**MAKALAH WEB PROGRAMMING II**

**SISTEM INFORMASI WEB PEMESANAN KOST ONLINE**



Disusun oleh :

1. Elsa Puspitasari (12190701)

2. Alfian Winuarto (12192437)

3. Andi Aprilia Sidiuzdah (12192611)

4. Andika Indra Prameswara (12191910)

5. Septian Yanuar (12190584)

Kelas : 12.3A.37

**PRODI SISTEM INFORMASI**

**UNIVERSITAS BINA SARANA INFORMATIKA**

**2020**

**DAFTAR ISI**

[**DAFTAR ISI** ii](#_Toc59684254)

[**KATA PENGANTAR** iii](#_Toc59684255)

[**BAB I** 1](#_Toc59684256)

[**PENDAHULUAN** 1](#_Toc59684257)

[**A.** **Latar Belakang** 1](#_Toc59684258)

[**B.** **Rumusan Masalah** 1](#_Toc59684259)

[**C.** **Tujuan Pembahasan Masalah** 1](#_Toc59684260)

[**D.** **Metode dan Teknik penulisan** 2](#_Toc59684261)

[**E.** **Sistematika penulisan** 2](#_Toc59684262)

[**BAB II** 3](#_Toc59684263)

[**LANDASAN TEORI** 3](#_Toc59684264)

[**A.** **Sejarah Codeigtiner** 3](#_Toc59684265)

[**B.** **Pengertian Framework** 3](#_Toc59684266)

[**C.** **Pengertian MVC (Model View Controller)** 3](#_Toc59684267)

[**D.** **Kelebihan Codeigtiner** 4](#_Toc59684268)

[**BAB III** 5](#_Toc59684269)

[**PEMBAHASAN** 5](#_Toc59684270)

[**A.** **Coding** 5](#_Toc59684271)

[**B.** **Tampilan** 32](#_Toc59684272)

[**BAB IV** 36](#_Toc59684273)

[**PENUTUP** 36](#_Toc59684274)

[**A.** **Kesimpulan** 36](#_Toc59684275)

[**B.** **Saran** 36](#_Toc59684276)

# **KATA PENGANTAR**

Puji syukur kami panjatkan atas kehadirat Allah SWT, yang telah memberikan rahmat dan karunia-Nya kepada kami sehingga kami dapat menyelesaikan makalah ini tepat pada waktunya. Dimana didalam makalah ini sebagai salah satu persyaratan untuk memperoleh nilai tugas khususnya tugas Project UAS Semester III pada mata kuliah Web Programming II. Adapun Judul Penulisan makalah ini adalah :

**“SISTEM INFORMASI WEB PEMESANAN KOST”**

Sebagai bahan penulisan diambil dari hasil observasi dari beberapa sumber literature yang mendukung penulisan ini. Kami menyadari tanpa bimbingan dan dorongan dari semua pihak, penulisan makalah ini tidak akan berjalan dengan lancar. Oleh sebab itu pada kesempatan ini, izinkan kami menyampaikan ucapan terima kasih kepada semua pihak yang membantu dalam mempermudah pembuatan makalah ini. Kami menyadari bahwa penulisan tugas makalah ini masih jauh sekali dari sempurna, untuk itu kami mohon kritik dan saran yang bersifat membangun demi kesempurnaan penulisan di masa yang akan datang.

Akhir kata semoga tugas makalah ini dapat berguna bagi kami khususnya dan pembaca pada umumnya.

Jakarta, Desember 2020

Penulis

# **BAB I**

# **PENDAHULUAN**

1. **Latar Belakang**

Web Application Framework (WAF), atau sering disingkat web framework, adalah suatu kumpulan kode berupa pustaka (library) dan alat (tool) yang dipadukan sedemikian rupa menjadi kerangka kerja (framework) guna memudahkan dan mempercepat proses pengembangan aplikasi.

Sebagian besar dari framework mengimplementasikan pola desain Model-View-Controller (MVC), yang memisahkan bagian kode untuk penanganan proses bisnis dengan bagian kode untuk keperluan presentasi (tampilan).

Codeigtiner adalah framework web untuk bahasa pemrograman PHP, yang dibuat oleh Rick Ellis pada tahun 2006, penemu dan pendiri EllisLab ([www.ellislab.com](http://www.ellislab.com)). EllisLab merupakan suatu tim kerja yang berdiri pada tahun 2002 yang bergerak di bidang pembuatan software dan tool untuk para pengembang web. Sejak tahun 2014 sampai sekarang., EllisLab telah menyerahkan hak kepemilikan codeigtiner ke British columbia Institue of Technology (BCIT) saat ini situs resmi codeigtiner adalah [www.codeigtiner.com](http://www.codeigtiner.com)

1. **Rumusan Masalah**

Berdasarkn latar belakang diatas, masalah dirumuskan sebagai berikut :

1. Mengetahui lebih dalam apa itu MVC.
2. Mengetahui kelebihan dari Codeigtiner.
3. Coding dari pembuatan sistem informasi berbasis web pemesanan kost.
4. Tampilan dari pembuatan sistem informasi berbasis web pemesanan kost
5. **Tujuan Pembahasan Masalah**
6. Untuk memenuhi persyaratan dalam memperoleh nilai UAS mata kuliah Web Programming II di semester III.
7. Untuk memperdalam ilmu dan pengetahuan mengenai materi di mata kuliah Web Programming II.
8. Untuk membuat pembaca lebih mengetahui dan memahami materi tentang penggunaan framework Codeigtiner melalui makalah ini dan penarikan kesimpulan yang lebih ditentukan oleh kelompok kami.
9. **Metode dan Teknik penulisan**
10. Metode Penulisan

Dalam pembuatan makalah ini, penulis menggunakan metode deskriptif, yaitu metode yang berdasarkan pengumpulan buku-buku dan media komunikasi lainnya yang berkaitan dengan masalah yang dibahas nantinya menjadi pegangan dan sumber.

1. Teknik Penulisan

Teknik yang digunakan dalam penyusunan karangan makalah ini adalah teknik bibliografi atau telaah pustaka yaitu sebuah teknik dengan menelaah atau mengkaji sumber-sumber tertulis.

1. **Sistematika penulisan**

Agar penulisan makalah ini lebih tersusun dan terarah dalam penyusunannya, maka penulis membaginya menjadi 4 bab :

BAB I : PENDAHULUAN, yang terdiri dari latar belakang, rumusan masalah, tujuan pembahasan masalah, metode dan teknik penulisan dan sistem matika penulisan.

BAB II : LANDASAN TEORI, yang terdiri sejarah Coteigtiner, pengertian Framework, MVC (Model View Controller), dan kelebihan Codeigtiner.

BAB III : PEMBAHASAN, yang terdiri dari lembar prosedur sistem yang berjalan, dan materi dari penggunaan Codeigtiner dalam judul Sistem Informasi Berbasis Web Pemesanan Kost.

BAB IV : PENUTUP, yang terdiri dari kesimpulan dan saran.

# **BAB II**

# **LANDASAN TEORI**

1. **Sejarah Codeigtiner**

Codeigtiner adalah aplikasi open source yang berupa framework dengan model MVC (Model View Controller) untuk membangun website dinamis dengan menggunakan PHP. Codeigtiner memudahkan developer untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuatnya dari awal. Codeigtiner diliris pertama kali pada 28 Februari 2006. Versi stabil terakhir adalah versi 3.0.6. yang diliris pada 12 Juni 2018.

1. **Pengertian Framework**

Framework secara sederhana dapat diartikan kumpulan dari fungsi-fungsi/prosedur-prosedur dan class-class untuk tujuan tertentu yang sudah siap digunakan sehinggga bisa lebih mempermudah dan mempercepat pekerjaan seorang pemrograman, tanpa harus membuat fungsi atau class dari awal.

Ada beberapa alasan mengapa menggunakan Framework :

* Mempercepat dan mempermudah pembangunan sebuah aplikasi web.
* Relatif memudahkan dalam proses maintenance karena sudah ada pola tertentu dalam sebuah framework (dengan syarat programmer mengikuti pola stantard yang ada).
* Umumnya framework menyediakan fasilitas-fasilitas yang umum dipakai sehingga kita tidak perlu membangun dari awal (misalnya validasi, ORM, pagination, multiple, database, scaffolding, pengaturan session, error handling, dll).
* Lebih bebas dalam pengembangan jika dibandingkan CMS.

1. **Pengertian MVC (Model View Controller)**

MVC merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berawal pada bahasa pemrograman Smalll Talk, MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, user interface, dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu MVC pattern dalam sutau aplikasi yaitu :

1. View, merupakan bagian yang menangani bagian presentation logic. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh controller. View berfungsi untuk menerima dan mempresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model.
2. Model, biasanya berhubungan langsung dengan database untuk memanipulasi data (insert, update, delete, search), menangani validasi dari bagian controller, namun tidak dapat berhubungan langsung dengan bagian view.
3. Controller, merupakan bagian yang mengatur hubungan antara bagian model dan bagian view, controller berfungsi untuk menerima request dan data dari user kemudian menentukan apa yang akan diproses oleh aplikasi.

