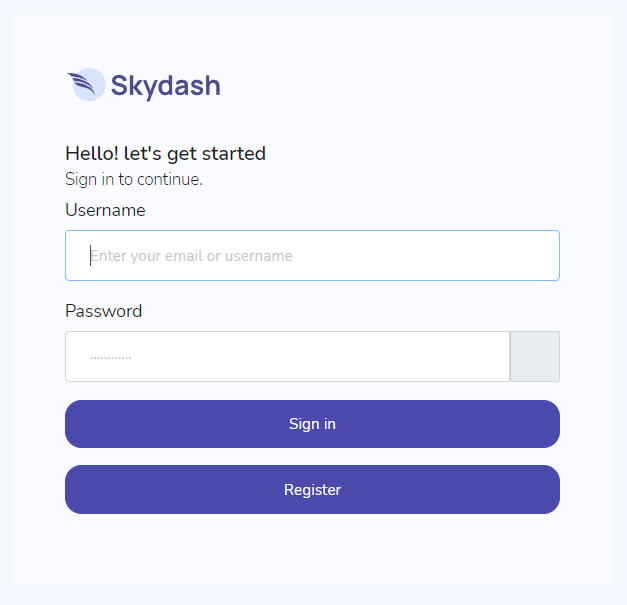
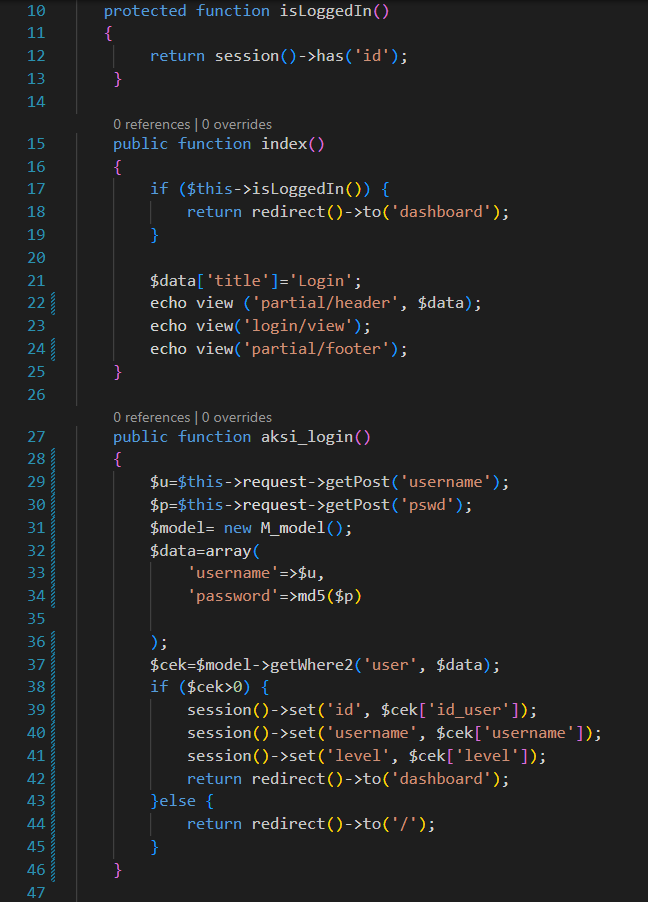
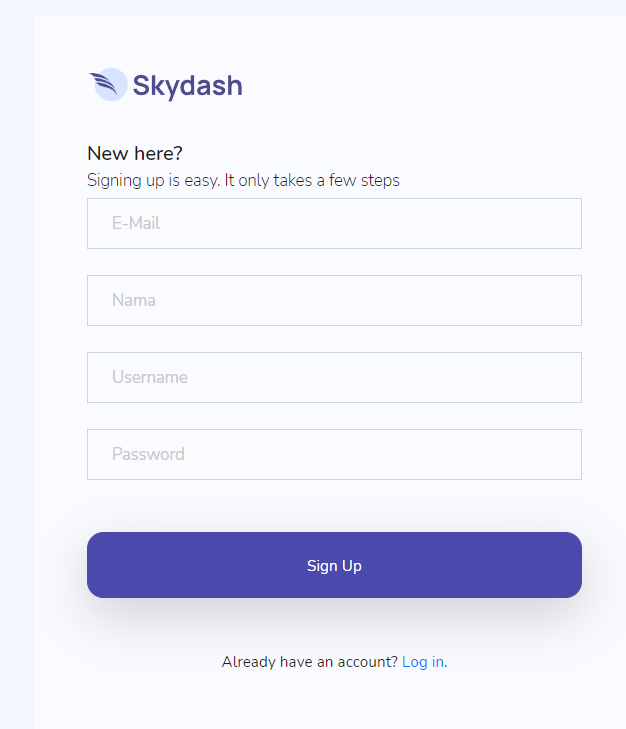
1. Login

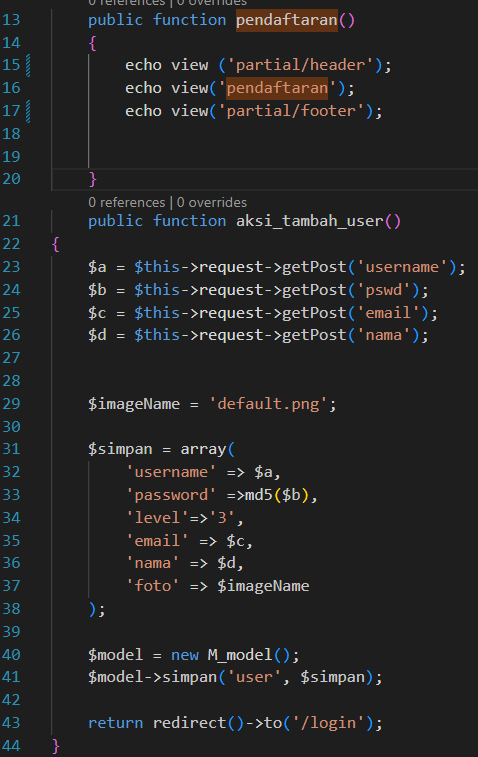




Kode PHP di atas mengimplementasikan fungsi-fungsi autentikasi pada aplikasi web. Fungsi `isLoggedIn` memeriksa apakah pengguna sudah login dengan mengecek keberadaan data 'id' dalam session. Fungsi `index` memastikan pengguna diarahkan ke halaman 'dashboard' jika sudah login, dan menampilkan halaman login jika belum, lengkap dengan header dan footer.Fungsi `aksi\_login` menangani proses login dengan memvalidasi data pengguna menggunakan model `M\_model`. Jika data cocok, sesi pengguna diinisialisasi dengan informasi seperti 'id', 'username', dan 'level', dan pengguna diarahkan ke halaman 'dashboard'. Jika tidak cocok, pengguna diarahkan kembali ke halaman login. Penting untuk memperhatikan bahwa penggunaan metode hash seperti md5 pada password tidak direkomendasikan untuk keamanan yang lebih baik; sebaiknya, gunakan metode hash yang lebih aman seperti bcrypt, dan pastikan mekanisme session yang digunakan aman untuk menghindari potensi risiko keamanan.

