# MODUL PRAKTIKUM WEB SEMESTER 6 PERTEMUAN 1 DATABASE, INSTALASI CODEIGNITER DAN PENGATURAN TEMPLATE

Alat dan Bahan:

1. Aplikasi Web Server (Xampp / Laragon)
2. CodeIgniter versi 3.1 (<https://codeload.github.com/bcit-ci/CodeIgniter/zip/3.1.13>)
3. Template Concept-Master (<https://github.com/puikinsh/concept/archive/refs/heads/master.zip>)
4. Text Editor (Sublime Text / Visual Studio Code)

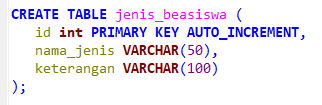
Tujuan:

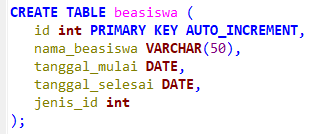
1. Instalasi Framework CodeIgniter Versi 3.1 untuk pengembangan Web
2. Menerapkan template concept-master untuk tampilan web

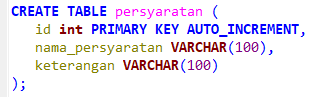
## Database

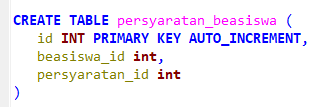
Tahapan pertama yang akan kita lakukan dalam pembuatan Sistem Informasi Manajemen Data Beasiswa (SIMDAWA) ini adalah membuat basis datanya terlebih dahulu beserta tabel-tabel yang diperlukan. Buat database baru pada MySQL kalian dengan nama: **beasiswa-app.**

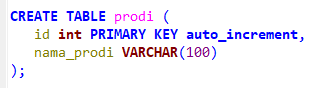
Kemudian buat beberapa tabel seperti pada gambar berikut:

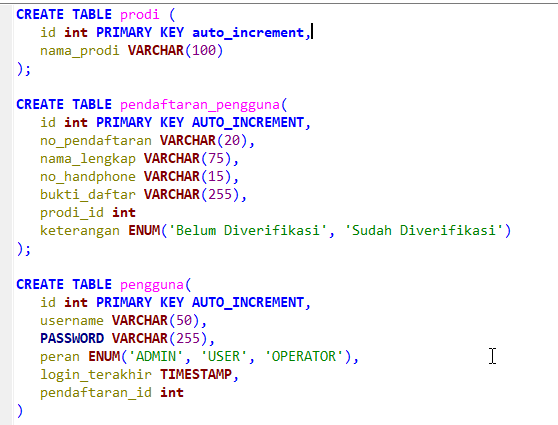












A picture containing graphical user interface

Description automatically generated

Text

Description automatically generated

## Buat Projek dan Install Framework CodeIgniter

Setelah database beasiswa\_app beserta tabel-tabelnya sudah dibuat, selanjutnya buat folder projek di lokasi web server kalian. Apabila menggunakan **laragon** di folder **www** sedangkan Xampp di folder **htdocs.**

1. Buat folder projek dengan nama **simdawa-app.**
2. Kemudian pindahkan file framework CodeIgniter yang sudah didownload ke dalam folder projek simdawa-app.
3. Lalu ekstrak file framework CodeIgniter tersebut sehingga di dalam folder simdawa-app menjadi seperti berikut:

A screenshot of a computer

Description automatically generated with medium confidence

1. Kalian bisa hapus saja folder user\_guide dan file CodeIgniter
2. Proses instalasi CodeIgniter sudah selesai, kalian bisa coba jalankan projek simdawa-app ini. Pastikan saat dijalankan tampil seperti berikut:

Graphical user interface, text, application, email

Description automatically generated

## Konfigurasi Framework CodeIgniter

Walaupun cukup dengan konfigurasi default codeigniter sudah dapat berjalan, tetapi untuk pembuatan aplikasi kita perlu melakukan beberapa konfigurasi agar dalam pengembangan bisa menjadi lebih mudah. Setidaknya konfigurasi yang perlu dilakukan ada pada file **autoload.php, config.php, database.php**, dan **route.php.** Keempat file ini ada di folder **application/config.**

* Autoload.php: konfigurasi ini untuk menentukan sumber daya apa yang akan dijalankan otomatis.
* Config.php: konfigurasi ini digunakan untuk mengatur alamat url aplikasi (base\_url).
* Database.php: konfigurasi ini digunakan untuk mengatur koneksi ke database, ada beberapa yang bisa kita atur di sini seperti hostname, username, password dan database yang mau digunakan.
* Route.php: konfigurasi ini untuk menentukan default controller awal.

1. **File Autoload.php**

File pertama yang akan kita atur adalah file **autoload.php.** Buka file **autoload**.**php**, kemudian cari bagian **libraries.** Lalu tambahkan **database** dan **session.** Setelah libraries, cari lagi bagian **helper** dan tambahkan **url.**

**Text

Description automatically generated**

**Text

Description automatically generated**

1. **File config.php**

Buka file config.php lalu pada bagian **base\_url** kalian isi sesuai dengan alamat projek yang dibuat yaitu: <http://localhost/simdawa-app>. Lalu pada bagian index kalian kosongkan saja isinya.

Text

Description automatically generatedText

Description automatically generated

1. **File database.php**

Konfigurasi berikutnya adalah mengatur koneksi ke database. Buka file database.php di folder config, lalu atur username, password, database sesuai database yang akan digunakan.

A screenshot of a computer

Description automatically generated with medium confidence

1. **File routes.php**

Buka file routes.php di folder config, kemudian atur pada bagian default\_controller diisi dengan home. Ini bertujuan, saat kita jalankan projek <http://localhost/simdawa-app/> maka controller yang dijalankan adalah home. Untuk sementara controller home belum kita buat. Akan kita buat pada step berikutnya.

Text

Description automatically generated

1. File .htaccess

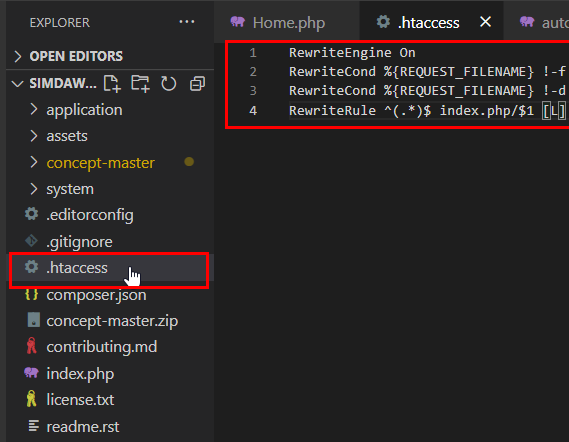
Kemudian kita juga perlu menambahkan 1 file yang mengatur agar supaya url projek menjadi lebih simple dengan menghilangkan index.php pada url. Yang awalnya <http://localhost/simdawa-app/index.php/home> akan kita ubah agar menjadi <http://localhost/simdawa-app/home> tanpa ada tambahan index.php. Untuk melakukan tersebut tambahkan 1 file di folder **simdawa-app** dengan nama file .htaccess kemudian ketik source code berikut:

RewriteEngine On

RewriteCond %{REQUEST\_FILENAME} !-f

RewriteCond %{REQUEST\_FILENAME} !-d

RewriteRule ^(.\*)$ index.php/$1 [L]



## Menggunakan Template concept-master

Pastikan kalian sudah mendownload template yang akan kita gunakan pada projek simdawa-app ini yaitu **concept-master.** Kemudian kalian letakkan file rar dari template tersebut ke dalam projek simdawa-app. Lalu klik kanan dan pilih Extract Here. Sehingga di dalam folder simdawa-app sudah bertambah 1 folder concept-master.

A screenshot of a computer

Description automatically generated with medium confidence

Masuk ke dalam folder concept-master, lalu kalian salin folder assest dan letakkan di dalam folder simdawa-app. Sehingga menjadi seperti berikut:

A screenshot of a computer

Description automatically generated with medium confidence

Sampai step ini, template concept-master sudah siap untuk kita gunakan di projek **simdawa-app.**

## Membuat Template

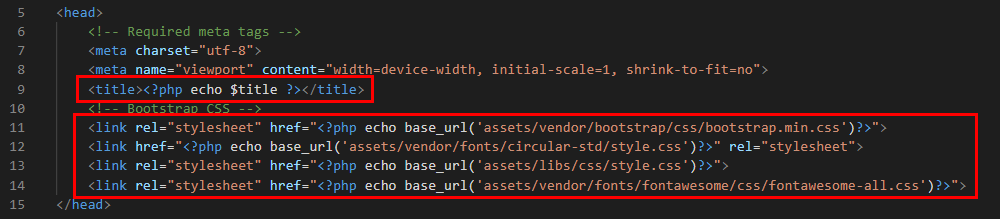
Buka text editor kalian (dalam modul ini akan digunakan text editor Visual Studio Code). Kemudian buka folder simdawa-app pada Visual Studio Code. Kemudian buat 1 buah folder di folder **application/views** dengan nama folder **template.**

**A screenshot of a computer

Description automatically generated with low confidence**

1. **Header.php**

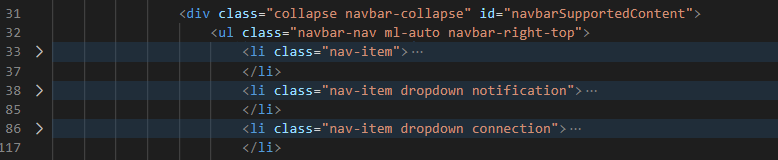
Pada folder **template,** buat 1 file dengan nama **header.php.** Untuk source codenya, kita akan salin dari template. Silahkan kalian cari file **blank-page-header.html** yang ada di folder template **concept-master**. Kemudian kalian salin source code dari baris 1 – 134 di file tersebut dan paste di file header.php. Setelah kalian salin, langkah pertama perbaiki bagian **<head>** agar pemanggilan file css nya bisa berjalan.



Pada bagian head di atas, ada variable $title yang nanti nilainya akan kita kirimkan melalui controller. Dan ada pemanggilan **base\_url().** Base\_url() ini sebelumnya sudah kita sesuaikan dengan alamat projek kita <http://localhost/simdawa-app/>. Dan base\_url() ini bisa kita panggil apabila sudah diaktifkan **url helper** di file autoload bagian helper.

