

## FRAMEWORK CODEIGNITER



### A. Pengertian

Menurut wikipedia bahasa Indonesia, CodeIgniter merupakan aplikasi sumber terbuka yang berupa framework PHP dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP.

Menurut wikipedia bahasa Inggris yang diterjemahkan, CodeIgniter adalah sebuah perangkat lunak open-source perkembangan pesat framework web, untuk digunakan dalam membangun situs web dinamis dengan PHP .

## **B. Latar Belakang**

Latar belakang permasalahan kali ini membuat aplikasi berbasis web yang bisa bekerja dengan baik, cepat, mudah, sederhana dan terstruktur adalah keinginan setiap pengembang aplikasi. Apalagi jika aplikasi yang dibangun bersekala cukup besar dan cukup rumit. Karena itulah mungkin sekarang saatnya anda melihat CodeIgniter, sebuah framework pengembangan PHP yang gratis, ringan, mudah diinstall, dan lengkap. CodeIgniter adalah sebuah Web Application Framework, yaitu aplikasi framework yang sengaja didesain untuk mendukung pengembangan web dinamis atau aplikasi berbasis web.

## **C. Tujuan**

Bertujuan mengenal dan memperdalam tentang framework codeigniter.

## **D. Tahap Pelaksanaan**

### **1. Pengertian**

CodeIgniter adalah framework PHP yang kuat dengan footprint yang sangat kecil, dibangun untuk pengembang yang membutuhkan toolkit sederhana dan elegan untuk membuat aplikasi web dengan fitur lengkap.

### **2. Alasan menggunakan Framework CI**

- **Kinerja yang luar biasa** : CI secara konsisten melebihi kebanyakan pesaingnya.
- **Tidak ada skala besar library monolitik** : tidak mencoba untuk menjadi segalanya bagi semua orang. Ini

adalah kerangka kerja MVC ramping, dengan kemampuan yang cukup untuk meningkatkan produktivitas Anda, sambil memberikan untuk pihak ketiga addons atau plugin untuk fungsi tambahan.

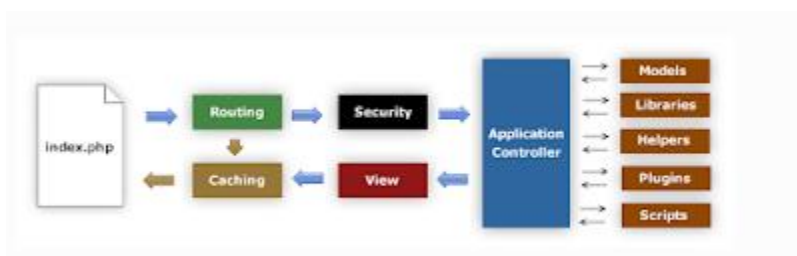
- **Hampir nol konfigurasi** : Banyak dari konfigurasi CodeIgniter dilakukan dengan konvensi, misalnya menempatkan model dalam folder "model". Masih ada sejumlah pilihan konfigurasi yang tersedia, melalui script di folder "config".
- **Menghabiskan lebih banyak waktu jauh dari komputer** : CodeIgniter mudah untuk belajar dan untuk mendapatkan mahir dengan.
- **Solusi sederhana atas kompleksitas** : CI mendorong MVC, namun tidak memaksa.
- **Tidak ada aturan coding ketat** : Gunakan sendiri coding dan penamaan konvensi Anda, dengan hanya beberapa peringatan yang berhubungan dengan konflik classname. CodeIgniter terlihat untuk memberdayakan Anda, tidak membelenggu Anda.
- **Kerangka dengan footprint yang kecil** : CodeIgniter 3 memiliki download 2MB, termasuk buku petunjuk.
- **Dokumentasi jelas** : Panduan Pengguna CodeIgniter dilengkapi dengan download. Ini berisi pengantar, tutorial, sejumlah "bagaimana" panduan, dan kemudian referensi dokumentasi untuk komponen yang membentuk kerangka.
- **Kompatibilitas dengan standart hosting** : CodeIgniter 3 hanya membutuhkan PHP 5.3.7, dan memainkan baik dengan hampir semua berbagi atau dedicated hosting platform. Banyak webapps perlu database, dan CodeIgniter mendukung paling umum, termasuk MySQL.