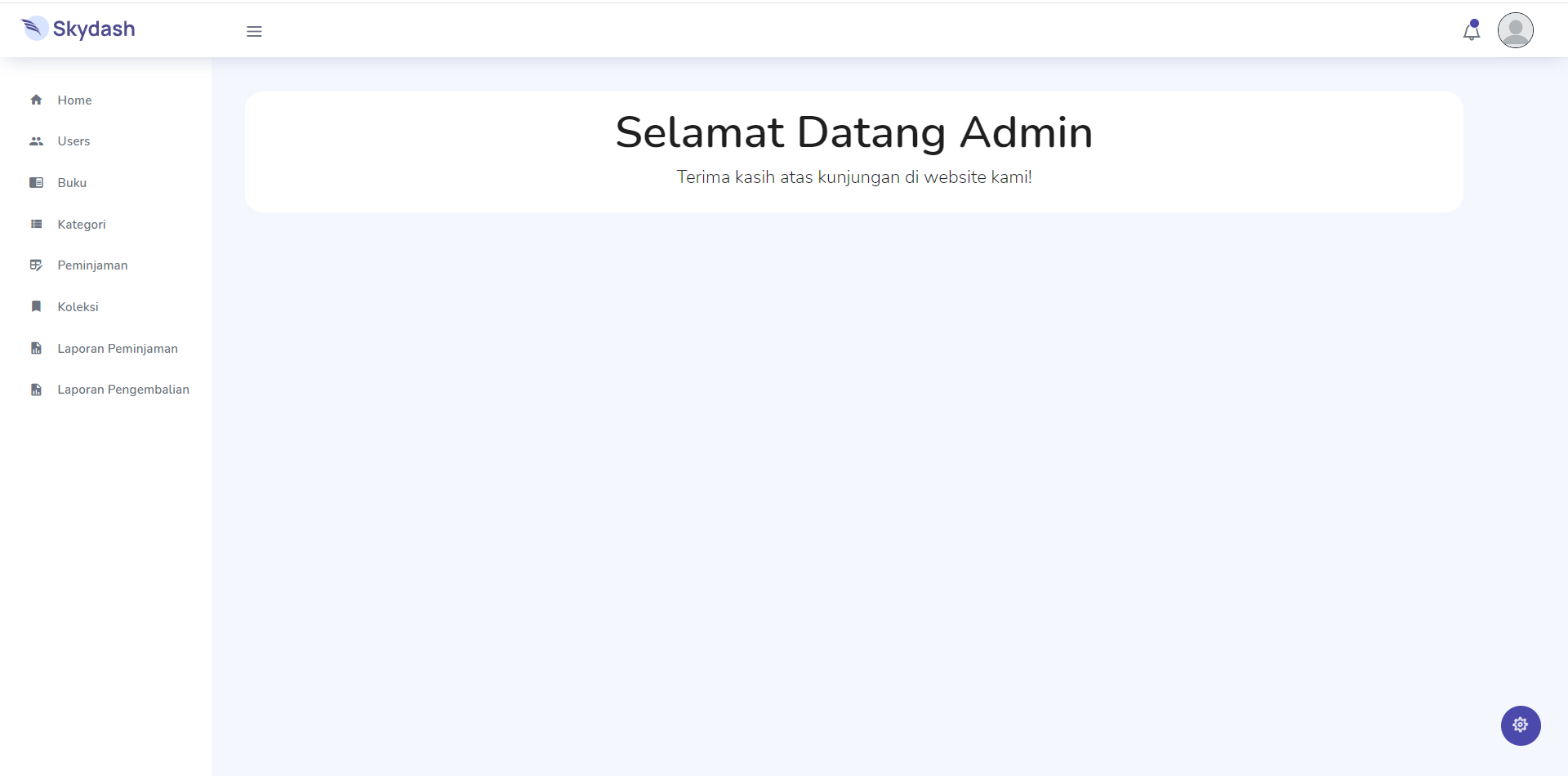
1. Register

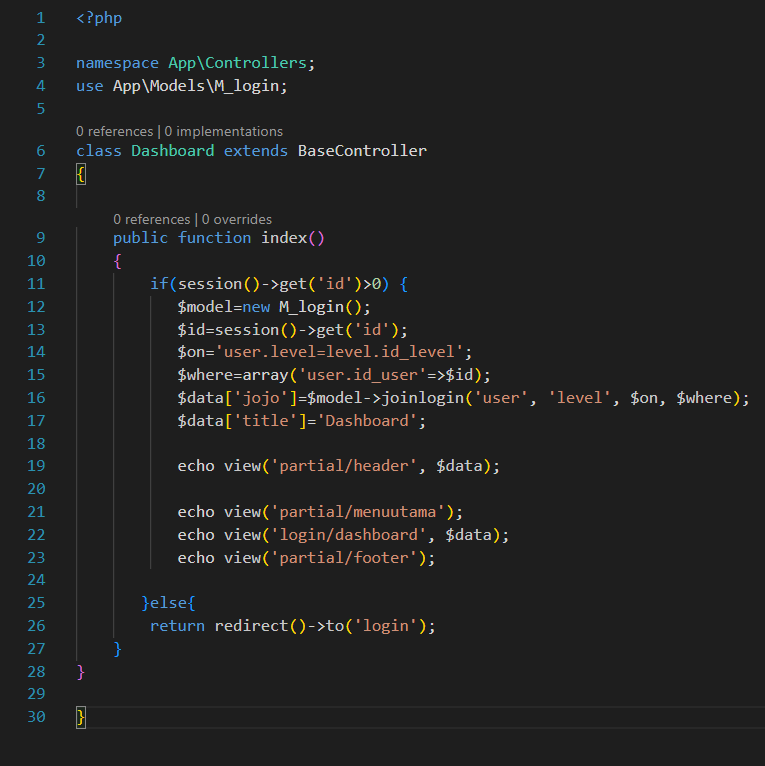




Metode pendaftaran pengguna pada fungsi pendaftaran() terdiri dari tiga tahap. Pertama, tampilan pembuka dan footer disisipkan menggunakan fungsi echo view() untuk membangun tata letak halaman. Kedua, fungsi aksi\_tambah\_user() menangani pengiriman formulir pendaftaran dengan mengambil data dari formulir menggunakan metode POST. Data tersebut kemudian dimasukkan ke dalam array yang akan disimpan ke dalam database, dengan kata sandi dienkripsi menggunakan md5. Terakhir, objek model M\_model dipanggil untuk menyimpan data pengguna baru ke dalam tabel 'user', termasuk pengaturan default untuk foto profil. Setelah proses pendaftaran selesai, pengguna dialihkan ke halaman login untuk masuk ke dalam sistem.

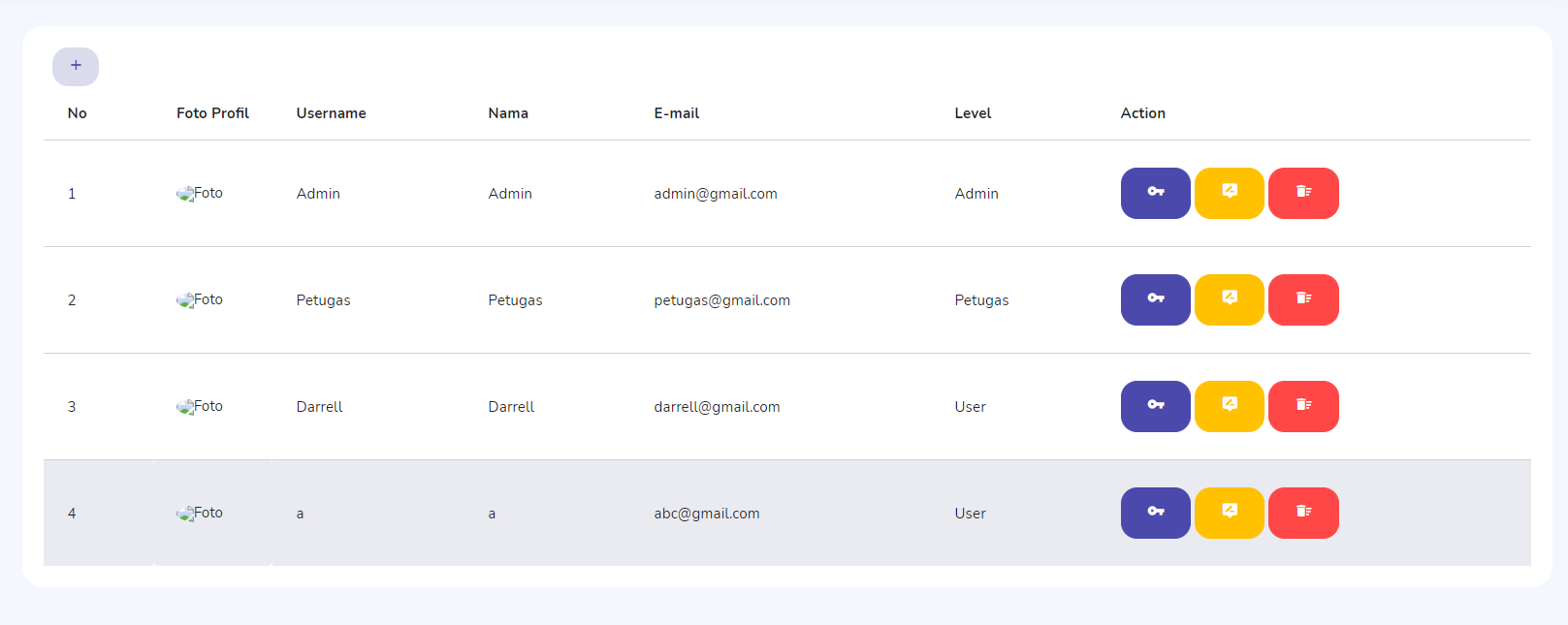
1. Dashboard





Kode PHP di atas adalah bagian dari kontroler "Dashboard" dalam suatu aplikasi web. Kontroler ini memeriksa apakah pengguna telah login berdasarkan sesi 'id'. Jika iya, kontroler membuat objek model, melakukan join antara tabel 'user' dan 'level' berdasarkan kondisi tertentu, dan menyiapkan data untuk tampilan dashboard. Jika pengguna belum login, dia diarahkan ke halaman login. Penggunaan tampilan parsial, seperti 'header', 'menuutama', 'dashboard', dan 'footer', memisahkan komponen-komponen tampilan untuk meningkatkan keterbacaan dan pemeliharaan kode.

