1. PRAKTIKUM
   1. Alat dan Bahan

Alat dan bahan yang harus disiapkan sebelum praktikum :

1. Bootstrap 4.4
2. SB-Admin 2
3. Jquery
4. *Web* *Browser*(Chrome, Mozila Firefox, Safari)
5. *Web* *Server* yang mendukung Apache dan MySql (XAMPP, Laragon, WAMP, MAMP)
6. Codeigniter 3.11
7. Composer (*optional*)
   1. Praktikum
      1. Konfigurasi Codeigniter

Setelah mendownload file codeigniter ganti namanya menjadi ci-kosan. Pada Codeigniter diperlukan beberapa konfigurasi. Buka folder ci-kosan menggunakan kode editor masing-masing, disini saya menggunakan Visual Studio Code.

* + - 1. Pada Folder ci-kosan/application/autoload

1. Cari array assosiatif auto-load libraries.

$autoload['libraries'] = array();

Kode 1 : Auto-Load Library

1. Lalu tambahkan session, database, form\_validation didalam *method* array agar dapat menggunakan *library* tersebut seperti gambar 4.2

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

Kode 2 : Panggil Library CI

1. Cari array assosiatif auto-load helper dan isi array dengan url, file, security, is\_login. Agar developer dapat memanggil function yang ada di dalam helper.

$autoload['helper'] = array('url', 'file', 'security', 'is\_login');

Kode 3 : Panggil Helper

* + - 1. Konfigurasi pada file ci-kosan/application/config

1. Cari array assosiatif base site url, lalu isikan http://localhost/ci-kosan/

$config['base\_url'] = 'http://localhost/ci-kosan/';

Kode 4 : Config Base Url

Keterangan :

* Gambar 4.5 merupakan array assosiatif yang akan mengirimkan string ke method base\_url, lalu method base\_url akan berisi string tersebut.

1. Cari array assosiatif index\_page lalu isi blank atau kosong didalam kutip

$config['index\_page'] = '';

Kode 5 : Config Index Page

* + - 1. Pada ci-kosan/application/database

1. Cari array assosiatif db default lalu isikan array hostname   
   = localhost, username = root, password kosongkan, database   
   = ci\_kosan.

$db['default'] = array(

'dsn'   => '',

'hostname' => 'localhost',

'username' => 'root',

'password' => '',

'database' => 'ci\_kosan',

'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

);

Kode 6 : Konfigurasi Database CI

* + - 1. Pada Folder ci-kosan/application/config/routes

1. Cari array assosiatif router default\_controller disii dengan string user

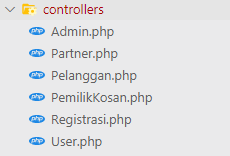
$route['default\_controller'] = 'user';

Kode 7 : Route Default Controller

Keterangan :

* $route['default\_controller'] berfungsi agar yang pertama kali dijalankan adalah *controller* *user*.
  + 1. Struktur Folder
       1. Folder Controllers

Pada **ci-kosan/application/controlers**, folder controllers akan berisi file role user(Admin, Pelanggan, Pemilik Kosan dan Partner) dan proses Registrasi.



Gambar 4.1 Struktur Folder Controllers

Buatlah file seperti yang tertera pada gambar. Untuk membuat controller ada beberapa aturan penamaan yang telah diterapkan oleh Codeigniter.

Struktur Controller :

<?php

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

class NamaController extends CI\_Controller

{

    public function \_\_construct()

    {

        parent::\_\_construct();

    }

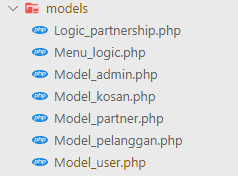
}

Kode 8 : Struktur Controller

Keterangan :

* Huruf pertama nama file controller harus huruf besar atau kapital, tidak boleh menggunakan spasi dan simbol.
* Nama class harus sama dengan nama file controller
* Sintak php boleh ditutup, boleh juga tidak.
* defined('BASEPATH') or exit('No direct script access allowed'); agar user jahat tidak bisa mengakses lewat url.
* Function \_\_construct(), digunakan agar developer tidak perlu memanggil code program(helper, library dan lain- sebagainya) berulang kali. Cukup menuliskan di function \_\_construct() maka helper dan library tersebut dapat dipanggil di function mana saja selama dalam satu controller.
  + - 1. Folder Models

Folder Model yang berada di **ci-kosan/application/models**, akan berisi role seperti controller, logika partnership dan logika menu.



Gambar 4.2 : Struktur Folder Models

Buat file, sesuaikan dengan gambar 4.10. Sebenarnya struktuk controller dan model itu sama aja, bahkan aturan penamaanya sama, yang membedakannya hanya perluasannya saja. Controller merupakan perluasan dari CI\_Controller sedangkan model perluasan dari CI\_Model.

Struktur Models :

<?php

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

class NamaModel extends CI\_Model

{

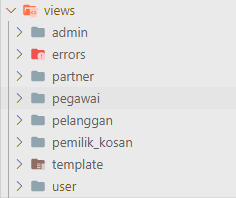
}

Kode 9 : Struktur Model

Keterangan :

* Penamaan Controller dan Model sama saja.
* Huruf pertama nama file model harus huruf besar atau kapital, tidak boleh menggunakan spasi dan simbol.
* Nama class harus sama dengan nama file model
* Sintak php boleh ditutup, boleh juga tidak.
* defined('BASEPATH') or exit('No direct script access allowed'); agar user jahat tidak bisa mengakses model lewat url.
* Biasanya nama *Model* dan *Controller* hampir sama dengan ketentuan jika controller itu mempunyai sesuatu hubungan.
  + - 1. Folder *Views*

Folder Views **ci-kosan/application/views**, berisi subfolder dari role user dan template(footer dan header). Agar developer tidak membuat halaman HTML yang berulang seperti memasukkan file css (lihat pada gambar 4.11) dan javascript (lihat gambar 4.13) diperlukan template, sehingga developer hanya perlu memanggil template tersebut di controller, template biasanya terbagi menjadi beberapa bagian, sesuai dengan keperluan.



Gambar 4.3 Struktur Folder Views

* + 1. Membuat Template *User*

Template user terbagi menjadi dua, yaitu header dan juga footer. Template user digunakan ketika halaman pertama kali diakses.

* + - 1. Header User

*Header user* merupakan elemen HTML yang berada di atas, biasanya berupa navbar yang menampilkan menu.

* + - * 1. Memasukkan CSS pada Header User

1. Buat sebuah file bernama header\_user.php pada folder   
   **ci-kosan/application/views/templates.**
2. Buat sebuah tag HTML 5 lalu sisipkan file css yang diperlukan diantara tag head dan penutup head seperti yang tertera pada gambar 4.11.