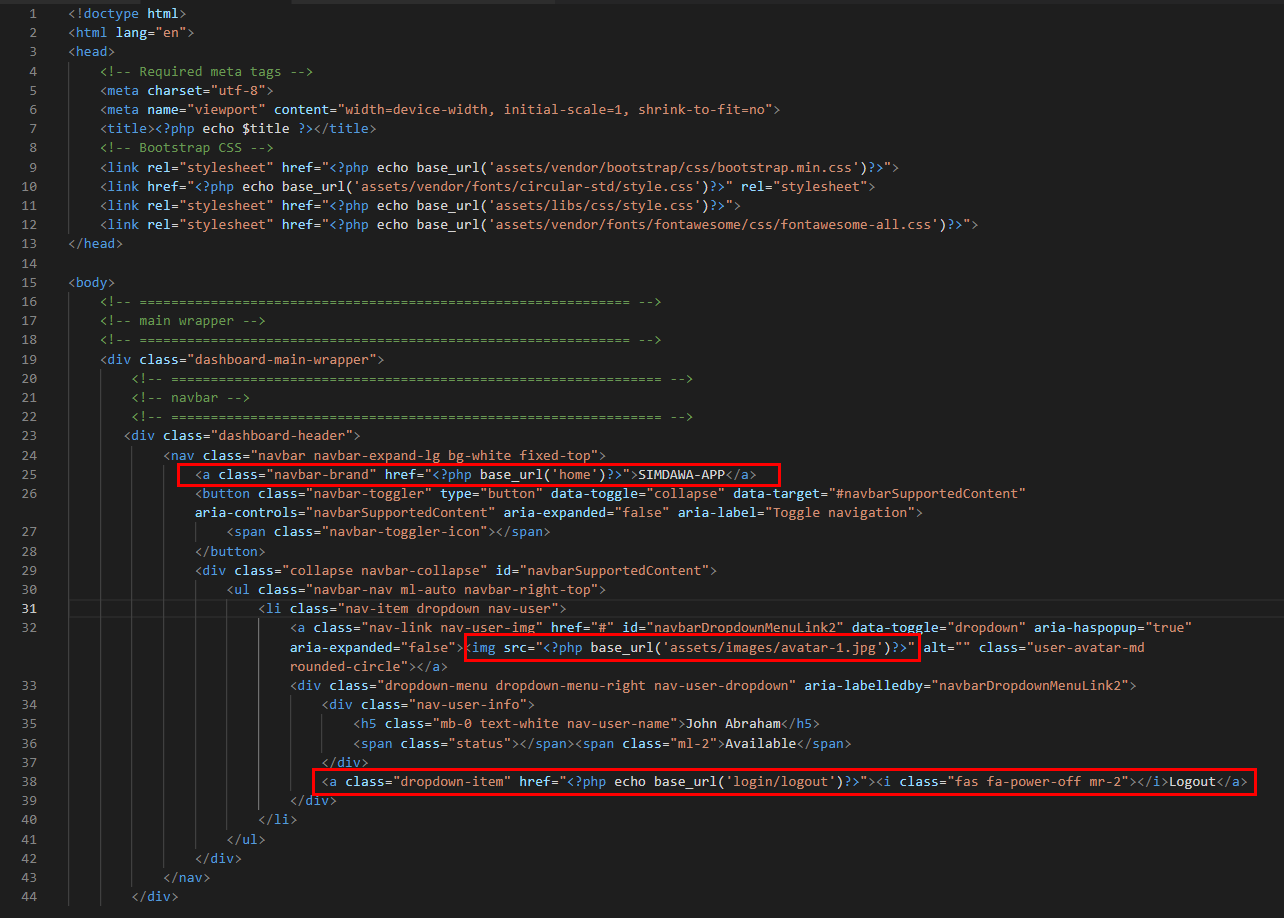
Setelah bagian head kalian perbaiki, selanjutnya kita akan hapus beberapa tag <li> pada header.php karena ada beberapa fungsi yang tidak kita gunakan dari bawaan template. Untuk menghapus tag <li> ini pastikan kalian menghapusnya dari tag pembuka <li> sampai tag penutupnya </li>. Berikut tag <li> yang kalian hapus.

1. Tag <li class=”nav-item”> yang berisi tentang search
2. Tag <li class=”nav-item dropdown notification”> yang berisi tentang notifikasi
3. Tag <li class=”nav-item dropdown connection”> yang berisi tentang connection



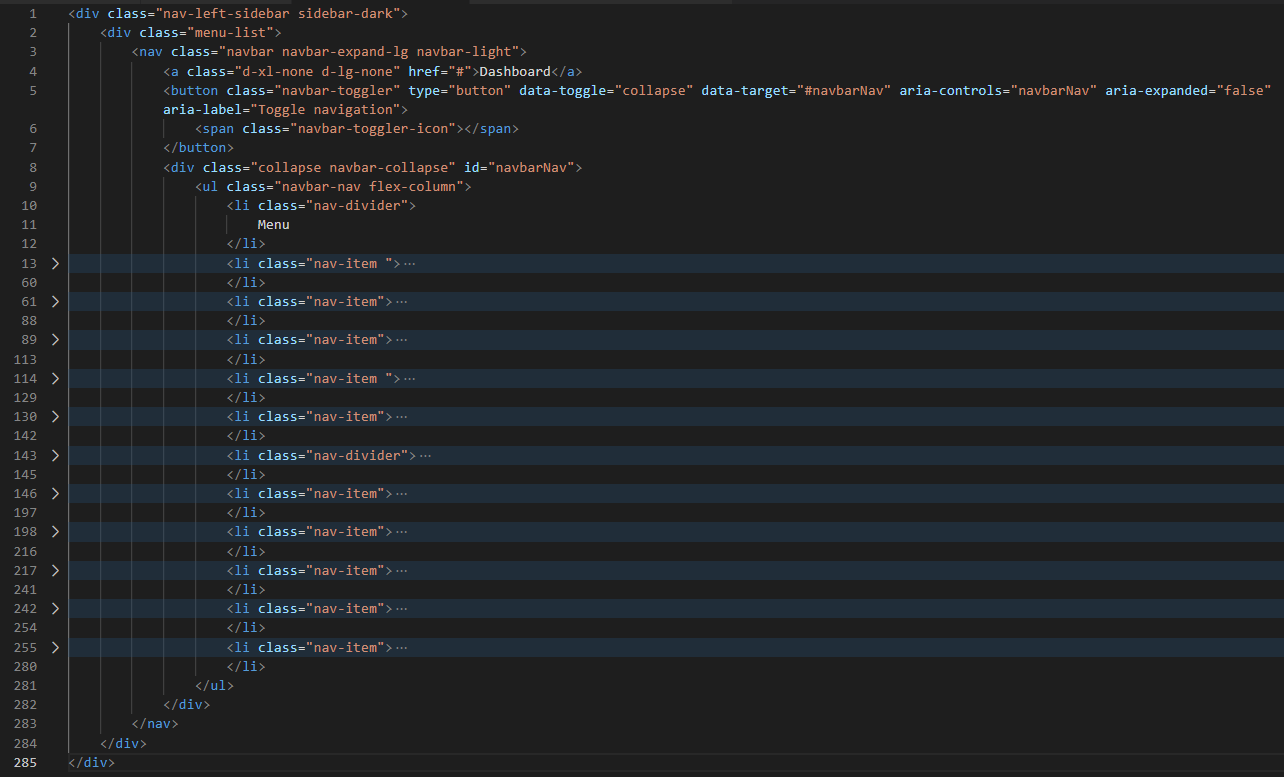
Gambar di atas adalah tag <li> yang perlu kalian hapus.

Berikut source code lengkap file header.php yang sudah diperbaiki.

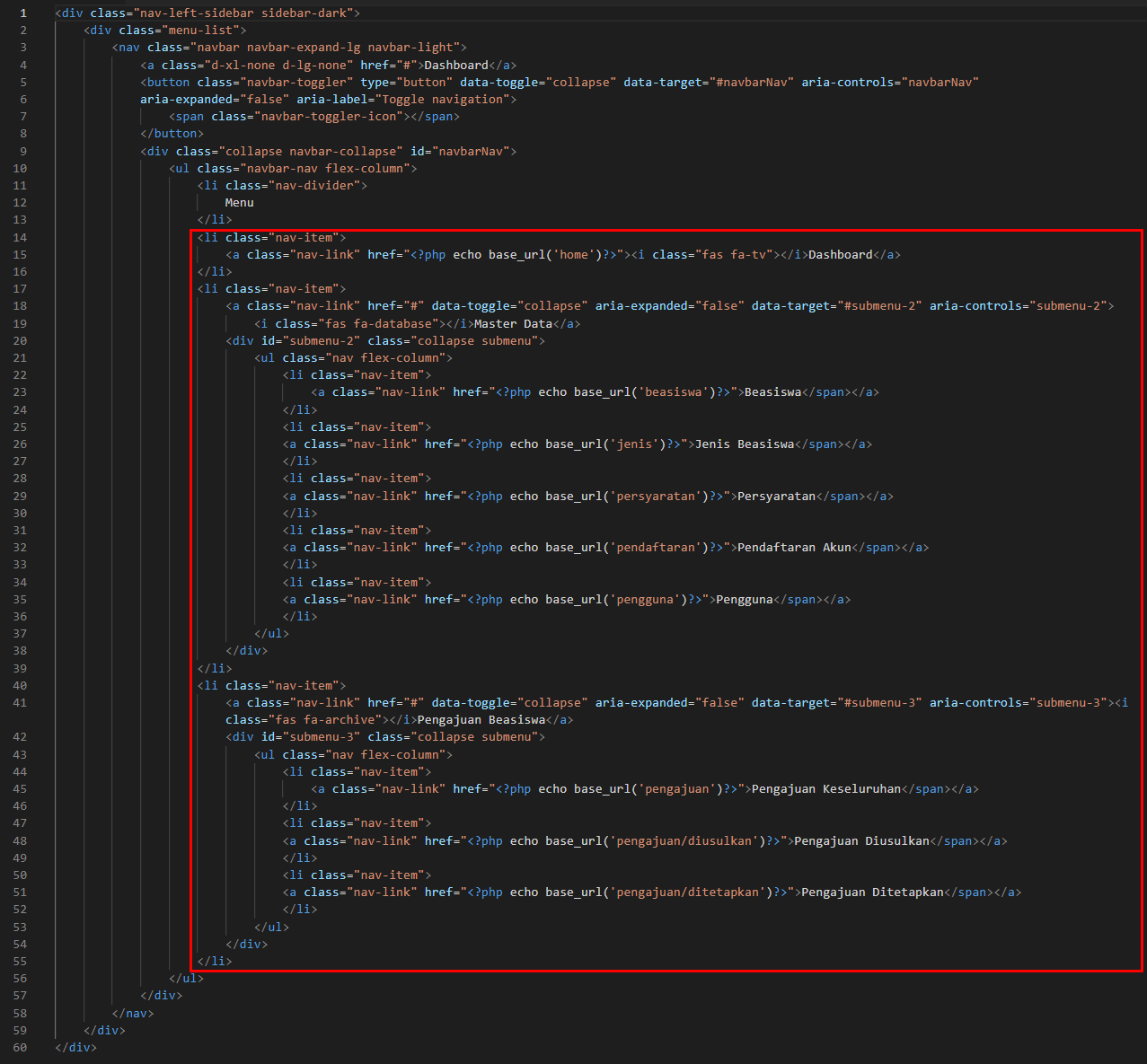


1. **Sidebar**

Berikutnya buat 1 file lagi di folder **template** dengan nama **sidebar.php.** Untuk source code **sidebar** ini, kalian salin lagi dari file **blank-page-header.html** dari baris 141 – 425. Dari source code yang salin tadi, berikut struktur source codenya:

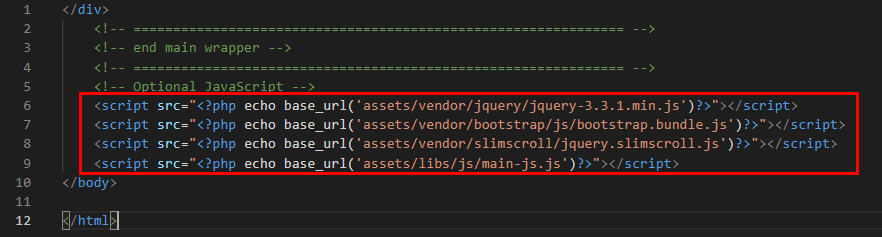


Terlihat dari source code yang disalin di atas, begitu banyak menu bawaan dari template concept-master. Kita akan perbaiki source code sidebar ini supaya menu yang dibuat hanya menu yang kita perlukan saja. Berikut source code sidebar.php setelah diperbaiki.