1. User

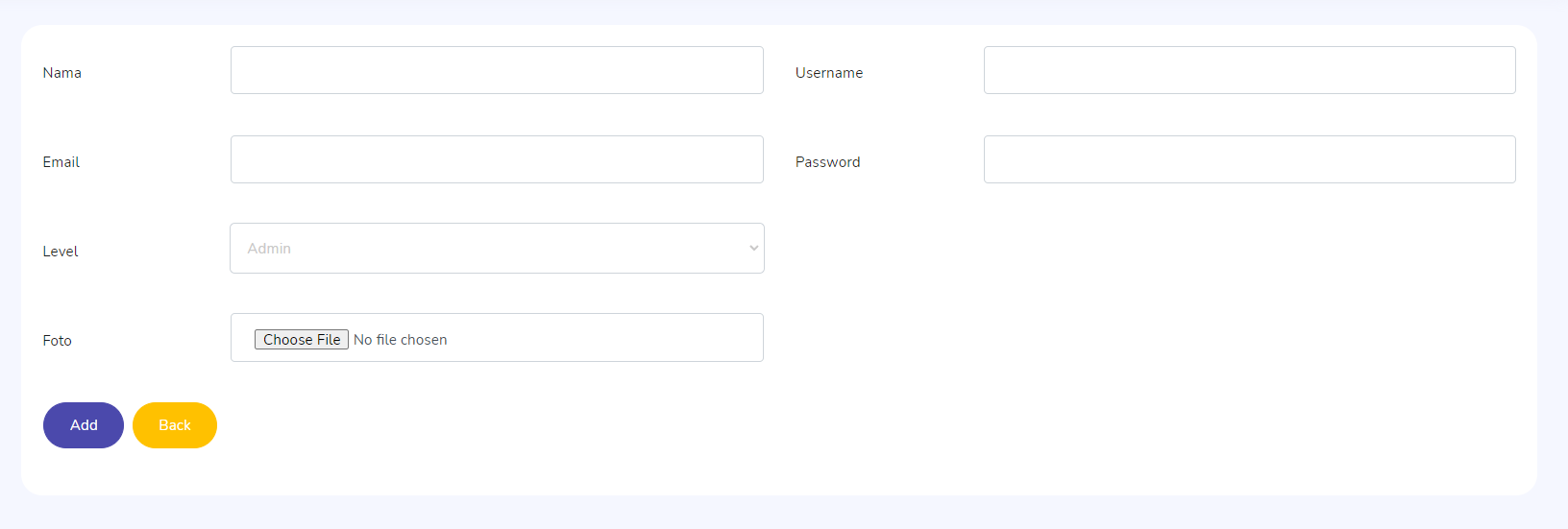


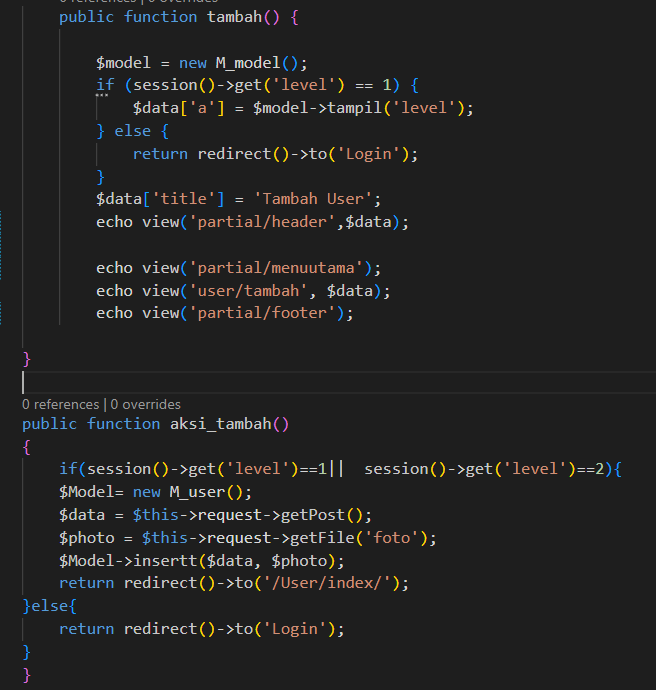


Kode PHP di atas adalah bagian dari kontroler "User" pada aplikasi web menggunakan CodeIgniter atau framework PHP serupa. Kontroler ini mengurus operasi terkait pengguna, seperti menampilkan data pengguna, mereset kata sandi, menambah, mengedit, memperbarui, dan menghapus pengguna.Pada metode `index`, kontroler memeriksa level pengguna yang login. Jika levelnya adalah 1, tampilan data pengguna dimuat dengan melakukan join antara tabel 'user' dan 'level'. Metode `reset\_password` mengizinkan reset kata sandi untuk pengguna level 1. Metode `tambah` menangani tampilan formulir tambah pengguna sesuai level.

Metode `aksi\_tambah` mengelola penambahan pengguna baru dengan memanfaatkan model `M\_user`. Metode `edit`, `update`, dan `delete` digunakan untuk mengedit, memperbarui, dan menghapus data pengguna sesuai tingkatan akses. Keseluruhan, kontroler ini mengimplementasikan operasi CRUD dengan mempertimbangkan level akses pengguna, memberikan keamanan dan fungsionalitas yang sesuai.

1. Tambah User

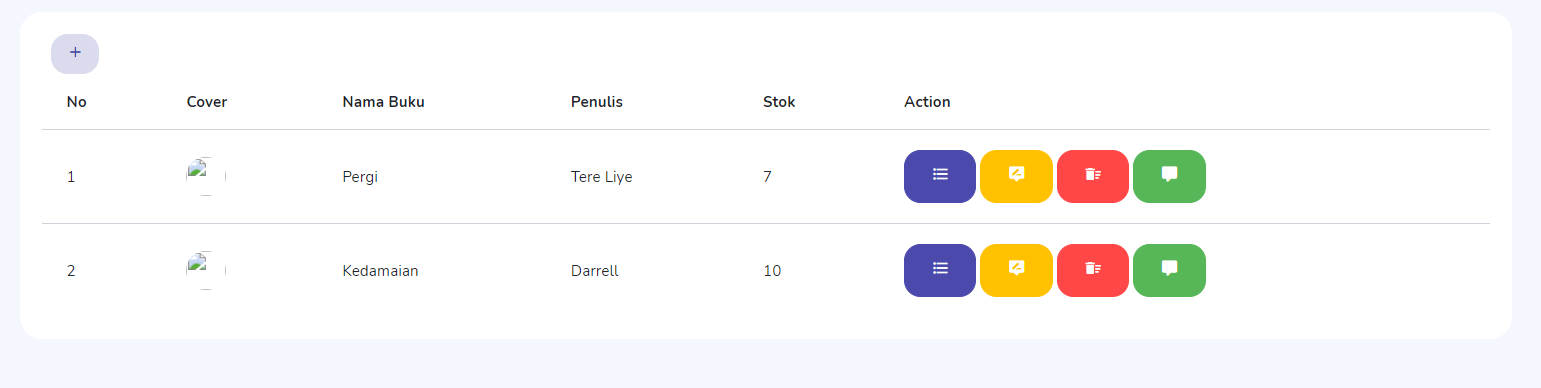


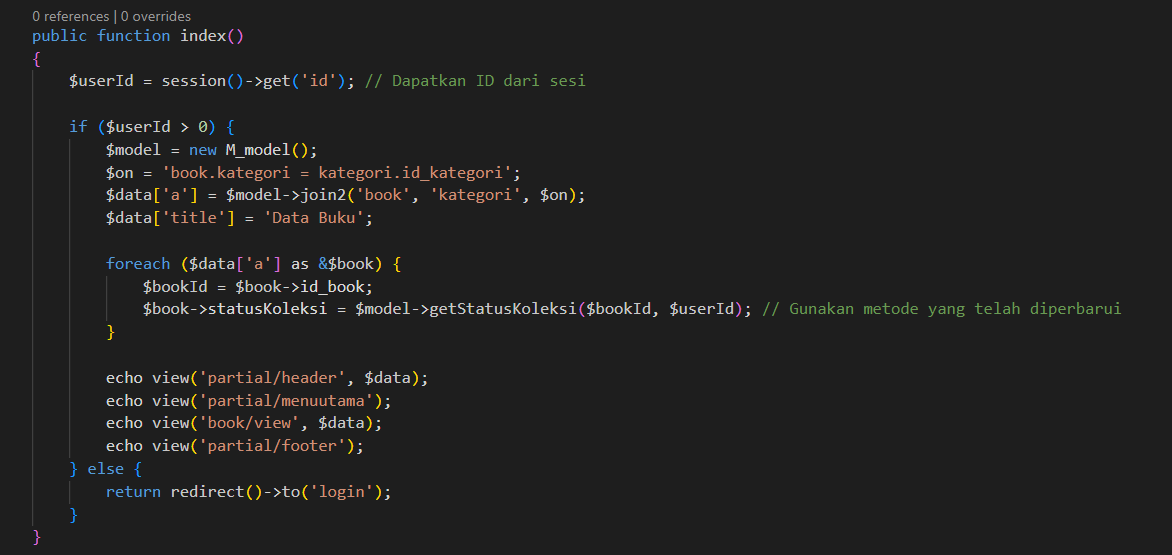


Metode `tambah` pada kontroler "User" bertanggung jawab untuk menampilkan formulir tambah pengguna. Terlebih dahulu, metode ini membuat objek model `M\_model` untuk mendapatkan data tingkat akses ('level'). Jika level pengguna yang sedang login adalah 1, maka data tingkat akses disimpan dalam array 'a', dan selanjutnya, tampilan formulir tambah pengguna dimuat. Jika bukan, pengguna diarahkan ke halaman login. Informasi judul halaman ('title') juga disiapkan untuk digunakan dalam tampilan.

Metode `aksi\_tambah` menangani proses penambahan pengguna baru. Pertama, metode ini memeriksa tingkat akses pengguna yang sedang login; jika levelnya adalah 1 atau 2, maka proses penambahan dilanjutkan. Data pengguna dan foto (jika ada) diambil dari input formulir menggunakan `getPost()` dan `getFile()`. Model `M\_user` kemudian digunakan untuk menyimpan data tersebut. Setelah berhasil, pengguna diarahkan kembali ke halaman tampilan data pengguna. Jika level pengguna tidak sesuai, pengguna diarahkan ke halaman login.Secara singkat, metode `tambah` menampilkan formulir tambah pengguna, dan metode`aksi\_tambah` mengelola proses penambahan pengguna baru dengan mempertimbangkan tingkat akses pengguna yang sedang login.