<head>

  <title> <?= $title; ?> </title>

  <link href="<?= base\_url('assets'); ?>/vendor/fontawesome-free/css/all.min.css" rel="stylesheet" type="text/css">

  <link href="<?= base\_url('assets'); ?>/css/sb-admin-2.min.css" rel="stylesheet">

  <link href="<?= base\_url('assets'); ?>/css/mystyle.css" rel="stylesheet">

  <link href="<?= base\_url('assets'); ?>/vendor/datatables/dataTables.bootstrap4.min.css" rel="stylesheet">

</head>

Kode 10 : Include CSS

Keterangan :

* Tag link berfungsi untuk memanggil file css.
* Sintak akan menerima string yang dikirim dari *controller* *user,* tergantung pada function, karena setiap page memiliki title yang berbeda.
  + - * 1. Navbar Header User

Membuat Navbar user, ketik

1. Tuliskan script dibawah tag penutup head(gambar 4.13).script berikut adalah body dari tag HTML header user.

<body>

  <nav class="navbar navbar-expand-sm navbar-dark fixed-top" id="scroll\_navbar">

    <div class="container-fluid">

      <a class="navbar-brand " href="#"><strong>JURAGAN KOSAN</strong></a>

      <strong>

        <ul class="navbar-nav">

          <li class="nav-item active">

            <a class="nav-link mr-1  fas fa-home" href="<?= base\_url() ?>"> Home<span class="sr-only"></span></a>

          </li>

          <li class="nav-item">

            <a class="nav-link mr-1  fas fa-building " href="<?= base\_url('user/listKosan') ?>"> Kosan</a>

          </li>

          <li class="nav-item">

            <a class="nav-link mr-1  fas fa-warehouse " href="<?= base\_url('user/listpartner') ?>"> Partner</a>

          </li>

          <li class="nav-item">

            <a class="nav-link mr-1 fas fa-sign-in-alt " href="<?= base\_url('user/login') ?>"> Login</a>

          </li>

          <li class="nav-item dropdown">

            <a class="nav-link fas fa-users mr-3" href="#" id="navbarDropdown" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

              Registrasi

            </a>

            <div class="dropdown-menu" aria-labelledby="navbarDropdown">

              <a class="dropdown-item" href="<?= base\_url('Registrasi/user') ?>">Pelanggan</a>

              <div class="dropdown-divider"></div>

              <a class="dropdown-item" href="<?= base\_url('Registrasi/partner') ?>">Partner</a>

              <div class="dropdown-divider"></div>

              <a class="dropdown-item" href="<?= base\_url('Registrasi/pemilikKosan') ?>">Pemilik Kosan</a>

            </div>

          </li>

        </ul>

      </strong>

    </div>

  </nav>

Kode 11 : Header User

* + - 1. Footer User

Footer user berisi file-file javascript dan function yang dibuat dengan javascript disimpan di footer.

1. Masukkan semua file javascript yang akan dipakai.

</body>

<script src="<?= base\_url('assets'); ?>/vendor/jquery/jquery.min.js"></script>

<script src="<?= base\_url('assets'); ?>/js/sb-admin-2.min.js"></script>

<script src="<?= base\_url('assets'); ?>/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<script src="<?= base\_url('assets'); ?>/vendor/jquery-easing/jquery.easing.min.js"></script>

Kode 12 : Insert file javascript

1. Membuat sebuah function scrollNavbar(). Ketika window/halaman di *scroll,* akan menambahkan sebuah kelas, yang berguna untuk memberi warna pada navbar.

  document.addEventListener("scroll", scrollNavbar);

  function scrollNavbar() {

    $(window).scroll(function() {

      if ($(this).scrollTop() > 300) {

        $('#scroll\_navbar').addClass('my-color')

      } else {

        $('#scroll\_navbar').removeClass('my-color')

      }

    })

  }

Kode 13 : Scroll Window

Keterangan :

* Sintak document.addEventListener("scroll", scrollNavbar); Apabila terjadi *event* *scroll* yang terjadi pada window/halaman *web*.
* Logika if, jika user scroll window pada titik y lebih dari 300 maka akan menambahkan kelas my-color, else kurang dari 300 hilangkan kelas my-color.
  + - 1. Folder Assets

Pada ci-kosan buatlah sebuah folder assets **ci-kosan/assets**, folder assets akan menyimpan css, javascript, dan juga foto.

* **Folder CSS**

Buatlah sebuah folder yang bernama **css**(huruf kecil/lowercase). Folder css akan berisi semua css maupun itu css bootstrap atau css yang dbuat sendiri.

* **Folder Image**

Buat sebuah folder yang nama **img**(lowercase), folder ini akan berisi semua foto yang dimasukkan oleh user seperti, bukti pembyaran, foto faktur, kosan, profile, ktp, dan lain sebagainya.

* **Folder Javascript**

Seperti halnya folder css, folder js berisi semua hal yang berkaitan dengan javascript.

* **Folder SCSS**

SCSS merupakan salah satu CSS preprocessor yang dapat memudahkan dalam penulisan sintak.

* **Folder Vendor**

Folder **vendor** berisi semua fitur-fitur yang ada di sb-admin 2 seperti bootstrap, chart.js, datatables, fontawesome, jquery, dan jquery-easing.

* + 1. Tampilan *Home* *User*

Home User adalah halaman HTML ketika pertama kali web diakses. Pada **ci-kosan/application/views/user**, buat sebuah file **index.php**.

* + - 1. Carousel dengan bootstrap

Carousel adalah elemen HTML yang berjalan secara otomatis atau bisa digerakkan oleh user dengan menekan tombol. Biasanya berupa beberapa foto, ketika diklik atau dibiarkan beberapa waktu, foto tersebut akan bergerak dari kiri ke kanan.

Pada file views **user/index.php** ketikkan script berikut :

<div class="row mr-0">

  <div class="bd-example">

    <div id="carouselExampleCaptions" class="carousel slide" data-ride="carousel">

      <ol class="carousel-indicators">

        <li data-target="#carouselExampleCaptions" data-slide-to="1" class="active"></li>

        <li data-target="#carouselExampleCaptions" data-slide-to="2"></li>

        <li data-target="#carouselExampleCaptions" data-slide-to="3"></li>

      </ol>

      <div class="carousel-inner">

        <div class="carousel-item active">

          <img src="<?= base\_url('assets/img/background.jpg') ?>" width="100%" height="450px" class="d-block w-100" alt="...">

          <div class="carousel-caption d-none d-md-block">

            <h5>First slide label</h5>

            <p>Nulla vitae elit libero, a pharetra augue mollis interdum.</p>

          </div>

        </div>

        <div class="carousel-item">

          <img src="<?= base\_url('assets/img/background2.jpg') ?>" width="100%" height="450px" class="d-block w-100" alt="...">

          <div class="carousel-caption d-none d-md-block">

            <h5>Second slide label</h5>

            <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>

          </div>

        </div>

        <div class="carousel-item">

          <img src="<?= base\_url('assets/img/background3.jpg') ?>" width="100%" height="450px" class="d-block w-100" alt="...">

          <div class="carousel-caption d-none d-md-block">

            <h5>Third slide label</h5>

            <p>Praesent commodo cursus magna, vel scelerisque nisl consectetur.</p>

          </div>

        </div>

      </div>

      <a class="carousel-control-prev" href="#carouselExampleCaptions" role="button" data-slide="prev">

        <span class="carousel-control-prev-icon" aria-hidden="true"></span>

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

      </a>

      <a class="carousel-control-next" href="#carouselExampleCaptions" role="button" data-slide="next">

        <span class="carousel-control-next-icon" aria-hidden="true"></span>

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

      </a>

    </div>

  </div>

</div>

Kode 14 : Foto Slide

* + - 1. Card Berbagai Kosan pada Home User

Menyediakan elemen HTML yang nantinya akan menampung card berbagai kosan.