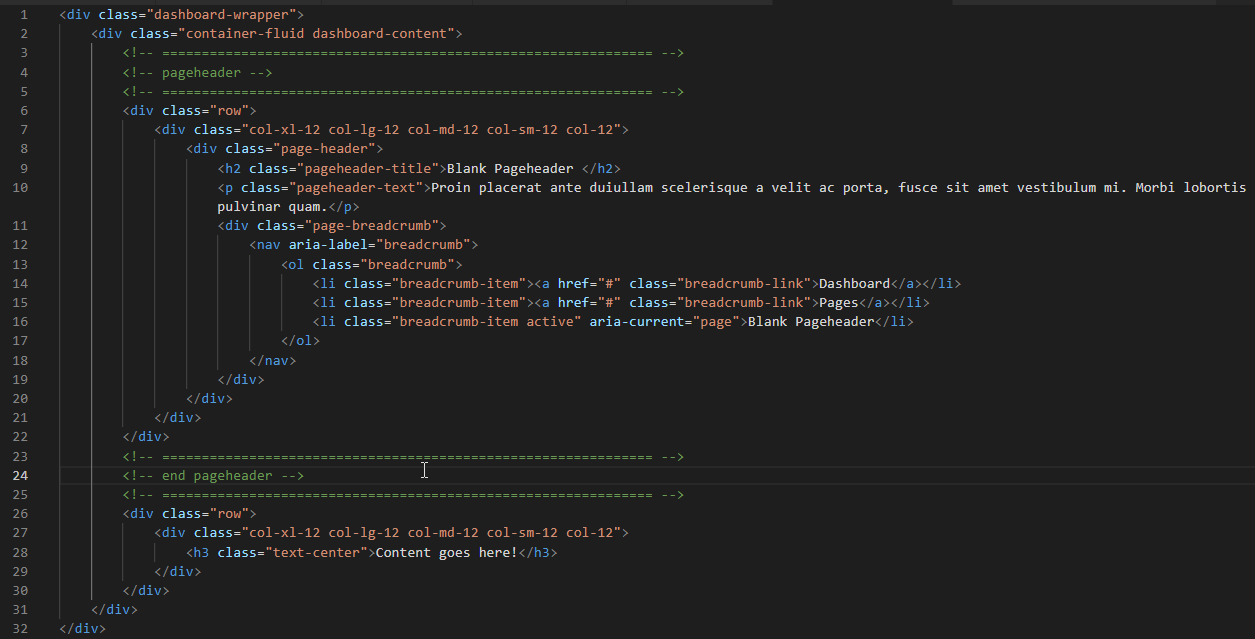
1. **Footer**

File berikutnya adalah footer.php. Salin source code dari **blank-page-header.html** dari source code baris 489-500. Kemudian perbaiki menjadi seperti berikut:



1. **Home\_View**

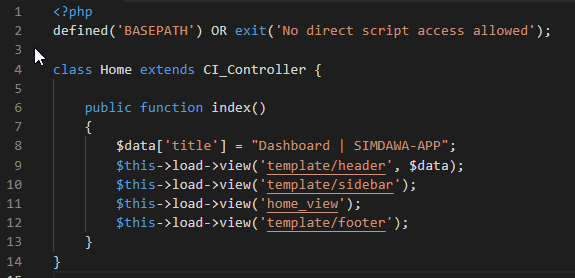
Buat file baru di folder **views** dengan nama file **home\_view.php.** Kemudian salin source code dari **blank-page-header.html** dari baris 432 – 485 ke **home\_view.php.** Setelah disalin, hapus pada bagian tag <div class=”footer”> sampai tutup </div>. sehingga source codenya menjadi seperti berikut:



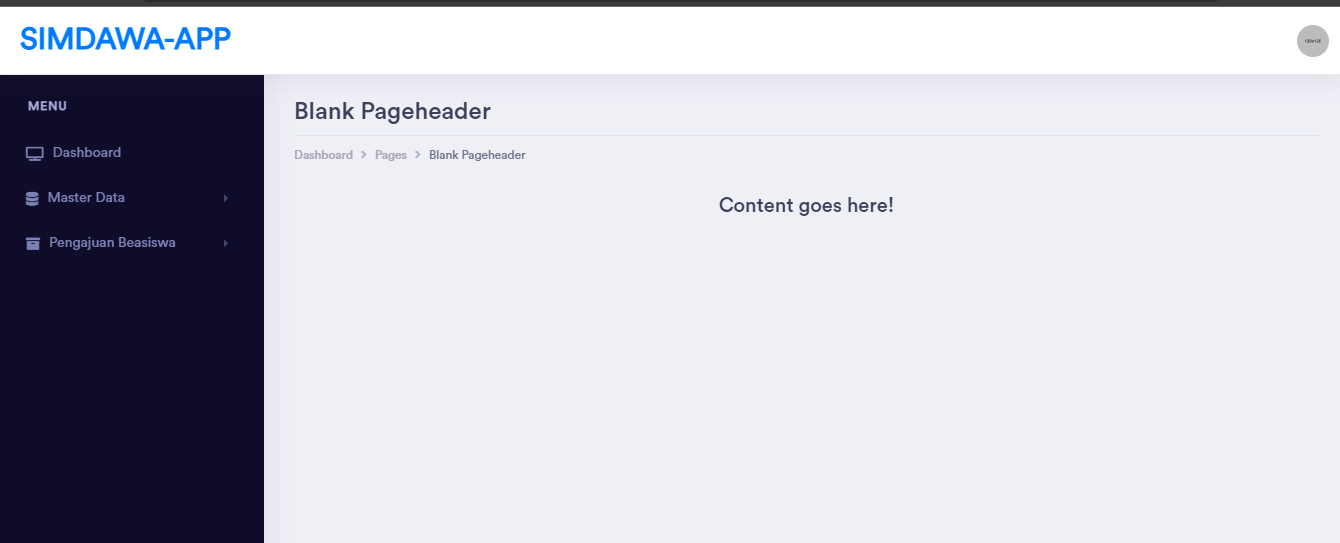
Semua file yang diperlukan untuk template awal sudah selesai.

## Controller Home

Setelah selesai semua template yang dikerjakan, selanjutnya kita akan coba jalankan template tersebut melalui controller home. Buat 1 file controller di folder **controller** dengan nama file **Home.php.** Kemudian ketik source code berikut:



Kemudian jalankan controller home tersebut dengan url: localhost/simdawa-app/home. Berikut tampilan program setelah dijalankan.



# MODUL PRAKTIKUM WEB SEMESTER 6 PERTEMUAN 2 CRUD

Tujuan:

1. Mahasiswa mampu membuat halaman untuk menampilkan data dari tabel database menggunakan CI
2. Mahasiswa mampu membuat halaman untuk proses tambah data ke tabel database menggunakan CI
3. Mahasiswa mampu membuat halaman untuk proses mengubah data di tabel database menggunakan CI
4. Mahasiswa mampu membuat halaman untuk proses menghapus data dari tabel database menggunakan CI

Pada modul pertemuan 2 ini akan dibahas tentang proses pembuatan CRUD pada tabel jenis beasiswa dengan menggunakan framework codeigniter. Sebelum melanjutkan modul ini, pastikan kalian sudah membuat tabel jenis\_beasiswa ini seperti yang sudah dibuat di modul pertemuan 1.

Karena ini adalah proses CRUD pertama kali, maka pada modul ini akan dibahas secara bertahap. Diharapkan kalian bisa mengikuti semua modul ini denagn baik.

## Controller Jenis

File yang pertama yang akan kita buat adalah file controller. Buat file baru di folder controller dengan nama Jenis.php. Kemudian ketik source code berikut:

Text

Description automatically generated

Berikutnya, kita akan membuat halaman model. Karena di controller **jenis** ada memanggil halaman JenisModel, maka kalian buat 1 file di folder **model** dengan nama **JenisModel.php.** Kemudian ketik souce code berikut:

Penjelasan Source Code:

## Baris 6-10: adalah konstruktor. Konstruktor adalah method pertama yang akan jalan saat controller ini dipanggil

## Baris 9: meload file model yaitu JenisModel. Dengan kita meload model, berarti kita bisa memanggil semua function yang ada di model tersebut.

## Baris 15: membuat data dengan indeks array jenis. Yang isi datanya diambil melalui function get\_jenis() di JenisModel. Artinya di JenisModel kita harus membuat function dengan nama get\_jenis()

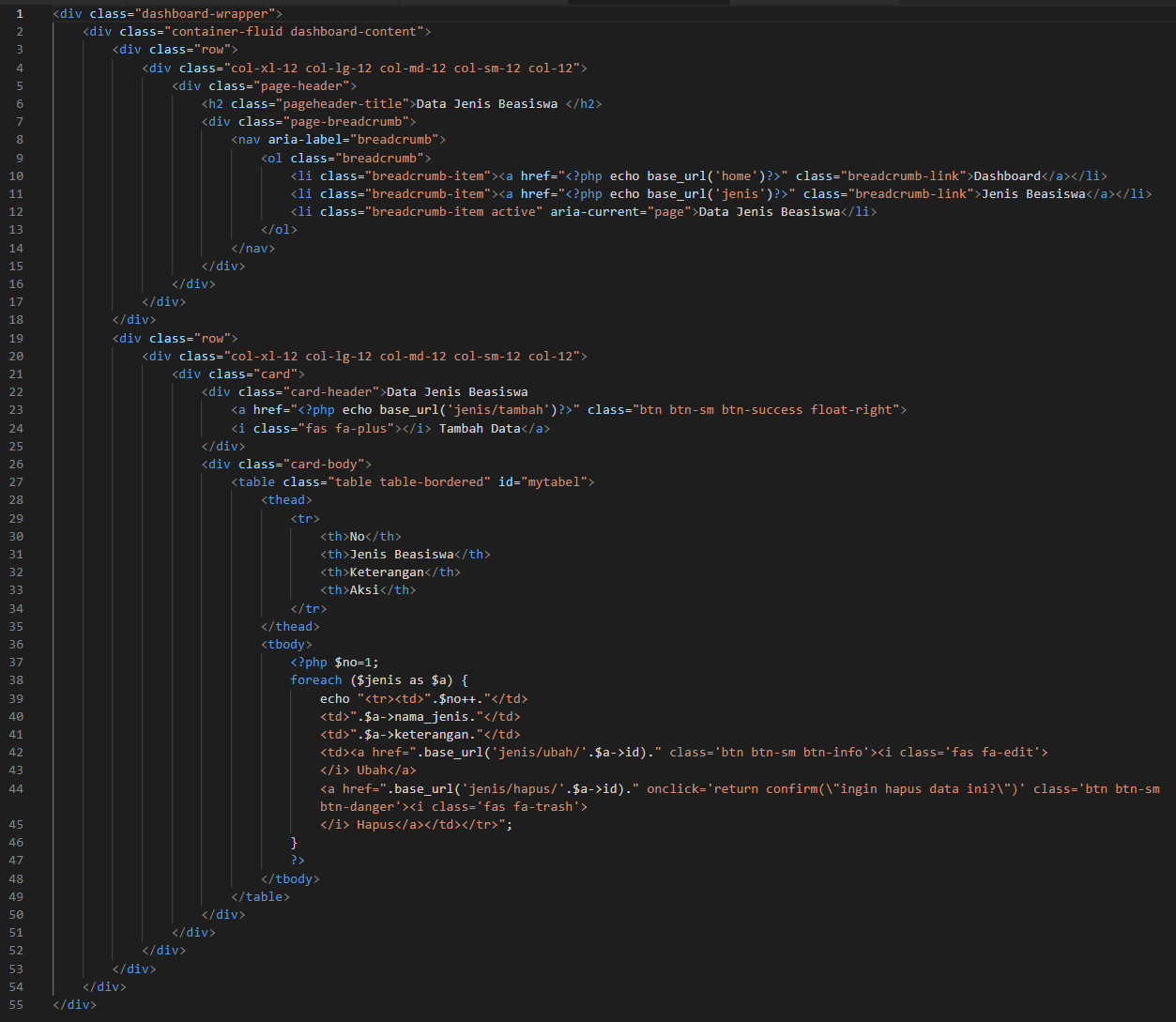
## Baris 18: memanggil halaman view dengan nama jenis\_read.php di folder views/jenis. Dengan mengirimkan data ke jenis\_read.