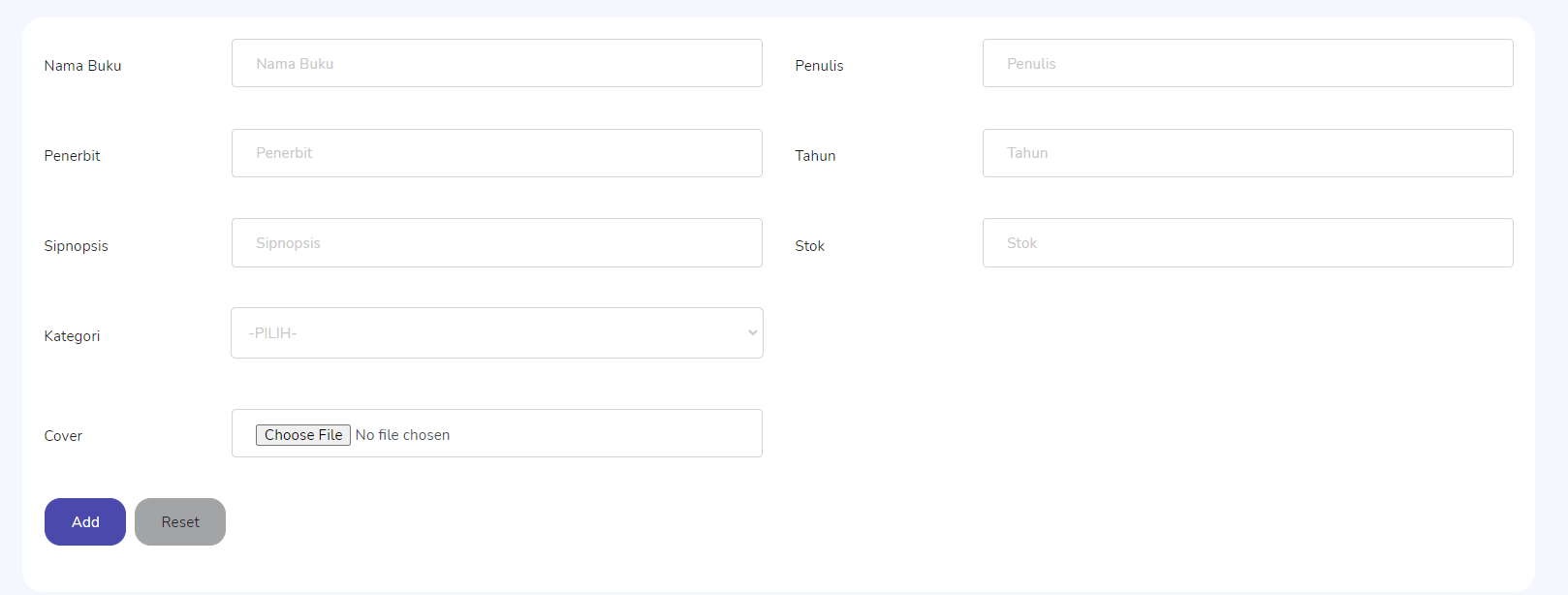
1. Buku





Metode `index` pada kontroler ini bertanggung jawab untuk menampilkan data buku. Jika pengguna telah login, kontroler membuat objek model dan melakukan operasi join antara tabel 'book' dan 'kategori'. Setiap buku mendapatkan status koleksi untuk pengguna yang sedang login melalui metode `getStatusKoleksi`. Hasilnya, data buku bersama dengan status koleksi disiapkan untuk tampilan. Selanjutnya, tampilan data buku dimuat menggunakan tampilan parsial seperti 'header', 'menuutama', 'view', dan 'footer'. Namun, jika pengguna tidak login, dia diarahkan kembali ke halaman login. Metode ini dirancang untuk memberikan informasi tentang status koleksi buku kepada pengguna yang telah login.

1. Tambah Buku

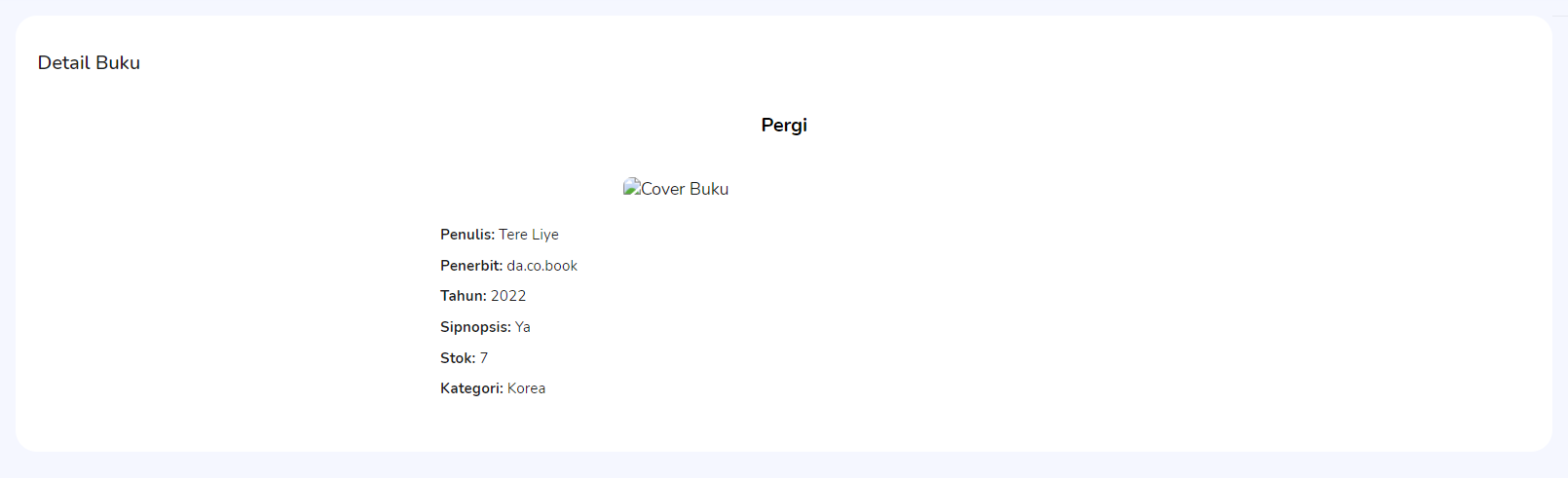


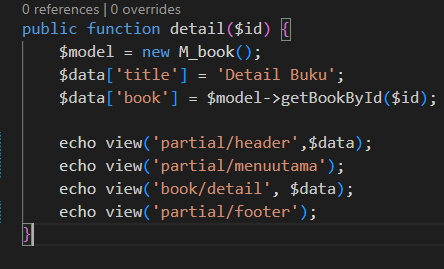


Metode `tambah` pada kontroler "Book" bertujuan untuk menampilkan formulir tambah buku. Dalam proses ini, objek model `M\_model` dibuat untuk mengambil data kategori buku. Selanjutnya, metode melakukan pemeriksaan tingkat akses pengguna, dan jika tingkat akses adalah 1 atau 2, data kategori disimpan dalam array 'a'. Formulir tambah buku kemudian dimuat, sedangkan jika tingkat akses tidak sesuai, pengguna diarahkan kembali ke halaman login. Judul halaman ('title') juga dipersiapkan untuk digunakan dalam tampilan.Di sisi lain, metode `aksi\_tambah` mengelola proses penambahan buku baru. Pertama, tingkat akses pengguna diperiksa; jika tingkat akses adalah 1 atau 2, data buku dan file cover diambil dari formulir. Objek model `M\_book` digunakan untuk menyimpan data tersebut. Setelah berhasil, pengguna diarahkan kembali ke halaman tampilan data buku. Namun, jika tingkat akses pengguna tidak sesuai, pengguna diarahkan kembali ke halaman login.

Dengan demikian, metode `tambah` dan `aksi\_tambah` dalam kontroler ini membentuk alur tambah buku yang memperhitungkan tingkat akses pengguna untuk memastikan keamanan dan fungsionalitas yang sesuai.