### 3. Fitur-fitur dalam CI

- MVC Berbasis Sistem
- Berat sangat Cahaya
- database kelas Pilihan lengkap dengan dukungan untuk beberapa platform.
- Query Builder database Dukungan
- Form dan Validasi Data
- Keamanan dan XSS Filtering+
- Manajemen sesi
- Email Mengirim Class. Mendukung -Lampiran, HTML / email Text, beberapa protokol (sendmail, SMTP, dan Mail) dan banyak lagi.
- Manipulasi Gambar Perpustakaan (cropping, mengubah ukuran, memutar, dll). Mendukung GD, ImageMagick, dan Netpbm.
- Berkas Mengunggah Kelas
- FTP Kelas
- lokalisasi
- pagination
- Enkripsi data
- benchmarking
- Halaman Caching penuh
- kesalahan Logging
- aplikasi Profiling
- kalender Kelas

- User Agent Kelas
- Zip Encoding Kelas
- Template Engine Kelas
- traceback Kelas
- XML-RPC Library
- Unit Pengujian Kelas
- Search-engine Friendly URL
- Fleksibel URI Routing
- Dukungan untuk Hooks dan Kelas Ekstensi
- perpustakaan besar "pembantu" fungsi.

#### 4. Alur kerja sistem data pada Framework CI



Fungsi-fungsi dari alur kerja sistem data dalam CI :

- index.php berfungsi sebagai controller depan, menginisialisasi sumber dasar yang dibutuhkan untuk menjalankan CodeIgniter.
- Router ini meneliti permintaan HTTP untuk menentukan apa yang harus dilakukan dengan hal itu.
- Jika file cache ada, dikirim langsung ke browser, melewati eksekusi sistem normal.

- **Keamanan.** Sebelum controller aplikasi dimuat, permintaan HTTP dan setiap pengguna diserahkan Data disaring untuk keamanan.
- **Controller beban model, perpustakaan inti, pembantu, dan sumber daya lain yang diperlukan untuk memproses permintaan khusus.**
- **Melihat diselesaikan diberikan kemudian dikirim ke web browser untuk dilihat.** Jika caching diaktifkan, pandangan-cache pertama sehingga pada permintaan selanjutnya dapat disajikan.