A screenshot of a computer

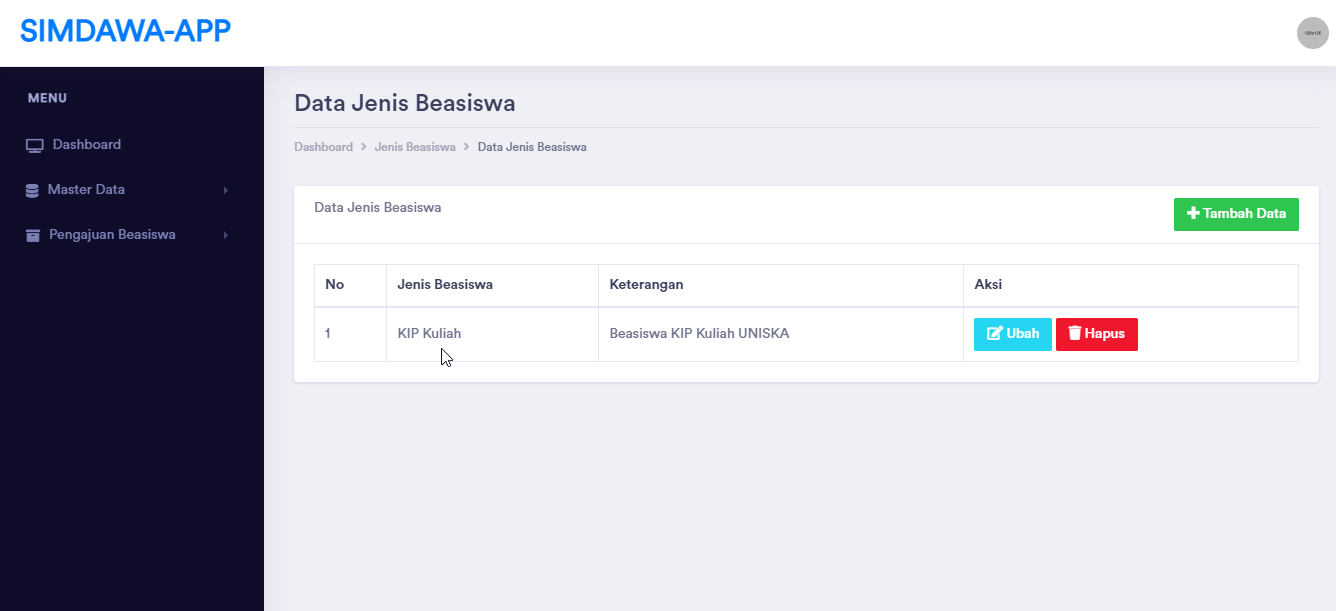
Description automatically generated with medium confidence

Dari Model di atas, di sini ada 1 fungsi dengan nama get\_jenis() yang tujuannya untuk mengambil data dari tabel jenis\_beasiswa. Dengan framework CI, untuk mengambil data dari tabel kita bisa gunakan $this🡪db🡪get().

Setelah model selesai, selanjutnya kita akan buat halaman untuk tampilan (view) data jenis beasiswanya. Di controller **jenis** file view nya dipanggil dari folder **views/jenis** berarti kita buat folder **jenis** di **views.** Kemudian buat file dengan nama **jenis\_read.php.** Berikut source codenya:



Setelah selesai mengerjakan jenis\_read.php tadi. Kalian tambahkan 1 data dulu di tabel jenis\_beasiswa agar hasil jalan programnya bisa terlihat. Silahkan kalian jalankan projek dengan url: localhost/simdawa-app/jenis.



## Tambah Data Jenis Beasiswa

Setelah proses tampil data jenis beasiswa selesai, selanjutnya kita akan buat halaman untuk proses tambah data beasiswa. Untuk menuju halaman tambah jenis beasiswa, di file jenis\_read.php sudah kita buat 1 link yang memanggil halaman tambah data. Berikut link nya:

Text

Description automatically generated

Dari link di atas memanggil controller **Jenis** function **tambah** Artinya, kita buat 1 function di controller Jenis dengan nama tambah. Berikut source codenya:

Text

Description automatically generatedDari function tambah tadi, kita akan membuat 1 file baru di folder **jenis** dengan nama file **jenis\_create.php**. Untuk source code **jenis\_create.php** ini bisa kalian copy dari source code **jenis\_read.php.** Kemudian kalian perbaiki source codenya sehingga menjadi seperti berikut:

Penjalasan:

* Baris 21-32: adalah function tambah. Untuk menjalankan function ini maka urlnya: localhost/simdawa-app/jenis/tambah
* Baris 22: untuk mengecek apakah tombol ‘create’ sudah diset/diklik. Bila sudah diklik kerjakan baris 23-24
* Baris 23: Memanggil fungsi **insert\_jenis()** di JenisModel
* Baris 24: Merefresh halaman kemudian menuju controller jenis atau function index
* Baris 26: Variabel title yang dikirimkan ke header
* Baris 29: Memanggil view jenis\_create di folder **views/jenis**

Text

Description automatically generated

Penjalasan:

* Baris 26: membuat form dengan action kosong. Artinya, saat button submit diklik, maka data yang diinput akan dikirim lagi ke function tambah
* Baris 29: inputan untuk nama jenis. Yang diperhatikan adalah bagian name nya, karena name ini akan kita panggil di model
* Baris 36: sama seperti baris 29
* Baris 40: button submit, untuk mengirimkan data yang diinput. Perhatikan name nya. Pastikan name nya ini sama dengan di function tambah controller Jenis di baris 22

Setelah source code jenis\_create.php selesai, coba kalian jalankan link tambah data. Berikut tampilan dari form tambah jenis beasiswa setelah dijalankan.

Graphical user interface, application, Teams

Description automatically generated

Apabila sudah tampil seperti di atas, selanjutnya kita perlu membuat function di JenisModel. Nama functionnya yaitu: **insert\_jenis().** Buka kembali file **JenisModel.php** kemudian tambahkan function **insert\_jenis()** dengan source code berikut:

Text

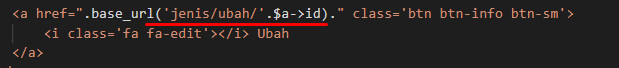
Description automatically generated

$this🡪input🡪post(‘nama\_jenis’) adalah cara CI mengambil nilai dari sebuah inputan. Sebenarnya kita juga bisa menggunakan cara PHP untuk mengambil nilai dari inputan yaitu: $\_POST[‘nama\_jenis’]

Setelah kalian tambahkan function **insert\_jenis()** di atas pada JenisModel, kalian sudah bisa untuk menambahkan data jenis beasiswa. Silahkan kalian coba untuk input data jenis beasiswa, kemudian klik simpan, maka data tersebut akan masuk ke tabel **jenis\_beasiswa.**

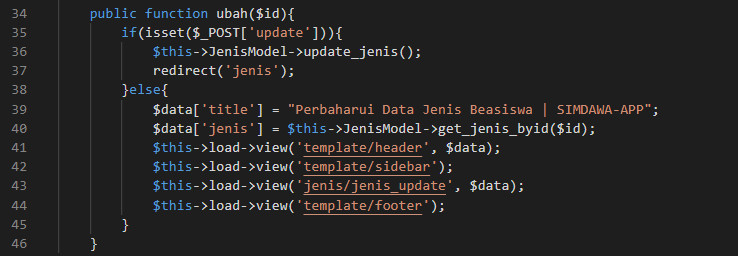
## Ubah Data Jenis Beasiswa

Sebelum melanjutkan ke proses ini, pastikan kalian sudah berhasil mengerjakan proses tambah data. Untuk proses ubah data ini, link ubah nya ada pada jenis\_read.php seperti berikut:



Dari link di atas, dapat diketahui bahwa, proses ubah dilakukan di controller **Jenis** function **ubah d**engan mengirimkan nilai $id.

Buka kembali controller Jenis, kemudian tambahkan function ubah dengan 1 parameter. Berikut source codenya:

****

Penjalasan:

* Baris 34-46 adalah function ubah. Karena di functionnya ada parameter $id, maka urlnya: localhost/simdawa-app/jenis/ubah/1. Nilai 1 ini nantinya akan ditampung $id
* Baris 35: untuk mengecek apakah tombol “update” sudah diset/diklik. Bila sudah diklik jalankan baris 36-37
* Baris 36: Memanggil fungsi **update\_jenis()** di JenisModel
* Baris 37: Merefresh halaman kemudian menuju controller jenis atau function index
* Baris 39: Variabel title yang dikirimkan ke header
* Baris 40: Variabel jenis yang nilainya diambil dari function **get\_jenis\_byid**() di JenisModel
* Baris 43: Memanggil view jenis\_update di folder **views/jenis** dengan mengirimkan nilai dari variable jenis.

Dari function ubah di atas, kita akan buat 2 buah function di JenisModel, yaitu: **update\_jenis()** dan **get\_jenis\_byid.** Buka kembali **JenisModel.php** kemudian ketik source code berikut:

Text

Description automatically generated

Source code yang berwarna hijau, hanya sebagai komentar / catatan saja.

Setelah JenisModel sudah kalian tambahkan, maka selanjutnya adalah membuat halaman view untuk ubah datanya. Buat 1 file baru di folder **views/jenis** dengan nama file **jenis\_update.php.** Untuk source codenya, bisa kalian salin dari **jenis\_create.php.** Kemudian diperbaiki sehingga menjadi seperti berikut:

Text

Description automatically generated

Text

Description automatically generated

Perhatikan nomor baris source code untuk menghindari kesalahan pengetikan. Dan source code yang bergaris merah adalah source yang perlu kalian perbaiki / tambahkan.

Apabila sudah selesai source cod di atas, kalian bisa jalankan untuk proses ubah data jenis beasiswa. Klik tombol ubah pada salah satu data. Maka akan tampil form ubah data dengan inputan yang sudah ada nilainya (value).

Graphical user interface, application, Teams

Description automatically generated

Terlihat pada gambar di atas, pada inputan sudah muncul data jenis beasiswa yang mau diperbaharui. Klik tombol simpan untuk mengubah data sesuai yang diinginkan.

## Hapus Data Jenis Beasiswa

Langkah terakhir pada proses CRUD ini adalah proses pembuatan hapus data jenis beasiswa. Proses hapus data ini dilakukan dengan mengklik tombol hapus pada tampil data. Link proses hapusnya adalah sebagai berikut:

Text

Description automatically generated

Pada source code di atas, ada tambahan onclick…, ini bertujuan untuk memunculkan konfirmasi terlebih dahulu sebelum menuju ke link href. Berdasarkan href di atas, maka proses hapus dilakukan pada controller Jenis function hapus.

Buka kembali controller Jenis kemudian tambahkan function **hapus()** seperti berikut ini:

Text

Description automatically generated

Dari function hapus di atas, proses hapus data dilakukan di function **delete**\_**jenis**() pada JenisModel dengan mengirimkan $id sebagai acuan data yang akan dihapus. Buka kembali JenisModel.php, kemudian tambahkan function delete\_jenis() seperti berikut:

Text

Description automatically generated

Setelah selesai mengerjakan function di atas, kalian sudah bisa menjalankan proses hapus data jenis beasiswa ini.