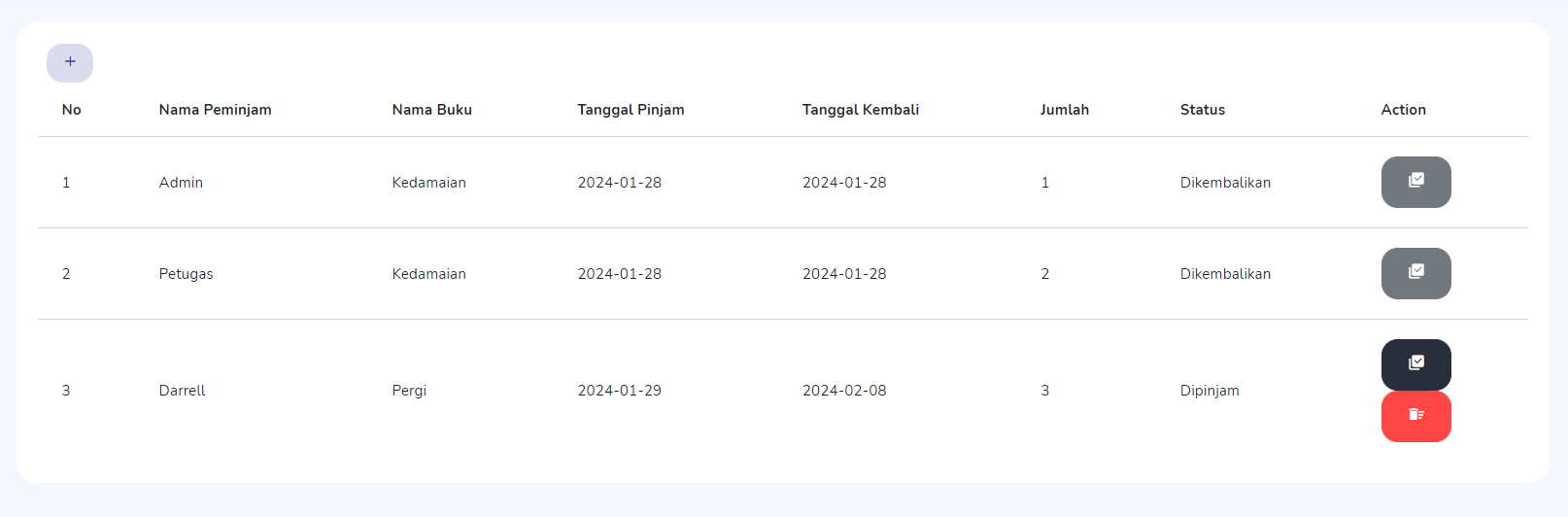
1. Detail Buku

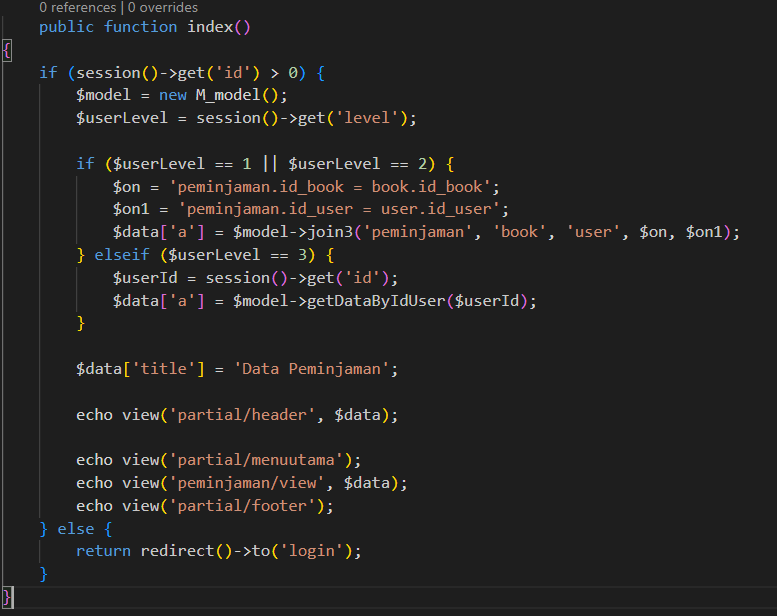




Metode `detail` pada kontroler "Book" berfungsi untuk menampilkan halaman detail buku berdasarkan ID buku yang diterima sebagai parameter. Dalam metode ini, objek model `M\_book` dibuat untuk mengakses data buku. Judul halaman ('title') diatur sebagai 'Detail Buku', dan data buku diperoleh melalui metode `getBookById` dengan menggunakan ID buku yang diterima. Selanjutnya, tampilan detail buku dimuat dengan menggunakan tampilan parsial seperti 'header', 'menuutama', 'detail', dan 'footer'. Informasi buku tersebut disajikan dalam tampilan, termasuk informasi yang diambil dari model menggunakan metode `getBookById`. Metode ini memberikan kemampuan untuk menampilkan detail lengkap dari buku berdasarkan ID-nya.Dengan demikian, metode `detail` dalam kontroler ini memungkinkan pengguna untuk melihat informasi lengkap mengenai suatu buku berdasarkan ID yang disediakan.

1. Data Peminjaman



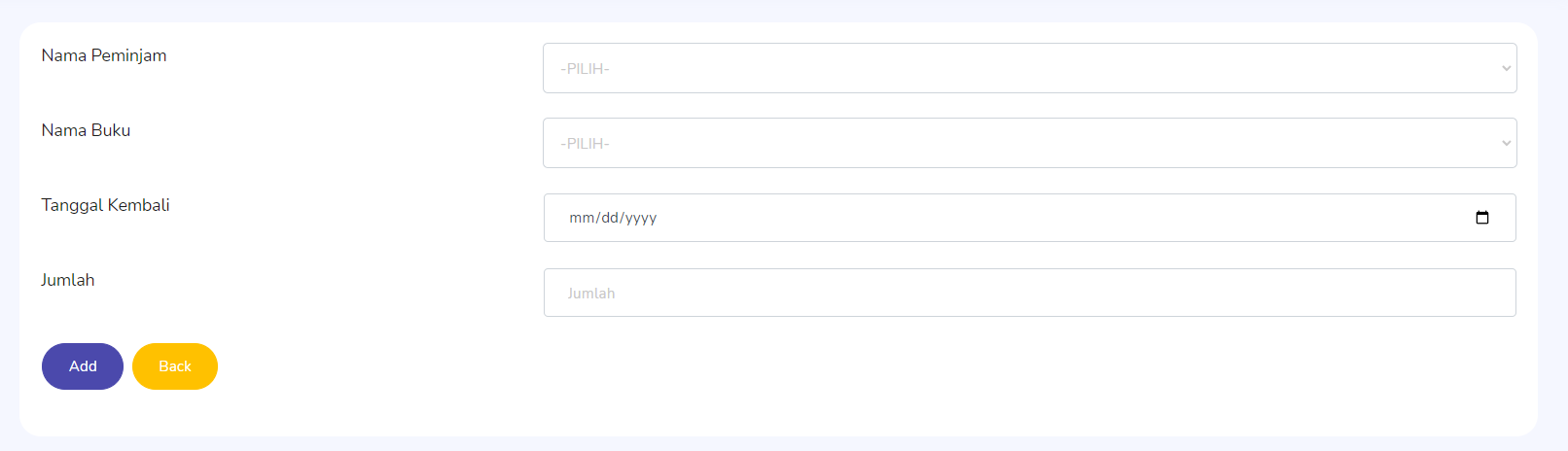


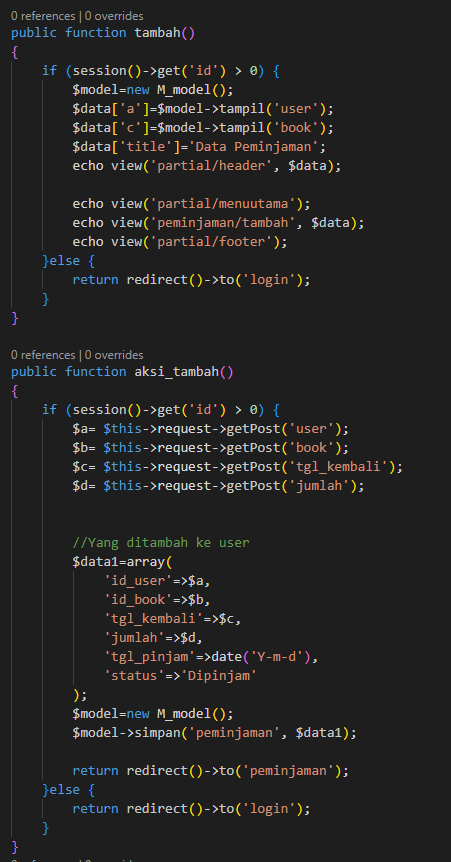
Metode `index` pada kontroler ini mengelola tampilan data peminjaman. Pertama, dilakukan pemeriksaan terhadap sesi ID pengguna; jika ID-nya lebih dari 0, menandakan bahwa pengguna telah login. Selanjutnya, objek model `M\_model` dibuat, dan level akses pengguna diambil dari sesi.

Jika level akses pengguna adalah 1 atau 2, kontroler melakukan operasi join antara tabel 'peminjaman', 'book', dan 'user' berdasarkan beberapa kondisi (ON clause). Data hasil join disimpan dalam array 'a' menggunakan metode `join3` pada model. Sedangkan jika level akses adalah 3, data peminjaman hanya diambil berdasarkan ID pengguna yang sedang login dengan menggunakan metode `getDataByIdUser`. Judul halaman ('title') disiapkan untuk tampilan, dan selanjutnya, tampilan data peminjaman dimuat menggunakan tampilan parsial seperti 'header', 'menuutama', 'view', dan 'footer'. Jika pengguna belum login, dia diarahkan kembali ke halaman login.

Secara keseluruhan, metode ini dirancang untuk menampilkan data peminjaman sesuai dengan level akses pengguna yang sedang login, baik itu level 1, 2, atau 3.

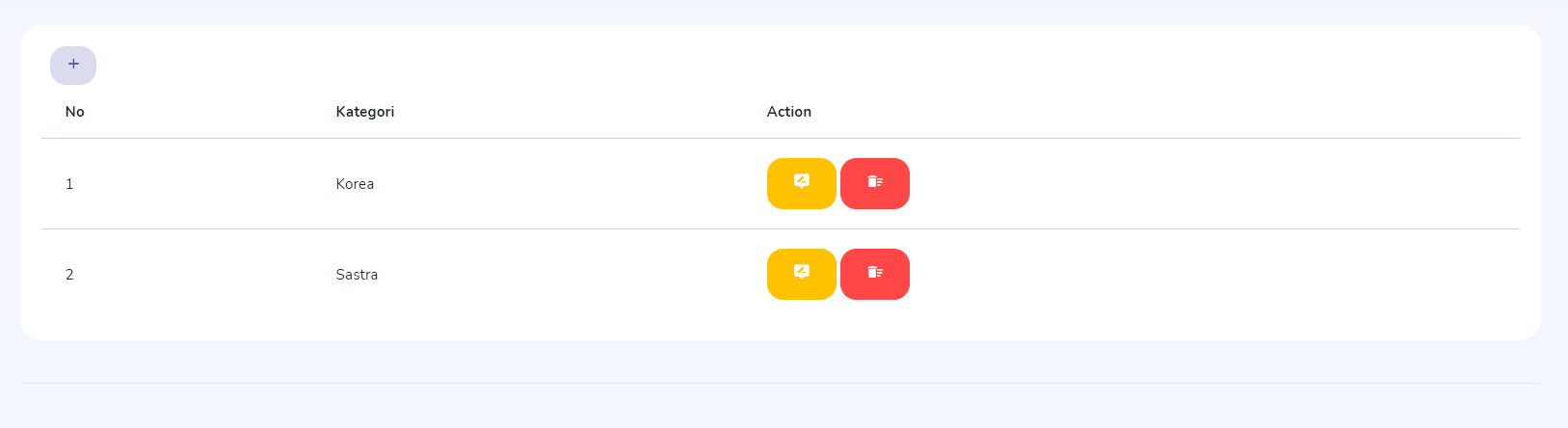
1. Tambah Peminjaman

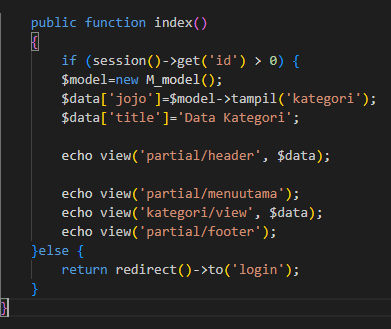




Metode `tambah` pada kontroler ini menampilkan formulir tambah peminjaman, memastikan bahwa pengguna telah login sebelumnya. Objek model `M\_model` digunakan untuk mengambil data pengguna dan data buku. Informasi tersebut disiapkan sebagai variabel 'a' dan 'c' dalam array 'data', bersama dengan judul halaman ('title'). Selanjutnya, formulir tambah peminjaman dimuat menggunakan tampilan parsial.Sementara itu, metode `aksi\_tambah` mengelola penambahan data peminjaman. Setelah memastikan bahwa pengguna telah login, data dari formulir, seperti ID pengguna, ID buku, tanggal kembali, dan jumlah, diambil dan disiapkan untuk disimpan dalam tabel 'peminjaman'. Objek model digunakan untuk menyimpan data, dan pengguna diarahkan kembali ke halaman data peminjaman setelah penambahan berhasil. Dengan langkah-langkah ini, kontroler ini memastikan bahwa hanya pengguna yang telah login yang dapat menambahkan data peminjaman, dan informasi peminjaman disimpan dengan sukses ke dalam sistem.