<div class="container text-dark mt-3 mb-5">

  <div class="row">

    <div class="col my-3">

      <h5>

        <i class="fa fa-building"></i> Menu

        <i class="fa fa-angle-right"></i> Berbagai Kosan

      </h5>

    </div>

  </div>

  <div class="row justify-content-start">

*<!— Card Berbagai Kosan -->*

  </div>

</div>

Kode 15 : Container Kosan

Keterangan :

* Pada tahap ini hanya kerangkanya saja, karena data kosan belum ada.
* Card ada diletakkan pada baris komentar “Card Berbagai Kosan”.
  + 1. Menu Management

Menu Management berguna untuk mengelola semua menu pada user. Menu management ini memakai *database*, sehingga developer hanya perlu memanggil tabel yang diperlukan

1. Nyalakan MySql di aplikasi xampp, dengan cara klik start pada module mysql
2. Ketikkan url berikut di web browser <http://localhost/phpmyadmin/>
3. Untuk membuat *database* baru, klik tombol baru pada *sidebar* sebelah kiri. Lalu beri nama ci\_kosan.
4. Buat empat tabel, dengan nama user, user\_role, user\_menu, user\_access\_menu.
   * + 1. Tabel user

Tabel user merupakan tabel yang menyimpan data username, password, dan role dari user. Role user akan dijelaskan pada tabel user\_role. Struktur tabel sebagai berikut :



Gambar 4.4 Struktur Tabel user

Keterangan :

* user\_id merupakan primary key yang di set auto\_increment agar tidak ada data yang redudansi.
* role\_id merupakan foreign key dari tabel user role.
* Tabel memang dibiarkan kosong karena tabel akan diisi saat user menginputkan data.

Ketikkan kueri pada menu SQL lalu run, maka mysql akan langsung mengenerate kueri menjadi sebuah tabel.

Query create tabel user :

CREATE TABLE `user` (

`user\_id` int(11) NOT NULL,

`username` varchar(128) NOT NULL,

`password` varchar(128) NOT NULL,

`role\_id` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `user`

ADD PRIMARY KEY (`user\_id`);

ALTER TABLE `user`

MODIFY `user\_id` int(11) NOT NULL AUTO\_INCREMENT;

* + - 1. Tabel user\_role

Tabel user\_role merupakan tabel yang menyimpan data role user, role terbagi menjadi empat, yaitu Admin, Partner, Pemilik Kosan, dan Pelanggan. Struktur tabel user\_role :



Kode 16 : Struktur Tabel user\_role

Query create tabel user\_role :

CREATE TABLE `user\_role` (

`role\_id` int(11) NOT NULL,

`role` varchar(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

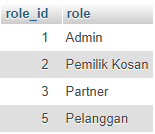
ALTER TABLE `user\_role`

ADD PRIMARY KEY (`role\_id`);

ALTER TABLE `user\_role`

MODIFY `role\_id` int(11) NOT NULL AUTO\_INCREMENT;

Data tabel user\_role :



Kode 17 : Data Tabel user\_role

Query insert tabel user\_role :

INSERT INTO `user\_role` (`role\_id`, `role`) VALUES

(1, 'Admin'),

(2, 'Pemilik Kosan'),

(3, 'Partner'),

(5, 'Pelanggan');

Keterangan :

* role\_id nomor 4 memang tidak ada karena saya telah memasukkan data dan menghapusnya. Field role\_id merupakan auto increment, jadi secara otomatis role\_id akan terus bertambah. Anda dapat mengabaika role\_id nomor 4, karena tidak ada pengaruhnya.
* Anda bisa mengganti role\_id pelanggan menjadi nomor 4, tetapi saat pengkodingan nanti diharapkan lebih teliti, karena saya tetap menggunakan role\_id nomor 5 untuk pelanggan.
  + - 1. Tabel user\_menu

Tabel user\_menu adalah tabel yang berisi menu user, sesuai dengan rolenya masing-masing, sehingga developer tidak perlu membuat halaman HTML yang berbeda-beda untuk setiap role user. Field menu berisikan nama-nama menu yang akan ditampilkan, field url adalah link dari menu atau ketika diklik kemana tujuan dari menu tersebut, dan icon adalah simbol dari menu.

Struktur tabel user\_menu :



Gambar 4.5 Struktur Folder user\_menu

Query create tabel user\_menu :

CREATE TABLE `user\_menu` (

`menu\_id` int(11) NOT NULL,

`menu` varchar(30) NOT NULL,

`url` varchar(30) NOT NULL,

`icon` varchar(128) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

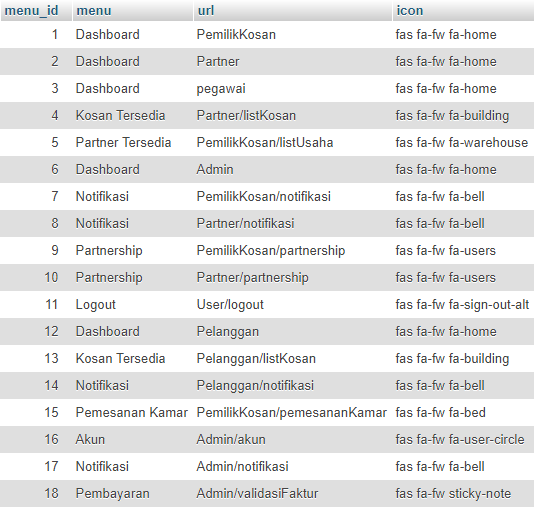
ALTER TABLE `user\_menu`

ADD PRIMARY KEY (`menu\_id`);

ALTER TABLE `user\_menu`

MODIFY `menu\_id` int(11) NOT NULL AUTO\_INCREMENT;

Data pada tabel user\_menu : Ubah lagi datanya



Gambar 4.6 Data Tabel user\_menu

Query Insert tabel user\_menu : Ubah lagi datanya

INSERT INTO `user\_menu` (`menu\_id`, `menu`, `url`, `icon`) VALUES

(1, 'Dashboard', 'PemilikKosan', 'fas fa-fw fa-home'),

(2, 'Dashboard', 'Partner', 'fas fa-fw fa-home'),

(3, 'Dashboard', 'pegawai', 'fas fa-fw fa-home'),

(4, 'Kosan Tersedia', 'Partner/listKosan', 'fas fa-fw fa-building'),

(5, 'Partner Tersedia', 'PemilikKosan/listUsaha', 'fas fa-fw fa-warehouse'),

(6, 'Dashboard', 'Admin', 'fas fa-fw fa-home'),

(7, 'Notifikasi', 'PemilikKosan/notifikasi', 'fas fa-fw fa-bell'),

(8, 'Notifikasi', 'Partner/notifikasi', 'fas fa-fw fa-bell'),

(9, 'Partnership', 'PemilikKosan/partnership', 'fas fa-fw fa-users'),

(10, 'Partnership', 'Partner/partnership', 'fas fa-fw fa-users'),

(11, 'Logout', 'User/logout', 'fas fa-fw fa-sign-out-alt'),

(12, 'Dashboard', 'Pelanggan', 'fas fa-fw fa-home'),

(13, 'Kosan Tersedia', 'Pelanggan/listKosan', 'fas fa-fw fa-building'),

(14, 'Notifikasi', 'Pelanggan/notifikasi', 'fas fa-fw fa-bell'),

(15, 'Pemesanan Kamar', 'PemilikKosan/pemesananKamar', 'fas fa-fw fa-bed'),

(16, 'Akun', 'Admin/akun', 'fas fa-fw fa-user-circle'),

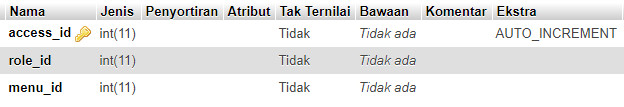
(17, 'Notifikasi', 'Admin/notifikasi', 'fas fa-fw fa-bell'),