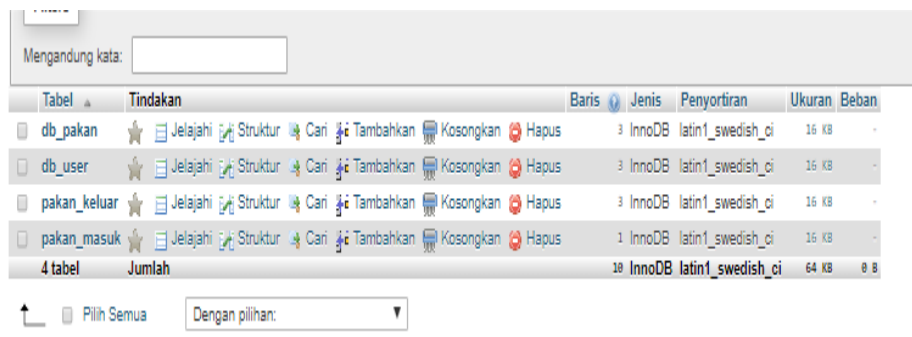
## 5. Tujuan Framework CI

- **Instansiasi dinamis.** Pada CodeIgniter, komponen yang dimuat dan rutinitas dieksekusi hanya jika diminta, bukan global. Tidak ada asumsi yang dibuat oleh sistem mengenai apa yang mungkin diperlukan di luar sumber daya minimal inti, sehingga sistem ini sangat ringan secara default. Peristiwa, seperti dipicu oleh permintaan HTTP, dan pengendali dan pandangan Anda merancang akan menentukan apa yang dipanggil.
- **Loose Coupling.** Coupling adalah sejauh mana komponen dari suatu sistem saling mengandalkan. Komponen kurang bergantung satu sama lain lebih dapat digunakan kembali dan fleksibel sistem menjadi. Tujuan kami adalah sistem yang sangat longgar ditambah.
- **Komponen Singularity.** Singularity adalah sejauh mana komponen memiliki tujuan yang difokuskan secara sempit. Dalam CodeIgniter, setiap kelas dan fungsinya sangat otonom untuk memungkinkan kegunaan maksimum.

# Proses Pembuatan Aplikasi Menggunakan Codeigneter

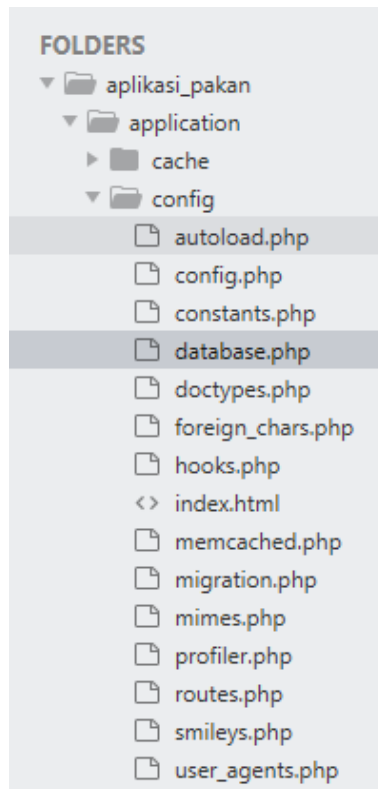
Langkah-langkah membuat sebuah aplikasi Pengelolaan dan Persediaan data Pakan konsentrat KPSBU(Koperasi Peternak Sapi Perah Bandung Utara) menggunakan codeigniter dan API.

Langkah awal dalam pembuatan aplikasi yaitu membuat sebuah database terlebih dahulu pada <http://localhost/phpmyadmin/index.php> disini kita membuat sebuah nama databasenya adalah aplikasi\_pakan, dalam database aplikasi\_pakan.



Tabel	Tindakan	Baris	Jenis	Penyortiran	Ukuran	Beban
db_pakan	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	3	InnoDB	latin1_swedish_ci	16 KB	-
db_user	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	3	InnoDB	latin1_swedish_ci	16 KB	-
pakan_keluar	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	3	InnoDB	latin1_swedish_ci	16 KB	-
pakan_masuk	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	1	InnoDB	latin1_swedish_ci	16 KB	-
4 tabel	Jumlah	10	InnoDB	latin1_swedish_ci	64 KB	0 B

Source code pada database.php di :



Berikut adalah source code databasenya ;

```
<?php  
  
defined('BASEPATH') OR exit('No direct script access  
allowed');  
  
/*  
  
| -----  
  
| DATABASE CONNECTIVITY SETTINGS
```



```
| -----  
  
| This file will contain the settings needed to access your  
| database.  
  
|  
  
| For complete instructions please consult the 'Database  
| Connection'  
  
| page of the User Guide.  
  
|  
  
| -----  
  
| EXPLANATION OF VARIABLES  
  
| -----  
  
|  
  
|      ['dsn']      The full DSN string describe a connection to  
| the database.  
  
|      ['hostname'] The hostname of your database server.  
  
|      ['username'] The username used to connect to the  
| database  
  
|      ['password'] The password used to connect to the  
| database
```

|        ['database'] The name of the database you want to connect to

|        ['dbdriver'] The database driver. e.g.: mysqli.

|                                Currently supported:

|                                cubrid, ibase, mssql, mysql, mysqli, oci8,

|                                odbc, pdo, postgres, sqlite, sqlite3, sqlsrv

|        ['dbprefix'] You can add an optional prefix, which will be added

|                                to the table name when using the Query Builder class

|        ['pconnect'] TRUE/FALSE - Whether to use a persistent connection

|        ['db\_debug'] TRUE/FALSE - Whether database errors should be displayed.