Graphical user interface, application, website

Description automatically generated

Tampilan setelah diklik button hapus, muncul konfirmasi untuk dilakukan proses hapus data.

Semua proses CRUD telah selesai, pastikan semua proses CRUD pada tabel jenis beasiswa ini berjalan dengan baik.

Berikut source code lengkap dari controller Jenis dan JenisModel:

Text

Description automatically generated

Text

Description automatically generated

Tugas:

Lakukan proses yang sama seperti pengerjaan tabel jenis beasiswa untuk pengerjaan tabel persyaratan. Tahap pengerjaannya:

1. Buat Controller Persyaratan

2. Buat Model PersyaratanModel

3. Buat Folder Persyaratan di Views

4. Buat File persyaratan\_read.php

5. Buat File persyaratan\_create.php

6. Buat File persyaratan\_update.php

Untuk source codenya, kalian bisa salin saja dari proses pengerjaan jenis beasiswa.

# MODUL PRAKTIKUM WEB SEEMSTER 6 PERTEMUAN 3 RELASI TABEL DAN REPORT

Tujuan:

1. Mahasiswa dapat menerapakan contoh kasus CRUD pada Tabel yang saling berelasi
2. Mahasiswa dapat membuat report

Alat dan Bahan

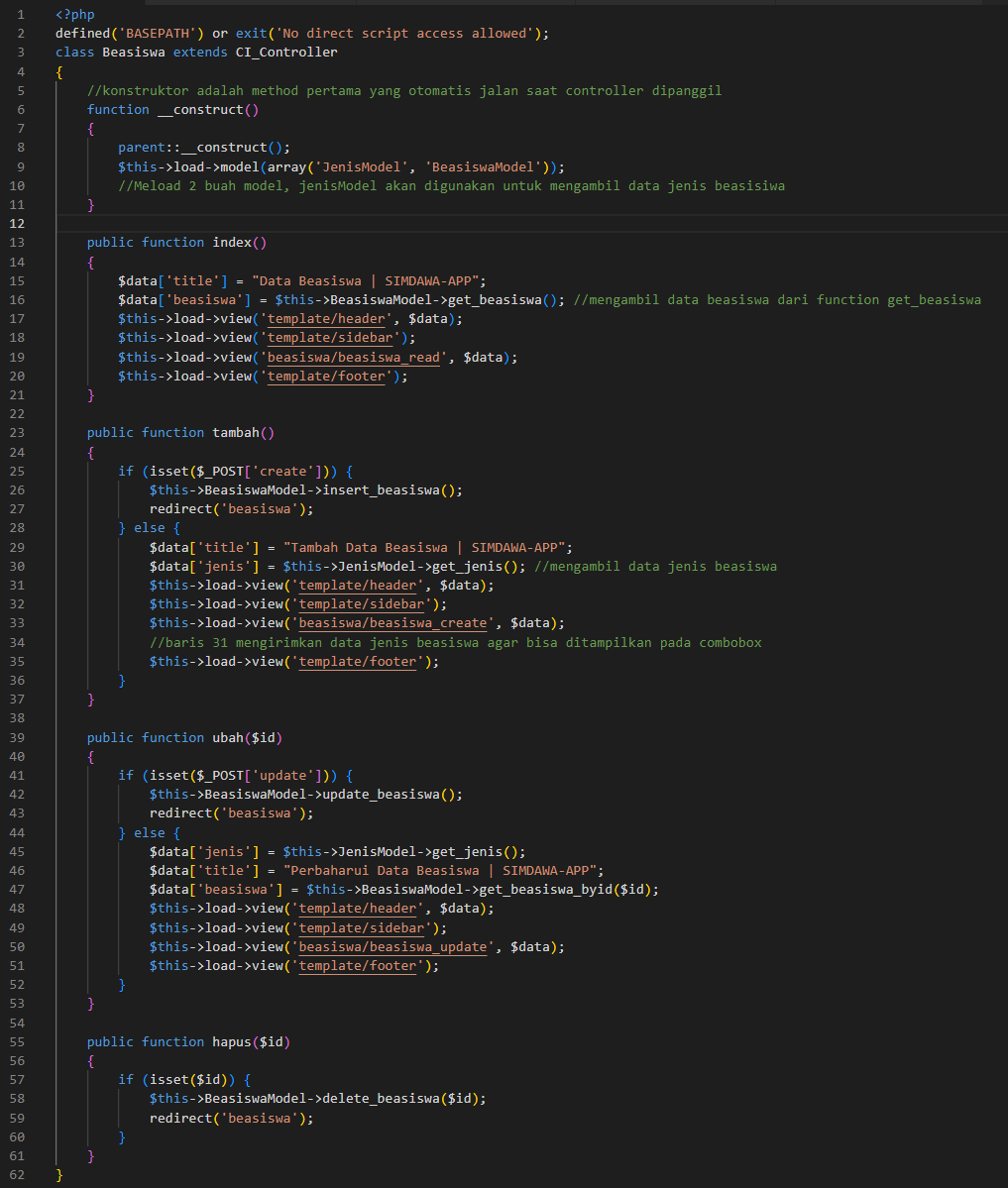
1. Libraries FPDF (<http://www.fpdf.org/en/dl.php?v=184&f=zip>)

Pendahuluan

Pada modul pertemuan 3 ini akan dibahas bagaimana menerapkan CRUD pada tabel yang saling berelasi. Untuk kasus tabel yang berelasi ini akan kita terapkan pada CRUD tabel Beasiswa yang berelasi dengan tabel Jenis Beasiswa. Kemudian dilanjutkan dengan pembuatan 1 contoh report dengan menggunakan libraries FPDF.

## Controller Beasiswa

Langkah pertama yang akan kita kerjakan adalah membuat controller Beasiswa terlebih dahulu. Silahkan buat file controller dengan nama Beasiswa. Kemudian source codenya bisa kalian salin dari source code controller Jenis yang sebelumnya sudah dikerjakan. Berikut source code controller Beasiswa:



Penjalasan:

* Source code pada controller Beasiswa sebenarnya hampir sama saja dengan controller Jenis. Hanya saja, ada beberapa perbedaan di antaranya:
  + Model yang dipanggil ada 2 (JenisModel dan BeasiswaModel) baris 9. JenisModel kita panggil karena kita perlu memanggil sebuah function yang ada di JenisModel yaitu function get\_jenis().
  + Pemanggilan function get\_jenis() baris 30 dan 45 bertujuan untuk mengambil data dari tabel jenis, lalu kemudian data tersebut dikirimkan ke beasiswa\_create dan beasiswa\_update untuk ditampilkan dalam bentuk combobox

## BeasiswaModel

Setelah controller, file selanjutnya yang akan kita buat adalah file Model. Buat 1 buah file model dengan nama BeasiswaModel. Kemudian ketik source code berikut:

Text

Description automatically generated

Text

Description automatically generated

Penjalasan:

* Source code BeasiswaModel ini sebenarnya hampir sama saja dengan model yang sebelumnya kita buat. Hanya saja, pada function get\_beasiswa() (baris 6-15) untuk mengambil data dari tabel beasiswa kita gunakan query SQL. Yang mana querynya adalah gabungan dari 2 tabel (join) dengan tabel jenis beasiswa
* Baris 11-14 adalah cara join kedua yang bisa kita lakukan untuk mengambil data beasiswa.
* Untuk proses insert dan update, jumlah datanya lebih banyak, karena kolom di tabel beasiswa ada 5 kolom yaitu (id, nama\_beasiswa, tanggal\_mulai, tanggal\_selesai, jenis\_id)

## Tampil Data Beasiswa

Setelah BeasiswaModel kalian selesaikan, selanjutnya yang akan kita buat adalah file yang digunakan untuk menampikan data beasiswa. Sesuai di controller Beasiswa baris 19 function **index**, buat 1 buah folder terlebih dahulu di folder views dengan nama **beasiswa.** Kemudian di folder tersebut, buat 1 file dengan nama **beasiswa\_read.php.** Untuk source code **beasiswa\_read.php** ini bisa kalian salin dari source code sebelumnya. Berikut source code lengkap dari file **beasiswa\_read.php:**

**Text

Description automatically generated**

Dari source code di atas, tidak banyak perbedaan dari file read yang sebelumnya. Yang membedakan hanya pada jumlah kolom di tabel saja.

Setelah selesai semua source code beasiswa\_read.php, kalian bisa jalankan halaman tampil data beasiswa ini. Jalankan projek simdawa-app, kemudian pada sidebar, pilih menu Beasiswa, yang mana menu Beasiswa ini akan mengarahkan halaman ke controller Beasiswa tanpa function artinya function index yang jalan. Berikut tampilan dari data beasiswa ini.

A screenshot of a computer

Description automatically generated

Apabila punya kalian datanya masih kosong, maka biarkan saja dulu. Karena proses tambah data beasiswa akan kita kerjakan berikutnya.

## Tambah Data Beasiswa

Proses selanjutnya yang akan kita kerjakan adalah proses tambah data beasiswa. Proses tambah data beasiswa ini sebenarnya sama saja seperti proses tambah data sebelumnya. Hanya saja, karena tabel beasiswa ini direlasikan ke tabel jenis\_beasiswa, maka kita perlu memasukan data jenis beasiswa ke proses tambah data ini.

Silahkan kalian buat file baru di folder views/beasiswa dengan nama file **beasiswa\_create.php.** Kalian bisa salin source code dari jenis\_create.php kemudian perbaiki source codenya sehingga menjadi seperti berikut:

Text

Description automatically generated

Source code yang bertanda kotak merah adalah source code untuk pembuatan form input data beasiswa.

Seletelah menyelesaikan source code beasiswa\_create.php di atas, kalian bisa jalankan proses tambah data beasiswa ini. Pada halaman beasiswa yang aktif, klik link untuk tambah data beasiswa, maka akan tampil seperti berikut:

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

Berikut tampilan isi dari combobox nama jenis beasiswa. Item yang ada di combobox itu diambil dari data tabel jenis beasiswa.

Coba isi data beasiswa lalu klik button simpan. Pastikan proses simpan data berhasil sehingga halaman akan diarahkan kembali ke tampil data beasiswa.

## Perbaharui Data Beasiswa

Proses selanjutnya yang akan kita kerjakan adalah proses ubah data beasiswa. Untuk proses ubah data beasiswa ini, buat file baru pada folder **views/beasiswa** dengan nama file **beasiswa\_update.php (**sesuai controller Beasiswa baris 50)**.** Source code **beasiswa\_update.php** bisa kalian salin dari **beasiswa\_create.php** dengan tambahan **value** pada inputan. Berikut source code dari **beasiswa\_update.php** ini:

Text

Description automatically generated

Source code yang berkotak merah adalah source code yang ditambahkan dari beasiswa\_create.php.

Setelah selesai mengerjakan beasiswa\_update.php, coba jalankan proses ubah data beasiswa dengan cara klik tombol ubah pada halaman tampil data. Berikut tampilan jalan program dari proses ubah data beasiswa.

Graphical user interface, application

Description automatically generated

Terlihat pada gambar di atas, pada form ubah data beasiswa, semua inputan terisi data sesuai data dari database. Untuk combobox jenis beasiswa, item yang terpilih sesuai data di tabel beasiswanya.

Lakukan proses ubah data, pastikan proses ubah data beasiswa berjalan dengan baik dengan berhasil mengubah data beasiswa.