(18, 'Pembayaran', 'Admin/validasiFaktur', 'fas fa-fw fa-sticky-note');

* + - 1. Tabel user\_access\_menu

Tabel user\_acees\_menu merupakan tabel gabungan dari tabel user dan tabel user\_menu, tabel user\_access\_menu berfungsi agar user tidak dapat mengakses menu user lain.

Struktur Tabel :



Gambar 4.7 Struktur Tabel user\_access\_menu

*Query* create tabel user\_access\_menu : Ubah lagi datanya

CREATE TABLE `user\_access\_menu` (

`access\_id` int(11) NOT NULL,

`role\_id` int(11) NOT NULL,

`menu\_id` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

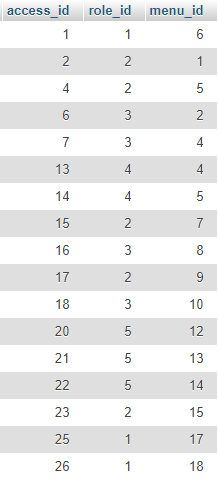
ALTER TABLE `user\_access\_menu`

ADD PRIMARY KEY (`access\_id`);

ALTER TABLE `user\_access\_menu`

MODIFY `access\_id` int(11) NOT NULL AUTO\_INCREMENT;

Data tabel user\_access\_menu :



Gambar 4.8 Data Tabel user\_access\_menu

*Query* insert tabel user\_access\_menu :

INSERT INTO `user\_access\_menu` (`access\_id`, `role\_id`, `menu\_id`) VALUES (1, 1, 6), (2, 2, 1), (4, 2, 5), (6, 3, 2), (7, 3, 4), (13, 4, 4), (14, 4, 5), (15, 2, 7), (16, 3, 8), (17, 2, 9), (18, 3, 10), (20, 5, 12), (21, 5, 13), (22, 5, 14), (23, 2, 15), (25, 1, 17), (26, 1, 18), (27,1,19);

* + 1. Template Admin, Partner, Pelanggan ,dan Pemilik Kosan

Role user Admin, Partner, Pelanggan dan Pemilik Kosan menggunakan template sb-admin2, jadi semua role tersebut memiliki header, sidebar dan topbar yang sama, hanya footernya saja yang berbeda.

* + - 1. Header

1. Pada Folder **ci-kosan/application/views/template,** buatlah file bernama header.php.
2. Beri jarak apabila membingungkan untuk dibaca.

<!DOCTYPE html>

<html lang="en">

<head>

  <title>

    <?= $title; ?>

  </title>

*<!-- Custom fonts for this template -->*

  <link href="<?= base\_url('assets'); ?>/vendor/fontawesome-free/css/all.min.css" rel="stylesheet" type="text/css">

  <link rel="stylesheet" type="text/css" href="<?php echo base\_url('assets'); ?>/css/bootstrap-select.css">

*<!-- Custom styles for this template -->*

  <link href="<?= base\_url('assets'); ?>/css/sb-admin-2.min.css" rel="stylesheet">

*<!-- Custom styles for this page -->*

  <link href="<?= base\_url('assets'); ?>/vendor/datatables/dataTables.bootstrap4.min.css" rel="stylesheet">

  <link href="<?= base\_url('assets'); ?>/css/mystyle.css" rel="stylesheet">

</head>

Kode 18 : Header

Keterangan :

* Pada tahap ini, header.php adalah file yang meload semua file CSS.
  + - 1. Topbar

Topbar berada pada atas window/halaman. Topbar berisi foto dan username user yang sedang login. Ketika user menekan foto akan muncul dropdown yang berisi function untuk logout dan link profile.

1. Pada Folder **ci-kosan/application/views/template,** buatlah file bernama **topbar.php.**
2. Lalu ketikkan kodingan berikut :

*<!-- Topbar -->*

<nav class="navbar navbar-expand navbar-light bg-white topbar static-top shadow">

*<!-- Topbar Navbar -->*

  <ul class="navbar-nav ml-auto">

*<!-- Nav Item - Search Dropdown (Visible Only XS) -->*

    <li class="nav-item dropdown no-arrow d-sm-none">

      <a class="nav-link dropdown-toggle" href="#" id="searchDropdown" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

        <i class="fas fa-search fa-fw"></i>

      </a>

*<!-- Nav Item - User Information -->*

    <li class="nav-item dropdown no-arrow">

      <a class="nav-link dropdown-toggle" href="#" id="userDropdown" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

        <span class="mr-2 d-none d-sm-inline text-gray-600 small">NamaUser</span>

        <img class="img-profile rounded-circle" src="<?= base\_url('assets/img/profile/default.jpg') ?>">

      </a>

*<!-- Dropdown - User Information -->*

      <div class="dropdown-menu dropdown-menu-right shadow animated--grow-in" aria-labelledby="userDropdown">

        <a class="dropdown-item" href="#">

          <i class="fas fa-user fa-sm fa-fw mr-2 text-gray-400"></i>

          Profile

        </a>

        <div class="dropdown-divider"></div>

        <a class="dropdown-item" href="#" data-toggle="modal" data-target="#logoutModal">

          <i class="fas fa-sign-out-alt fa-sm fa-fw mr-2 text-gray-400"></i>

          Logout

        </a>

      </div>

    </li>

  </ul>

</nav>

*<!-- End of Topbar -->*

Kode 19 : Topbar

* + - 1. Sidebar

Sidebar merupakan elemen HTML yang berada pada sisi window/halaman, biasanya terdapat pada sisi kiri dan menyimpan menu user.