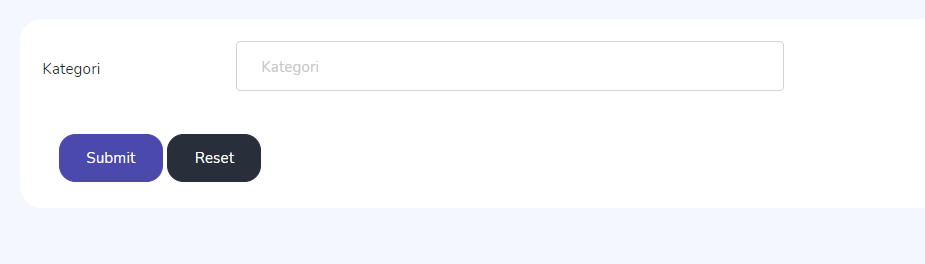
1. Kategori

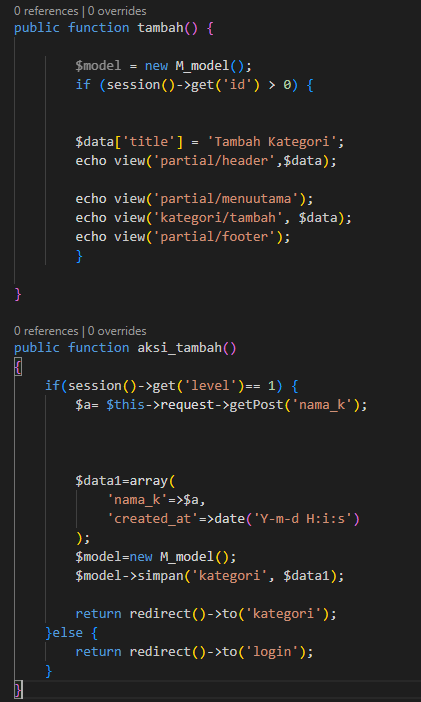




Metode `index` pada kontroler ini menangani tampilan data kategori. Pertama, dilakukan pemeriksaan terhadap sesi ID pengguna; jika ID-nya lebih dari 0, menandakan bahwa pengguna telah login. Jika kondisi tersebut terpenuhi, objek model `M\_model` dibuat, dan data kategori diambil melalui metode `tampil`. Data tersebut kemudian disiapkan dalam array 'jojo' bersama dengan judul halaman ('title').Selanjutnya, tampilan data kategori dimuat menggunakan tampilan parsial seperti 'header', 'menuutama', 'view', dan 'footer'. Jika pengguna belum login, mereka diarahkan kembali ke halaman login. Dengan langkah-langkah ini, kontroler memastikan bahwa hanya pengguna yang telah login yang dapat mengakses dan melihat data kategori.

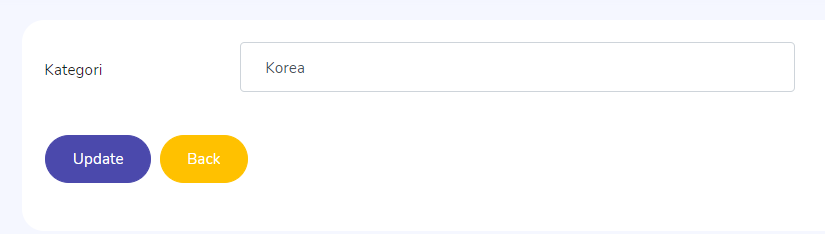
1. Tambah Kategori

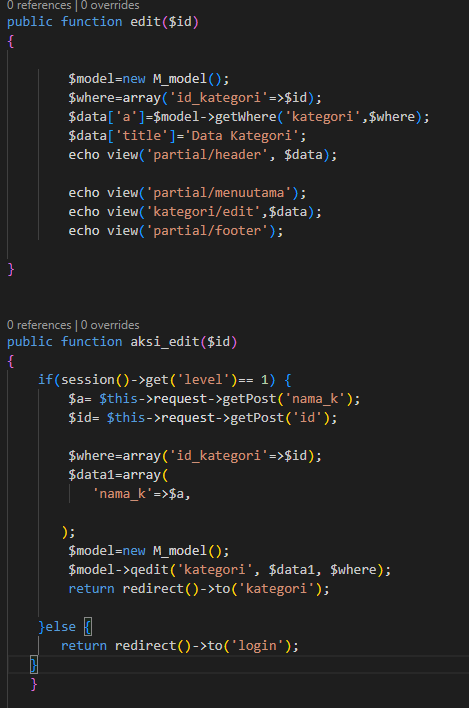




Metode `aksi\_tambah` pada kontroler ini mengelola proses penambahan kategori baru. Pertama, dilakukan pemeriksaan tingkat akses pengguna; jika tingkat akses adalah 1, menandakan bahwa pengguna memiliki hak untuk menambahkan kategori. Data nama kategori ('nama\_k') diambil dari formulir menggunakan `$this->request->getPost()` dan disiapkan dalam variabel 'a'. Selanjutnya, data kategori disusun dalam array 'data1' dengan mencakup nama kategori dan informasi waktu dibuat ('created\_at') yang diperoleh menggunakan `date('Y-m-d H:i:s')`. Objek model `M\_model` kemudian digunakan untuk menyimpan data kategori baru ke dalam tabel 'kategori'.Jika proses penambahan berhasil, pengguna diarahkan kembali ke halaman data kategori. Jika tingkat akses tidak sesuai, pengguna diarahkan kembali ke halaman login.Dengan langkah-langkah ini, kontroler ini memastikan bahwa hanya pengguna dengan tingkat akses 1 yang dapat menambahkan kategori baru, dan informasi kategori disimpan dengan sukses dalam sistem.

1. Edit Kategori

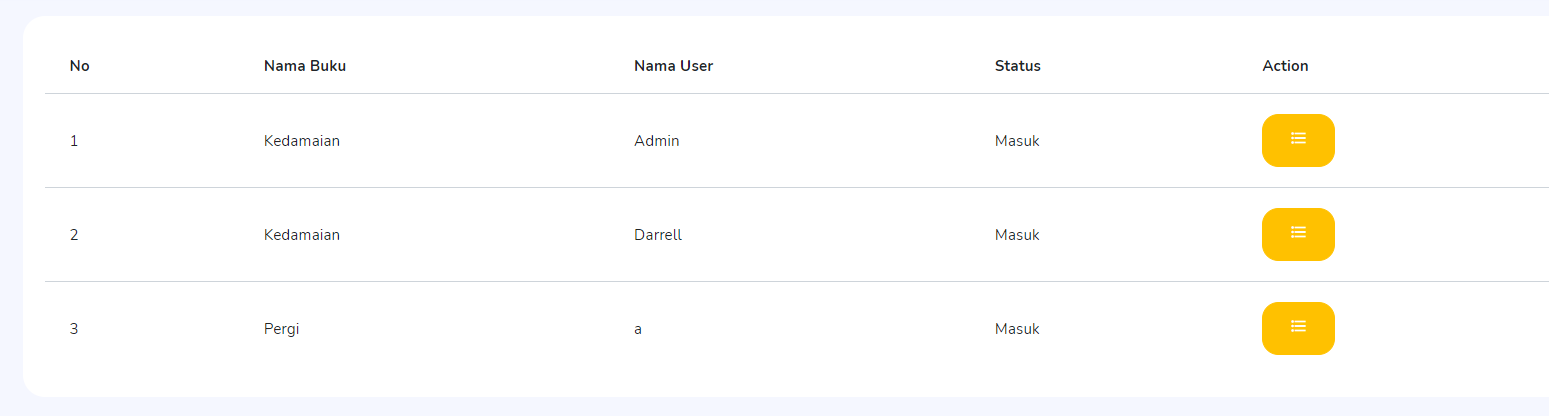


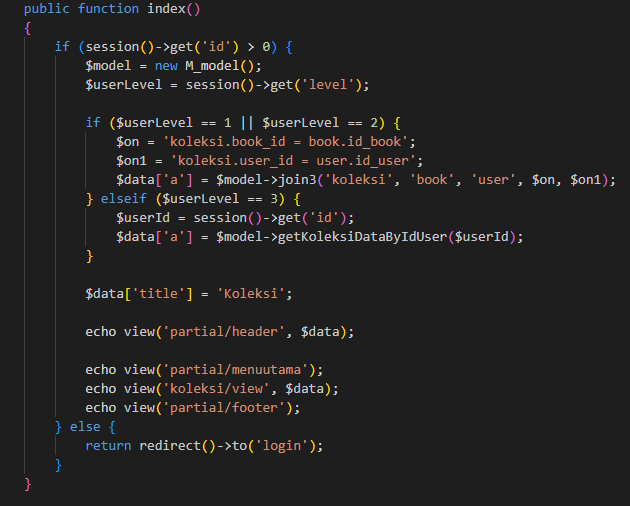


Metode `edit` pada kontroler ini menangani tampilan formulir pengeditan kategori berdasarkan ID kategori yang diterima sebagai parameter. Objek model `M\_model` dibuat, dan data kategori yang sesuai dengan ID tersebut diambil menggunakan metode `getWhere`. Informasi tersebut kemudian disiapkan dalam array 'a', bersama dengan judul halaman ('title'). Selanjutnya, formulir edit kategori dimuat menggunakan tampilan parsial.