|        ['cache\_on'] TRUE/FALSE - Enables/disables query caching

|        ['cachedir'] The path to the folder where cache files should be stored

|       ['char\_set'] The character set used in communicating with the database

|       ['dbcollat'] The character collation used in communicating with the database

|                       NOTE: For MySQL and MySQLi databases, this setting is only used

|                       as a backup if your server is running PHP < 5.2.3 or MySQL < 5.0.7

|                       (and in table creation queries made with DB Forge).

|                       There is an incompatibility in PHP with mysql\_real\_escape\_string() which

|                       can make your site vulnerable to SQL injection if you are using a

|                       multi-byte character set and are running versions lower than these.

|                       Sites using Latin-1 or UTF-8 database character set and collation are unaffected.

|       ['swap\_pre'] A default table prefix that should be swapped with the dbprefix

|       ['encrypt']   Whether or not to use an encrypted connection.

|

|                       'mysql'   (deprecated),   'sqlsrv'   and  
'pdo/sqlsrv' drivers accept TRUE/FALSE

|                       'mysqli' and 'pdo/mysql' drivers accept an  
array with the following options:

|

|                       'ssl\_key'   - Path to the private key  
file

|                       'ssl\_cert'   - Path to the public key  
certificate file

|                       'ssl\_ca'    - Path to the certificate  
authority file

|                       'ssl\_cpath' - Path to a directory  
containing trusted CA certificates in PEM format

|                       'ssl\_cipher' - List of \*allowed\*  
ciphers to be used for the encryption, separated by colons (':')

|                       'ssl\_verify' -    TRUE/FALSE;  
Whether verify the server certificate or not

|

| [compress] Whether or not to use client compression  
(MySQL only)

| ['stricton'] TRUE/FALSE - forces 'Strict Mode' connections

- | - good for ensuring strict SQL while developing

- | `['ssl_options']` Used to set various SSL options that can be used when making SSL connections.

| 'failover' array - A array with 0 or more data for connections if the main should fail.

| ['save\_queries'] TRUE/FALSE - Whether to "save" all executed queries.

NOTE: Disabling this will also effectively disable both

| `$this->db->last_query()` and profiling of DB queries.

When you run a query, with this setting set to TRUE (default),

CodeIgniter will store the SQL statement for debugging purposes.



```
'hostname' => 'localhost',

'username' => 'root',

'password' => "",

'database' => 'aplikasi_pakan',

'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,

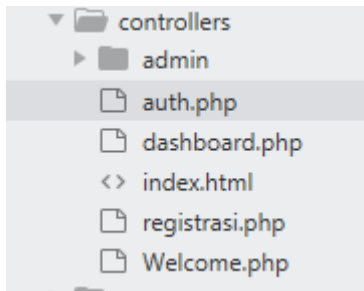
'stricton' => FALSE,

'failover' => array(),

'save_queries' => TRUE
```

```
);
```

Setelah itu tambahkan login pada Controller ;



Berikut adalah source codenya ;

```
<?php

/**
 *
 */

class Auth extends CI_Controller
{

    public function login()
    {
```



```

        $this->form_validation-
>set_rules('username','Username','required',['required'      =>
'Username Harus diisi!']);

        $this->form_validation-
>set_rules('password','Password','required',['required'      =>
'Password Harus diisi!']);

        if ($this->form_validation->run() == FALSE)

        {

            $this->load->view('templates/header');

            $this->load->view('form_login');

            $this->load->view('templates/footer');

        }else{

            $auth = $this->model_auth->cek_login();

            if($auth == FALSE)

            {

                $this->session-
>set_flashdata('pesan','<div
                                class="alert alert-
danger alert-dismissible fade show"