Dengan menggunakan prinsip MVC suatu aplikasi dapat dikembangkan sesuai dengan kemampuan developernya, yaitu programmer yang menanganin bagian model dan controller, sedangkan designer yang menangani bagian view, sehingga penggunaan arsitektur MVC dapat meningkatkan maintanablity dan organisasi kode. Walaupun demikian membutuhkan komunikasi yang baik antara programmer dan designer dalam menangani variabel-variabel yang akan ditampilkan.

1. **Kelebihan Codeigtiner**

Ada beberapa kelebihan codeigtiner (CI) dibandingkan dengan Framework PHP lain :

* Performa sangat cepat : salah satu alasan tidak menggunakan framework adalah karena eksekusinya yang lebih lambat daripada PHP from the scract, tapi Codeigtiner sangat cepat bahkan mungkin bisa dibilang Codeigtiner merupakan framework yang paling cepat dibanding framework lain.
* Konfigurasi yang sangat minim (nearly zero configuration) : tentu saja untuk menyesuaikan dengan database dan keleluasaan routing tetap diijinkan melakukan konfigurasi dengan mengubah beberapa file konfigurasi seperti database.php atau autoload.php, namun untuk menggunakan codeigtiner dengan setting standart, anda hanya perlu merubah sedikit saja file pada folder config.
* Banyak komunitas : dengan banyaknya komunitas CI ini, memudahkan kita untuk berinteraksi dengan yang lain, baik itu bertanya atau teknologi terbaru.
* Dokumentasi yang sangat lengkap : setiap paket instalasi codeigtiner sudah disertai user guide yang sangat bagus dan lengkap untuk dijadikan permulaan, bahasanya pun masih dipahami.
* Dan banyak lagi yang lainnya.

# **BAB III**

# **PEMBAHASAN**

Bab ini berisi project pembuatan Sistem Informasi menggunakan framework Codeigtiner dengan pola MVC (Model View Controller) yang berjudul “Pemesanan Kost”.

1. **Coding**

Disini kami akan menampilkan beberapa contoh coding yang digunakan dalam membuat Program Sistem Informasi kami dengan nama folder kos\_online pada xampp/htdoc.

1. File Routes

Variabel tersebut harus kita isi dengan controller utama yang akan kita gunakan sebagai tampilan utama website kita. Standartnya, variabel tersebut akan terisi dengan **welcome** yang akan menampilkan halaman utama Codeigtiner. Dalam program kami menggunakan file controller kos.php.

**$route['default\_controller'] = "kos";**

**$route['404\_override'] = '';**

1. File Config

Untuk menyesuaikan url yang kita gunakan atau jika kita sudah upload ke hosting url tersebut kita sesuaikan dengan nama domain.

**// AutoConfigure BaseURL**

**$config['base\_url'] = "http://".$\_SERVER['HTTP\_HOST'];**

**$config['base\_url'] .= preg\_replace('@/+$@','',dirname($\_SERVER['SCRIPT\_NAME'])).'/';**

**// OR**

**// $config['base\_url'] = '/CodeIgniter/';**

1. File Database

Untuk konfigurasi untuk koneksi ke database.

**$db['default']['hostname'] = 'localhost';**

**$db['default']['username'] = 'root';**

**$db['default']['password'] = '';**

**$db['default']['database'] = 'db\_kos\_t\_on\_line';**

**$db['default']['dbdriver'] = 'mysql';**

Seperti yang ditampilkan diatas bahwa usernamenya adalah root sebagai default username aplikasi xampp dan nama database yang digunakan adalah db\_kos\_t\_on\_line.

1. File Controller

Controller merupakan kumpulan intruksi aksi yang menghubungkan model dan view, jadi user tidak akan berhubungan dengan model secara langsung, intinya data yang tersimpan di database (model) diambil oleh controller dan kemudian controller pula yang menampilkannya ke view. Jadi controller lah yang mengelola kontruksi.

Ada beberapa file controller pada project kita, berikut hanya satu controller yg akan kita simpan disini sebagai controller pertama yang akan muncul pada halaman awal

<?php

class Kos extends CI\_Controller {

public function \_\_construct()

{

parent::\_\_construct();

$this->load->helper('form');

$this->load->helper('url');

$this->load->database();

$this->load->library('pagination');

$this->load->library('session');

$this->load->model('m\_kos');

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

}

public function index()

{

$comp=array(

"slider"=>$this->html\_slider(),

"kos"=>$this->html\_konten(),

"footer"=>$this->html\_footer(),

);

$this->load->view("index",$comp);

}

public function search()

{

// get search string

$search = ($this->input->post("kata\_pencarian"))? $this->input->post("kata\_pencarian") : "NIL";

$search = ($this->uri->segment(3)) ? $this->uri->segment(3) : $search;

// pagination settings

$config = array();

$config['base\_url'] = site\_url("c\_kos/search/$search");

$config['total\_rows'] = $this->m\_kos->get\_kos\_count($search);

$config['per\_page'] = "12";

$config["uri\_segment"] = 4;

$choice = $config["total\_rows"]/$config["per\_page"];

$config["num\_links"] = floor($choice);

// integrate bootstrap pagination

$config['full\_tag\_open'] = '<ul class="pagination pagination-lg">';

$config['full\_tag\_close'] = '</ul>';

$config['first\_link'] = false;

$config['last\_link'] = false;

$config['first\_tag\_open'] = '<li>';

$config['first\_tag\_close'] = '</li>';

$config['prev\_link'] = 'Prev';

$config['prev\_tag\_open'] = '<li class="prev">';

$config['prev\_tag\_close'] = '</li>';

$config['next\_link'] = 'Next';

$config['next\_tag\_open'] = '<li>';

$config['next\_tag\_close'] = '</li>';

$config['last\_tag\_open'] = '<li>';

$config['last\_tag\_close'] = '</li>';

$config['cur\_tag\_open'] = '<li class="active"><a href="#">';

$config['cur\_tag\_close'] = '</a></li>';

$config['num\_tag\_open'] = '<li>';

$config['num\_tag\_close'] = '</li>';

$this->pagination->initialize($config);

$data['page'] = ($this->uri->segment(4)) ? $this->uri->segment(4) : 0;

// get books list

$data['booklist'] = $this->m\_kos->get\_kos($config['per\_page'], $data['page'], $search);

$data['pagination'] = $this->pagination->create\_links();

$comp=array(

"slider"=>$this->html\_slider(),

"kos"=>$this->load->view("kos",$data,true), /// html\_konten dan yg lainya adalah fungtion html\_konten yg dbawah.

"footer"=>$this->html\_footer(),

);

$this->load->view("index",$comp);

}

public function html\_slider()

{

$idb=$this->uri->segment(3);

$data['page']="0";

$config['per\_page']='5';

$data=array(

"slider"=>$this->m\_kos->kos\_new\_slider($config["per\_page"], $data['page'], $idb),

);

return $this->load->view('slider',$data,true);

}

public function html\_konten()

{

$config['base\_url'] = site\_url('kos/index');

$config['total\_rows'] = $this->m\_kos->get\_kos\_html\_konten\_count();

$config['per\_page'] = "12";

$config["uri\_segment"] = 3;

$choice = $config["total\_rows"]/$config["per\_page"];

$config["num\_links"] = floor($choice);

// integrate bootstrap pagination

$config['full\_tag\_open'] = '<ul class="pagination pagination-lg">';

$config['full\_tag\_close'] = '</ul>';

$config['first\_link'] = false;

$config['last\_link'] = false;

$config['first\_tag\_open'] = '<li>';

$config['first\_tag\_close'] = '</li>';

$config['prev\_link'] = 'Prev';

$config['prev\_tag\_open'] = '<li class="prev">';

$config['prev\_tag\_close'] = '</li>';

$config['next\_link'] = 'Next';

$config['next\_tag\_open'] = '<li>';

$config['next\_tag\_close'] = '</li>';

$config['last\_tag\_open'] = '<li>';

$config['last\_tag\_close'] = '</li>';

$config['cur\_tag\_open'] = '<li class="active"><a href="#">';

$config['cur\_tag\_close'] = '</a></li>';

$config['num\_tag\_open'] = '<li>';

$config['num\_tag\_close'] = '</li>';

$this->pagination->initialize($config);

$data['page'] = ($this->uri->segment(3)) ? $this->uri->segment(3) : 0;

// get books list

$data['booklist'] = $this->m\_kos->get\_kos\_html\_konten($config["per\_page"], $data['page'], NULL);

$data['pagination'] = $this->pagination->create\_links();

// load view

return $this->load->view('kos',$data,true);

}

public function new\_kos()

{

$config['base\_url'] = site\_url('kos/new\_kos');

$config['total\_rows'] = $this->m\_kos->get\_kos\_html\_konten\_count();

$config['per\_page'] = "12";

$config["uri\_segment"] = 3;

$choice = $config["total\_rows"]/$config["per\_page"];

$config["num\_links"] = floor($choice);

// integrate bootstrap pagination

$config['full\_tag\_open'] = '<ul class="pagination pagination-lg">';

$config['full\_tag\_close'] = '</ul>';

$config['first\_link'] = false;

$config['last\_link'] = false;

$config['first\_tag\_open'] = '<li>';

$config['first\_tag\_close'] = '</li>';

$config['prev\_link'] = 'Prev';

$config['prev\_tag\_open'] = '<li class="prev">';

$config['prev\_tag\_close'] = '</li>';

$config['next\_link'] = 'Next';

$config['next\_tag\_open'] = '<li>';

$config['next\_tag\_close'] = '</li>';

$config['last\_tag\_open'] = '<li>';

$config['last\_tag\_close'] = '</li>';

$config['cur\_tag\_open'] = '<li class="active"><a href="#">';

$config['cur\_tag\_close'] = '</a></li>';

$config['num\_tag\_open'] = '<li>';

$config['num\_tag\_close'] = '</li>';

$this->pagination->initialize($config);

$data['page'] = ($this->uri->segment(3)) ? $this->uri->segment(3) : 0;

// get books list

$data['booklist'] = $this->m\_kos->kos\_new($config["per\_page"], $data['page'], NULL);

$data['pagination'] = $this->pagination->create\_links();

$comp=array(

"slider"=>$this->html\_slider(),

"kos"=>$this->load->view("kos",$data,true), /// html\_konten dan yg lainya adalah fungtion html\_konten yg dbawah.

"footer"=>$this->html\_footer(),

);

$this->load->view("index",$comp);

}

public function kategori($id)

{

$kt=$this->uri->segment(3);

$config['base\_url'] = site\_url("kos/kategori/$kt");

$config['total\_rows'] = $this->m\_kos->get\_kos\_kt($kt);

$config['per\_page'] = "12";

$config["uri\_segment"] = 4;

$choice = $config["total\_rows"]/$config["per\_page"];

$config["num\_links"] = floor($choice);

// integrate bootstrap pagination

$config['full\_tag\_open'] = '<ul class="pagination pagination-lg">';

$config['full\_tag\_close'] = '</ul>';

$config['first\_link'] = false;

$config['last\_link'] = false;

$config['first\_tag\_open'] = '<li>';

$config['first\_tag\_close'] = '</li>';

$config['prev\_link'] = 'Prev';

$config['prev\_tag\_open'] = '<li class="prev">';

$config['prev\_tag\_close'] = '</li>';

$config['next\_link'] = 'Next';

$config['next\_tag\_open'] = '<li>';

$config['next\_tag\_close'] = '</li>';

$config['last\_tag\_open'] = '<li>';

$config['last\_tag\_close'] = '</li>';

$config['cur\_tag\_open'] = '<li class="active"><a href="#">';

$config['cur\_tag\_close'] = '</a></li>';

$config['num\_tag\_open'] = '<li>';

$config['num\_tag\_close'] = '</li>';

$this->pagination->initialize($config);

$data['page'] = ($this->uri->segment(4)) ? $this->uri->segment(4) : 0;

// get books list

$data['booklist'] = $this->m\_kos->kos\_kt($config['per\_page'], $data['page'], $kt);

$data['pagination'] = $this->pagination->create\_links();

$comp=array(

"slider"=>$this->html\_slider(),

"kos"=>$this->load->view("kos",$data,true), /// html\_konten dan yg lainya adalah fungtion html\_konten yg dbawah.

"footer"=>$this->html\_footer(),

);

$this->load->view("index",$comp);

}

public function lihat\_kos($id)

{

$this->m\_kos->update\_view($id);

$cek=$this->m\_kos->lihat\_kos("where kd\_kos\_t='$id'");

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

$data=array(

"kost"=>$this->m\_kos->lihat\_kos("where tb\_u\_ser\_kos\_t.no\_ktp\_us=tb\_kos\_t\_onl.no\_ktp\_us and tb\_kos\_t\_onl.id\_kmp=tb\_kmps.id\_kmp and tb\_kos\_t\_onl.kd\_kos\_t='$id'")->result(),

);

$comp=array(

"slider"=>"",

"kos"=>$this->load->view("lihat\_kos",$data,true),

"footer"=>$this->html\_footer(),

);

$this->load->view("index",$comp);

}else{

show\_404();

}

}

public function html\_footer()

{

$idb=$this->uri->segment(3);

$data['page']="0";

$config['per\_page']='4';

$comp=array(

"recent"=>$this->m\_kos->kos\_new($config["per\_page"], $data['page'], $idb),

"kosl"=>$this->m\_kos->kos\_l($config["per\_page"], $data['page'], $idb),

"kosp"=>$this->m\_kos->kos\_p($config["per\_page"], $data['page'], $idb),

"kontak"=>$this->m\_kos->kontak()->result(),

);

return $this->load->view('footer',$comp,true);

}

public function komentar(){

$nm=$\_POST['nama'];

$e\_m=$\_POST['email'];

$sub=$\_POST['sub'];

$kom=$\_POST['komen'];

date\_default\_timezone\_set("Asia/Jakarta");

$tgl=date('Y-m-d H:i:s');

if($nm=="" or $e\_m=="" or $sub=="" or $kom==""){

$this->session->set\_flashdata('pesan','<div class="alert alert-danger alert-dismissible" role="alert">

<button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">&times;</span></button>

<strong>Gagal! </strong> Data tidak boleh ada yang kosong.</div>');

redirect('kos/kontak');

}

$data\_insert=array(

'nm'=>$nm,

'em'=>$e\_m,

'sub'=>$sub,

'kom'=>$kom,

'tgl'=>$tgl,

'id'=>''

);

$res=$this->m\_kos->simpan\_komen('tb\_kmen\_usr',$data\_insert);

if($res >= 1){

$this->session->set\_flashdata('pesan','<div class="alert alert-success alert-dismissible" role="alert">

<button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">&times;</span></button>

<strong>Sukses! </strong> Komentar berhasil dikirim.</div>');

redirect('kos/kontak');

}else{

$this->session->set\_flashdata('pesan','<div class="alert alert-danger alert-dismissible" role="alert">

<button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">&times;</span></button>

<strong>Gagal! </strong> Komentar gagal dikirim.</div>');

redirect('kos/kontak');

}

}

public function populer()

{

$config['base\_url'] = site\_url('kos/populer');

$config['total\_rows'] = $this->m\_kos->get\_kos\_html\_konten\_count();

$config['per\_page'] = "12";

$config["uri\_segment"] = 3;

$choice = $config["total\_rows"]/$config["per\_page"];

$config["num\_links"] = floor($choice);

// integrate bootstrap pagination

$config['full\_tag\_open'] = '<ul class="pagination pagination-lg">';

$config['full\_tag\_close'] = '</ul>';

$config['first\_link'] = false;

$config['last\_link'] = false;

$config['first\_tag\_open'] = '<li>';

$config['first\_tag\_close'] = '</li>';

$config['prev\_link'] = 'Prev';

$config['prev\_tag\_open'] = '<li class="prev">';

$config['prev\_tag\_close'] = '</li>';

$config['next\_link'] = 'Next';

$config['next\_tag\_open'] = '<li>';

$config['next\_tag\_close'] = '</li>';

$config['last\_tag\_open'] = '<li>';

$config['last\_tag\_close'] = '</li>';

$config['cur\_tag\_open'] = '<li class="active"><a href="#">';

$config['cur\_tag\_close'] = '</a></li>';

$config['num\_tag\_open'] = '<li>';

$config['num\_tag\_close'] = '</li>';

$this->pagination->initialize($config);

$data['page'] = ($this->uri->segment(3)) ? $this->uri->segment(3) : 0;

// get books list

$data['booklist'] = $this->m\_kos->kos\_pop($config["per\_page"], $data['page'], NULL);

$data['pagination'] = $this->pagination->create\_links();

$comp=array(

"slider"=>$this->html\_slider(),

"kos"=>$this->load->view("kos",$data,true), /// html\_konten dan yg lainya adalah fungtion html\_konten yg dbawah.

"footer"=>$this->html\_footer(),

);

$this->load->view("index",$comp);

}

public function kontak()

{

$data=array(

"kontak"=>$this->m\_kos->kontak()->result(),

);

$comp=array(

"slider"=>"",

"kos"=>$this->load->view("kontak",$data,true),

"footer"=>$this->html\_footer(),

);

$this->load->view("index",$comp);

}

public function daftar()

{

$data=array();

$comp=array(

"slider"=>"",

"kos"=>$this->load->view("daftar",$data,true),

"footer"=>$this->html\_footer(),

);

$this->load->view("index",$comp);

}

public function mendaftar(){

$noktp=$\_POST['nktp'];

$nm=$\_POST['nml'];

$jk=$\_POST['jk'];

$nohp=$\_POST['nohp'];

$alm=$\_POST['alm'];

$user=$\_POST['user'];

$pas=md5($\_POST['pas']);

date\_default\_timezone\_set("Asia/Jakarta");

$tgl=date('Y-m-d H:i:s');

if($noktp=="" or $nm=="" or $jk=="" or $nohp=="" or $alm=="" or $user=="" or $pas==""){

$this->session->set\_flashdata('pesan','<div class="alert alert-danger alert-dismissible" role="alert">

<button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">&times;</span></button>

<strong>Gagal! </strong> Data tidak boleh ada yang kosong.</div>');

redirect('kos/daftar');

}

$data\_insert=array(

'no\_ktp\_us'=>$noktp,

'nml\_us'=>$nm,

'je\_kel\_us'=>$jk,

'no\_hp\_us'=>$nohp,

'alm\_us'=>$alm,

'ua\_ks\_user'=>$user,

'ua\_ks\_pas'=>$pas,

'lev\_ua'=>'3',

'ft\_us'=>'',

'tgl\_daf'=>$tgl

);

$res=$this->m\_kos->mendaftar('tb\_u\_ser\_kos\_t',$data\_insert);

if($res >= 1){

$this->session->set\_flashdata('pesan','<div class="alert alert-success alert-dismissible" role="alert">

<button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">&times;</span></button>

<strong>Sukses! </strong> Anda Berhasil mendaftar silahkan login dengan username dan password yang anda daftarkan.</div>');

redirect('kos/daftar');

}else{

$this->session->set\_flashdata('pesan','<div class="alert alert-danger alert-dismissible" role="alert">

<button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">&times;</span></button>

<strong>Gagal! </strong> Anda gagal untuk mendaftar.</div>');

redirect('kos/daftar');

}

}

public function pesan\_kos($id){

$idu=$this->session->userdata('noktp');

if($idu==""){

$this->session->set\_flashdata('pesan','<div class="alert alert-danger alert-dismissible" role="alert">

<button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">&times;</span></button>

<strong>Gagal! </strong> Untuk melakukan pemesanan,anda harus login terlebih dahulu.</div>');

redirect('kos/lihat\_kos/'.$id);

}

}

}

1. File Model

Model adalah Kelas yang merepresentasikan atau memodelkan tipe datayang akan digunakan oleh aplikasi. Berikut coding model kami :

<?php

class M\_kos extends CI\_Model{

function \_\_construct()

{

parent::\_\_construct();

}

public function kos\_pop($limit, $start, $st = NULL){

if ($st == "NIL") $st = "";

$sql = "select \* from tb\_kos\_t\_onl where stt\_ks='2' and kd\_kos\_t NOT IN(select kd\_kos\_t from tb\_psn\_usr\_kost) ORDER BY v\_kos\_t DESC limit " . $start . ", " . $limit;

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

return $query->result();

}

public function kos\_new\_slider($limit, $start, $st = NULL){

if ($st == "NIL") $st = "";

$sql = "select \* from tb\_kos\_t\_onl,tb\_slider where kd\_kos\_t<>'$st' and stt\_ks='2' and kd\_kos\_t NOT IN(select kd\_kos\_t from tb\_psn\_usr\_kost) ORDER BY tgl\_pst DESC limit " . $start . ", " . $limit;

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

return $query->result();

}

public function kos\_new($limit, $start, $st = NULL){

if ($st == "NIL") $st = "";

$sql = "select \* from tb\_kos\_t\_onl where kd\_kos\_t<>'$st' and stt\_ks='2' and kd\_kos\_t NOT IN(select kd\_kos\_t from tb\_psn\_usr\_kost) ORDER BY tgl\_pst DESC limit " . $start . ", " . $limit;

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

return $query->result();

}

public function kos\_kt($limit, $start, $st = NULL){

if ($st == "NIL") $st = "";

$sql = "select \* from tb\_kos\_t\_onl where ktg\_r='$st' and stt\_ks='2' and kd\_kos\_t NOT IN(select kd\_kos\_t from tb\_psn\_usr\_kost) limit " . $start . ", " . $limit;

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

return $query->result();

}

public function kos\_l($limit, $start, $st = NULL){

if ($st == "NIL") $st = "";

$sql = "select \* from tb\_kos\_t\_onl where ktg\_r='Pria' and kd\_kos\_t<>'$st' and stt\_ks='2' and kd\_kos\_t NOT IN(select kd\_kos\_t from tb\_psn\_usr\_kost) limit " . $start . ", " . $limit;

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

return $query->result();

}

public function kos\_p($limit, $start, $st = NULL){

if ($st == "NIL") $st = "";

$sql = "select \* from tb\_kos\_t\_onl where ktg\_r='Wanita' and kd\_kos\_t<>'$st' and stt\_ks='2' and kd\_kos\_t NOT IN(select kd\_kos\_t from tb\_psn\_usr\_kost) limit " . $start . ", " . $limit;

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

return $query->result();

}

///////////////////////////////////////////////////////coding untuk menampilkan , pagination & pecarian //////////////////////////

function get\_kos($limit, $start, $st = NULL)

{

if ($st == "NIL") $st = "";

$sql = "select tb\_kmps.nm\_kmps,tb\_kos\_t\_onl. \* from tb\_kos\_t\_onl,tb\_kmps where nm\_kos\_t like '%$st%' or alm\_ks like '%$st%' or nm\_kmps like '%$st%' and tb\_kos\_t\_onl.id\_kmp=tb\_kmps.id\_kmp and tb\_kos\_t\_onl.stt\_ks='2' and tb\_kos\_t\_onl.kd\_kos\_t NOT IN(select kd\_kos\_t from tb\_psn\_usr\_kost) limit " . $start . ", " . $limit;

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

return $query->result();

}

function get\_kos\_html\_konten($limit, $start, $st = NULL)

{

if ($st == "NIL") $st = "";

$sql = "select \* from tb\_kos\_t\_onl where tb\_kos\_t\_onl.kd\_kos\_t NOT IN(select kd\_kos\_t from tb\_psn\_usr\_kost) and tb\_kos\_t\_onl.stt\_ks='2' limit " . $start . ", " . $limit;

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

return $query->result();

}

function get\_kos\_html\_konten\_count($st = NULL)

{

if ($st == "NIL") $st = "";

$sql = "select \* from tb\_kos\_t\_onl where tb\_kos\_t\_onl.kd\_kos\_t NOT IN(select kd\_kos\_t from tb\_psn\_usr\_kost) and tb\_kos\_t\_onl.stt\_ks='2'";

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

return $query->num\_rows();

}

function get\_kos\_count($st = NULL)

{

if ($st == "NIL") $st = "";

$sql = "select tb\_kmps.nm\_kmps,tb\_kos\_t\_onl. \* from tb\_kos\_t\_onl,tb\_kmps where tb\_kos\_t\_onl.nm\_kos\_t like '%$st%' or tb\_kos\_t\_onl.alm\_ks like '%$st%' or tb\_kmps.nm\_kmps like '%$st%' and tb\_kos\_t\_onl.id\_kmp=tb\_kmps.id\_kmp and tb\_kos\_t\_onl.stt\_ks='2' and tb\_kos\_t\_onl.kd\_kos\_t NOT IN(select kd\_kos\_t from tb\_psn\_usr\_kost) ";

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

return $query->num\_rows();

}

function get\_kos\_kt($st = NULL)

{

if ($st == "NIL") $st = "";

$sql = "select \* from tb\_kos\_t\_onl where ktg\_r='$st' and kd\_kos\_t NOT IN(select kd\_kos\_t from tb\_psn\_usr\_kost) ";

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

return $query->num\_rows();

}

public function lihat\_kos($where=""){

$data= $this->db->query('select tb\_kmps.nm\_kmps,tb\_u\_ser\_kos\_t.nml\_us,tb\_u\_ser\_kos\_t.no\_hp\_us,tb\_kos\_t\_onl. \* from tb\_kos\_t\_onl,tb\_kmps,tb\_u\_ser\_kos\_t '.$where);

return $data;

}

public function kontak($where=""){

$data= $this->db->query('select \* from tb\_kon\_ta\_k '.$where);

return $data;

}

public function rekening($where=""){

$data= $this->db->query('select \* from tb\_rek\_ks\_awk '.$where);

return $data;

}

public function kampus($where=""){

$data= $this->db->query('select \* from tb\_kmps '.$where);

return $data;

}

public function update\_view($id){

$this->db->set('v\_kos\_t','v\_kos\_t+1',False);

$this->db->where('kd\_kos\_t',$id);

$this->db->update('tb\_kos\_t\_onl');

}

public function update\_user\_lev($id){

$this->db->set('lev\_ua','2',False);

$this->db->where('no\_ktp\_us',$id);

$this->db->update('tb\_u\_ser\_kos\_t');

}

public function data\_kos\_adm($where=""){

$data= $this->db->query('select tb\_u\_ser\_kos\_t.nml\_us,tb\_kos\_t\_onl. \* from tb\_kos\_t\_onl,tb\_u\_ser\_kos\_t '.$where);

return $data;

}

public function cek\_log\_user($table ,$where ){

return $this->db->get\_where($table ,$where );

}

public function cek\_log\_adm($table ,$where ){

return $this->db->get\_where($table ,$where );

}

public function cek\_kos\_usr($table ,$where ){

return $this->db->get\_where($table ,$where );

}

public function simpan\_komen($tabelName,$data){

$res = $data = $this->db->insert($tabelName,$data);

return $res;

}

public function mendaftar($tabelName,$data){

$res = $data = $this->db->insert($tabelName,$data);

return $res;

}

public function simpan\_kos($tabelName,$data){

$res = $data = $this->db->insert($tabelName,$data);

return $res;

}

public function profil($where=""){

$data= $this->db->query('select \* from tb\_u\_ser\_kos\_t '.$where);

return $data;

}

public function profil\_adm($where=""){

$data= $this->db->query('select \* from tb\_ad\_min\_kos '.$where);

return $data;

}

public function edit\_profil($tabelName,$data,$where){

$res = $data = $this->db->update($tabelName,$data,$where);

return $res;

}

public function pesan\_kos($tabelName,$data){

$res = $data = $this->db->insert($tabelName,$data);

return $res;

}

public function data\_pesanan($where=""){

$sql = "select tb\_kos\_t\_onl.nm\_kos\_t,tb\_kos\_t\_onl.ktg\_r,tb\_kos\_t\_onl.alm\_ks,tb\_kos\_t\_onl.hrg\_ks,tb\_psn\_usr\_kost. \* from tb\_kos\_t\_onl,tb\_psn\_usr\_kost ".$where;

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

return $query;

}

public function j\_pesanan($where=""){

$sql = "select \* from tb\_psn\_usr\_kost ".$where;

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

return $query->num\_rows();

}

public function j\_trans($where=""){

$sql = "select \* from tb\_psn\_usr\_kost,tb\_trans\_usr\_kost ".$where;

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

return $query->num\_rows();

}

public function data\_transaksi($where=""){

$sql = "select tb\_psn\_usr\_kost.tgl\_psn,tb\_psn\_usr\_kost.metod\_pemb,tb\_trans\_usr\_kost.tgl\_trans\_us,tb\_trans\_usr\_kost.no\_fak\_trans\_us,tb\_trans\_usr\_kost.tot\_bay,tb\_trans\_usr\_kost.stt\_t,tb\_trans\_usr\_kost.cek,tb\_kos\_t\_onl. \* from tb\_psn\_usr\_kost,tb\_trans\_usr\_kost,tb\_kos\_t\_onl ".$where;

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

return $query;

}

public function batal\_psn($tabelName,$where){

$res = $data = $this->db->delete($tabelName,$where);

return $res;

}

public function kos\_usr($where=""){

$sql = "select tb\_kmps.nm\_kmps,tb\_kos\_t\_onl. \* from tb\_kmps,tb\_kos\_t\_onl ".$where;

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

return $query;

}

public function kos\_tampil($where=""){

$sql = "select \* from tb\_kos\_t\_onl ".$where;

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

return $query->result();

}

public function del\_kos\_us($tabelName,$where){

$res = $data = $this->db->delete($tabelName,$where);

return $res;

}

public function psn\_ks\_usr($where=""){

$sql = "select tb\_u\_ser\_kos\_t.nml\_us,tb\_u\_ser\_kos\_t.je\_kel\_us,tb\_kos\_t\_onl.ktg\_r,tb\_kos\_t\_onl.hrg\_ks,tb\_psn\_usr\_kost. \* from tb\_u\_ser\_kos\_t,tb\_psn\_usr\_kost,tb\_kos\_t\_onl ".$where;

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

return $query;

}

public function tran\_ks\_usr($where=""){

$sql = "select tb\_u\_ser\_kos\_t.nml\_us,tb\_kos\_t\_onl.ktg\_r,tb\_psn\_usr\_kost.metod\_pemb,tb\_kos\_t\_onl.hrg\_ks,tb\_kos\_t\_onl.kd\_kos\_t,tb\_trans\_usr\_kost. \* from tb\_u\_ser\_kos\_t,tb\_psn\_usr\_kost,tb\_kos\_t\_onl,tb\_trans\_usr\_kost ".$where;

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

return $query;

}

public function j\_pesanan\_kos($where=""){

$sql = "select \* from tb\_psn\_usr\_kost,tb\_kos\_t\_onl ".$where;

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

return $query->num\_rows();

}

public function j\_trans\_kos($where=""){

$sql = "select \* from tb\_psn\_usr\_kost,tb\_trans\_usr\_kost,tb\_kos\_t\_onl ".$where;

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

return $query->num\_rows();

}

public function edit\_kos\_usr($tabelName,$data,$where){

$res = $data = $this->db->update($tabelName,$data,$where);

return $res;

}

public function data\_usr($where=""){

$sql = "select \* from tb\_u\_ser\_kos\_t ".$where;

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

return $query;

}

public function cari\_kd\_kos($where=""){

$sql ="select RIGHT(kd\_kos\_t,6) AS kdmax from tb\_kos\_t\_onl ORDER BY kd\_kos\_t DESC LIMIT 1".$where;

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

return $query;

}

public function cari\_kd\_pesann($where=""){

$sql ="select RIGHT(id\_psn\_ks\_us,6) AS kdmax from tb\_psn\_usr\_kost ORDER BY id\_psn\_ks\_us DESC LIMIT 1".$where;

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

return $query;

}

public function cari\_kd\_trans($where=""){

$sql ="select RIGHT(no\_fak\_trans\_us,6) AS kdmax from tb\_trans\_usr\_kost ORDER BY no\_fak\_trans\_us DESC LIMIT 1".$where;

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

return $query;

}

public function cek\_pesann($where=""){

$sql ="select \* from tb\_psn\_usr\_kost ".$where;

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

return $query;

}

public function metod\_pemb($tabelName,$data,$where){

$res = $data = $this->db->update($tabelName,$data,$where);

return $res;

}

public function rek($where=""){

$sql ="select \* from tb\_rek\_ks\_awk ".$where;

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

return $query;

}

public function nohp\_pem($where=""){

$sql = "select tb\_u\_ser\_kos\_t.no\_hp\_us,tb\_u\_ser\_kos\_t.nml\_us,tb\_kos\_t\_onl.no\_ktp\_us,tb\_kos\_t\_onl.hrg\_ks,tb\_psn\_usr\_kost. \* from tb\_psn\_usr\_kost,tb\_kos\_t\_onl,tb\_u\_ser\_kos\_t ".$where;

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

return $query;

}

public function trans\_kos($tabelName,$data){

$res = $data = $this->db->insert($tabelName,$data);

return $res;

}

public function pesan\_kontak\_us($where=""){

$sql ="select \* from tb\_kmen\_usr".$where;

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

return $query;

}

public function cek\_trans($where=""){

$sql ="select \* from tb\_trans\_usr\_kost ".$where;

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

return $query;

}

public function data\_trans\_adm($where=""){

$sql ="select tb\_u\_ser\_kos\_t.nml\_us,tb\_psn\_usr\_kost.id\_psn\_ks\_us,tb\_trans\_usr\_kost. \* from tb\_trans\_usr\_kost,tb\_psn\_usr\_kost,tb\_u\_ser\_kos\_t ".$where;

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

return $query;

}

public function data\_psn\_adm($where=""){

$sql ="select tb\_u\_ser\_kos\_t.nml\_us,tb\_psn\_usr\_kost. \* from tb\_psn\_usr\_kost,tb\_u\_ser\_kos\_t ".$where;

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

return $query;

}

public function bg\_slider($where=""){

$sql ="select \* from tb\_slider ".$where;

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

return $query;

}

public function c\_trans($tabelName,$data,$where){

$res = $data = $this->db->update($tabelName,$data,$where);

return $res;

}

public function kos\_confirm($tabelName,$data,$where){

$res = $data = $this->db->update($tabelName,$data,$where);

return $res;

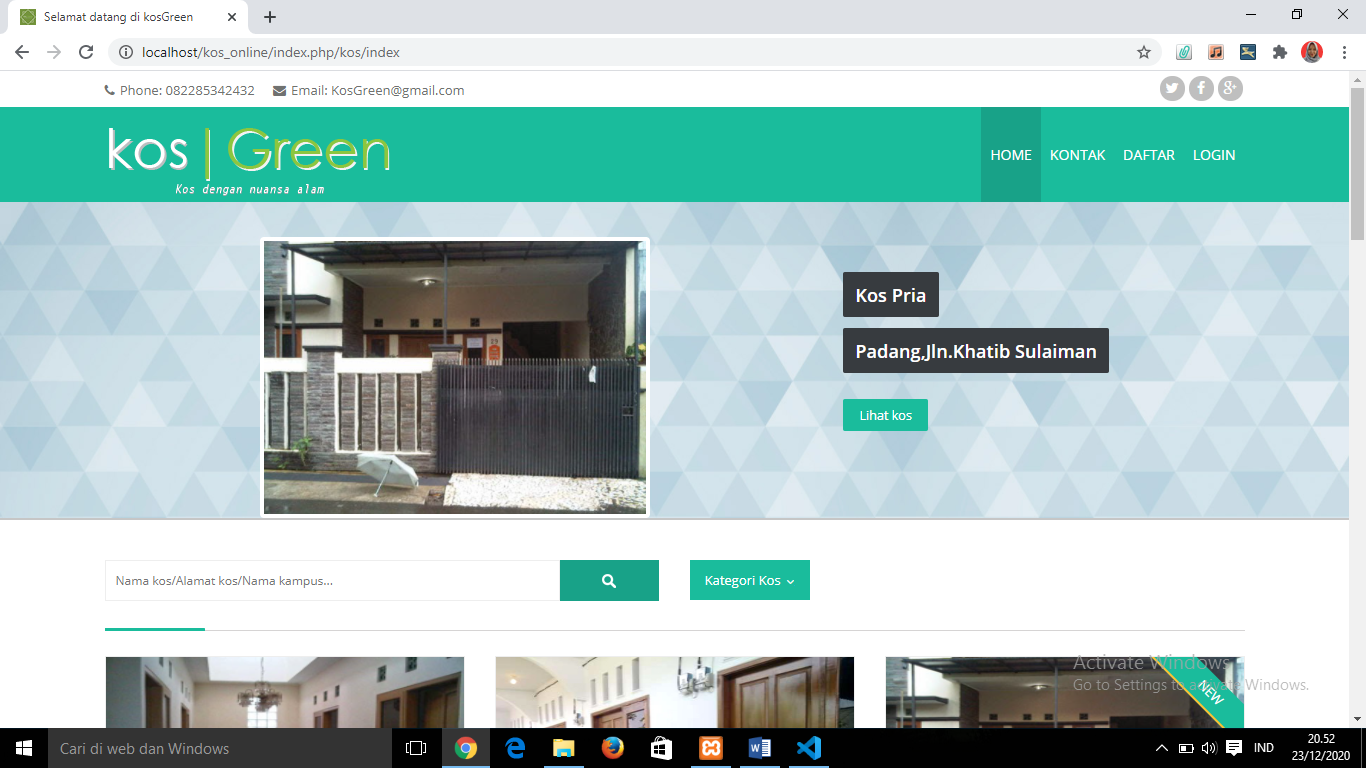
}

}

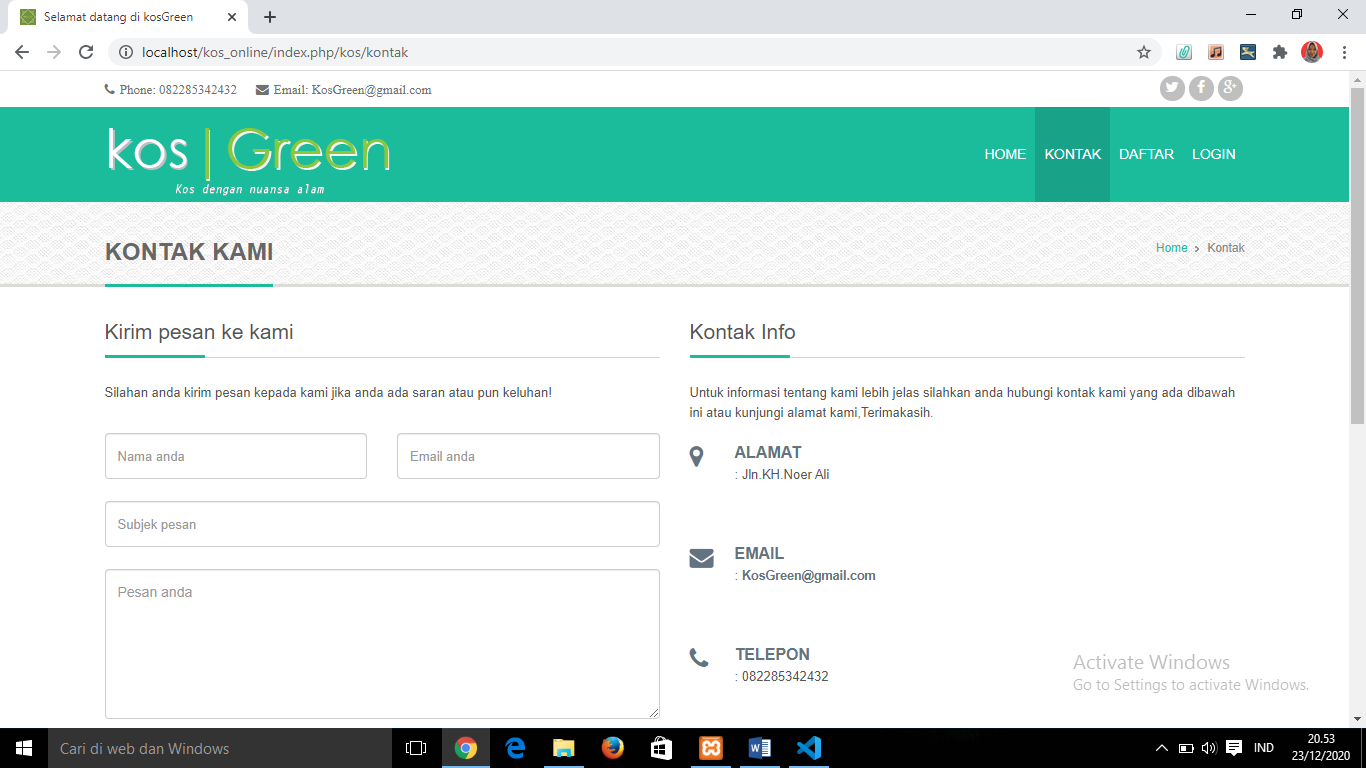
1. File View

View merupakn bagian yang menangani halaman user interface atau halaman yang mundul pada user. Terdapat banyak file view.

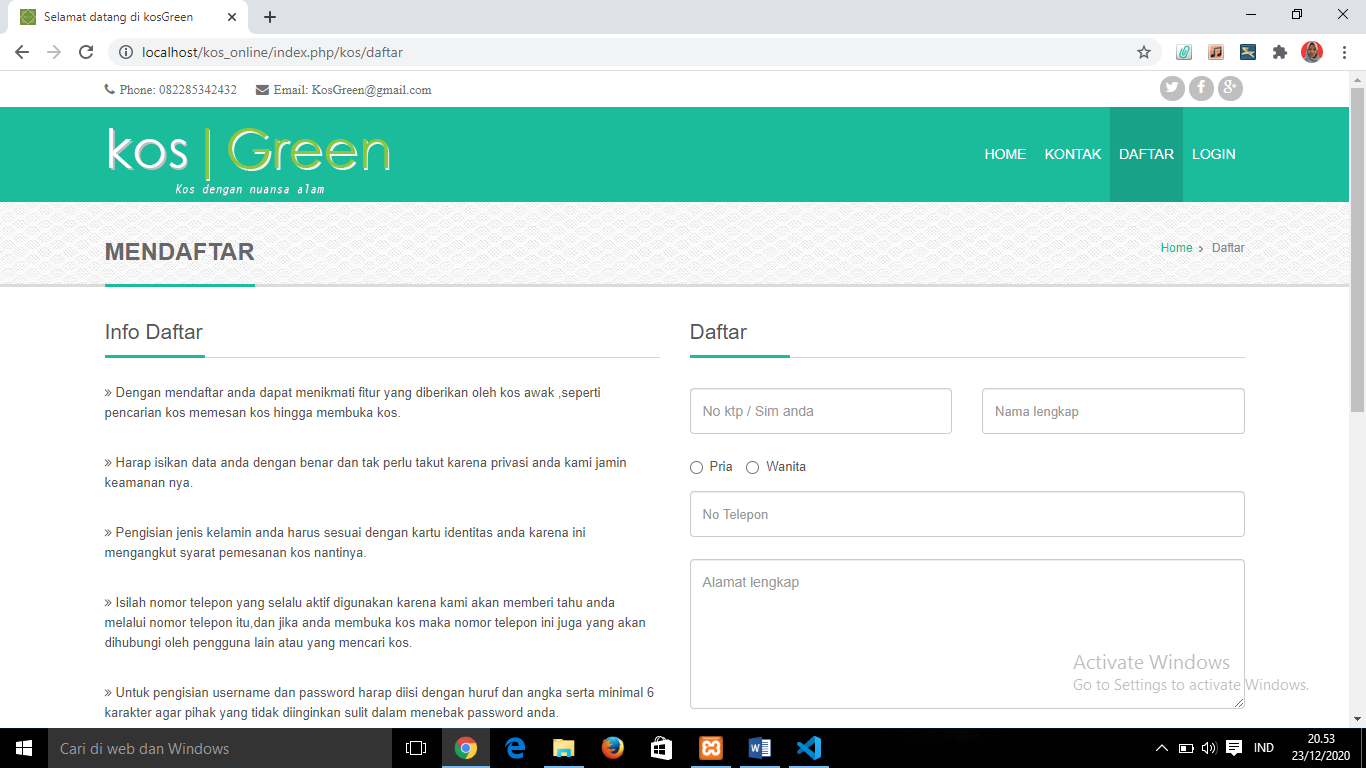
1. **Tampilan**
2. Halaman Home



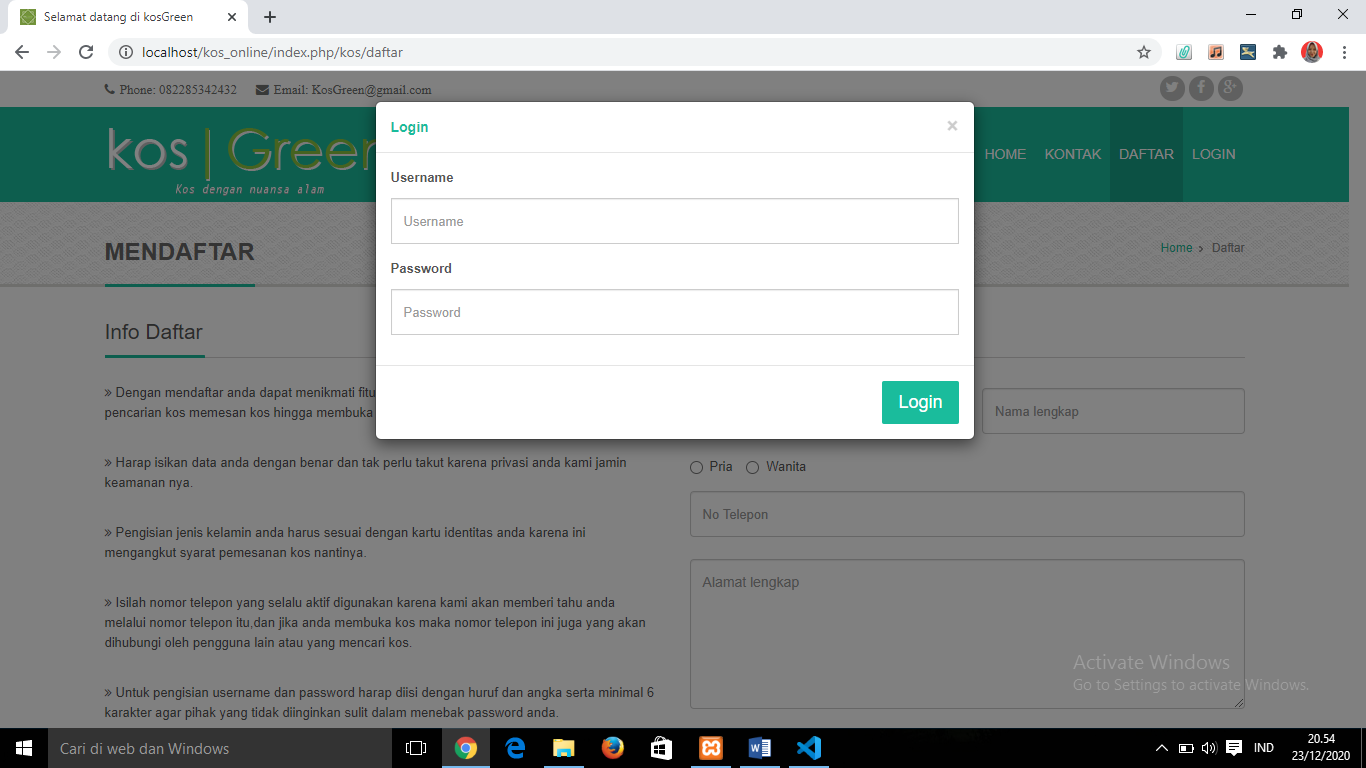
1. Halaman Kontak



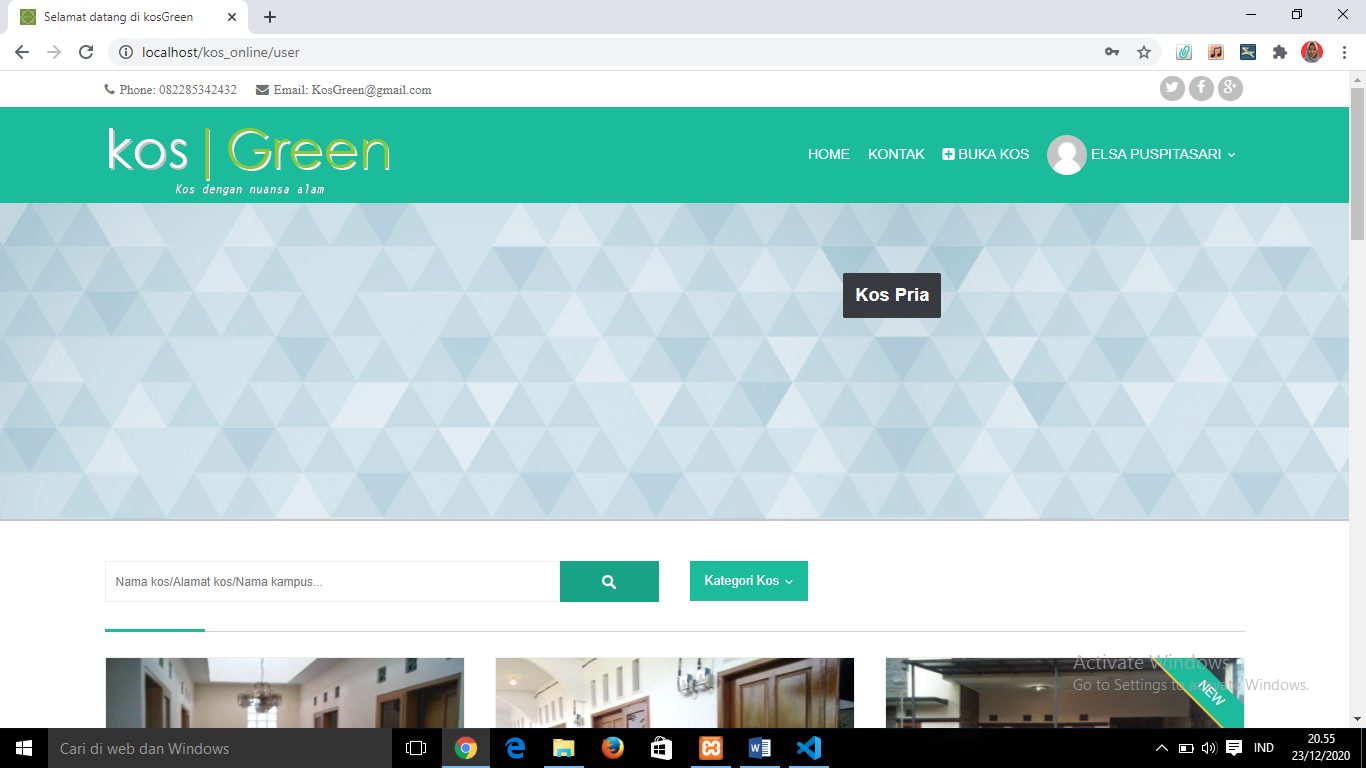
1. Halaman Daftar



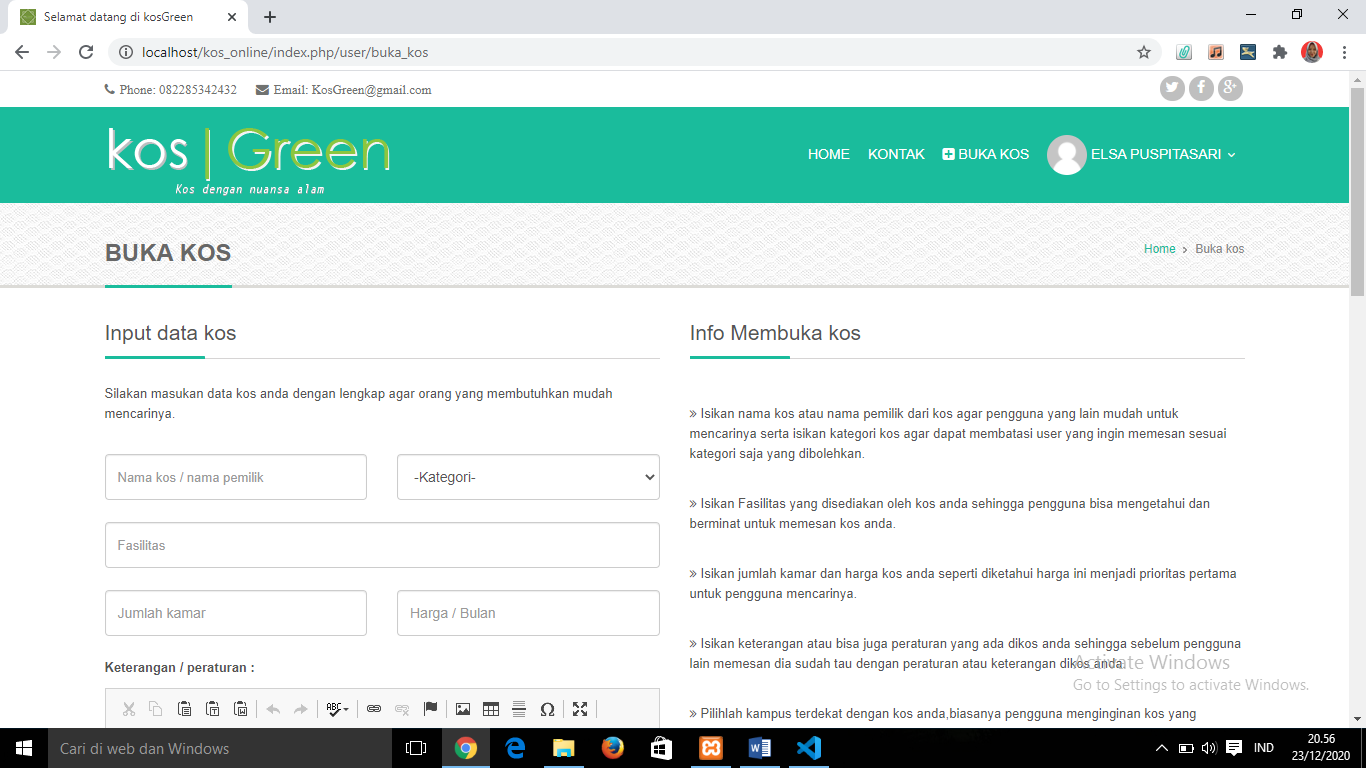
1. Halaman Login



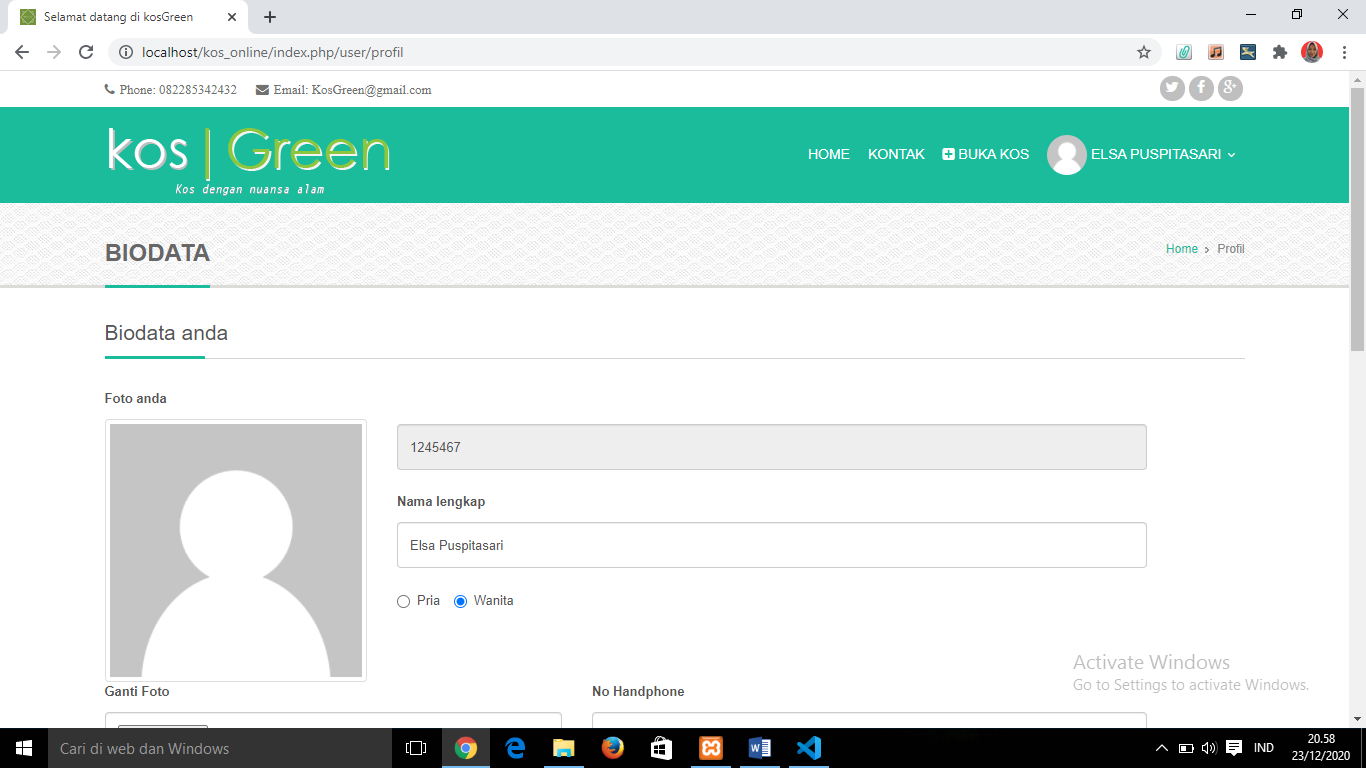
1. Halaman sesudah Login



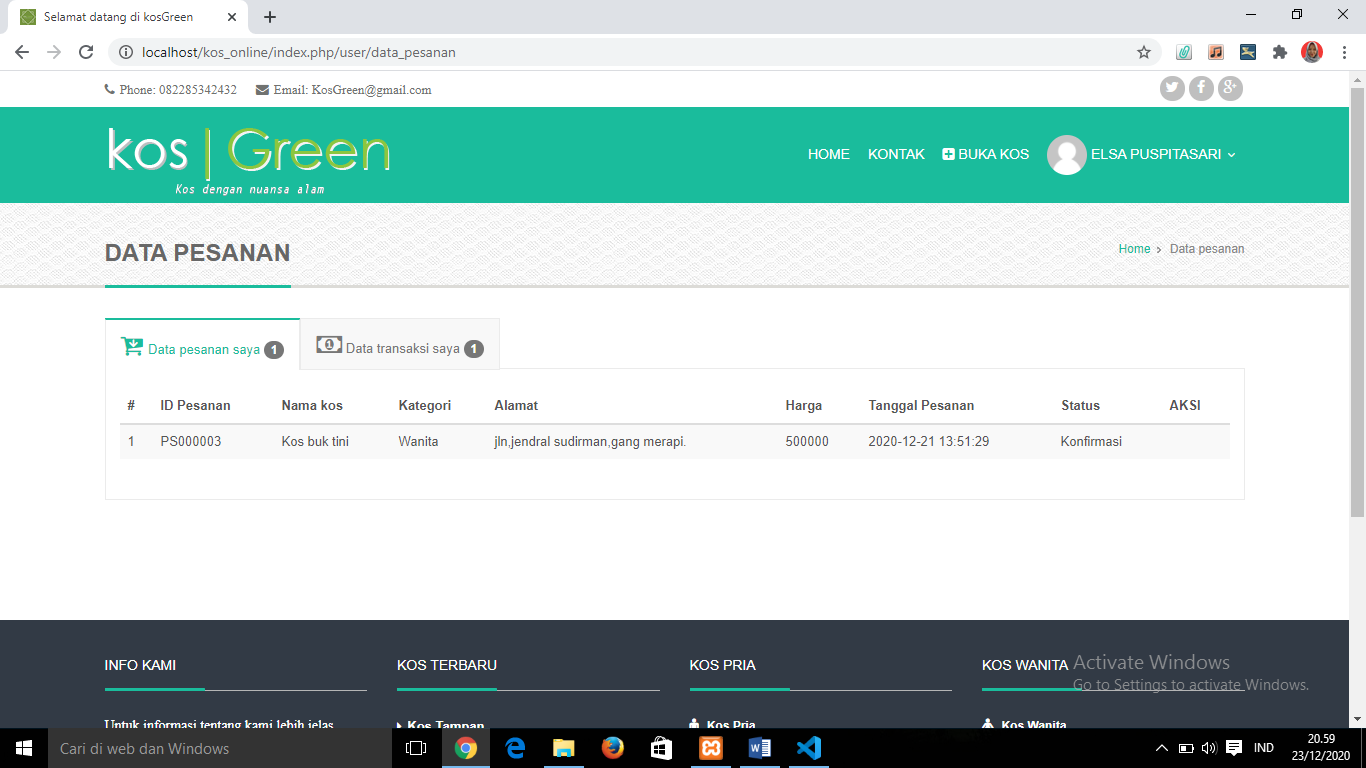
1. Halaman Buka Kos



1. Halaman Profil



1. Halaman Pesanan



# **BAB IV**

# **PENUTUP**

1. **Kesimpulan**

Codeigtiner adalah aplikasi open sourse yang berupa framework dengan model MVC (Model View Controller) untuk membangun website dinamis dengan menggunakan PHP. Codeigtiner memiliki lebih banyak kelebihan daripada framework lainnya.

Website KosGreen ini dapat sangat berguna bagi semua kalangan, terutama pembisnis kost. Pemilik atau pembisnis kost tisak harus menyiapkan brosur untuk promosi karena pada era ini semua dapat kita akses melalui web.

1. **Saran**

Dari pembahasan yang telah kami paparkan diatas masih banyak kekurangan karena kami masih dalam tahap belajar dan masih banyak fitur-fitur yang kurang karena keterbatasan kami.

Untuk itu kami juga memberikan saran untuk kami dan juga para mahasiswa :

1. Harus lebih mendalami materi mengenai codeigtiner, karena kami menyadari masih banyak kekurangan dari program kami.
2. Agar makalah kami jadi acuan belajar bagi pembaca.