Metode `aksi\_edit` mengelola proses pengeditan kategori. Pemeriksaan tingkat akses dilakukan; jika tingkat akses adalah 1, pengguna diizinkan untuk mengedit kategori. Data nama kategori ('nama\_k') diambil dari formulir, bersama dengan ID kategori ('id'). Array 'data1' disusun dengan informasi yang akan diperbarui. Objek model `M\_model` menggunakan metode `qedit` untuk melakukan perubahan pada tabel 'kategori' berdasarkan ID.Jika proses pengeditan berhasil, pengguna diarahkan kembali ke halaman data kategori. Jika tingkat akses tidak sesuai, pengguna diarahkan kembali ke halaman login.Dengan langkah-langkah ini, kontroler ini memastikan bahwa hanya pengguna dengan tingkat akses 1 yang dapat mengedit kategori, dan perubahan pada informasi kategori diterapkan ke dalam sistem.

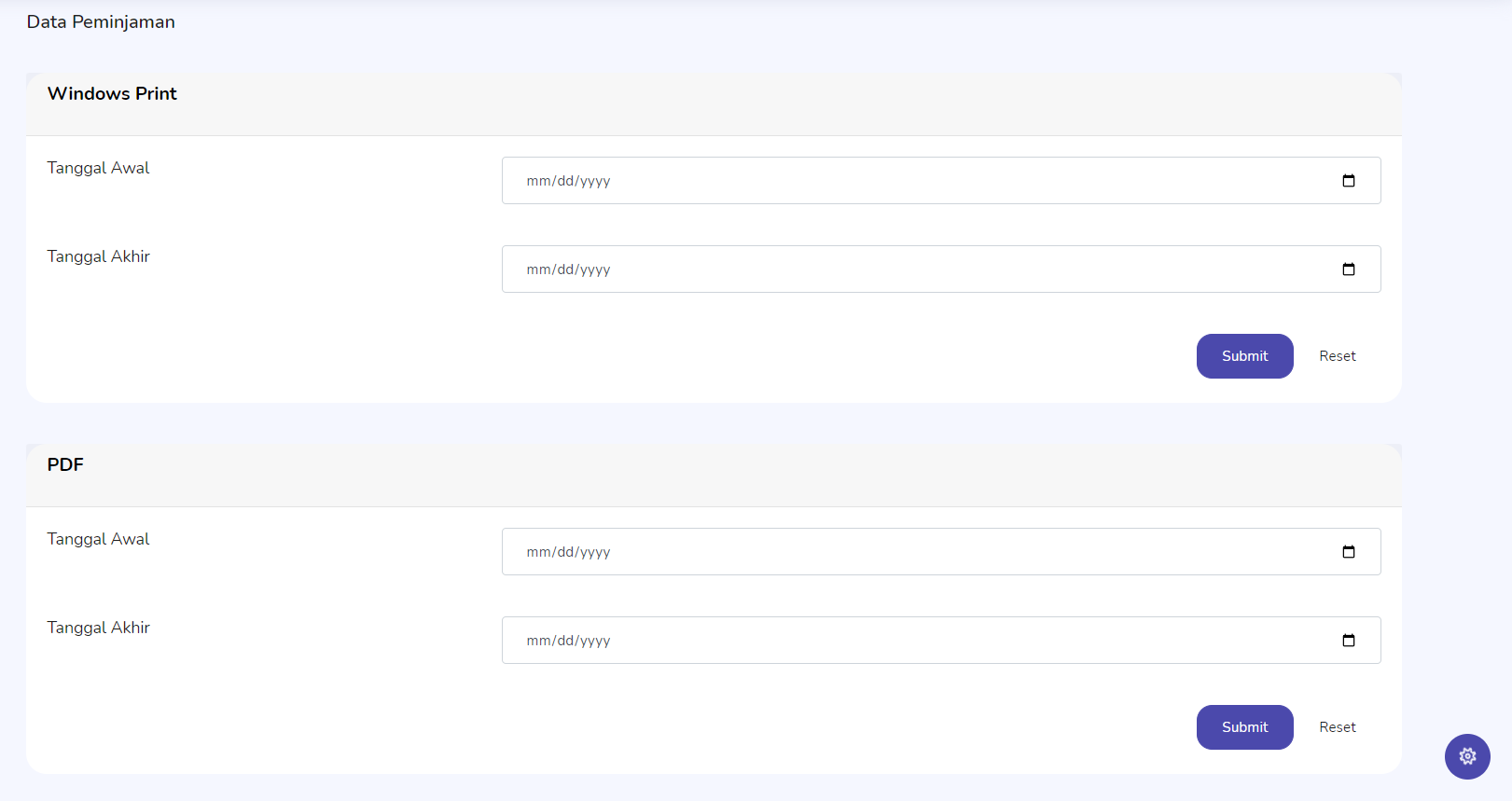
1. Koleksi

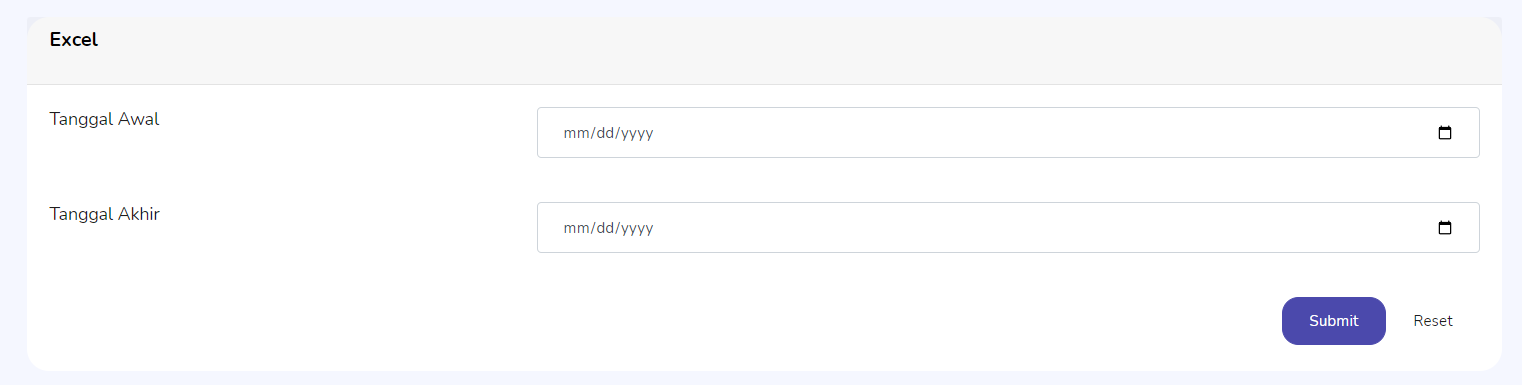


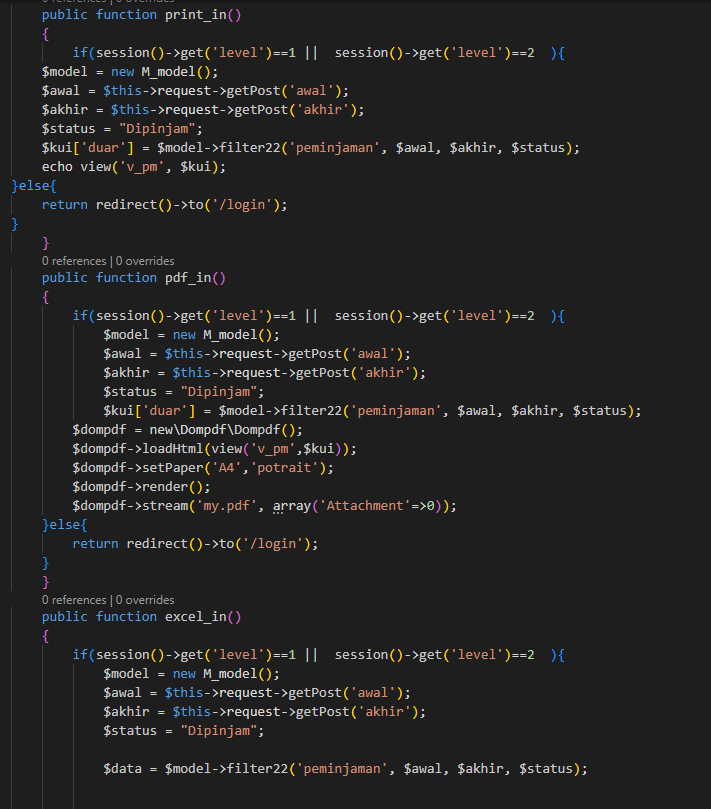


Fungsi `index` pada kontroler ini mengelola tampilan halaman koleksi untuk pengguna yang telah login. Jika sesi aktif, kontroler memeriksa level pengguna. Jika level 1 atau 2, dilakukan join tabel 'koleksi', 'book', dan 'user'. Jika level 3, data koleksi diambil berdasarkan ID pengguna. Judul halaman ditetapkan, dan tiga view dipanggil. Jika tidak ada sesi aktif, pengguna diarahkan ke halaman login.

