf dengan menyertakan data pesanan yang ditemukan sebelumnya.

Anda dapat menentukan ukuran kertas dan orientasi file PDF yang akan dihasilkan. Dalam contoh ini, ukuran kertas diatur sebagai 'A4' dan orientasinya adalah 'landscape'.

Fungsi load\_html() dari library dompdf digunakan untuk memuat konten HTML yang akan dikonversi menjadi file PDF. Konten HTML yang akan dimuat adalah output dari $this->output->get\_output().

Setelah memuat konten HTML, Anda memanggil metode render() untuk menghasilkan file PDF.

Terakhir, Anda menggunakan metode stream() untuk mengirim file PDF yang dihasilkan ke browser pengguna. Nama file PDF disetel sebagai "bukti-order-$user\_id.pdf" dan opsi 'Attachment' diatur sebagai 0 untuk menghindari bahwa file PDF tersebut diunduh sebagai lampiran.

**File application/views/order/index.php**

<**div** class="relative overflow-x-auto">

    <**div** class="flex justify-center">

        <**div** class="w-full md:w-2/3 card text-center shadow-lg p-3 mb-5 bg-white rounded">

            <**div** class="text-center m-5">My Order</**div**>

            <**div** class="overflow-x-auto mt-10">

                <**table** class="table-auto w-full text-sm text-left">

                    <**thead**>

                        <**tr**>

                            <**th** class="border border-gray-800 p-2">Order ID</**th**>

                            <**th** class="border border-gray-800 p-2">Email</**th**>

                            <**th** class="border border-gray-800 p-2">Postal Code</**th**>

                            <**th** class="border border-gray-800 p-2">Country Code</**th**>

                            <**th** class="border border-gray-800 p-2">Total Product</**th**>

                            <**th** class="border border-gray-800 p-2">Total Price</**th**>

                            <**th** class="border border-gray-800 p-2">Delete</**th**>

                        </**tr**>

                    </**thead**>

                    <**tbody**>

                        <?php

                        $subtotal = 0; *// Inisialisasi subtotal*

                        $totalProducts = 0; *// Inisialisasi total produk*

                        foreach ($order as $row) {

                        ?>

                            <**tr**>

                                <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->order\_id ?></**td**>

                                <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->email ?></**td**>

                                <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->postal\_code ?></**td**>

                                <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->country\_code ?></**td**>

                                <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->total\_product ?></**td**>

                                <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->total\_price ?></**td**>

                                <**td** class="border border-gray-800 px-4 py-2">

                                    <**a** onclick="return confirm('Are you sure you want to delete this record?');" href="<?= base\_url('order/delete/' . $row->order\_id); ?>">

                                        <**i** style="color: red" class="far fa-trash-alt"></**i**>

                                    </**a**>

                                </**td**>

                            </**tr**>

                        <?php

                        }

                        ?>

                    </**tbody**>

                </**table**>

            </**div**>

            <**div** class="flex justify-center mt-5">

                <**a** href="<?= base\_url('order/generatePdf/'); ?>" class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded">

                    <**span** class="fas fa-file-pdf"></**span**>  Generate PDF

                    </**a**>

            </**div**>

        </**div**>

    </**div**>

</**div**>

Kode ini menggunakan foreach untuk mengulang setiap pesanan dalam array $order yang diberikan.

Setiap pesanan ditampilkan dalam baris tabel dengan menggunakan elemen <tr>.

Untuk setiap pesanan, atribut-atribut seperti order\_id, email, postal\_code, country\_code, total\_product, dan total\_price diambil dari objek $row dan ditampilkan dalam sel tabel menggunakan elemen <td>.

Terdapat juga kolom "Delete" yang berisi tautan untuk menghapus pesanan. Tautan ini akan mengarahkan pengguna ke URL order/delete/{order\_id} ketika diklik, dengan order\_id sebagai parameter.

Di bagian bawah tabel, terdapat tautan "Generate PDF" yang mengarahkan pengguna ke URL order/generatePdf/. Tautan ini digunakan untuk menghasilkan file PDF dari pesanan menggunakan fungsi generatePdf() yang telah Anda definisikan sebelumnya.

**File application/views/order/pdf.php**

<**style**>

    .table-container {

        overflow-x: auto;

    }

    .table-container table {

        width: 100%;

        border-collapse: collapse;

    }

    .table-container th,

    .table-container td {

        padding: 8px;

        text-align: left;

    }

    .table-container th {

        background-color: #f2f2f2;

    }

    .table-container .center {

        text-align: center;

    }

    @media only screen and (max-width: 600px) {

        .table-container table,

        .table-container th,

        .table-container td {

            font-size: 12px;

        }

    }

</**style**>

<**div** class="table-container">

    <**table** border="1">

        <**tr**>

            <**td**>

                <**table** border="1">

                    <**tr**>

                        <**th**>Order ID</**th**>

                        <**th**>Email</**th**>

                        <**th** >Postal Code</**th**>

                        <**th**>Country Code</**th**>

                        <**th** >Total Product</**th**>

                        <**th** >Total Price</**th**>

                    </**tr**>

                    <?php

                    $no = 1;

                    foreach ($order as $row) {

                    ?>

                        <**tr**>

                            <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->order\_id ?></**td**>

                            <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->email ?></**td**>

                            <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->postal\_code ?></**td**>

                            <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->country\_code ?></**td**>

                            <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->total\_product ?></**td**>

                            <**td** class="border border-gray-800 px-4 py-2"><?php echo $row->total\_price ?></**td**>

                        </**tr**>

                    <?php $no++;

                    } ?>

                </**table**>

            </**td**>

        </**tr**>

        <**tr**>

            <**td**>

                <**hr**>

            </**td**>

        </**tr**>

        <**tr**>

            <**td** class="center">

                <?= date('d M Y H:i:s') ?>

            </**td**>

        </**tr**>

    </**table**>

</**div**>

Gaya CSS di dalam tag <style> digunakan untuk memperbarui tampilan tabel. Gaya ini akan menghasilkan tabel yang responsif dengan overflow-x auto saat layar kecil (lebar maksimum 600px). Selain itu, gaya juga menentukan pengaturan tampilan sel dan latar belakang header tabel.

Struktur tabel di dalam tag <div class="table-container"> tetap sama dengan yang sebelumnya, tetapi dengan beberapa perubahan markup.

Setiap pesanan masih ditampilkan dalam baris tabel menggunakan elemen <tr>, dan atribut-atribut pesanan seperti order\_id, email, postal\_code, country\_code, total\_product, dan total\_price tetap ditampilkan dalam sel tabel menggunakan elemen <td>.

Penambahan variabel $no digunakan untuk mengatur nomor urut pada setiap baris pesanan.

Di bagian bawah tabel, terdapat baris terpisah dengan elemen <hr> dan baris yang menampilkan tanggal dan waktu saat ini menggunakan fungsi date('d M Y H:i:s').