```

```

                                role="alert">

                                Username atau Password

Anda Salah!

                                <button    type="button"
class="close" data-dismiss="alert" aria-label="Close">

                                <span                                aria-
hidden="true">&times;</span>

                                </button>

                                </div>');

                                redirect('auth/login');

                                }else{

                                $this->session-
>set_userdata('username',$auth->username);

                                $this->session-
>set_userdata('role_id',$auth->role_id);

                                switch($auth->role_id){

                                case            1            :

                                redirect('admin/dashboard_admin');

                                break;

```

```

                                case      2      :
redirect('dashboard');

                                break;

                                default: break;

                                }

                                }

                                }

}

public function logout(){

    $this->session->sess_destroy();

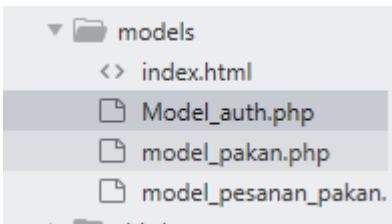
    redirect('auth/login');

}

}

```

Setelah itu tambahkan models untuk login.



Pada models login berikut source codenya;

```
<?php

/**
 *
 */

class Model_auth extends CI_Model

{

    public function cek_login()

    {

        $username = set_value('username');

        $password = set_value('password');

        $result = $this->db->where('username',$username)
```

```

>where('password',$password)
-
->limit(1)
-

>get('db_user');

        if($result->num_rows() > 0){

            return $result->row();

        }else{

            return array();

        }

    }

}

```

Tambahkan source code pada view:

```

<body class="bg-gradient-primary">

<div class="container">

```



```
<form method="post" action="<?php echo
base_url('auth/login') ?>" class="user">

    <div class="form-group">

        <input type="text" class="form-control form-
control-user" id="exampleInputEmail" aria-
describedby="emailHelp" placeholder="Masukkan username
Anda..." name="username">

        <?php echo form_error('username', '<div
class="text-danger small ml-2">','</div'); ?>

    </div>

    <div class="form-group">

        <input type="password" class="form-control
form-control-user" id="exampleInputPassword"
placeholder="Masukkan Password Anda..." name="password">

        <?php echo form_error('password', '<div
class="text-danger small ml-2">','</div'); ?>

    </div>

    <div class="form-group">

        <div class="custom-control custom-checkbox
small">
```

```
        <input type="checkbox" class="custom-control-
input" id="customCheck">

        </div>

    </div>

    <button type="submit" class="btn btn-primary
form-control">

        Login

    </button>

</form>

<hr>

<div class="text-center">

    <a class="small" href="<?php echo
base_url('registrasi/index'); ?>">Silahkan Buat Akun!</a>

</div>

</div>

</div>

</div>

</div>

</div>

</div>
```



```
</div>
```

```
</div>
```

```
</div>
```

```
<!-- Bootstrap core JavaScript-->
```

```
<script src="vendor/jquery/jquery.min.js"></script>
```

```
<script  
src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
```

```
<!-- Core plugin JavaScript-->
```

```
<script                                src="vendor/jquery-  
easing/jquery.easing.min.js"></script>
```

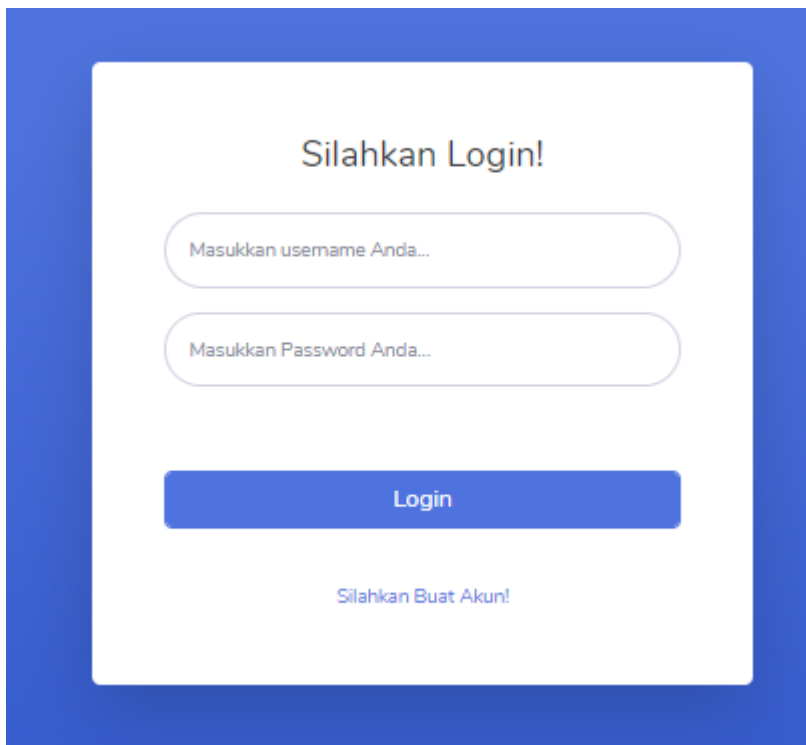
```
<!-- Custom scripts for all pages-->
```

```
<script src="js/sb-admin-2.min.js"></script>
```

```
</body>
```

```
</html>
```

Pertama kita akan membuka [http://localhost/aplikasi\\_pakan/index.php/auth/login/](http://localhost/aplikasi_pakan/index.php/auth/login/), dan akan muncul tampilan login sebagai berikut :

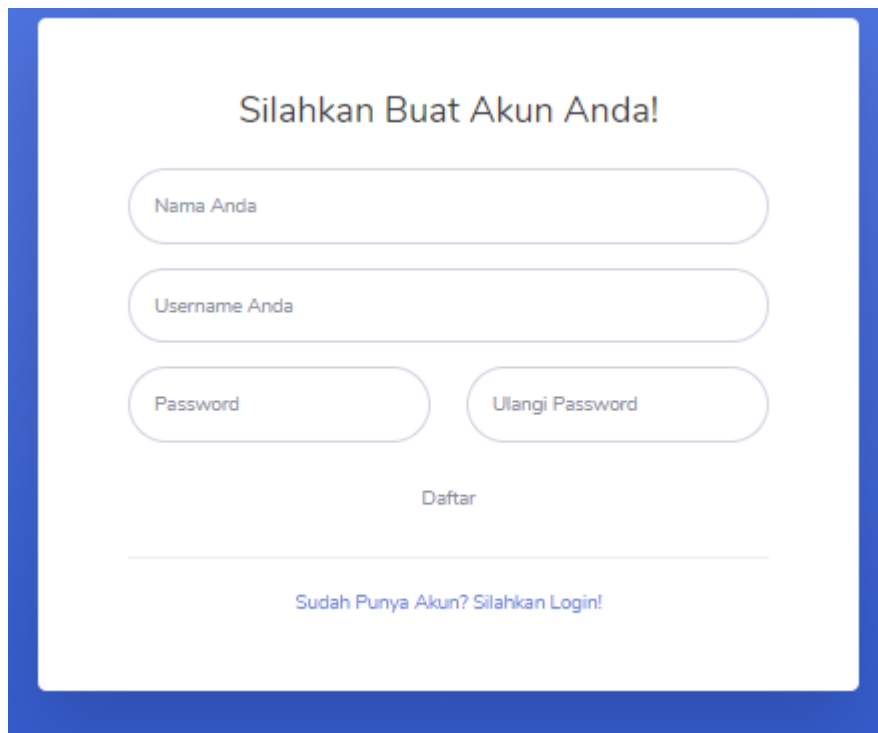


The image shows a login form with a blue border. The title is "Silahkan Login!". Below the title are two input fields: "Masukkan username Anda..." and "Masukkan Password Anda...". Below these fields is a blue button labeled "Login". At the bottom of the form is a link that says "Silahkan Buat Akun!" in blue text.

Gambar di atas merupakan tampilan awal login.

Setelah itu kita akan membuat register admin dan Pegawai.

Ketika Pegawai yang ingin masuk tetapi tidak memiliki akun maka pegawai akan register terlebih . Maka kita buka [http://localhost/aplikasi\\_pakan/registrasi/index](http://localhost/aplikasi_pakan/registrasi/index) maka akan tampil halaman seperti gambar dibawah ini



The image shows a registration form with a blue border. The title is "Silahkan Buat Akun Anda!". Below the title are four input fields: "Nama Anda", "Username Anda", "Password", and "Ulangi Password". Below these fields is a "Daftar" button. At the bottom, there is a link that says "Sudah Punya Akun? Silahkan Login!".

Source code pada register ada Controller .

```
<?php
```

```
class Registrasi extends CI_Controller{
```

```
    public function index()
```

```
    {
```

```
        $this->form_validation-  
>set_rules('nama', 'Nama', 'required',  
['required' => 'Nama Wajib Diisi!']);  
        $this->form_validation-  
>set_rules('username', 'Username',  
'required', ['required' => 'Username  
Wajib Diisi!']);  
        $this->form_validation-  
>set_rules('password_1', 'Password',  
'required|matches[password_2]', [  
            'required' =>  
'Password Wajib Diisi!',  
            'matches'=> 'Password  
Tidak Cocok!'  
        ]);  
        $this->form_validation-  
>set_rules('password_2', 'Password',  
'required|matches[password_1]');  
  
        if($this->form_validation-  
>run() == FALSE) {  
            $this->load-  
>view('templates/header');  
            $this->load-  
>view('registrasi');
```

```

        $this->load-
>view('templates/footer');
    } else {
        $data = array(
            'id' => '',
            'nama' => $this-
>input->post('nama'),
            'username' =>
$this->input->post('username'),
            'password' =>
$this->input->post('password_1'),
            'role_id' => 2,
        );

        $this->db->insert('db_user',
$data);
        redirect('auth/login');
    }
}
}

```

Setelah itu pada views,tambahkan source code berikut.

```
<body class="bg-gradient-primary">
```



```
<?php echo form_error('nama',  
'<div class="text-danger small ml-2">',  
'</div>') ?>
```

```
</div>
```

```
<div class="form-group">  
    <input type="text" class="form-  
control form-control-user"  
id="exampleInputEmail"  
placeholder="Username Anda"  
name="username">
```

```
<?php echo  
form_error('username', '<div class="text-  
danger small ml-2">', '</div>') ?>  
</div>
```

```
<div class="form-group row">  
    <div class="col-sm-6 mb-3 mb-  
sm-0">
```

```
        <input type="password"  
class="form-control form-control-user"  
id="exampleInputPassword"  
placeholder="Password" name =  
"password_1">
```

```
<?php echo  
form_error('password_1', '<div class="text-  
danger small ml-2">', '</div>') ?>  
</div>
```

```
<div class="col-sm-6">  
    <input type="password"  
class="form-control form-control-user"  
id="exampleRepeatPassword"  
placeholder="Ulangi Password"  
name="password_2">  
</div>  
</div>
```

```
<button type="submit"  
class="btn btn-primary btn-user btn-  
block">Daftar</button>
```

```
</form>  
<hr>  
<div class="text-center">  
    <a class="small" href="<?php  
echo base_url('auth/login') ?>">Sudah  
Punya Akun? Silahkan Login!</a>
```





Login pada aplikasi Pakan KPSBU sebagai berikut.

ADMIN

Dashboard

Data Master Pakan

Data Stok Pakan

Data Pesanan Pakan

Search for...

q

Selamat Datang admin

Logout

Input Data Pesanan Pakan

NO	ID PAKAN	JENIS PAKAN	TANGGAL KELUAR	JUMLAH	NAMA PEMESAN	ALAMAT PEMESAN		
1	1101	konsertrat 2	2019-12-03	12	arifji	lembang		
2	1103	konsertrat 12	2019-12-11	12	ade	lembang		
3	1104	konsertrat 1	2020-01-09	12	ade	bandung		





















































































































