1. Laporan Peminjaman



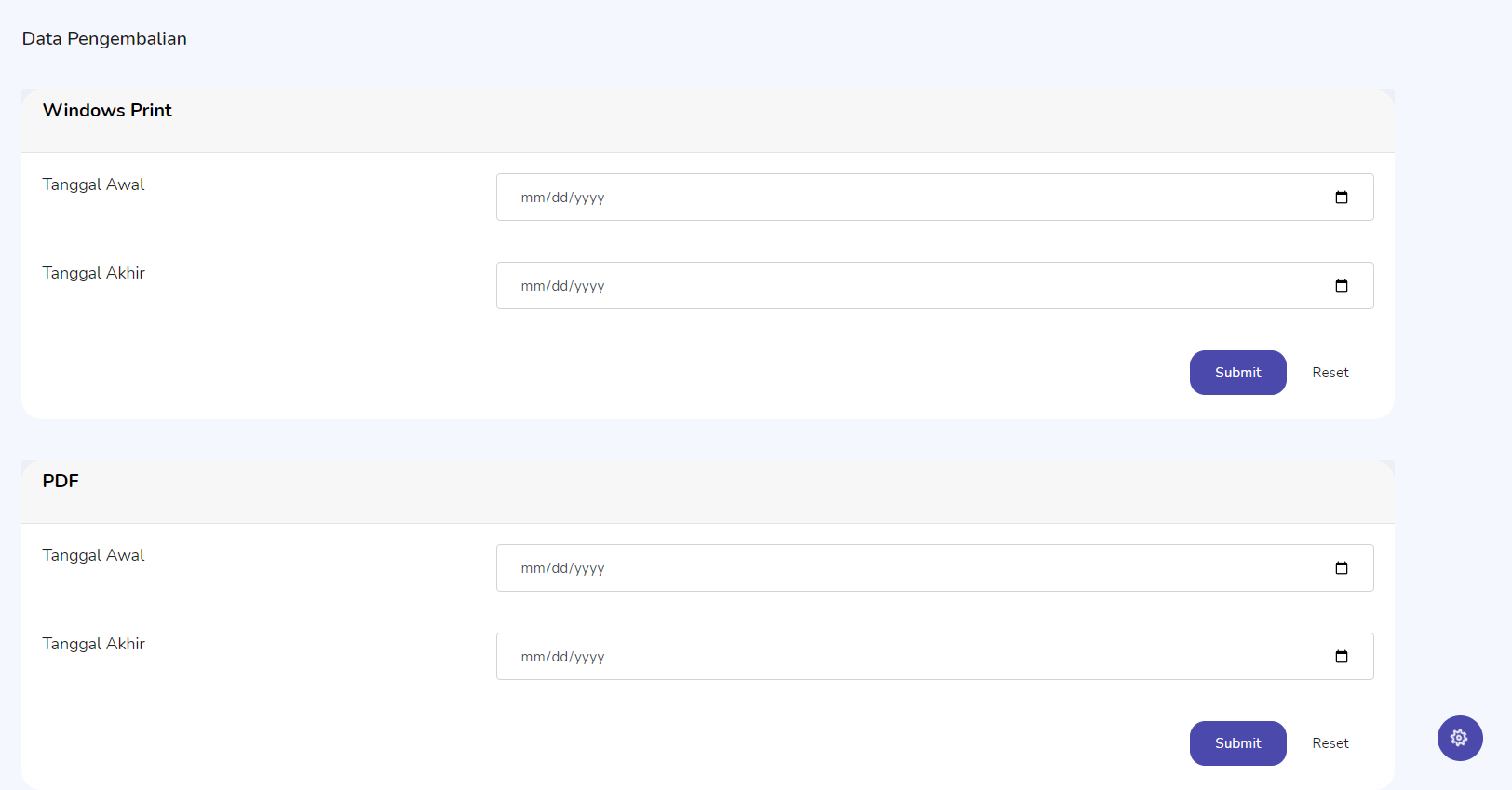


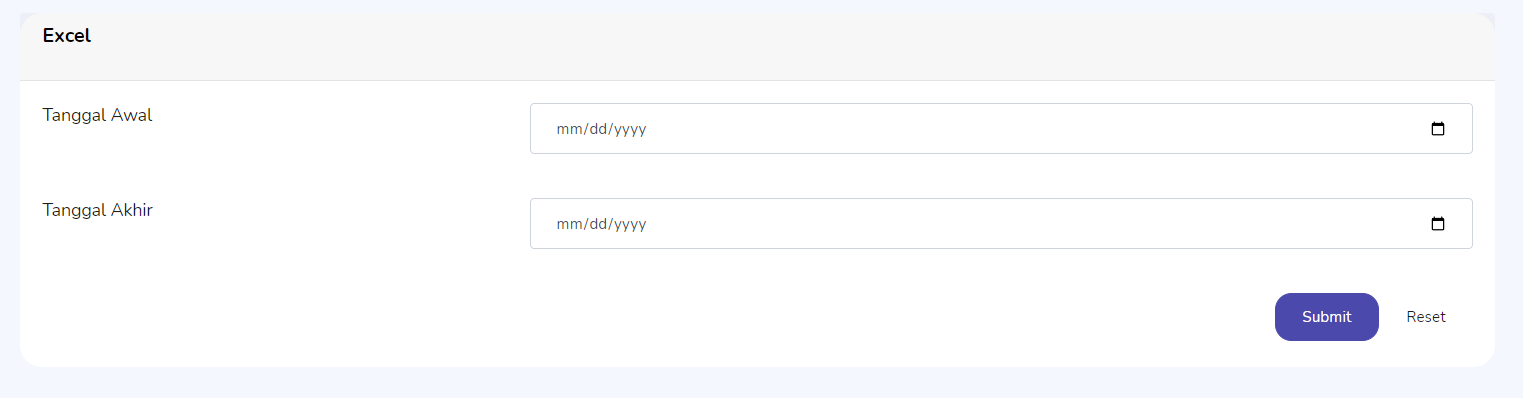


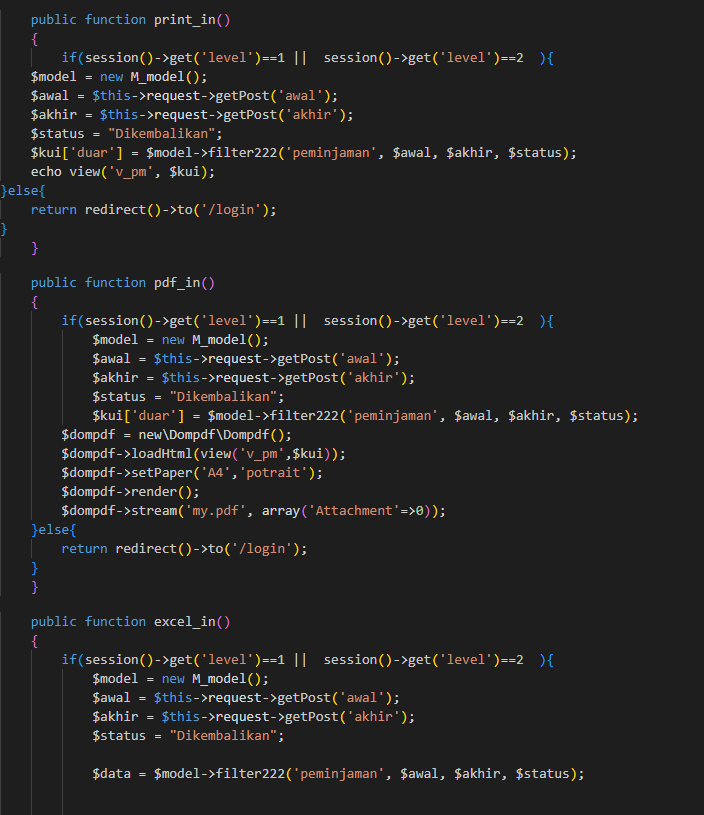
Kode PHP di atas adalah bagian dari sebuah controller dalam framework PHP CodeIgniter yang digunakan untuk mengelola laporan peminjaman. Controller ini mencakup beberapa metode, seperti `index`, `print\_in`, `pdf\_in`, dan `excel\_in`. Controller ini terhubung dengan model `M\_model` untuk mengambil data yang diperlukan.Metode `index` bertanggung jawab untuk menampilkan halaman utama laporan peminjaman. Jika level pengguna adalah 1 atau 2 (admin atau pengguna tertentu), maka halaman dengan filter peminjaman akan ditampilkan. Jika tidak, pengguna akan diarahkan kembali ke halaman login.

Metode `print\_in` dan `pdf\_in` digunakan untuk menghasilkan laporan peminjaman dalam format cetak dan PDF. Kedua metode ini memvalidasi level pengguna sebelum memproses permintaan. Data laporan diambil dari model berdasarkan rentang tanggal dan status "Dipinjam". Metode `pdf\_in` menggunakan library Dompdf untuk menghasilkan file PDF dari tampilan laporan. Metode `excel\_in` bertugas membuat laporan peminjaman dalam format Excel. Seperti metode sebelumnya, ini memeriksa level pengguna sebelum melanjutkan. Data diambil dari model dan disusun ke dalam spreadsheet menggunakan library PhpSpreadsheet. Setelah itu, spreadsheet disimpan sebagai file Excel dan dikirim ke pengguna sebagai respons dengan menggunakan header PHP.Kode ini menunjukkan implementasi dasar dari operasi CRUD untuk laporan peminjaman dalam sebuah aplikasi berbasis web dengan menggunakan PHP dan CodeIgniter.

1. Laporan Pengembalian







Fungsi print\_in() digunakan untuk mengambil data filter pengembalian dari model dan menampilkannya pada view bernama 'v\_pm'. Data ini kemudian dapat dicetak.Fungsi pdf\_in() menggunakan pustaka Dompdf untuk membuat file PDF dari view 'v\_pm'. Data yang diambil dan ditampilkan pada PDF juga merupakan hasil dari filter pengembalian.Fungsi excel\_in() membuat file Excel dengan menggunakan PhpSpreadsheet. Data hasil filter pengembalian ditampilkan dalam format tabel pada file Excel, yang kemudian dapat diunduh oleh pengguna.Penting untuk dicatat bahwa terdapat pengujian level akses sebelum menjalankan fungsi-fungsi ini, sehingga hanya pengguna dengan level akses tertentu yang dapat mengakses dan menggunakan fitur laporan pengembalian. Jika pengguna tidak memiliki level akses yang sesuai, mereka akan diarahkan kembali ke halaman login.