1. Pada Folder **ci-kosan/application/views/template,** buatlah file bernama sidebar.php.

<body id="page-top">

*<!-- Page Wrapper -->*

  <div id="wrapper">

*<!-- Sidebar -->*

    <ul class="navbar-nav bg-gradient-primary sidebar sidebar-dark accordion" id="accordionSidebar">

*<!-- Sidebar - Brand -->*

      <a class="sidebar-brand d-flex align-items-center justify-content-center text-white">

        <div class=" sidebar-brand-icon rotate-n-15">

          <i class="fas fa-laugh-wink"></i>

        </div>

        <div class="sidebar-brand-text mx-auto">Juragan Kosan</div>

      </a>

      <hr class="sidebar-divider">

*<!-- Nav Item - Dashboard -->*

      <?php foreach ($menu as $m) : ?>

        <li class="nav-item active ">

          <a class="nav-link pt-0" href="<?= base\_url() ?><?= $m['url'] ?>">

            <i class="<?= $m['icon'] ?>"></i>

            <span><?= $m['menu'] ?></span>

            <?php if ($m['menu'] == "Notifikasi") : ?>

              <sup id="hitung\_notifikasi"></sup>

            <?php endif; ?>

          </a>

        </li>

        <hr class="sidebar-divider">

      <?php endforeach; ?>

      <li class="nav-item active ">

        <a class="nav-link pt-0" href="<?= base\_url('user/logout') ?>">

          <i class="fas fa-fw fa-sign-out-alt"></i>

          <span>Logout</span>

        </a>

      </li>

*<!-- Sidebar Toggler (Sidebar) -->*

      <div class="text-center d-none d-md-inline">

        <button class="rounded-circle border-0" id="sidebarToggle"></button>

      </div>

    </ul>

*<!-- End of Sidebar -->*

*<!-- Content Wrapper -->*

    <div id="content-wrapper" class="d-flex flex-column">

*<!-- Main Content -->*

      <div id="content">

Kode 20 : Sidebar

* + 1. Membuat Form Pendafataran Akun

Form pendaftaran berfungsi untuk menginputkan informasi Partner, Pelanggan dan Pemilik Kosan ke Database.

* + - 1. Halaman Registrasi Akun

Registrasi akun (Partner, Pelanggan dan Pemilik Kosan) untuk memiliki user interfaces yang sama, yang membedakannya hanya proses dibelakangnya, yaitu ketika menginputkan role user.

1. Pada folder **ci-kosan/application/views/user**, buatlah sebuah file bernama **registrasi\_akun.php**.
2. Script kode sebagai berikut,

<div class="position-bg load-pic-registrasi set-img text-white">

  <div class="container">

    <div class="row justify-content-center">

      <div class="col-md-8 p-4 mt-4">

        <h3 class="text-center text-light">

          <strong>Registrasi Akun <?= $role ?></strong>

          <hr class="mb-0">

        </h3>

        <?= $this->session->flashdata('message'); ?>

        <form action="" method="post">

          <div class="form-group-sm row">

            <div class="col-md-4">

              <label><strong>Username</strong></label>

              <input type="text" name="username" value="<?= set\_value('username') ?>" class="form-control form-control-sm" autocomplete="username">

              <?= form\_error('username', '<small class="text-danger ">', '</small>'); ?>

            </div>

            <div class="col-md-4">

              <label><strong>Password</strong></label>

              <input type="password" name="password" value="<?= set\_value('password') ?>" class="form-control form-control-sm" autocomplete="current-password">

              <?= form\_error('password', '<small class="text-danger ">', '</small>'); ?>

            </div>

            <div class="col-md-4">

              <label><strong>Nama Lengkap</strong></label>

              <input type="text" name="nama\_lengkap" value="<?= set\_value('nama\_lengkap') ?>" class="form-control form-control-sm">

              <?= form\_error('nama\_lengkap', '<small class="text-danger ">', '</small>'); ?>

            </div>

          </div>

          <div class="form-group-sm row mt-1">

            <div class="col-md-4">

              <label><strong>Tempat Lahir</strong></label>

              <input type="text" name="tempat\_lahir" value="<?= set\_value('tempat\_lahir') ?>" class="form-control form-control-sm">

              <?= form\_error('tempat\_lahir', '<small class="text-danger ">', '</small>'); ?>

            </div>

            <div class="col-md-4">

              <label><strong>Tanggal Lahir</strong></label>

              <input type="date" name="tanggal\_lahir" value="<?= set\_value('tanggal\_lahir') ?>" class="form-control form-control-sm" />

              <?= form\_error('tanggal\_lahir', '<small class="text-danger ">', '</small>'); ?>

            </div>

            <div class="col-md-4">

              <div class="row">

                <div class="col">

                  <label><strong>Jenis Kelamin</strong></label>

                </div>

              </div>

              <div class="row  ml-1">

                <?php $jenis\_kelamin = ""; ?>

                <div class="custom-control custom-radio custom-control-inline">

                  <input id="customRadioInline1" name="jenis\_kelamin" type="radio" class="custom-control-input <?php if ($jenis\_kelamin == 'Laki-Laki') echo "checked='checked'"; ?>" value="Laki-laki" />

                  <label class="custom-control-label" for="customRadioInline1">Laki-laki</label>

                </div>

                <div class="custom-control custom-radio custom-control-inline">

                  <input id="customRadioInline2" name="jenis\_kelamin" type="radio" class="custom-control-input <?php if ($jenis\_kelamin == 'Perempuan') echo "checked='checked'"; ?>" value="Perempuan" />

                  <label class="custom-control-label" for="customRadioInline2">Perempuan</label>

                </div>

                <?= form\_error('jenis\_kelamin', '<small class=" text-danger ">', '</small>'); ?>

              </div>

            </div>

          </div>

          <small>\*Tempat dan tanggal lahir harus sesuai dengan KTP</small>

          <div class="form-group-sm row">

            <div class="col-md-4">

              <label><strong>Pekerjaan</strong></label>

              <input type="text" name="pekerjaan" value="<?= set\_value('pekerjaan') ?>" class="form-control form-control-sm">

              <?= form\_error('pekerjaan', '<small class="text-danger ">', '</small>'); ?>

            </div>

            <div class="col-md-4">

              <label><strong>Nomor Hp</strong></label>

              <input type="text" name="nomor\_telepon" class="form-control form-control-sm" value="<?= set\_value('nomor\_telepon') ?>">

              <?= form\_error('nomor\_telepon', '<small class="text-danger ">', '</small>'); ?>

            </div>

          </div>

          <div class="form-group-sm row mt-1">

            <div class="col-md-6">

              <label><strong>Nama Jalan atau Gedung</strong></label>

              <input type="text" name="jalan" class="form-control form-control-sm" value="<?= set\_value('jalan') ?>">

              <?= form\_error('jalan', '<small class="text-danger ">', '</small>'); ?>

            </div>

            <div class="col-md-3">

              <label><strong>Provinsi</strong></label>

              <select selected class="form-control form-control-sm" id="provinsi" name="provinsi">

                <option selected disabled> - </option>

                <?php foreach ($provinsi as $p) : ?>

                  <option value="<?= $p['provinsi']; ?>"><?= $p['provinsi']; ?></option>

                <?php endforeach; ?>

              </select>

              <?= form\_error('provinsi', '<small class="text-danger ">', '</small><br>'); ?>

            </div>

            <div class="col-md-3">

              <label><strong>Kabupaten</strong></label>

              <select selected class="form-control form-control-sm" id="kabupaten" name="kabupaten">

                <option selected disabled> - </option>

              </select>

              <?= form\_error('kabupaten', '<small class="text-danger ">', '</small><br>'); ?>

            </div>

          </div>

          <small>\*Diharapkan untuk mengisikan alamat domisili atau alamat sekarang, alamat akan digunakan untuk rekomendasi kosan terdekat</small>