## Hapus Data Beasiswa

Proses hapus data beasiswa sama saja seperti proses hapus data pada tabel yang lain. Klik tombol hapus pada tampil data beasiswa, muncul konfirmasi, klik yes, maka halaman akan diarahkan ke base\_url(‘beasiswa/hapus/1’) baris 62 pada beasiswa\_read.php, yang mana artinya link tersebut menuju ke controller **beasiswa** jalankan function **hapus** dengan mengirimkan **id** sebagai acuan yang akan dihapus.

## Report Data Beasiswa

Selanjutnya yang akan kita lakukan pada modul 3 ini adalah membuat report atau cetak data. Contoh kasus yang akan kita kerjakan adalah membuat report data beasiswa. Untuk membuat report ini pastikan kalian sudah mengunduh library FPDF yang sudah dibagikan di alat dan bahas di atas. Berikut proses pembuatan report data beasiswa ini:

1. Letakkan library FPDF yang sudah diunduh ke folder **application/third\_party.**

Graphical user interface, application

Description automatically generated

1. Kemudian klik kanan pada file tersebut lalu pilih **Extract To fpdf**

Graphical user interface, text, application

Description automatically generated

1. Setelah langkah 2 dilakukan, maka akan muncul 1 folder dengan nama **fpdf** di folder **third\_party.**
2. Kemudian buka kembali Visual Studio Code lalu buat 1 file baru di folder **application/libraries** dengan nama file adalah **pdf.php**

Graphical user interface, application

Description automatically generated

1. Pada file pdf.php tersebut, ketik source code berikut:

Graphical user interface

Description automatically generated

Tujuan dari source code di atas adalah untuk memanggil file fpdf.php yang ada di folder **third\_party/fpdf**

1. Selanjutnya buka kembali controller Beasiswa lalu tambahkan pemanggilan file pdf yang dibuat tadi beserta tambahkan juga 1 buah function dengan nama cetak. Berikut source code dari controller Beasisswa:

Text

Description automatically generated

Source code yang bergaris merah saja yang ditambahkan di controller Beasiswa.

1. Dari function cetak yang ditambahkan di controller Beasiswa, kita perlu menambahkan 1 file baru di folder **views/beasiswa** dengan nama file **beasiswa\_print.php.** Ketik source code berikut ini:

Text

Description automatically generated

Untuk penjelasan source code di atas, bisa dipahami dari komentar yang sudah ditambahkan

1. Setelah beasiswa\_print sudah selesai, kita perlu menambahkan 1 buah link button yang ditujukan untuk memanggil beasiswa\_print tersebut. Buka kembali beasiswa\_read.php, kemudian tambahkan 1 tag <a href> di bawah link tambah data. Berikut source code beasiswa\_php setelah ditambahkan link untuk cetak data.

Text

Description automatically generated

Tambahkan yang bergaris merah saja.

Setelah beasiswa\_read.php kalian tambahkan source code seperti di atas, coba jalankan kembali projek simdawa-app kemudian pilih menu beasiswa. Berikut tampilan halaman data beasiswa yang sudah ditambahkan link Cetak Data.

Graphical user interface, application, table

Description automatically generated

Lalu klik button Cetak Data di atas, maka akan tampil report data beasiswa seperti berikut:

Graphical user interface

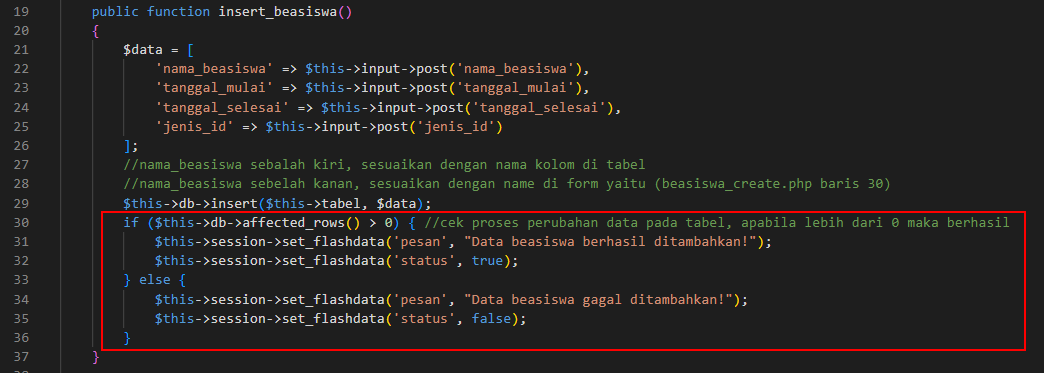
Description automatically generated

Hasil tampilan cetak report data beasiswa. Pastikan kalian sudah berhasil membuat report seperti di atas.

## Menambahkan Notifikasi

Seperti yang sudah kalian kerjakan pada 3 buah CRUD sebelumnya yaitu tabel beasiswa, tabel jenis beasiswa dan tabel persyaratan. Semua proses CRUD tersebut belum kita tambahkan notifikasi/pemberitahuan bahwa proses tambah, ubah dan hapus berhasil dilakukan. Untuk menambahkan notifikasi tersebut kita bisa gunakan **$this🡪session🡪set\_flashdata(‘name’,’isi pesan’).** Untuk menambahkan notifikasi ini, akan dicontohkan pada proses tambah data beasiswa saja. Berikut langkah-langkah yang dilakukan untuk menambahkan notifikasi ini:

1. Buka file BeasiswaModel, kemudian pada function insert\_beasiswa tambahkan beberapa baris seperti berikut:



Tambahkan source code yang bergaris merah saja.

1. Kemudian buat 1 file baru di folder **views/template** dengan nama file **notifikasi.php.** Lalu ketik source code berikut:

A screenshot of a computer

Description automatically generated

1. Setelah notifikasi.php selesai, selanjutnya file **notifikasi.php** ini akan kita panggil di file **beasiswa\_read.php.**
2. Buka kembali **beasiswa\_read.php** kemudian tambahkan source code seperti berikut:

A picture containing text

Description automatically generated

Tambahkan source code yang bergaris merah saja.

1. Proses penambahan notifikasi sudah selesai. Coba jalankan proses tambah data beasiswa, setelah data berhasil ditambahkan maka akan muncul notifikasi seperti gambar berikut:

Graphical user interface, application, table

Description automatically generated

Tugas:

1. Lakukan proses yang sama seperti pembuatan report data beasiswa untuk membuat report data jenis beasiswa dan report data persyaratan.
2. Tambahkan notifikasi pada semua proses tambah, ubah dan hapus pada tabel jenis\_beasiswa dan persyaratan seperti di proses tambah data beasiswa.

# MODUL PRAKTIKUM WEB SEMESTER 6 PERTEMUAN 4 PENDAFTARAN PENGGUNA, LOGIN DAN VERIFIKASI AKUN

Tujuan

1. Mahasiswa diharapkan dapat membuat halaman untuk pendaftaran pengguna yang di dalamnya terdapat proses upload data
2. Mahasiswa dapat menambahkan proses upload data pada sistem dengan menggunakan framework CodeIgniter
3. Mahasiswa dapat membuat halaman proses Login menggunakan framework CodeIgniter

Alat dan Bahan

1. File sign-up.html dan login.html dari file concept-master

Pada modul pertemuan 4 ini, akan kita bahas tentang proses pembuatan pendaftaran pengguna agar nantinya pengguna/mahasiswa bisa login ke sistem. Setelah pendaftaran pengguna selesai, akan dilanjutkan lagi ke proses pembuatan login. Kemudian terakhir pembuatan proses verifikasi akun pendaftaran pengguna serta pembatasan menu-menu berdasarkan pengguna yang login. Sebelum melanjutkan pengerjaan di modul 4 ini, pastikan kalian sudah membuat tabel yang sudah diuraikan di pertemuan 1.

## Preview Jalan Program Pendaftaran

Sebelum mengerjakan proses pendaftaran, akan ditampilkan terlebih dahulu hasil proses tampilan program pendaftaran ini. Berikut tampilan hasil jalan program proses pendaftaran:

1. Tampilan Halaman Pendaftaran

Graphical user interface, application, Teams

Description automatically generated

1. Tampilan Pemberitahuan No Pendaftaran Sudah Ada Di Sistem

Graphical user interface, application, Teams

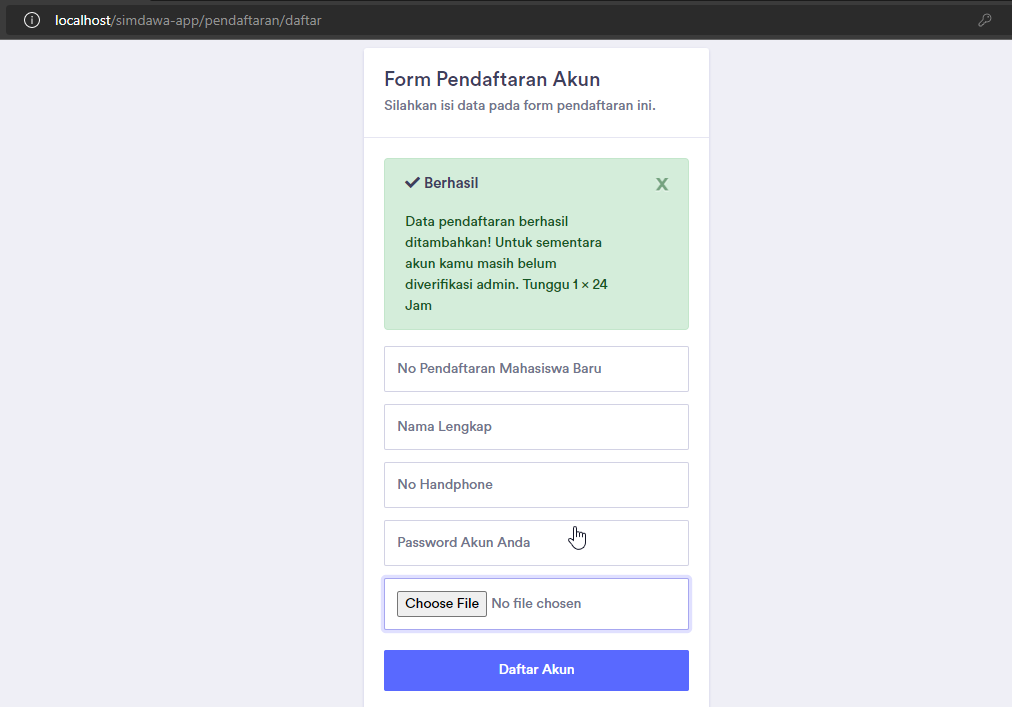
Description automatically generated

1. Tampilan Pemberitahuan Tipe File yang Diupload Salah

Graphical user interface, application

Description automatically generated

1. Tampilan Pemberitahun Pendaftaran Sukses



## Folder Upload Bukti Daftar

Untuk proses pendaftaran ini, seperti yang terlihat di gambar pada subjudul A, pendaftaran memerlukan upload bukti daftar sebegai verifikasi. Untuk itu, kita perlu membuat 1 buah folder yang nantinya digunakan untuk menampung file yang diupload oleh pendaftar.