          <div class="form-group-sm row mt-1">

          </div>

          <div class="form-group-sm row mt-1">

            <div class="col-md-3">

              <label><strong>Kecamatan</strong></label>

              <select selected class="form-control form-control-sm" id="kecamatan" name="kecamatan">

                <option selected disabled> - </option>

              </select>

              <?= form\_error('kecamatan', '<small class="text-danger ">', '</small><br>'); ?>

            </div>

            <div class="col-md-3">

              <label><strong>Kelurahan</strong></label>

              <select selected class="form-control form-control-sm" id="kelurahan" name="kelurahan">

                <option selected disabled> - </option>

              </select>

              <?= form\_error('kelurahan', '<small class="text-danger ">', '</small><br>'); ?>

            </div>

          </div>

          <button type="submit" class="form-control-sm col-md-4 mt-2 float-right btn btn-primary text-light btn-user btn-block">

            <strong>Registrasi</strong>

          </button>

        </form>

      </div>

    </div>

  </div>

</div>

Kode 21 : Halaman Registrasi

Keterangan :

* <?= $role ?> berfungsi untuk mengubah agar tulisan tersebut dinamis(bisa berubah-ubah) sesuai dengan role pendafataran yang diinginkan user.
* Method set\_value() digunakan agar user tidak perlu mengisi kembali ketika sudah menginputkan, kecuali form tersebut di refresh.
* form\_error() berfungsi untuk menampilkan error apabila terdapat form yang tidak mengikuti aturan yang telah ditetapkan.
* Sintak HTML <div class="position-bg load-pic-registrasi set-img text-white">, berfungsi untuk mengatur background gambar.
* Berikut adalah script css yang digunakan untuk membuat background, script ini berada pada file mystyle.css di folder css

.set-img {

    height: 100%;

    width: 100%;

}

.load-pic-registrasi {

    background-image: url(images/video.jpg);

}

.position-bg {

    min-height: 380px;

    background-position: center;

    background-repeat: no-repeat;

    background-size: cover;

    position: relative;

}

Kode 22 : MyStyle Registrasi

* + - 1. Dropdown Alamat

Tabel alamat menampung berbagai alamat yang ada di Indonesia, tabel alamat akan dibuat sebagai dropdown chain, yaitu ketika salah satu menu di dropdown dipilih, menu pada dropdown yang berkaitan akan berubah sesuai dengan data yang disediakan. Untuk membuat dropdown alamat kita memerlukan.

* + - * 1. Tabel Provinsi

Tabel provinsi berisi semua provinsi yang ada di indonesia, disini hanya disediakan beberapa saja untuk contoh.

Query create tabel provinsi :

CREATE TABLE `provinsi` (

`id` int(11) NOT NULL,

`provinsi` varchar(128) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `provinsi`

ADD PRIMARY KEY (`id`);

Data tabel provinsi :



Gambar 4.9 Data Tabel provinsi

Query Insert tabel provinsi :

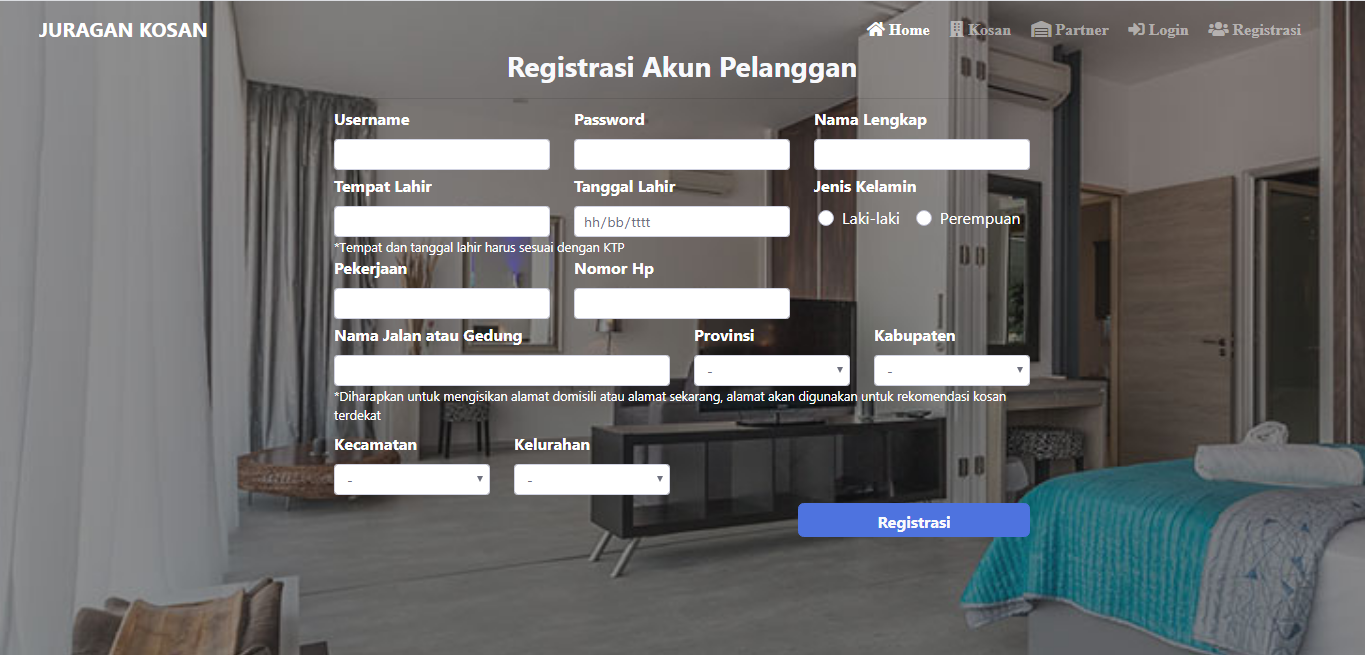
INSERT INTO `provinsi` (`id`, `provinsi`) VALUES

(1, 'Jawa Barat'),

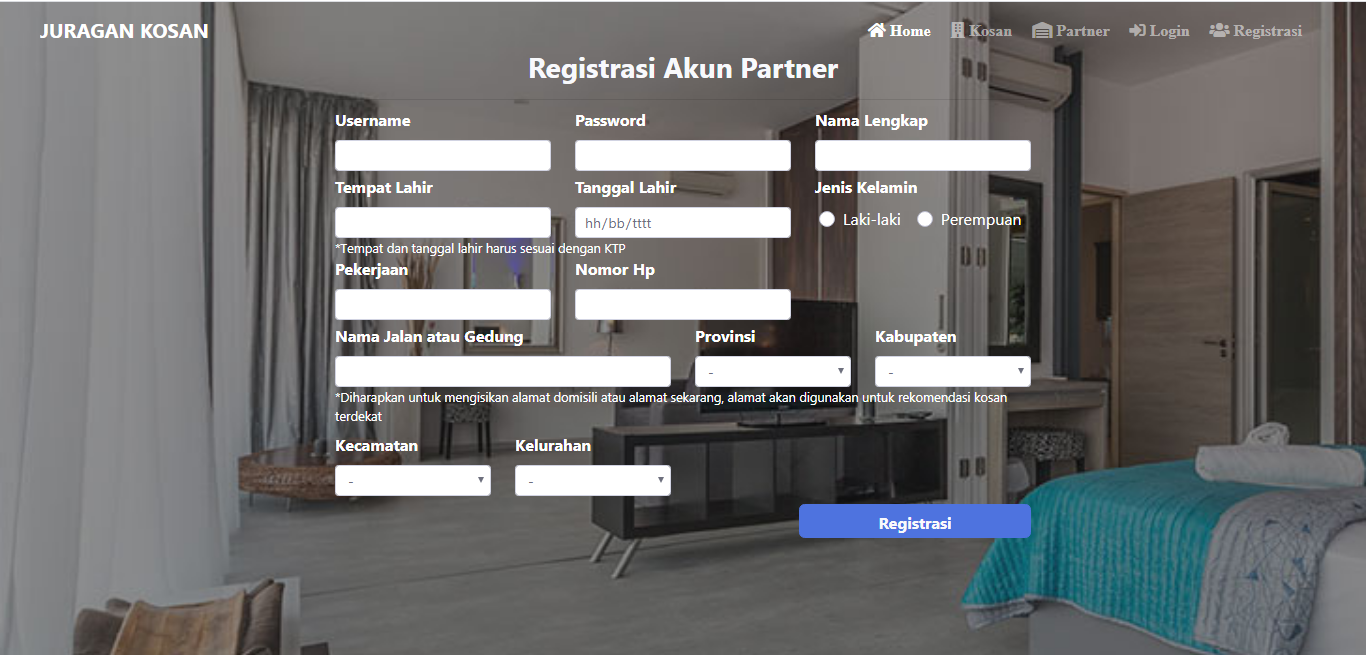
(2, 'Jawa Timur'),

(3, 'Jawa Tengah');

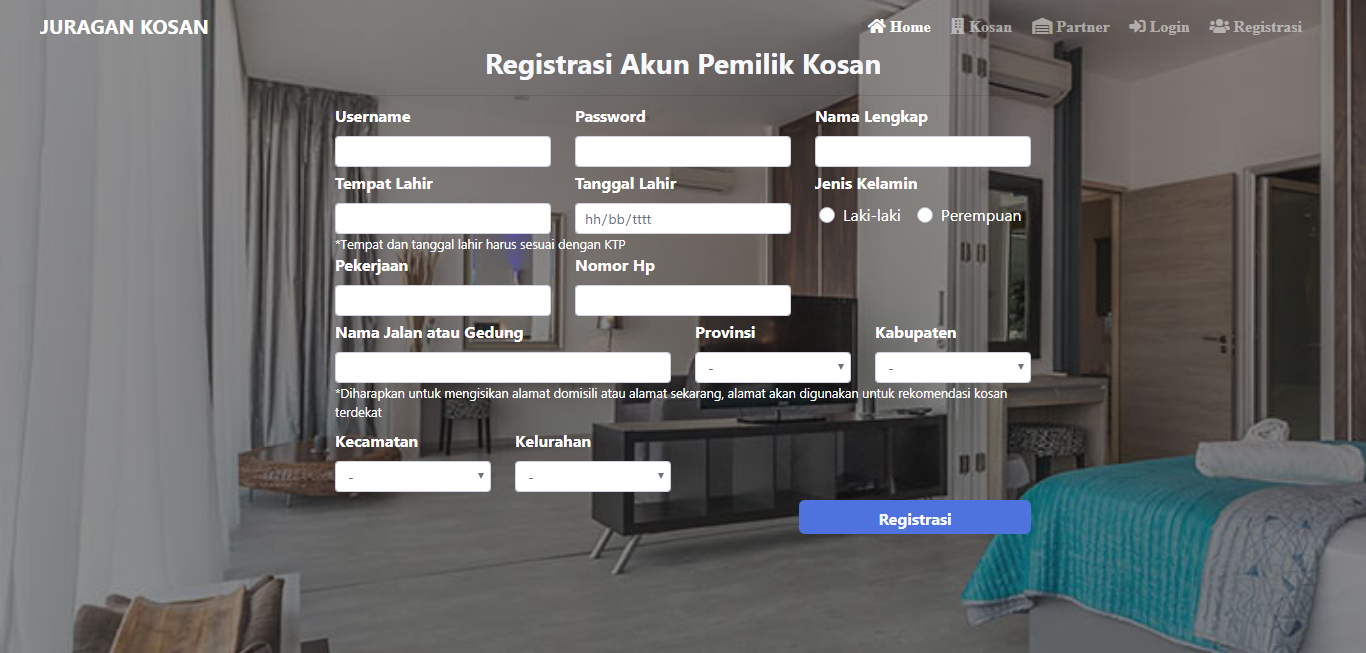
* + - * 1. Tabel Kabupaten
        2. Tabel Kecamatan
        3. Tabel Kelurahan
        4. Tabel Alamat
        5. View Alamat



Gambar 4.10 UI Registrasi Akun Pelanggan



Gambar 4.11 UI Registrasi Akun Partner



Gambar 4.12 UI Registrasi Akun Pemilik Kosan

* + 1. Membuat Form Login

Semua user(Admin, Partner, Pemilik Kosan dan Pelanggan) login pada form yang sama, setelah user mengetikkan username dan password, maka sistem akan mengecek apa role user tersebut, lalu akan mengarahkannya(*redirect*) sesuai dengan role masing-masing.

1. Pada folder **ci-kosan/application/views/user** buatlah file bernama **login.php** lalu ketikkan kode HTML berikut :

<div class="position-bg load-pic-login">

  <div class="set-img">

    <div class="container">

      <div class="row justify-content-center">

        <div class="col-lg-6 my-5">

          <div class="p-5 my-auto">

            <h3 class="text-center mb-4 text-light">

              <strong>Login User</strong>

            </h3>

            <?= $this->session->flashdata('message'); ?>

            <form class="user" method="POST">

              <div class="form-group">

                <input type="text" name="username" class="form-control form-control-user" placeholder="Username" value="<?= set\_value('username') ?>">

                <?= form\_error('username', '<small class="text-danger pl-3">', '</small>'); ?>

              </div>

              <div class="form-group">

                <input type="password" name="password" class="form-control form-control-user" placeholder="Password" autocomplete="on">

                <?= form\_error('password', '<small class="text-danger pl-3">', '</small>'); ?>

              </div>

              <button type="submit" class="btn my-color text-light btn-user btn-block border">

                <strong>Login</strong>

              </button>

            </form>

          </div>

        </div>

      </div>

    </div>

  </div>

</div>

Kode 23 : Form Login

Keterengan :

* <div class="position-bg load-pic-login"> berfungsi untuk meload css sendiri yang telah dibuat di mystyle.cs.
* <?= $this->session->flashdata('message'); ?> akan menampilkan error ketika username dan password tidak terdaftar.
* Function set\_value() berfungsi untuk mengisi form tersebut dengan username yang telah diinputkan user
* Function form\_error() akan menampilkan error sesuai dengan yang telah di setting oleh developer, contoh : akan muncul peringatan, apabila user tidak mengisi username atau password.