Graphical user interface, application

Description automatically generatedGraphical user interface, text

Description automatically generatedBuat 1 buah folder di projek **simdawa-app** dengan nama **upload.** Kemudian di folder tersebut tambahkan lagi 1 buah folder dengan nama **bukti\_daftar.** Berikut tampilan struktur folder setelah ditambahkan 2 buah folder ini:

## Controller Pendaftaran

Proses pertama yang akan kita kerjakan pada modul 4 ini adalah proses pembuatan pendaftaran pengguna. Untuk mengerjakan ini, buat controller **Pendaftaran** terlebih dahulu dan tambahkan 1 function **daftar**. Berikut source code dari controller **Pendaftaran** ini:

Text

Description automatically generated

Dari source code controller Pendaftaran tersebut bisa kita lihat bahwa:

1. Buat model dengan nama **PendaftaranModel** (sesuai baris 8)
2. Buat function **cek\_nopendaftaran** pada model tersebut (sesuai baris 20)
3. Buat function **upload\_bukti** pada model PendaftaranModel (sesuai baris 25)
4. Buat function **insert\_pendaftaran** pada model PendaftaranModel (sesuai baris 27)
5. Buat folder **pendaftaran** di folder **views.** Dan buat file di folder tersebut dengan nama **daftar\_create.php** (sesuai baris 38)

## PendaftaranModel

Langkah berikutnya buat model dengan nama **PendaftaranModel.** Berikut source code dari PendaftaranModel ini:



Dari source code PendaftaranModel di atas, terlihat ada bertambah 1 buah function yaitu **insert\_pengguna.** Function **insert\_pengguna** ini digunakan untuk menambahkan data ke tabel pengguna yang nanti fungsinya untuk proses login. Function **insert\_pengguna** ini akan jalan apabila proses penambahan data ke pendaftaran pengguna sukses (baris 48). Username untuk login ini diambil dari no\_pendaftaran (baris 50), dan untuk passwordnya menggunakan enkripsi md5 (baris 51).

## File View daftar\_create

Controller dan model sudah kita selesaikan, selanjutnya membuat halaman untuk form pendaftaran penggunanya. Buat 1 buah folder terlebih dahulu di folder **views** dengan nama folder **pendaftaran.** Kemudian di folder tersebut tambahkan 1 file dengan nama **daftar\_create.php (**sesuai file yang dipanggil di controller Pendaftaran baris 38)

Untuk membuat halaman tampilan form pendaftaran ini, kalian salin source codenya dari file **sign-up.html** yang ada di folder template **concept-master** ke file **daftar\_create.php.** Kemudian kalian perbaiki source codenya sehingga menjadi seperti berikut:

Text

Description automatically generated

Text

Description automatically generated

Perhatikan nomor baris agar tidak terjadi kesalahan pada program.

Setelah 3 file ini kalian selesaikan, kalian sudah bisa menjalankan proses pendaftaran pengguna ini. Untuk menjalankannya, karena ini belum dibuatkan link menuju form pendaftaran, maka menjalankannya langsung kalian isi saja di url dengan link: **localhost/simdawa-app/pendaftaran/daftar.** Hasil jalan program bisa dilihat pada gambar di sub judul A poin 1.

Silahkan kalian coba lakukan proses pendaftaran, pastikan semua proses pendaftaran berhasil dan data pedantarannya masuk ke 2 tabel, yaitu: tabel pendaftaran\_pengguna dan tabel pengguna.

## Verifikasi Akun Pendaftan

Setelah semua proses pendaftaran pengguna berhasil kalian kerjakan, berikutnya akan kita lanjutkan lagi projek ini pada proses verifikasi akun pendaftar. Proses pendaftaran yang sudah berhasil, belum bisa masuk ke sistem, karena status akunnya masih “**Belum Diverifikasi”.** Untuk mengatur Verifikasi ini akan dilakukan oleh akun operator atau kepala bagian kemahasiswaan.

Sebelum kita lakukan pengerjaan proses verifikasi akun, kalian perbaiki dulu tabel pendaftaran\_pengguna pada kolom keterangan menjadi dengan 3 buah enum yaitu: “Sudah Diverifikasi”, “Belum Diverifikasi”, “Akun Dibatalkan” dengan menjalankan query sebagai berikut:

**ALTER** **TABLE** pendaftaran\_pengguna

**CHANGE** keterangan keterangan **ENUM**('Sudah Diverifikasi','Belum Diverifikasi', 'Akun Dibatalkan')

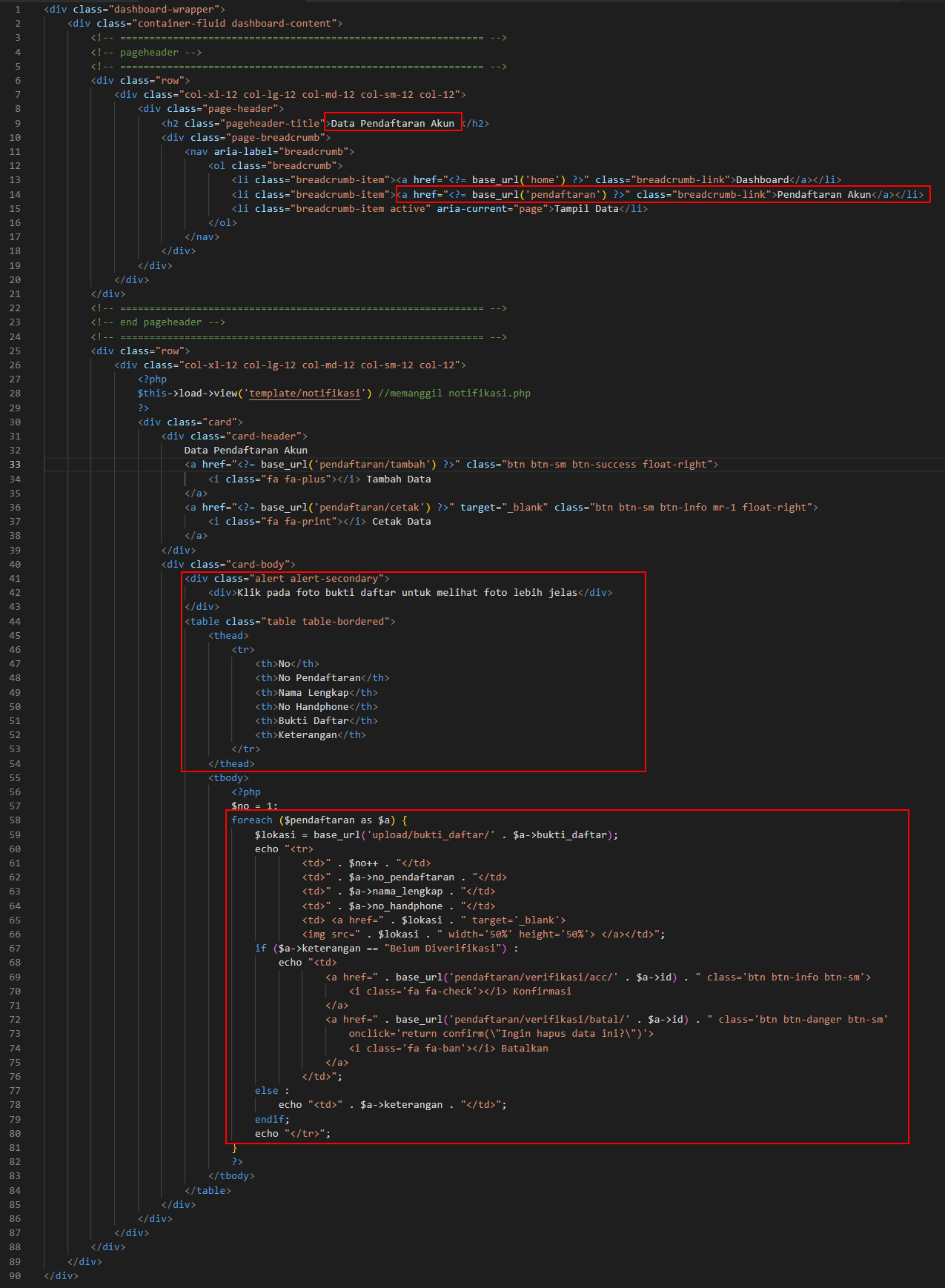
Selanjutnya Buka kembali file controller **Pendaftaran,** kemudian pada function **index** kalian tambahkan seperti berikut:

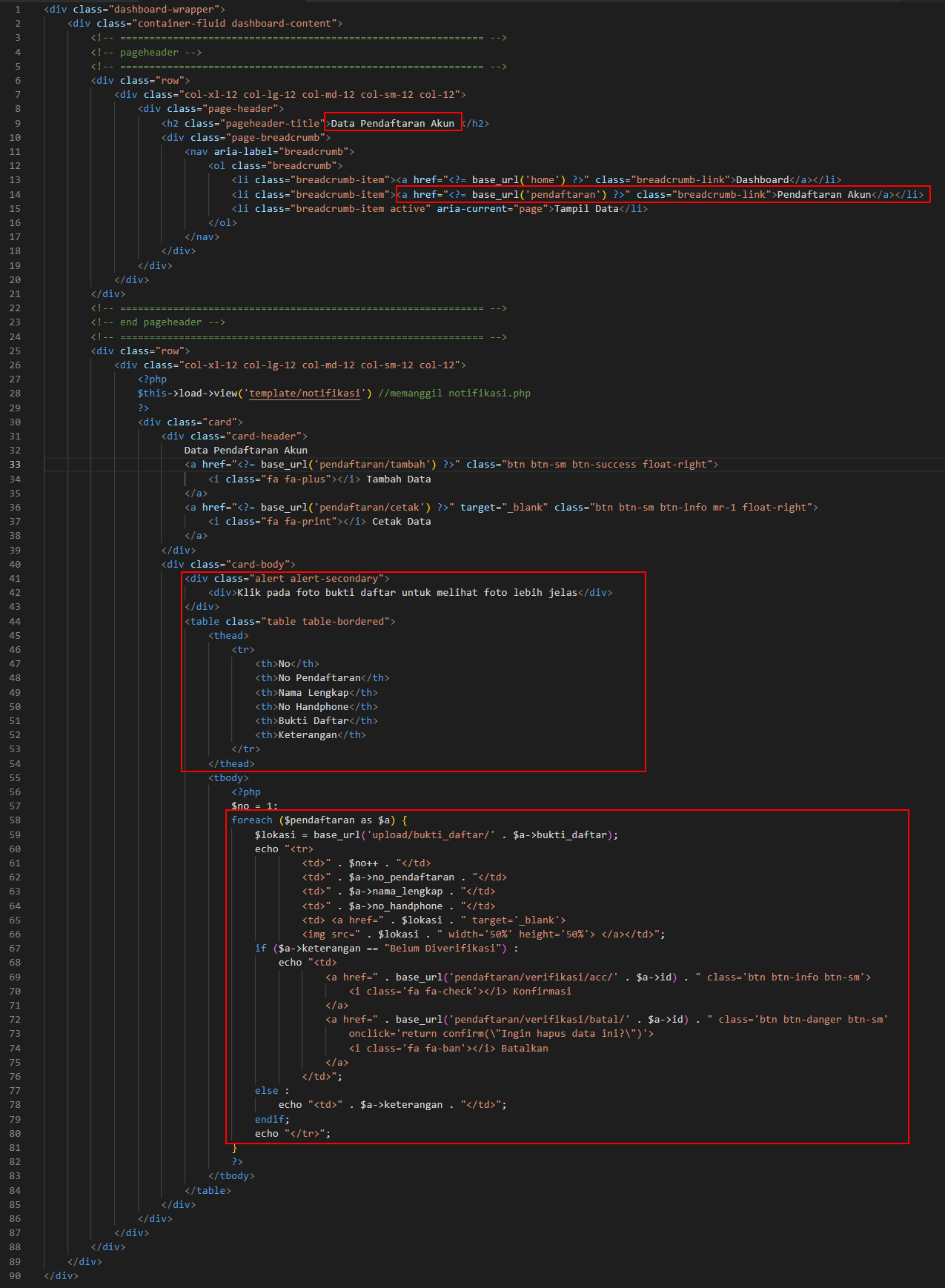


Dari source code di atas, kita perlu menambahkan 1 buah function di PendaftaranModel dengan nama get\_**pendaftaran**. Dan perlu membuat 1 file lagi dengan nama **pendaftaran**\_**read.php** di folder **pendaftaran**.