1. Buka Controller user buat sebuah function login. Function ini akan dijalankan ketika user berada pada menu login. Apabila semua persyaratan telah terpenuhi, maka function proses\_login(); akan dijalankan.

public function login()

    {

        $this->check\_privilege();

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

            'username',

            'Username',

            'trim|required',

            [

                'required' => 'Please, type your username'

            ]

        );

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

            'password',

            'Password',

            'trim|required',

            [

                'required' => 'Need password to login'

            ]

        );

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

            $data['title'] = 'Login';

            $this->load->view('template/header\_user', $data);

            $this->load->view('user/login');

            $this->load->view('template/footer\_user');

        } else {

            $this->proses\_login();

        }

    }

Kode 24 : Function login

Keterangan :

* $this->check\_privilege(); memanggil sebuah function yang ada dalam satu controller.
* $this->form\_validation->set\_rules() aturan yang dibuat oleh developer agar user mengetahui kalau terjadi kesalahan. Dan akan ditampilkan dengan function form\_error().
* $this->load->view() menggabunggkan semua file.
* $data['title'] = 'Login'; Array assosiatif yang menyimpan tipe data string lalu dikirim ke file header\_user.

1. Buat sebuah function bernama check\_privilege dengan hak akses private. Function ini bertugas untuk mengecek role user, dan mengarahkannya ke link masing-masing role. Jadi tidak bisa mengakses halaman login ketika user sudah login.

private function check\_privilege()

    {

        if ($this->session->userdata('user\_id')) {

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

            if ($role\_id == 1) {

                redirect('admin');

            } else if ($role\_id == 2) {

                redirect('PemilikKosan');

            } else if ($role\_id == 3) {

                redirect('partner');

            } else if ($role\_id == 4) {

                redirect('pegawai');

            } else if ($role\_id == 5) {

                redirect('pelanggan');

            } else {

                $this->blocked();

            }

        }

    }

Kode 25 : Function checkPrivilege

Keterangan :

* Sebenarnya pada versi aplikasi ini, role pegawai tidak dipakai. Apabila selanjutnya terdapat update, role pegawai mungkin jadi pertimbangan.
* Jika di tabel user\_role pelanggan role\_id = 4, kamu perlu menyesuaikan logika tersebut.

1. Buat function proses\_login di controller user dengan hak akses private. Function proses\_login berguna untuk mengecek inputan user, apakah username atau password terdapat di database, jika tidak akan muncul peringatan. Setelah data user terverifikasi oleh sistem, sistem akan mengeset atau menyimpan user yang login dengan session.

private function proses\_login()

    {

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

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

        $user        = $this->db->get\_where('user', ['username' => $username])->row\_array();

*// Jika user ada*

        if ($user) {

*// cek password*

            if (password\_verify($password, $user['password'])) {

                $data =

                    [

                        'user\_id'      => $user['user\_id'],

                        'role\_id'      => $user['role\_id'],

                    ];

                $this->session->set\_userdata($data);

                $id = $this->Model\_user->getDataAkun();

                $cek\_nik = $this->db->get\_where('tabel\_akun', ['akun\_id' => $id['akun\_id']])->result\_array();

*// jika Admin*

                if ($user['role\_id'] == 1) {

                    redirect('admin');

*// Jika Aplikasi*

                } elseif ($user['role\_id'] == 2) {

                    ($cek\_nik[0]['nik'] == null) ? $this->cek\_nik() : redirect('PemilikKosan');

*// Jika partner*

                } else if ($user['role\_id'] == 3) {

                    ($cek\_nik[0]['nik'] == null) ? $this->cek\_nik() : redirect('partner');

*// Jika Pegawai*

                } else if ($user['role\_id'] == 4) {

                    ($cek\_nik[0]['nik'] == null) ? $this->cek\_nik() : redirect('pegawai');

                } else if ($user['role\_id'] == 5) {

                    ($cek\_nik[0]['nik'] == null) ? $this->cek\_nik() : redirect('pelanggan');

                } else {

                    redirect('user/login');

                }

            } else {

                $this->session->set\_flashdata(

                    'message',

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

                    <strong>Password salah</strong>

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

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

                    </button>

                    </div>'

                );

                redirect('user/login');

            }

        } else {

            $this->session->set\_flashdata(

                'message',

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

                <strong>Username tidak terdaftar</strong>

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

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

                </button>

                </div>'

            );

            redirect('user/login');

        }

    }

Kode 26 : Function proses\_login

Keterangan :

* $data berguna untuk mengambil id akun yang sedang *login*.
* $user mengecek apakah ada *username* tersebut pada *database*.
* $this->session->set\_userdata($data); berguna untuk menyimpan $data ke dalam *session*.
* $cek\_nik mengecek apakah user yang login sudah mengisi informasi NIK.
* password\_verify($password, $user['password']) berguna untuk mencocokkan *password* yang dimasukkan user dengan *password* yang teleh di *hash*.
  + 1. Dashboard Pemilik Kosan
    2. Form Pendaftaran Kosan
    3. Profile Kosan
    4. Menampilkan Usaha di Menu Usaha Tersedia (Pemilik Kosan)
    5. Form Pendafataran Usaha *Partner*
    6. Dashboard Partner
    7. Membuat *Profile* Usaha *Partner*
    8. Form Pengusulan Partnership(Pemilik Kosan)
    9. Konfirmasi Partnership(Partner)
       1. Notifkasi Penerimaan Partnership
       2. Notifikasi Penolakan Partnership
    10. Pengajuan Pemberhentian Partnership(Pemilik Kosan)
    11. Konfirmasi Pengajuan Pemberhentian *Partnership* (*Partner*)
        1. Negosiasi Partnership
        2. Tolak Negosiasi Partnership
    12. Konfirmasi Negosiasi(Pemilik Kosan)
        1. Terima Negosiasi
        2. Berhentikan Partnership
    13. Menampilkan Kosan di Menu Kosan Tersedia (Partner)
    14. Form Pengusulan *Partnership*(*Partner*)
    15. Konfirmasi *Partnership*(Pemilik Kosan)
        1. Notifkasi Penerimaan Partnership
        2. Notifikasi Penolakan Partnership
    16. Pengajuan Pemberhentian Partnership(Partner)
    17. Konfirmasi Pengajuan Pemberhentian *Partnership*(Pemilik Kosan)
        1. Negosiasi Partnership
        2. Tolak Negosiasi Partnership
    18. Konfirmasi Negosiasi(Partner)
        1. Terima Negosiasi
        2. Berhentikan Partnership
    19. Menampilkan Kosan pada Menu Kosan Tersedia (Pelanggan)
    20. Form Pemesanan Kamar(Pelanggan)
    21. Konfirmasi Pemesanan Kamar(Pemilik Kosan)
    22. Notifikasi dan Upload Bukti Pembayaran (Pelanggan)
    23. *Approval* Bukti Pembayaran dan Kirim Faktur(Admin)
    24. *Download* Faktur(Pelanggan)
    25. *Upload* File Faktur(Pemilik Kosan)
    26. *Approval* Faktur Pemilik Kosan(Admin)