Silahkan kalian tambahkan sendiri 1 function di PendaftaranModel dengan nama function get\_pendaftaran (source code sama seperti get\_jenis di JenisModel).

Kemudian kalian buat 1 file baru di folder pendaftaran dengan nama **pendaftaran\_read.php** dan salin source codenya dari **beasiswa\_read.php.** Lalu perbaiki sehingga menjad seperti berikut:

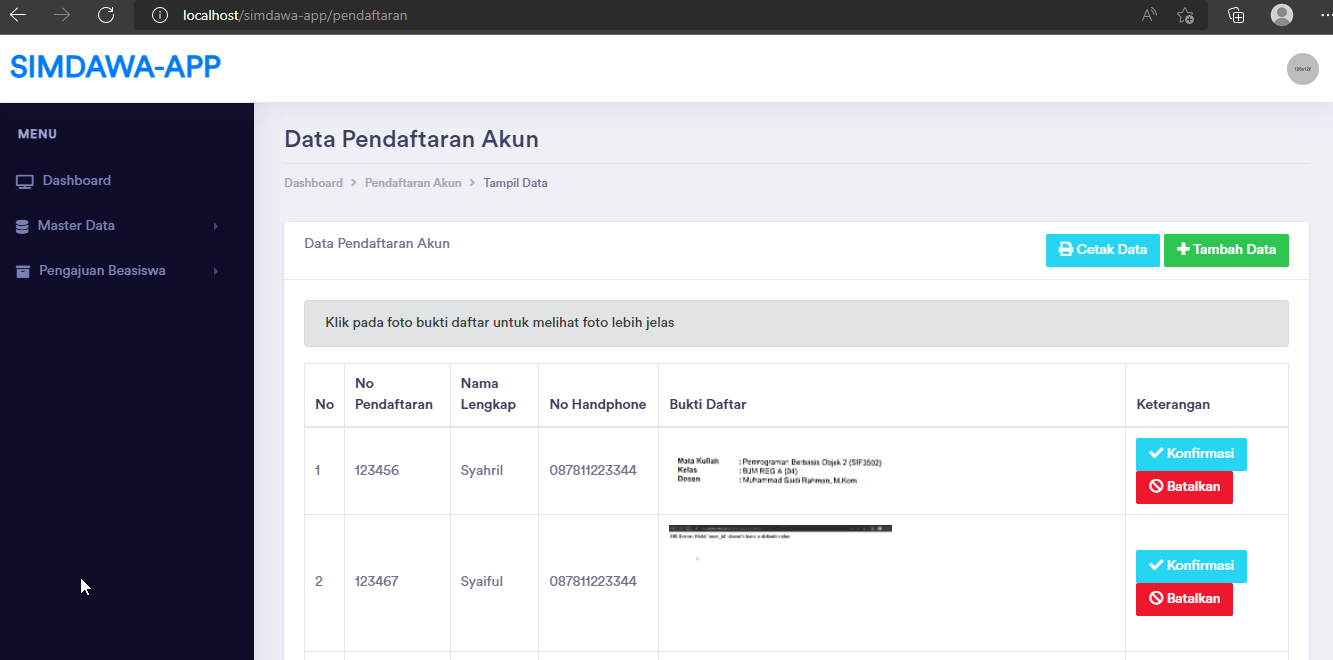




Penjelasan Source Code:

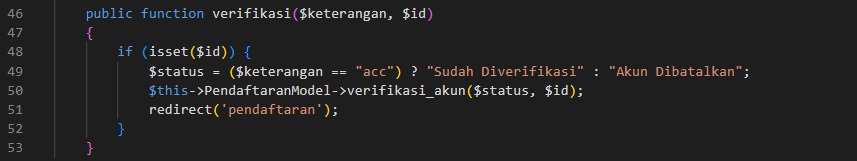
* Source code di atas hampir sama saja seperti file read lainnya, yang membedakan ada pada baris 59 dan baris 67 sampai 80.
* Baris 59 adalah membuat variable $lokasi yang datanya diisi dari lokasi file foto yang sudah diupload.
* Baris 67 sampai 80 digunakan untuk memunculkan tombol konfirmasi dan batalkan jika kolom keterangan masih bernilai Belum Diverifikasi. Untuk proses konfirmasi akun dilakukan pada controller Pendaftaran di function verifikasi (baris 67 & 72).
* Baris 78 akan menampilkan isi kolom keterangan kalau kolom keterangan nilainya selain Belum Diverifikasi.

Apabila sudah kalian selesaikan source code di atas, kalian bsa jalankan halaman pendaftaran ini dengan mengklik menu pendaftaran di bagian sidebar. Berikut hasil jalan programnya.

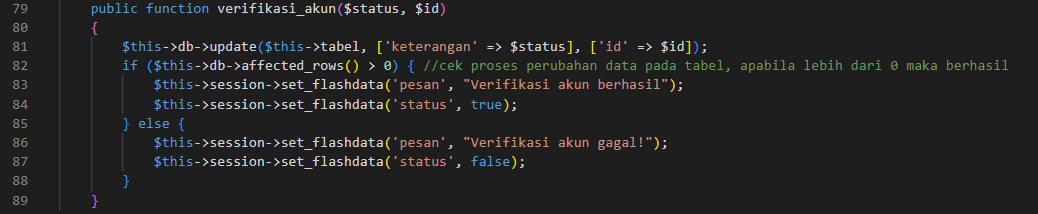


Dari gambar di atas terlihat, terdapat image yang menampilkan bukti daftar yang telah diupload. Apabila kalian klik image tersebut, maka akan tampil ukuran aslinya (baris 65-65). Dan terdapat 2 buah tombol untuk Konfirmasi dan Batalkan.

Tombol Konfirmasi dan Batalkan tersebut akan diarakan ke controller Pendaftaran function verifikasi (baris 69 & 72). Untuk itu kalian perlu tambahkan 1 function seperti berikut ini di controller Pendaftaran.



Baris 49 adalah percabangan yang menggunakan operator ternary (?). Yang maksudnya adalah variable $status akan bernilai “Sudah Diverifikasi” jika variable $keterangan nilainya sama dengan “acc”. Selanjutnya proses verifikasi akun dilakukan pada PendaftaranModel pada function **verifikasi\_akun** (baris 50). Untuk itu, kalian tambahkan 1 function tersebut di PendaftaranModel dengan source code seperti berikut:



Jika semua source code di atas sudah kalian selesaikan, maka selesailah sudah proses dari Verifikasi Akun ini. Silahkan kalian coba jalan proses verifikasi akun, pastikan semuanya berjalan dengan baik.

## Pembuatan Akun Operator dan Admin

Proses pembuatan akun Operator dan Admin bisa kalian buat sendiri dari halaman Pengguna dengan menambahkan link tambah data untuk proses pembuatan akun.

## Form Login

Proses berikutnya yang akan kita kerjakan adalah Form Login. Pada Form Login ini akan kita berikan beberapa validasi seperti pendaftar yang login namun akunnya belum diverifikasi maka akan muncul “akun anda belum diverifikasi”. Username atau password salah maka akan muncul “Username dan Password tidak ditemukan di sistem”.

Step pertama yang perlu kita lakukan adalah membuat controller Login terlebih dahulu. Silahkan kalian buat controller Login ini kemudian ketik source code berikut:

Text

Description automatically generated

Dari source code di atas, seperti yang kalian lihat di sana ada pemanggilan LoginModel (baris 10). Kemudian pada LoginModel tersebut kita buat function dengan nama cek\_login (baris 15). Dan baris 35 ada pemanggilan view dengan nama **login\_view** yang ada di folder **views/login.**

**Langkah** berikutnya yang akan kita lakukan adalah sesuai baris 10 pada controller Login, kita perlu membuat model dengan nama **LoginModel.** Berikut source code dari LoginModel tersebut:

Text

Description automatically generated

Semua penjelasan source code LoginModel ini sudah disertakan pada bagian komentarnya. Intinya dari LoginModel ini akan mencek akun di tabel pengguna berdasarkan username dan password yang diinput. Apabila username dan passwordnya ada di sistem, maka akan dicek lagi keterangan akun dari pengguna tersebut di tabel **pendaftaran\_pengguna**. Apabila kolom **keterangan** Belum Diverifikasi maka login tidak bisa dilakukan (baris 17).

Setelah LoginModel di atas kalian selesaikan, langkah terakhir yang akan kita lakukan untuk melakukan proses login ini adalah membuat view dengan nama **login\_view.php** dan diletakkan di folder **views/login.** Silahkan kalian buat folder dan file tersebut. Untuk source tampilan halaman Login, kita akan gunakan tampilan dari template **concept-master** pada file **login.html**. Berikut source code **login\_view.php** setelah dimodifikasi dari file template.

Text

Description automatically generated

Pastikan source code yang kalian ketik sudah sesuai dengan source code di atas.

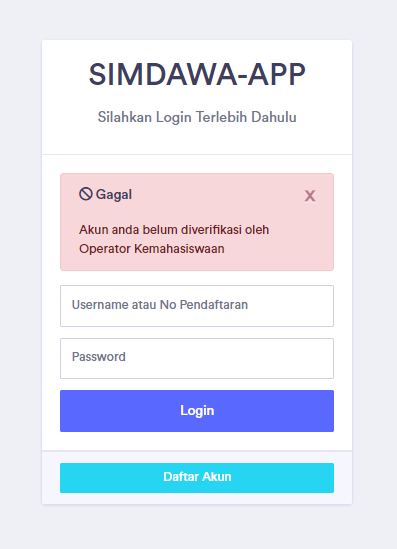
Setelah semua file kalian selesaikan (Login Controller, LoginModel, Login\_View). Kalian sudah bisa untuk menjalankan proses login ini. Jalankan di browser dengan url: **localhost/simdawa-app/login.** Berikut hasil jalan program proses login:

1. Tampilan Halaman Login

Graphical user interface, application

Description automatically generated

1. Tampilan Validasi Akun Belum Diverifikasi



1. Tampilan Username dan Password Salah

Graphical user interface, application

Description automatically generated

## Pembatasan Hak Akses

Dari halaman login yang sudah kita kerjakan, kita dapat mengetahui, siapa yang sedang login ke dalam sistem. Di sistem yang kita buat ini terdiri dari 3 jenis akun yaitu: ADMIN, OPERATOR dan USER. Akun pengguna dengan peran USER hanya bisa melihat data saja dan melakukan pengajuan beasiswa. Sedangkan Akun dengan peran OPERATOR dapat melakukan proses CRUD namun tidak bisa menambahkan akun pengguna. Dan Akun ADMIN dapat melakukan semua proses CRUD termasuk menambahkan akun pengguna.

Untuk melakkukan pembatasan tersebut, kita perlu mengecek dari **session peran** (baris 21 di controller Login).

1. Pembatasan Menu

Pembatasan pertama akan kita atur pada menu, di mana USER hanya dapat mengakses menu Beasiswa saja. Buka kembali file **sidebar.php** kemudian perbaiki source codenya dengan menambahkan source code yang bertanda merah.



Kotak merah pertama digunakan untuk membatasi agar menu Jenis Beasiswa, Persyaratan, Pendaftaran Akun hanya bisa diakses selain USER.

Kotak merah kedua digunakan untuk membatasi agar menu Pengguna hanya bisa diakses oleh akun ADMIN.

1. Pembatasan CRUD

Berikutnya buka file beasiswa\_read.php, pada file ini akan kita modifikasi agar proses Tambah, Ubah, Hapus dan Cetak hanya bisa dilakukan selain akun USER. Berikut source codenya:

