BAB IV

HASIL DAN PEMBAHASAN

4.1 Pengembangan Sistem

Sistem pengambilan ide skripsi berbasis *website* dikembangkan dengan menggunakan Bahasa pemrograman PHP dan *framework* Codeigniter. Codeigniter menggunakan metode MVC, meliputi *Model, View*, dan *Controller*. *Model* berisi fungsi-fungsi untuk mengakses *database*. *View* berfungsi untuk mengatur antarmuka pengguna. *Controller* berfungsi sebagai penghubung antara *model* dan *view*.

4.2 Pembuatan Sistem Pengambilan Ide Skripsi

Berdasarkan rancangan yang telah dibahas pada bab 3, maka dibuatlah sistem pengambilan ide skripsi menggunakan Bahasa pemrograman PHP dan *framework* Codeigniter.

4.2.1 Halaman Beranda

Beranda adalah tampilan awal ketika pengguna mengakses *website*. Tampilan beranda dapat dilihat pada Gambar 4.1.



Gambar 4. 1 Tampilan beranda

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Beranda extends CI_Controller {
    public function index()
    {
        sthis->load->view('beranda');
    }
}
```

Gambar 4. 2 Coding controller beranda

```
<?php
$currentPage= 'beranda';
include('/layout/header.php');
?>
    <section id="(imgber)anda">
    </section>
<?php include('/layout/footer.php'); ?>
```

Gambar 4.3 Coding tampilan beranda

Gambar 4.2 memperlihatkan *coding controller* Beranda yang memanggil tampilan beranda dalam folder *views*. *Coding* tampilan beranda dapat dilihat pada Gambar 4.3. Pada *coding* tampilan beranda ini, dapat dilihat bahwa tampilan beranda menggunakan *header* dan *footer* yang dipanggil melalui fungsi *include*.

4.2.2 Halaman Registrasi Mahasiswa

Halaman registrasi mahasiswa adalah halaman dimana mahasiswa dapat mendaftarkan akun. Mahasiswa harus mengisi nama, NIM, prodi, fakultas, alamat, nomor hp, jumlah sks, IPK, username, email, dan password. Tampilan halaman registrasi mahasiswa dapat dilihat pada Gambar 4.4.

Construction of the second sec					
Gocalhost:81/ideskripsi/Register	C ⁴ Q, Search	合自じ	+	â×-	• =
	Pendaftaran Akun Mahasiswa				
	The second se				
	Nama				
	NIM				
	Prodi 👻				
	Fakultas				
	Alamat				
	14.10				
	NO. HP				
	Jumlah SKS				
	IPK (contoh: 3.5)				
	Constant of the second s				

Gambar 4. 4 Tampilan registrasi mahasiswa

Coding controller registrasi mahasiswa dapat dilihat pada Gambar 4.5. Dalam fungsi *index* dapat dilihat bahwa sistem melakukan validasi input, yaitu input harus diisi, panjang minimal dan maksimal, input unik, input *integer*, pencocokan input, dan validasi *email*. Jika validasi benar, data akan dimasukkan ke dalam *database* melalui fungsi daftar yang ada dalam M_account pada folder *views*. *Coding* fungsi daftar dapat dilihat pada Gambar 4.6.

Pada Gambar 4.8 dapat dilihat *coding* tampilan register mahasiswa. Pada input NIM, nomor HP, jumlah SKS, dan IPK terdapat attribut *onkeypress* yang mana pengguna hanya dapat mengetikkan karakter yang telah ditentukan dalam javascript pada Gambar 4.7.

Gambar 4. 5 Coding register mahasiswa



```
- -
<script type="text/javascript">
function isNumber(evt) {
evt = (evt) ? evt : window.event;
var charCode = (evt.which) ? evt.which : evt.keyCode;
if (charCode > 31 && (charCode < 48 || charCode > 57)) {
   return false;
}
return true;
}
function isFloat(evt) {
    var theEvent = evt || window.event;
var key = theEvent.keyCode || theEvent.which;
          key = String.fromCharCode(key);
          if (key.length == 0) return;
          var regex = /^[0-9.\b]+$/;
          if (!regex.test(key)) {
               theEvent.returnValue = false;
               if (theEvent.preventDefault) theEvent.preventDefault();
          }
</script>
```

Gambar 4.7 Coding javascript pada register mahasiswa

```
<section class="container">
   <div class="login">
    <hi>>Pendaftaran Akun Mahasiswa</hi>
    <form method="post">
     <?php echo form_open('register');?>
     <input type="text" name="name" placeholder="Nama" value="<?php echo set_value('name'); ?>" required />
 <?php echo form_error('name'); ?> 
     cinput type="text" name="nim" placeholder="NIM" value="<?php echo set_value('nim'); ?>" onkeypress="return isNumber(event)" required /

cect nume=nim pideenoider.

cect nume=nim pideenoider.

<select name="prodi" class="form-control" required>
<sption value="" disabled selected style="display: none;">Prodi</option>
<option value="reknologi Informasi">Teknologi Informasi</option>
</select>
</re>

cp> <?php echo form_error('prodi'); ?> 
      <select name="fakultas" class="form-control" required>
<option value="" disabled selected style="display: none;">Fakultas</option>
<option value="Teknik">Teknik</option>
</select>

 <?php echo form_error('fakultas'); ?> 
     <input type="text" name="alamat" placeholder="Alamat" value="<?php echo set_value('alamat'); ?>" required />

(p) 
     <input type="text" name="nohp" placeholder="No. HP" value="<?php echo set_value('nohp'); ?>" onkeypress="return isNumber(event)" requi

echo form_error('nohp'); >> 

     <input type="text" name="sks" placeholder="Jumlah SKS" value="<?php echo set_value('sks'); ?>" onkeypress="return isNumber(event)" req
     echo form_error('sks'); >> 
     time="ipk" placeholder="IPK (contoh: 3.5)" value="<?php echo set_value('ipk'); ?>" onkeypress="return isFloat(event
 </php echo form_error('ipk'); ?> 
     >
     <input type="text" name="username" placeholder="Username" value="<?php echo set_value('username'); ?>" required />

     <input type="text" name="email" placeholder="Email" value="<?php echo set_value('email'); ?>" required />

cho form_error('email'); ?> 
     <cp>type="password" name="password" placeholder="Password" value="<?php echo set_value('password'); ?>" required />

cp> <?php echo form_error('password'); ?> 
     type="password" name="password_conf" placeholder="Confirm Password" value="<?php echo set_value('password_conf'); ?>" required

echo form_error('password_conf'); ?> 
     <input type="hidden" name="idusers" value="2"/>

<input type="submit" name="btnSubmit" value="Daftar" />
```

Gambar 4.8 Coding tampilan register mahasiswa

4.2.3 Halaman Registrasi Dosen

Halaman registrasi dosen adalah halaman dimana dosen dapat mendaftarkan akun. Dosen harus mengsisi nama, NIDN, prodi, fakultas, alamat, nomor hp, pendidikan terakhir, keahlian, username, email, dan password. Tampilan halaman registrasi dosen dapat dilihat pada Gambar 4.9.

N		0.0			40	v
D. Incompare more subserve providence		L C T A states	¥ 8	• •	đ,	~ -
	Pendaftaran Akun I	Dosen				
	Nama					
	NIDN					
	Prod					
	Fakultas	*				
	Alamat					
	No. HP					
	Pendidkan Terakhir	v				
	Keahlan					

Gambar 4.9 Tampilan registrasi dosen

Coding controller registrasi dosen dapat dilihat pada Gambar 4.10. Dalam fungsi *index* dapat dilihat bahwa sistem melakukan validasi input, yaitu input harus diisi, panjang minimal dan maksimal, input unik, input *integer*, pencocokan input, dan validasi *email*. Jika validasi benar, data akan dimasukkan ke dalam *database* melalui fungsi daftardosen yang ada dalam M_account pada folder *views*. *Coding* fungsi daftar dapat dilihat pada Gambar 4.11.

Pada Gambar 4.13 dapat dilihat *coding* tampilan register mahasiswa. Pada input NIDN dan nomor HP terdapat attribut *onkeypress* yang mana pengguna hanya dapat mengetikkan karakter yang telah ditentukan dalam javascript pada Gambar 4.12.







```
<script type="text/javascript">
function isNumber(evt) {
evt = (evt) ? evt : window.event;
var charCode = (evt.which) ? evt.which : evt.keyCode;
if (charCode > 31 && (charCode < 48 || charCode > 57)) {
    return false;
    }
    return true;
}
</script>
```



```
<section class="container">
    <div class="login">
    <div class="login">
    <hi>Pendaftaran Akun Dosen</hi>
    <form method="post">
     <?php echo form_open('register');?>
      <input type="text" name="name" placeholder="Nama" value="<?php echo set_value('name'); ?>" required/>
      echo form_error('name'); ?> 
      <input type="text" name="nidn" placeholder="NIDN" value="<?php echo set_value('nidn'); ?>" onkeypress="return isNumber(event)" required
      echo form_error('nidn'); ?> 
      cselect name="prodi" class="form-control" required>
coption value="" disabled selected style="display: none;">Prodi</option>
coption value="Teknologi Informasi">Teknologi Informasi</option>
</select>
 <?php echo form_error('prodi'); ?> 
      <cp><selct name="fakultas" class="form-control" required>
<coption value="" disabled selected style="display: none;">Fakultas</option>
<option value=""reknik">Teknik</option>
</select>
 <?php echo form_error('fakultas'); ?> 
      ciput type="text" name="alamat" placeholder="Alamat" value="<?php echo set_value('alamat'); ?>" required />

echo form_error('alamat'); 

<input type="text" name="nohp" placeholder="No. HP" value="<?php echo set_value('nohp'); ?>" onkeypress="return isNumber(event)" requir

      <selct name="pendidikanterakhir" class="form-control" required>
<option value="" disabled selected style="display: none;">Pendidikan Terakhir</option>
<option value="S2">S2</option>
<option value="S2">S2</option>
        </select>
       <rp><input type="text" name="keahlian" placeholder="Keahlian" value="<?php echo set_value('keahlian'); ?>" required />
        <?php echo form_error('keahlian'); ?> 
       <cp><input type="text" name="username" placeholder="Username" value="<?php echo set_value('username'); ?>" required />

echo form_error('username'); ?> 
      type="text" name="email" placeholder="Email" value="<?php echo set_value('email'); ?>" required />

<?php echo form_error('email'); ?> 
      type="password" name="password" placeholder="Password" value="<?php echo set_value('password'); ?>" required />

</php echo form_error('password'); ?> 

<input type="password" name="password_conf" placeholder="Confirm Password" value="<?php echo set_value('password_conf'); ?>" required

<?php echo form_error('password_conf'); ?> 

      <input type="hidden" name="id_users" value="3"/>
```

Gambar 4. 13 Coding tampilan register dosen

4.2.4 Halaman Login

Halaman *login* berfungsi untuk memberikan keamanan di dalam sistem. Pengguna harus mengisi *username* dan *password* yang benar untuk dapat masuk ke dalam sistem. Tampilan halaman *login* dapat dilihat pada Gambar 4.14.

🗡 ide Skripsi X 🖓 localhost/81/127.0.0.1/ide X 🕂			- 0 ×
(Decalhost/81/ideskripsi/Login	C Q, Search	合自 🛡 🖡	÷ * × = ≡
	Lasta		
	Lògin		
	litemame		
	sauranne.		
	Password		
	Login		

Gambar 4. 14 Tampilan halaman login

Pada Gambar 4.15, dapat dilihat bahwa sistem melakukan validasi *login*. Jika *username* dan *password* sudah diisi, sistem akan melakukan validasi *login* menggunakan fungsi *login* dalan *library* simple_login. *Coding* fungsi *login* dapat dilihat pada Gambar 4.16.

```
<?php
     defined('BASEPATH') OR exit('No direct script access allowed');
     class Login extends CI_Controller {
          public function index() {
                // Fungsi Login
                $valid = $this->form validation;
               svalu = stils->rorm_valuation;
susername = $this->input->post('username');
spassword = $this->input->post('password');
$valid->set_rules('username','Username','required');
$valid->set_rules('password','Password','required');
                if($valid->run()) {
                     $this->simple_login->login($username,$password);
                }
                 // End fungsi login
               $this->load->view('account/v_login');
          }
          public function logout(){
               $this->simple_login->logout();
          }
    }
```

Gambar 4. 15 Coding controller Login

public function login(\$username, \$password) {

3

```
//cek username dan passi
 $query = $this->CI->db->get_where('mahasiswa',array('username_mhs'=>$username,'password_mhs' => md5($password)));
$query2 = $this->CI->db->get_where('dosen',array('username_dsn'=>$username,'password_dsn' => md5($password)));
$query3 = $this->CI->db->get_where('admin',array('username_adm'=>$username,'password_adm' => md5($password)));
 if($query->num_rows() == 1) {
          squety-snum__os() == 1/
//ambil da user berdasar username
$row = $this-xCI-xdb-xquery('SELECT nim_mhs FROM mahasiswa where username_mhs = "'.$username.'"');
$row2 = $this-xCI-xdb-xquery('SELECT id_users FROM mahasiswa where username_mhs = "'.$username.'"');
$row3 = $this-xCI-xdb-xquery('SELECT nama_mhs FROM mahasiswa where username_mhs = "'.$username.'"');
          $admin = $row->row();
$tipe = $row2->row();
$nama = $row3->row();
          $id
                        = $admin->nim mhs;
           $tipeuser = $tipe->id_users;
$namauser= $nama->nama_mhs;
               /set session use
          //set session user
$this->CL->session->set_userdata('username', $username);
$this->CL->session->set_userdata('id_login', uniqid(rand()));
$this->CL->session->set_userdata('id_user', $id);
$this->CL->session->set_userdata('tipeuser'); $tipeuser);
$this->CL->session->set_userdata('namauser', $namauser);
          //redirect ke halaman dashboard
redirect(site_url('dashboard'));
  elseif($query2->num_rows() == 1) {
          //ambil data user berdasar username
frow = $this->CL=>db=>query('SELECT nidn_dsn FROM dosen where username_dsn = "'.$username.'");
frow2 = $this->CL=>db=>query('SELECT id_users FROM dosen where username_dsn = "'.$username.'");
frow3 = $this->CL=>db=>query('SELECT nama_dsn FROM dosen where username_dsn = "'.$username.'");
          $admin = $row->row();
$tipe = $row2->row();
$nama = $row3->row();
          $id = $admin->nidn_dsn;
$tipeuser = $tipe->id_users;
$namauser= $nama->nama_dsn;
            //set session use
          //set session user
fthis-vCI->session->set_userdata('username', $username);
fthis-vCI->session->set_userdata('id_login', uniqid(rand()));
fthis-vCI->session->set_userdata('id_user', $id);
fthis-vCI->session->set_userdata('tiguser', $tiguser);
sthis-vCI->session->set_userdata('namauser', $namauser);
            //redirect ke halaman dashboa
           redirect(site_url('dashboard'));
   elseif($query3->num_rows() == 1) {
          //ambil data user berdasar username
Srow = $this->CI->db->query('SELECT id_adm FROM admin where username_adm = "'.$username.'"');
Srow2 = $this->CI->db->query('SELECT id_users FROM admin where username_adm = "'.$username.'"');
$row3 = $this->CI->db->query('SELECT nama_adm FROM admin where username_adm = "'.$username.'"');
                                                       berdasar u
          $admin = $row->row();
$tipe = $row2->row();
$nama = $row3->row();
           $id = $admin->id_adm;
$tipeuser = $tipe->id_users;
$namauser= $nama->nama_adm;
          $id
              //set session user
          //set session user
$this->CL->session->set_userdata('username', $username);
$this->CL->session->set_userdata('id_login', uniqid(rand()));
$this->CL->session->set_userdata('id_user', $id);
$this->CL->session->set_userdata('tipeuser', $tipeuser);
$this->CL->session->set_userdata('namauser', $namauser);
          //redirect ke halaman dashboard
redirect(site_url('dashboard'));
}
else{
            //iika tidak ada, set notifikasi dalam flashdata.
           $this->CI->session->set_flashdata('sukses','Username atau password anda salah, silakan coba lagi.. ');
                   edirect ke halaman login
          redirect(site_url('login'));
 }
   return false;
```

Gambar 4. 16 Coding fungsi login

Jika validasi *login* benar, sistem mengalihkan ke *controller dashboard*. *Coding controller dashboard* dapat dilihat pada Gambar 4.17. Pada fungsi *index*, dapat dilihat bahwa tiap tipe akun diarahkan ke halaman *dashboard* masing-masing.

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Dashboard extends CI_Controller{
   function __construct(){
       parent::__construct();
       $this->simple_login->cek_login();
   }
   7/load halaman dashboard
   public function index(){
         if($this->session->userdata('tipeuser')== '2') {
            redirect(site_url('Mahasiswa'));
        }
        elseif($this->session->userdata('tipeuser') == '3') {
            redirect(site_url('Dosen'));
        }
        elseif($this->session->userdata('tipeuser') == '1') {
            redirect(site_url('Administrator'));
        3
   }
}
```

Gambar 4. 17 Coding controller dashboard

4.2.5 Halaman Dashboard Administrator

Halaman *dashboard* administrator adalah halaman awal yang tertampil ketika administrator melakukan *login* ke dalam sistem. Pada *dashboard* administrator, terdapat menu manajemen akun mahasiswa dan dosen. Tampilan halaman *dashboard* administrator dapat dilihat pada Gambar 4.18.

) () localhost 81/ideskripsi/	Administrator		C ^t Q, Search	☆ 自 ♥	↓ ☆ X =
dministrator					🛔 Tatan
Dashboard		Administrator			
Manajemen Akun	۰.	Administrator			
		Selamat datang di dashboard Administrator!			

Gambar 4. 18 Tampilan dashboard administrator

Pada menu manajemen akun mahasiswa, Administrator dapat melihat data akun mahasiswa, melakukan pencarian, pengurutan data, aktivasi akun, mengubah *password*, dan menghapus akun mahasiswa. Tampilan dapat dilihat pada Gambar 4.19, Gambar 4.20, Gambar 4.21, Gambar 4.22, Gambar 4.23, Gambar 4.24.

🕴 Administrator 🛛 🗙	Jocalhost 81 / 127.0.0.1 / ide ×	+					- 6
(i) localhost 81/ideskripsi/Adminis	trator/data_mahasiowa		C Q, Search		合自 🛡	∔ ń X	- 😣
Administrator							
n Dashboard	Data Ma	hasiswa					
🚠 Manajemen Akun	Data Ma	1114515774					
Akun Mahasiswa							
Akun Dosen	Type your se	arch word				Q Search	
	Total records - 40						
	NIM	Nama	Prodi	Fakultas	Status	Action	1
	20130140006	Angga Putra Handika	Teknik Informatika	Teknik	ACTIVE	52	
	20130140276	Anggita Nur	Teknik Informatika	Teknik	NON ACTIVE	<u> </u>	8
	20130140130	Bachtiar Madya Perma	Teknik Informatika	Teknik	ACTIVE	N	
	20120130165	Citra Lestari	Teknik informatika	Teknik	NON ACTIVE	S /	
	20130140005	Dicky Faizal Huda	Teknik Informatika	Teknik	ACTIVE	8 /	
	20120130162	Dinda kirana	Teknik Informatika	Teknik	ACTIVE	S /	
	20130140139	Dirga Rama Sudira	Teknik Informatika	Teknik	ACTIVE	5 /	
	20130140134	Faizal Azhari	Teknik Informatika	Teknik	NON ACTIVE	S /	
	20130140013	Fathania Firwan Firdaus	Teknik Informatika	Teknik	ACTIVE		

Gambar 4. 19 Tampilan manajemen akun mahasiswa

dministrator Dashboard						
I Dashboard						👗 Tatan
	Data Mah	aciewa				
Manajemen Akun 👻	Data Man	a313Wa				
Akun Mahasiswa						and the second se
Akun Dosen	safira					Q Search
	Total records - 1					
	NIM	Nama	Prodi	Fakultas	Status	Action



🕴 Administrator 🛛 🗙	📑 localhost:81 / 127.0.0.1 / ide 🗙	+				- 0	×
(Iocalhost 81/ideskripsi/Admin	istrator/data_mahasiswa/NIM_MHS/de	uc71/	C Q Searc	ħ	合自 🛡	∔ * X = Ø	=
Administrator						🛔 Tatania	* :
2 Dashboard	Data Ma	hasiswa					
🛦 Manajemen Akun		IIIdSISWa					
Akun Mahasiswa						and the second se	
Akun Dosen	Type your se	arch word				Q Search	
	Total records - 40						
	NIM	Nama	Prodi	Fakultas	Status	Action	
	20120130162	Dinda kirana	Teknik Informatika	Teknik	ACTIVE	S 🖉 🖻	
	20120130165	Citra Lestari	Teknik Informatika	Teknik	NON ACTIVE	S / 8	
	20130140001	Kwon Jlyong	Teknik Informatika	Teknik	ACTIVE	S / 8	
	20130140002	Tom Holland	Teknik Informatika	Teknik	NON ACTIVE	S / 8	
	20130140005	Dicky Faizal Huda	Teknik Informatika	Teknik	ACTIVE	S 🖉 😫	
	20130140006	Angga Putra Handika	Teknik Informatika	Teknik	ACTIVE	S / 8	
	20130140007	Nadya Aprillia Ardiyanti	Teknik Informatika	Teknik	ACTIVE	S / 8	
	20130140008	Sigid Nur Hidayat	Teknik Informatika	Teknik	ACTIVE	S / 8	
	20130140009	Sinta Dela	Teknik Informatika	Teknik	NON ACTIVE	S 🖉 🖻	

Gambar 4. 21 Tampilan data mahasiswa urut berdasarkan NIM

🕴 Administrator 🛛 🗙		+					- 0
🜔 🛈 localhost:81/ideskripsi/Admin	nistrator/data_mahasiswa		C Q, Search		合自 🛡	+ +)	x 🗕 🐼
Administrator						Ű.	🌢 Tatania
R Dashboard	Data M	hacicwa					
🛦 Manajemen Akun		anasiswa					
Akun Mahasiswa	Activated Sucess	tully					
Akun Dosen							
	Type your s	sarch word				Q Searc	•
	Total records - 40		200	T-1			
	NIM	Nama	Prodi	Pakultas	status	Acti	on
	20130140276	Anggita Nur	Teknik informatika	Teknik	NON ACTIVE		
	20130140130	Bachtiar Madya Perma	Teknik Informatika	Teknik	ACTIVE	a 2	R
	20120130165	Citra Lestari	Teknik Informatika	Teknik	NON ACTIVE	R /	8
	20130140005	Dicky Faizal Huda	Teknik Informatika	Teknik	ACTIVE	a /	R
	20120130162	Dinda kirana	Teknik Informatika	Teknik	ACTIVE	S /	R
	20130140139	Dirga Rama Sudira	Teknik Informatika	Teknik	ACTIVE	<u> </u>	8
	20120140124	Esital Atheri	Taknik Informatika	Takolk	NON ACTIVE		

Gambar 4. 22 Tampilan berhasil aktivasi akun mahasiswa

(localhost 81/ideskripsi/Adminis	trator/edit_mhs/20130140006	C Q Search	☆	e 🛡	+	ń	X	- 50	=
Administrator							1		
 Dashboard Manajemen Akun 	Ubah Data	20130140006 Angga Putra Handika							
	Fakutas Prodi Password	Teknik Teknik Informatika 11toscos70652401100036072516266661 V Submit D Reset							

Gambar 4. 23 Tampilan ubah password akun mahasiswa

F Administrator	× mik	ocalhost81/127.0.0.1/ide ×	+				-	5 X
(i) localhost:81/ideskripsi/Ad	dministrator/	data_mahasiswa		C Q, Search		☆ 自 ♥	+ ÷ ×=	≈ ≡
Administrator							🛔 Tat	ania -
Dashboard		Data Ma	haciewa					
🚠 Manajemen Akun	19	Data Ma	IIIdSISWa					
Akun Mahasiswa		Deleted Successful	N					
Akun Dosen								
		Type your se	arch word				Q Search	
		Total records - 39						
		NIM	Nama	Prodi	Fakultas	Status	Action	
		20130140006	Angga Putra Handika	Teknik informatika	Teknik	ACTIVE	S 2 8	
		20130140276	Anggita Nur	Teknik informatika	Teknik	NON ACTIVE	N Z B	
		20130140130	Bachtiar Madya Perma	Teknik informatika	Teknik	ACTIVE	S 2 8	
		20120130165	Citra Lestari	Teknik informatika	Teknik	NON ACTIVE	S 2 8	
		20130140005	Dicky Faizal Huda	Teknik informatika	Teknik	ACTIVE	S 2 8	
		20120130162	Dinda kirana	Teknik Informatika	Teknik	ACTIVE	S 2 8	
		20130140139	Dirga Rama Sudira	Teknik Informatika	Teknik	ACTIVE	S 🖊 🖻	
		00100140194	Esital Athari	Tabnik informatika	Tabuik	NON ACTIVE		

Gambar 4. 24 Tampilan berhasil menghapus akun mahasiswa

Pada Gambar 4.25, dapat dilihat bahwa fungsi data_mahasiswa memanggil fungsi *get_mahasiswa* dan *record_count* yang ada pada M_Mhs dalam folder *models*. Fungsi get_mahasiswa digunakan untuk mendapatkan data mahasiswa dari *database*. Sedangkan fungsi *record_count* digunakan untuk menghitung jumlah data mahasiswa. *Coding* fungsi *get_mahasiswa* dan *record_count* dapat dilihat pada Gambar 4.26. public function data_mahasiswa()

\$data = array(); \$data['sort_cols'] = array('NIM_MHS' => 'NIM', 'NAMA_MHS' => 'Nama', 'PROI_MHS' => 'Prodi', 'FAKULTAS_MHS' => 'Fakultas', 'STATUS'=>'Status'

); \$config = array(); //base_url(). 'index.php/questions/page/'.\$sortfield.'/'.\$order.'/', \$search_string = \$this->input->post('search');

\$config["per_page"] = 10; //max number of page links \$config['num_links'] = 2; //use page number as parameter \$config['use_page_numbers'] = TRUE;

\$data['search_string'] = '';
if(!empty(\$search_string)) {

\$this->uri->segment(6, \$this->uri->segment(5, 1));
\$data['search_string'] = \$this->uri->segment(5, \$search_string);

}
//set default page uri
\$page_uri = 5;

if(!empty(\$data['search_string']))
\$page_uri = 6;

\$config["uri_segment"] = \$page_uri;

\$config["total_rows"] = \$this->mahasiswa->record_count(\$data['search_string']);

\$data['page'] = \$this->uri->segment(\$page_uri, 1);

\$data['sort_by'] = \$this->uri->segment(3, 'NAMA_VHS'); SorderBy = \$this->uri->segment(4, "desc"); Soffset = (\$data['page']-1) * \$config['per_page']; \$data['total_rows'] = \$config['total_rows']; if(SorderBy == "asc") \$data['sort_order'] = "desc"; else \$data['sort_order'] = "asc"; \$config['base_url"] = base_url().'Administrator/data_mahasiswa/'.\$data['sort_by'].'/'.\$orderBy.'/'.\$data['search_string']; \$config['full_tag_close'] = ' \$config['first_tag_open'] = 'all class="pagination">; \$config['first_tag_open'] = 'all class="prev page">'; \$config['first_tag_open'] = 'all class="prev page">; \$config['first_tag_open'] = 'all class="prev page">; \$config['first_tag_open'] = 'all class="next page">; \$config['last_tag_open'] = 'all class="next page">; \$config['last_tag_open'] = 'all class="next page">; \$config['next_tag_open'] = 'all clas

\$config['cur_tag_open'] = '';
\$config['cur_tag_close'] = '';

\$config['num_tag_open'] = '';
\$config['num_tag_close'] = '';

\$data["data"] = \$this->mahasiswa->get_mahasiswa(\$config["per_page"], \$offset, \$data['sort_by'], \$data['sort_order'], \$data['search_string'

\$this->pagination->initialize(\$config);
\$data["links"] = \$this->pagination->create_links();

```
$this->load->view('admin/v_datamahasiswa', $data);
}
```

Gambar 4. 25 Coding data mahasiswa

```
function get_mahasiswa($per_page, $offset, $sortfield, $orderBy, $search_string, $id=0)
  ſ
    if(empty($id)){
      //echo $per_page.'fff'.$offset.'fff'.$sortfield.'fff'.$orderBy;
     if(!empty($search_string)) {
       $this->db->like('NAMA_MHS',$search_string);
        $this->db->or_like('NIM_MHS',$search_string);
       $this->db->or_like('PRODI_MHS',$search_string);
       $this->db->or_like('FAKULTAS_MHS',$search_string);
     $this->db->order_by("$sortfield", "$orderBy");
     $this->db->limit($per_page,$offset);
     $query = $this->db->get('mahasiswa');
      if ($query->num_rows() > 0) {
        foreach ($query->result() as $row) {
         $data[] = $row;
        3
     return $data;
    3
    return false;
    } else {
   $query = $this->db->get_where('mahasiswa', array('NIM_MHS' => $id));
   return $query->row_array();
   3
 }
public function record_count($search_string) {
if(!empty($search_string)) {
 $this->db->like('NAMA_MHS',$search_string);
 $this->db->or_like('NIM_MHS',$search_string);
 $this->db->or_like('PRODI_MHS',$search_string);
 $this->db->or_like('FAKULTAS_MHS',$search_string);
}
  return $this->db->count_all_results("mahasiswa");
}
```



Pada Gambar 4.27, dapat dilihat *coding* fungsi *edit_mhs* dalam *controller* Administrator memanggil fungsi select dalam *model* mahasiswa untuk mengambil data mahasiswa dengan parameter nim_mhs. Administrator dapat melakukan *reset password* mahasiswa melalui fungsi *profile_mhs* yang ada pada *model* mahasiswa. *Password* disimpan ke dalam *database* menggunakan enkripsi md5. Fungsi *select* dan *profile_mhs* dapat dilihat pada Gambar 4.28.

functio	n edit_mhs(\$nim_mh s)
	<pre>if(\$_POST==NULL) { \$data['data_mhs'] = \$this->mahasiswa->select(\$nim_mhs); \$this->load->view('admin/v_resetmahasiswa',\$data); }else { \$data['password_mhs'] = md5(\$this->input->post('password_mhs')); \$nim_mhs = \$this->input->post('nim_mhs'); \$this->mahasiswa->profile_mhs(\$nim_mhs, \$data); redirect('Administrator/data_mahasiswa'); }</pre>

Gambar 4. 27 Coding fungsi edit mahasiswa

```
function select($id_user)
{
    return $this->db->get_where('mahasiswa', array('nim_mhs'=>$id_user))->row();
}
//UPDATE
function profile_mhs($iduser, $data)
{
    $this->db->where(array('nim_mhs' => $iduser, 'nim_mhs !=' => 0));
    $this->db->update('mahasiswa', $data);
}
```

Gambar 4. 28 Coding fungsi select dan profile_mhs

Pada Gambar 4.29, dapat dilihat fungsi *delete_mhs* dalam *controller* Administrator yang memanggil fungsi *delete_mhs* dalam *model* mahasiswa. Fungsi *delete_mhs* dalam *model* mahasiswa dapat dilihat pada Gambar 4.30.

```
public function delete_mhs($id)
{
    $this->mahasiswa->delete_mhs($id);
    $this->session->set_flashdata('message', 'Deleted Sucessfully');
    redirect('Administrator/data_mahasiswa');
}
```

Gambar 4. 29 Coding fungsi delete_mhs dalam controllers

```
public function delete_mhs($id)
{
    $this->db->where('NIM_MHS', $id);
    return $this->db->delete('mahasiswa');
}
```

Gambar 4. 30 Coding fungsi delete_mhs dalam models

Pada Gambar 4.31 dapat dilihat fungsi aktivasi akun mahasiswa dalam *controller* Administrator memanggil fungsi *act_mhs* dalam *model* mahasiswa. Fungsi *act_mhs* dapat dilihat pada Gambar 4.32.

```
function aktivasi_mhs($nim_mhs)
{
    $this->mahasiswa->act_mhs($nim_mhs);
    $this->session->set_flashdata('message', 'Activated Sucessfully');
    redirect('Administrator/data_mahasiswa');
}
```

Gambar 4. 31 Coding aktivasi mahasiswa

```
function act_mhs($id)
{
    $this->db->set('STATUS_MHS', '1');
    $this->db->where('NIM_MHS', $id);
    $this->db->update('mahasiswa');
}
```

Gambar 4. 32 Fungsi act_mhs

Pada menu manajemen akun dosen, Administrator dapat melihat data akun dosen, melakukan pencarian, pengurutan data, aktivasi akun, mengubah *password*, dan menghapus akun dosen. Tampilan dapat dilihat pada Gambar 4.33, Gambar 4.33, Gambar 4.35, Gambar 4.36, Gambar 4.37, Gambar 4.38.

Administrator	×	iocalhost:81 / 127.0.0.1 / ide 兴	+							- 8	No.
(i) localhost 81/ideskrips	i/Administrator	/deta_dosen		C 9.9	Search	合 自 5	5 +	ń	X	97	
Administrator											
Dashboard		Data Da	con								Ī
A Manajemen Akun	×	Dala Du	isen								
Akun Dosen	Type your se	sarch word					Q Se	arch			
		Total records - 9									
		NIDN	Nama	Prodi	Fakultas	Status		A	tion		
		0518048401	Aprilia Kurniati	Teknik Informatika	Teknik	NON ACTIVE		94	1		
		0526047401	Asroni	Teknik Informatika	Teknik	ACTIVE		4	/ 8		
		0515038702	Cahya Damarjati	Teknik Informatika	Teknik	ACTIVE		9	/ 8		
		0707108402	Chayadi Oktomy	Teknik Informatika	Teknik	ACTIVE		9	/ 8		
		0502026801	Dwijoko Purbohadi	Teknik Informatika	Teknik	ACTIVE		٩,	/ 8		
		0522046701	Eko Prasetyo	Teknik Informatika	Teknik	ACTIVE		4	/ 8		
		0511116901	Haris Setyawan	Teknik Informatika	Teknik	ACTIVE		9	/ 8		
		0521037603	Helmi Zain Nuri	Teknik Informatika	Teknik	NON ACTIVE		٩,	/ 8		
		0503068601	Reza Giga Isnanda	Teknik Informatika	Teknik	ACTIVE		٩,			

Gambar 4. 33 Tampilan manajemen akun dosen

Administrator >	× local	lhost:81 / 127.0.0.1 / ide 兴 📗	+					- 1	×
(Iocalhost 81/ideskripsi/Adm	ninistrator/dat	a_dosen/		C Q Sea	rch	合自 🛡	∔ ŝ	X — §	* ≡
Administrator								📥 Та	tania +
Dashboard		Data Daa	o p						
📥 Manajemen Akun	~	Data Dos	en						
Akun Mahasiswa							_		
Akun Dosen		eko					Q Se	earch	
		Total records 1							
		NIDN	Nama	Prodi	Fakultas	Status	Act	tion	
		0522046701	Eko Prasetyo	Teknik Informatika	Teknik	ACTIVE	A	/ 8	

Gambar 4. 34 Tampilan pencarian data dosen

F Administrator ×	nus los	calhost:81 / 127.0.0.1 / ide 🗙	+						C		
③ localhost:81/ideskripsi/Admi	inistrator/d	lata_dosen/NIDN_DSN/desc/1	(C C	2, Search	合自	0 1	Ŷ	×	8	
Administrator									🛔 Та		
B Dashboard		Data Da									
🛦 Manajemen Akun		Data Do	sen								
Akun Mahasiswa									_		
Akun Dosen		Type your se	arch word					Q, Sea	irch		
		NIDN	Nama	Prodi	Fakultas	Status		Ac	tion		
		0502026801	Dwijoko Purbohadi	Teknik Informatika	Teknik	ACTIVE		٩.	/ 8		
		0503068601	Reza Giga Isnanda	Teknik Informatika	Teknik	ACTIVE		٩,	/ 8		
		0511116901	Haris Setyawan	Teknik Informatika	Teknik	ACTIVE		4	/ 8		
		0515038702	Cahya Damarjati	Teknik Informatika	Teknik	ACTIVE		94	8		
		0518048401	Aprilia Kumiati	Teknik Informatika	Teknik	NON ACTIVE		۹,	/ 8		
		0521037603	Helmi Zain Nuri	Teknik Informatika	Teknik	NON ACTIVE		٩	/ 8		
		0522046701	Eko Prasetyo	Teknik Informatika	Teknik	ACTIVE		٩,	/ 8		
		0526047401	Asroni	Teknik Informatika	Teknik	ACTIVE		4	- 8		
		0707108402	Chavadi Oktomy	Teknik Informatika	Teknik	ACTIVE		4			

Gambar 4. 35 Tampilan data dosen urut berdasarkan NIDN

🕴 Administrator 🛛 🗙	localhost81 / 127.0.0.1 / idr 🗙	+					-	
localhost:81/ideskripsi/Admini	istrator/data_dosen		C 9.	Search	合自	9 ¥ A	х –	R
ministrator							🔺 Tata	nia -
Dashboard	Data Da	con						
Aanajemen Akun		Sell						
Vkun Mahasiswa	Activated Sucess	ulv						
Vkun Dosen								
						0.0		
	Type your se	arch word				CC Se	arch	
	Total records - 9							
	NIDN	Nama	Prodi	Fakultas	Status	A	ction	
	0518048401	Aprilia Kumiati	Teknik Informatika	Teknik	NON ACTIVE	4	28	
	0526047401	Asroni	Teknik Informatika	Teknik	ACTIVE	4	8	
	0515038702	Cahya Damarjati	Teknik Informatika	Teknik	ACTIVE	4	18	
	0707108402	Chayadi Oktomy	Teknik Informatika	Teknik	ACTIVE	4	28	
	0502026801	Dwijoko Purbohadi	Teknik Informatika	Teknik	ACTIVE	4	/ 8	
	0522046701	Eko Prasetyo	Teknik Informatika	Teknik	ACTIVE	٩	28	
	0511116901	Haris Setyawan	Teknik Informatika	Teknik	ACTIVE	4	2 8	
	0521037603	Halmi 7sin Nuri	Taknik Informatika	Taknik	NON ACTIVE			

Gambar 4. 36 Tampilan berhasil aktivasi akun dosen

🕴 Administrator 🛛 🗙	localhost81/127.0.0.1/ide × +		- Ø ×
O localhost:81/ideskripsi/Administra	ntor/edit_dsn/0515038702	C ^e Q ₄ Search	☆ 白 ♥ ♣ ★ X ➡ ⅔ ☰
Administrator			🛓 Tatania 👻
 ֎ Dashboard ▲ Manajemen Akun 	Ubah Data		
	NDN Nama Pakultas Prodi Password	osisoaaroz Cahya Damarjali Teknik Teknik Informatika 219656011a89e19258344e06;7226073 Vibbrat D Reset	

Gambar 4. 37 Tampilan ubah password akun dosen

🕴 Administrator 🛛 🗙	a localhost:81 / 127.0.0.1 / ide ×	+					- 8
•) • i localhost:81/ideskripsi/Admin	nistrator/data_dosen		C Q	Search	☆ 自	∞ + •	e X 🗕 😒
dministrator							💧 Tatania
Dashboard	Data Da	con					
, Manajemen Akun 🛛 👻	Data Do	Sell					
Akun Mahasiswa	Deleted Sucessful	īγ					
Akun Dosen							
	Type your se	arch word				a	Search
	Total records - 8						
	NIDN	Nama	Prodi	Fakultas	Status		Action
	0518048401	Aprilia Kurniati	Teknik Informatika	Teknik	NON ACTIVE	5	8
	0526047401	Asroni	Teknik informatika	Teknik	ACTIVE	5	N 🖊 🖻
	0515038702	Cahya Damarjati	Teknik Informatika	Teknik	ACTIVE		
	0707108402	Chayadi Oktomy	Teknik Informatika	Teknik	ACTIVE		V 🖉 🛱
	0502026801	Dwijoko Purbohadi	Teknik Informatika	Teknik	ACTIVE		
	0522046701	Eko Prasetyo	Teknik Informatika	Teknik	ACTIVE	E.	
	0511116901	Haris Setyawan	Teknik Informatika	Teknik	ACTIVE		
	0521027602	Holmi Zaio Nuri	Tokolk informatika	Tokolk	NON ACTIVE		

Gambar 4. 38 Tampilan berhasil menghapus akun dosen

Pada Gambar 4.39, dapat dilihat bahwa fungsi *data_dosen* memanggil fungsi *get_dosen* dan *record_count* yang ada pada M_Dsn dalam folder *models*. Fungsi *get_dosen* digunakan untuk mendapatkan data dosen dari *database*. Sedangkan fungsi *record_count* digunakan untuk menghitung jumlah data dosen. *Coding* fungsi *get_dosen* dan *record_count* dapat dilihat pada Gambar 4.40.

```
public function data_dosen()
  $data = array();
$data['sort_cols'] = array(
    'NIDN_DSN' => 'NIDN',
    'NAMA_DSN' => 'Nama',
    'PRODI_DSN' => 'Prodi',
    'FAKULTAS_DSN' => 'Fakultas'
  $config["per_page"] = 10;
  //max number of page links
$config['num_links'] = 2;
//use page number as parameter
$config['use_page_numbers'] = TRUE;
  $data['search_string'] =
  if(!empty($search_string)) {
     $this->uri->segment(6, $this->uri->segment(5, 1));
$data['search_string'] = $this->uri->segment(5, $search_string);
  } elseif($this->uri->segment(5) != null && !empty($this->uri->segment(5)) && $this->uri->segment(6) != null) {
    $data['search_string'] = $this->uri->segment(5);
  3
  }
//set default page uri
$page_uri = 5;
  if(!empty($data['search_string']))
$page_uri = 6;
  $config["uri_segment"] = $page_uri;
  $config["total_rows"] = $this->dosen->record_count($data['search_string']);
  $data['page'] = $this->uri->segment($page_uri, 1);
  $data['sort_by'] = $this->uri->segment(3, 'NAMA_DDSN');
$orderBy = $this->uri->segment(4, "desc");
$offset = ($data['page']-1) * $config['per_page'];
$data['total_rows'] = $config['total_rows'];
if($orderBy == "asc") $data['sort_order'] = "desc"; else $data['sort_order'] = "asc";
 $config["base_url"] = base_url().'Administrator/data_dosen/'.$data['sort_by'].'/'.$orderBy.'/'.$data['search_string'];
$config['full_tag_open'] = ';
$config['full_tag_close'] = '';
$config['full_tag_close'] = '&li class="prev page">;
$config['first_tag_open'] = '&li class="prev page">;
$config['first_tag_close'] = '';
  $config['last_link'] = 'Last »';
$config['last_tag_open'] = '';
$config['last_tag_close'] = '';
  $config['next_link'] = 'Next →';
$config['next_tag_open'] = '';
$config['next_tag_close'] = '';
  $config['prev_link'] = '← Previous';
$config['prev_tag_open'] = '';
$config['prev_tag_close'] = '';
 $config['cur_tag_open'] = '<a href="">';
$config['cur_tag_close'] = '</a>';
 $config['num_tag_open'] = '';
$config['num_tag_close'] = '';
 $data["data"] = $this->dosen($config["per_page"], $offset, $data['sort_by'], $data['sort_order'], $data['search_string']);
 $this->pagination->initialize($config);
$data["links"] = $this->pagination->create_links();
 $this->load->view('admin/v_datadosen', $data);
```

3

Gambar 4. 39 Coding data dosen

```
function get_dosen($per_page, $offset, $sortfield, $orderBy, $search_string, $id=0)
  {
    if(empty($id)){
       /echo $per_page.'fff'.$offset.'fff'.$sortfield.'fff'.$orderBy;
      if(!empty($search_string)) {
    $this->db->like('NAMA_DSN',$search_string);
        $this->db->or_like('NIDN_DSN',$search_string);
        $this->db->or_like('PRODI_DSN',$search_string);
        $this->db->or_like('FAKULTAS_DSN',$search_string);
      $this->db->order_by("$sortfield", "$orderBy");
      $this->db->limit($per_page,$offset);
      $query = $this->db->get('dosen');
      if ($query->num_rows() > 0) {
        foreach ($query->result() as $row) {
          $data[] = $row;
        3
      return $data;
    }
    return false;
    } else {
    $query = $this->db->get_where('dosen', array('NIDN_DSN' => $id));
    return $query->row_array();
    }
 }
public function record_count($search_string) {
if(!empty($search_string)) {
  $this->db->like('NAMA_DSN',$search_string);
 $this->db->or_like('NIDN_DSN',$search_string);
$this->db->or_like('PRODI_DSN',$search_string);
 $this->db->or_like('FAKULTAS_DSN',$search_string);
}
   return $this->db->count_all_results("dosen");
3
```

Gambar 4. 40 Coding model data mahasiswa

Pada Gambar 4.41, dapat dilihat *coding* fungsi *edit_dsn* dalam *controller* Administrator memanggil fungsi *select* dalam *model* dosen untuk mengambil data dosen dengan parameter nidn_dsn. Administrator dapat melakukan *reset password* dosen melalui fungsi *profile_dsn* yang ada pada *model* dosen. *Password* disimpan ke dalam *database* menggunakan enkripsi md5. Fungsi *select* dan *profile_dsn* dapat dilihat pada Gambar 4.42.

```
function edit_dsn($nidn_dsn)
{
    if($_POST==NULL) {
        $data['data_dsn'] = $this->dosen->select($nidn_dsn);
        $this->load->view('admin/v_resetdosen',$data);
    }else {
        $data['password_dsn'] = md5($this->input->post('password_dsn'));
        $nidn_dsn = $this->input->post('nidn_dsn');
        $this->dosen->profile_dsn($nidn_dsn, $data);
        redirect('Administrator/data_dosen');
    }
}
```

Gambar 4. 41 Coding fungsi edit dosen

```
function select($id_user)
{
    return $this->db->get_where('dosen', array('NIDN_DSN'=>$id_user))->row();
}
//UPDATE
function profile_dsn($iduser, $data)
{
    $this->db->where(array('NIDN_DSN' => $iduser, 'NIDN_DSN !=' => 0));
    $this->db->update('dosen', $data);
}
```

Gambar 4. 42 Coding fungsi select dan profile_dsn

Pada Gambar 4.43, dapat dilihat fungsi *delete_dsn* dalam *controller* Administrator yang memanggil fungsi *delete_dsn* dalam *model* dosen. Fungsi *delete_dsn* dalam *model* dosen dapat dilihat pada Gambar 4.44.

```
public function delete_dsn($id)
{
    $this->dosen->delete_dsn($id);
    $this->session->set_flashdata('message', 'Deleted Sucessfully');
    redirect('Administrator/data_dosen');
}
```

Gambar 4. 43 Coding fungsi delete_dsn dalam controllers

```
public function delete_dsn($id)
{
    $this->db->where('NIDN_DSN', $id);
    return $this->db->delete('dosen');
}
```

Gambar 4. 44 Coding fungsi delete_dsn dalam models

Pada Gambar 4.45, dapat dilihat fungsi aktivasi akun dosen dalam *controller* Administrator yang memanggil fungsi *act_dsn* dalam *model* dosen. Fungsi *act_dsn* dalam *model* dosen dapat dilihat pada Gambar 4.46.

```
function aktivasi_dsn($nidn_dsn)
{
    $this->dosen->act_dsn($nidn_dsn);
    $this->session->set_flashdata('message', 'Activated Sucessfully');
    redirect('Administrator/data_dosen');
}
```

Gambar 4. 45 Coding aktivasi akun dosen

```
function act_dsn($id)
{
    $this->db->set('STATUS_DSN', '1');
    $this->db->where('NIDN_DSN', $id);
    $this->db->update('dosen');
}
```

Gambar 4. 46 Fungsi act_dsn

4.2.6 Halaman Dashboard Mahasiswa

Halaman *dashboard* mahasiswa adalah halaman utama untuk mahasiswa setelah *login*. Pada halaman ini terdapat menu profil, dimana mahasiswa dapat mengubah data diri. Tampilan dapat dilihat pada Gambar 4.41 dan 4.42.



Gambar 4. 47 Tampilan dashboard mahasiswa

D localbort 81/ideckrine:/Maharinaa/ood	File mbr/20120140010		0	Q Security		◇ 白 図	1.4	x = 6	. =
D rocanosco i/ racskrips/ manasiswa/ pro	ine_mins/20150140010		Ç	~ 550101		и н •	• •	× = *	-
Ide SKRIPSI		Beranda	Ide Skripsi	Pengumuman	Tentang Kami	Dashboard	Log Ou	t	
							-		а,
				M	enu				
Profil				Profi	1				
NIM	20130140010								
Nama	Safira Nira Yuma								
Prodi	Teknik Informatika		~						
Fakultas	Teknik		~						
Alamat	Padokan Lor, Tirtonirmolo,Kasihan, bantu	I							
No. hp	085794279088								
SKS	135								
IPK	3.75								
Username	safirany								
Email	safiranira@gmail.com								

Gambar 4. 48 Tampilan ubah profil mahasiswa

Pada Gambar 4.43, dapat dilihat fungsi untuk mengubah profil mahasiswa dalam *controller* mahasiswa. Jika validasi benar, data mahasiswa dalam *database* akan diperbarui.

```
public function profile_mhs($id_user)
{
               //GET REQUIRED DATA FROM DB
              $data['data_mhs'] = $this->M_Mhs->select($id_user);
              $this->load->view('mhs/Vm_mhs-profile', $data);
public function profile_mhs_update()
             $this->form_validation->set_rules('nama', 'NAMA MAHASISWA', 'required');
$this->form_validation->set_rules('prodi', 'PRODI', 'required');
$this->form_validation->set_rules('fakultas', 'FAKULTAS', 'required');
$this->form_validation->set_rules('alamat', 'ALAMAT', 'required');
$this->form_validation->set_rules('alamat', 'required');
$this->form_validation->set_rules('alamat', 'required');
$this->form_validation->set_rules('alamat', 'required');
$this->form_validation->set_rules('required');
$this->form_validation->set_rules('required');
$this->form_validation->set_rules('required');
$this->form_validation->set_rules('required');
$this->form_validation->set_rules('required');
$this->form_validation->set_rules('required');
$this->form_validation->set_rules('required');
$this->form_validation->set_rules('required');
$this->form_validation->set_rules('required');
$this->form_validation->set_rule
             $this->form_validation->set_rules('nohp','NO HP','required');
$this->form_validation->set_rules('sks','SKS','required');
#this >form_validation_set_rules('ks','TK','required');
             $this->form_validation->set_rules('ipk', 'IPK', 'required');
$this->form_validation->set_rules('username', 'USERNAME', 'required');
              $this->form_validation->set_rules('email','EMAIL','required');
              if($this->form_validation->run() == FALSE)
              {
                              $this->load->view('mhs/Vm_mhs-profile');
              3
              else
              {
                            $data['nama_mhs'] = $this->input->post('nama');
$data['prodi_mhs'] = $this->input->post('prodi');
$data['fakultas_mhs'] = $this->input->post('fakultas');
$data['alamat_mhs'] = $this->input->post('alamat');
                            $data['alamat_mhs'] = $this->input->post('ala
$data['alamat_mhs'] = $this->input->post('ala
$data['nohp_mhs'] = $this->input->post('nohp');
$data['sks_mhs'] = $this->input->post('sks');
$data['ipk_mhs'] = $this->input->post('ipk');
                             $data['username mhs']
                                                                                                                       = $this->input->post('username');
                             $data['email_mhs'] = $this->input->post('email');
                             $id mhs
                                                                                                          = $this->input->post('nim');
                             $this->M_Mhs->profile_mhs($id_mhs, $data);
                             redirect(site_url('Mahasiswa'));
              3
}
```



4.2.7 Halaman Dashboard Dosen

Halaman *dashboard* dosen adalah halaman utama untuk dosen setelah *login*. Pada halaman ini terdapat beberapa menu, meliputi profil, data ide skripsi, dan seleksi mahasiswa. Pada menu profil, dosen dapat mengubah data diri. Menu data ide skripsi menampilkan data ide skripsi dosen yang bersangkutan. Pada menu ide skripsi, dosen dapat melakukan pencarian, pengurutan, penambahan, pengubahan, dan penghapusan data ide skripsi. Menu seleksi mahasiswa menampilkan data mahasiswa yang mengambil ide skripsi. Pada menu seleksi mahasiswa, dosen dapat memilih mahasiswa untuk kemudian memberikan keputusan diterima atau tidak diterima. Tampilan dapat dilihat pada Gambar 4.50, Gambar 4.51, dan Gambar 4.52, Gambar 4.53, Gambar 4.54, Gambar 4.55, Gambar 4.56, Gambar 4.57, Gambar 4.58, Gambar 4.59, Gambar 4.60.





localhost:81/ideskripsi/Dosen/profile_dsi	/0511116901		C	Q, Search		☆自♥	+ 1	r X -	8
Ide SKRIPSI		Beranda	lde Skripsi	Pengumuman	Tentang Kami	Dashboard	Log	Out	
Drofil				Me	nu				
PIOII				Profil					
NIDN	0511116901			Data Id	e Skripsi				
Nama	Haris Setyawan			Seleksi	Mahasiswa				
Prodi	Teknik Informatika		~						
Fakultas	Teknik		~						
Alamat	Klaten								
	085794299789								
No. hp									
No. hp Pendidikan Terakhir	53								
No. hp Pendidikan Terakhir Keahlian	S3 Programming								
No. hp Pendidikan Terakhir Keahlian Username	53 Programming harisetvawan								

Gambar 4. 51 Tampilan profil dosen

				C Q Searc	h		ê 🛡	1 +	Ĥ	Х -
Ide SKRIPSI			Beranda Ide	Skripsi Penj	gumuman Tentang Kam	i D	ashboar	rd I	Log Ou	ıt
		-		~						
Data Ide S	kripsi				Menu					
Can ida skrinsi			0.50	are b	Data Ide Skripsi				-	
contract antipar					Seleksi Mahasiswa					
			+ Tambah (de Skripsi						
			Mata Kuliah							
lde Skripsi	Deskripsi	Kuota	Pendukung	Action						
Ide Skripsi Aplikasi kasir mart	Deskripsi aplikasi desktop windows	Kuota 0	Pendukung PAW, IBD	Action						
Ide Skripsi Aplikasi kasir mart Sistem pembuangan limbah	Deskripsi aplikasi desktop windows website sistem pembuangan limbah	Kuota 0 1	Pendukung PAW, IBD PKW, IBD	Action						

Gambar 4. 52 Tampilan menu data ide skripsi

Ide SKRIPSI Beranda Katorija Pengumumani Tentang Kami Dashboard Log Out	Ide SKRIPSI								
Data Ide Skripsi web Q search Imbah Met Skripsi Selesi Mahasiswa imbah Met Skripsi Selesi Mahasiswa				Beranda Id	e Skripsi 🛛 Peng	gumuman Tentang Kami	i Dashboard	d Log (Out
Nether Osearch Imbah Deskripsi Kuok Mata Kuliah Series Stetem pembuangan t PKV, IBD Imbah									
Impair Deskripsi Steren pembuangan 1 PKW, IBD 1	D. C. L. C.	0 # 2012#0				Menu			
Lata Ide Skripsi Lata Ide Skripsi Lata Ide Skripsi Ide Skripsi Deskripsi Mata Kullah Pendukung Kuota Sistem pembuangan Imbah website sistem pembuangan 1 PKW, IBD Image	Data Ide Sk	ripsi				Profil			
Colspan="2">Seleksi Mahasiswa Seleksi Mahasiswa Ide Skripsi Deskripsi Mata Kuliah Pendukung Action Sistem pembuangan Iimbah website sistem pembuangan 1 PKV, IBD Image: Colspan="2">Colspan="2">Seleksi Mahasiswa	web			0.5	arch	Data Ide Skripsi			
International Internationa International International International International In	inco			4 .5		Seleksi Mahasiswa			
Ide Skripsi Deskripsi Kuota Mata Kuliah Pendukung Action Sistem pembuangan Imbah website sistem pembuangan Imbah 1 PKW, IBD Imbah				+ Tambah	lde Skripsi				
Stetem pembuangan website sistem pembuangan 1 PKW, IBD Z	ide Skripsi	Deskripsi	Kuota	Mata Kuliah Pendukung	Action				
	Sistem pembuangan limbah	website sistem pembuangan limbah	1	PKW, IBD	28				



(i) leathert	1 //destation//Desses/data_ide//DE	SVDIDS((dates/1))			0	Canada	~ m m .	* X =	6.8	
U localitosci	n nestripse besen data_loter be_	3888-37 0650 17				Secret		n / =	ar.	
le	de SKRIPSI			Beranda Ide	e Skripsi	Pengumuman Tentang	Kami Dashboard	Log Out		
	Data Ide S	kripsi				Menu				
	Cari ida ekrinei			0.54	arch	Data Ide Skripsi		_		
	Carride skripsi			Q Se	arui	Seleksi Mahasis	wa			
	Ide Skripsi	Deskripsi	Kuota	+ Tambah I Mata Kuliah Pendukung	de Skripsi					
	Aplikasi kasir mart	aplikasi desktop windows	0	PAW, IBD	/ 8					
	Sistem pembuangan limbah	website sistem pembuangan limbah	1	PKW, IBD	2 8					
	Sistem Perizinan Kuliah	Membutuhkan 3 Mahasiswa- Logic- Database- HCI	2	PKW, IBD, HCI	28					

Gambar 4. 54 Tampilan data ide skripsi dosen urut judul



Gambar 4. 55 Tampilan tambah ide skripsi

Ide SKRIPSI		Beranda Ide Skripsi	Pengumuman Tentang Kar	ni Dashboard Log Ou	ıt
E Contraction of the second se					
			Menu		
Edit Ide Sk	ripsi		menu		
			Data Ide Skripsi		
Ide Skripsi	Sistem pembuangan limbah		Seleksi Mahasiswa		
Deskripsi	website sistem pembuangan limbah				
Kuota	1				
Mata Kuliah Repdukupg	PKW, IBD				
- Chukung					
	Submit DiReset				

Gambar 4. 56 Tampilan ubah ide skripsi



Gambar 4. 57 Tampilan menghapus ide skripsi

localhost:81/ii	deskripsi/Dosen/data_ide				G 0	Q, Search			+	ń	× -	s.
Ide	SKRIPSI			Beranda Ide	Skripsi	Pengumuman Tentang Kami	Da	shboard	L	og Ou		
	Data Ide S	kripsi				Menu				-		
	Deleted Sucessfully					Profil Data Ide Skripsi						
	Cari ide skripsi			Q, Sei	arch	Seleksi Mahasiswa				-		
				+ Tambah k	de Skripsi							
	Ide Skripsi	Deskripsi	Kuota	Mata Kuliah Pendukung	Action							
	Sistem pembuangan Imbah	website sistem pembuangan limbah	21	PKW, IBD	28							
1	Sistem Perizinan Kuliah	Membutuhkan 3 Mahasiswa- Logic-	2	PKW, IBD, HCI	18							





Gambar 4. 59 Tampilan menu seleksi mahasiswa

localhost 81/ideskripsi/Dosen/seleksi_mhs/17	C Q. Search 🛱 🛍 🖁 🗍 👘	+ ×- ≡
Ide SKRIPSI	Beranda Ide Skripsi Pengumuman Tentang Kami Login Dashboard	
Soloksi Mahasiswa	Menu	
Seleksi Manasiswa	Profil	
NIM 20130140015	Data ide Skripsi	
Status	Seleksi Mahasiswa	
	Log Out	
🛩 Submit		

Gambar 4. 60 Tampilan seleksi mahasiswa

Pada Gambar 4.61, dapat dilihat fungsi untuk mengubah profil dosen dalam *controller* dosen. Jika validasi benar, data dosen dalam *database* akan diperbarui.

```
public function profile_dsn($id_user)
                      //GET REQUIRED DATA FROM DB
$data['data_dsn'] = $this->M_Dsn->select($id_user);
$this->load->view('dsn/Vm_dsn-profile', $data);
3
public function profile_dsn_update()
                     $this->form_validation->set_rules('nama', 'NAMA MAHASISWA', 'required');
$this->form_validation->set_rules('prodi', 'PRODI', 'required');
$this->form_validation->set_rules('alamat', 'ALAMAT', 'required');
$this->form_validation->set_rules('nom ', 'No HP', 'required');
$this->form_validation->set_rules('pendidikanterakhir', 'PENDIDIKAN TERAKHIR', 'required');
$this->form_validation->set_rules('keahlian', 'KEAHLIAN', 'required');
$this->form_validation->set_rules('keahlian', '
                       if($this->form_validation->run() == FALSE)
                       {
                                              $this->load->view('dsn/Vm_dsn-profile');
                       }
                       else
                       {
                                           $data['nama_dsn'] = $this->input->post('nama');
$data['prodi_dsn'] = $this->input->post('prodi');
$data['fakultas_dsn'] = $this->input->post('fakultas');
$data['alamat_dsn'] = $this->input->post('alamat');
$data['nohp_dsn'] = $this->input->post('pendidikanterakhir');
$data['keahlian_dsn'] = $this->input->post('pendidikanterakhir');
$data['username_dsn'] = $this->input->post('keahlian');
$data['username_dsn'] = $this->input->post('username');
$data['username_dsn'] = $this->input->post('username');
                                               $data['email_dsn'] = $this->input->post('email');
                                                                                                                                                                     = $this->input->post('nidn');
                                              $id dsn
                                              $this->M_Dsn->profile_dsn($id_dsn, $data);
redirect(site_url('Dosen'));
à
```

Gambar 4. 61 Coding ubah profil dosen

Pada Gambar 4.62, dapat dilihat bahwa fungsi *data_ide* memanggil fungsi *select_ide* yang ada pada *model* ide. Fungsi *select_ide* digunakan untuk mendapatkan data ide berdasarkan NIDN dosen dari *database. Coding* fungsi *select ide* dapat dilihat pada Gambar 4.63. \$config["uri_segment"] = \$page_uri; \$config["total_rows"] = \$this->ide->record_count(\$data['search_string']); \$data['page'] = \$this->uri->segment(\$page_uri, 1); \$data['sort_by'] = \$this->uri->segment(3, 'IDE_SKRIPSI'); \$orderBy = \$this->uri->segment(4, "desc"); \$offset = (\$data['page']-1) * \$config['per_page']; \$data['total_rows'] = \$config['total_rows']; if(\$orderBy == "asc") \$data['sort_order'] = "desc"; else \$data['sort_order'] = "asc"; \$config["base_url"] = base_url().'Dosen/data_ide/'.\$data['sort_by'].'/'.\$order8y.'/'.\$data['search_string']; \$config['full_tag_close'] = '
 '
 \$config['first_link'] = 'blaquo; First'; \$config['first_tag_close'] = '
 '
 \$config['first_tag_close'] = '
 '; \$config['last_link'] = 'Last »'; \$config['last_tag_open'] = ''; \$config['last_tag_close'] = ''; \$config['next_link'] = 'Next →'; \$config['next_tag_open'] = ''; \$config['next_tag_close'] = ''; \$config['prev_link'] = '← Previous'; \$config['prev_tag_open'] = ''; \$config['prev_tag_close'] = ''; \$config['cur_tag_open'] = ''; \$config['cur_tag_close'] = ''; \$config['num_tag_open'] = '';
\$config['num_tag_close'] = '';

} elseif(\$this->uri->segment(5) != null && !empty(\$this->uri->segment(5)) && \$this->uri->segment(6) != null) {
 \$data['search_string'] = \$this->uri->segment(5);

\$data["data"] = \$this->ide->select_ide(\$config["per_page"], \$offset, \$data['sort_by'], \$data['sort_order'], \$data['sert_string'];

\$this->pagination->initialize(\$config);
\$data["links"] = \$this->pagination->create_links();

\$this->load->view('dsn/v_idedosen', \$data);

public function data_ide()
{

\$config["per_page"] = 10;
//max number of page link //max number of page links
\$config['num_links'] = 2;
//use page number as parameter
\$config['use_page_numbers'] = TRUE;

\$data['search_string'] = '';
if(!empty(\$search_string)) {

}
//set default page uri
\$page_uri = 5;

if(!empty(\$data['search_string']))
\$page_uri = 6;

}

{
 //GET REQUIRED DATA FROM DB
 Sdata = array();
 Sdata['title'] = 'Ide Skripsi';
 Sdata['sort_cols'] = array(
 'IDE_SKRIPSI' => 'Ide Skripsi',
 'DESKRIPSI' => 'Deskripsi',
 'KUOTA_AMBIL' => 'Kuota',
 'MATKUL'=> 'Mata Kuliah Pendukung'
}

\$this->uri->segment(6, \$this->uri->segment(5, 1));
\$data['search_string'] = \$this->uri->segment(5, \$search_string);

}

Gambar 4. 62 Coding data ide dosen

62

```
function select_ide($per_page, $offset, $sortfield, $orderBy, $search_string, $id=0)
{
    if(empty($id)){
                      _page.'fff'.$offset.'fff'.$sortfield.'fff'.$orderBy;
           cho $per
      if(!empty($search_string)) {
        $this->db->like('IDE_SKRIPSI',$search_string);
$this->db->or_like('DESKRIPSI',$search_string);
      J
$this->db->order_by("$sortfield", "$orderBy");
$this->db->limit($per_page,$offset);
      $this->db->select("ID_IDE_JOE_SKRIPSI, DESKRIPSI, KUOTA_AMBIL, NIDN_DSN, MATKUL");
$this->db->from('ideskripsi');
      $this->db->where('NIDN_DSN', $this->session->userdata('id_user'));
$query = $this->db->get();
      if ($query->num_rows() > 0) {
  foreach ($query->result() as $row) {
           $data[] = $row;
         j,
      return $data;
    3
    return false;
   } else {
   $query = $this->db->get_where('ideskripsi', array('ID_IDE' => $id));
    return $query->row_array();
    3
}
```



Pada Gambar 4.64, dapat dilihat bahwa fungsi *add_ide* pada *controller* IdeSkripsi memanggil fungsi *add_ide* pada *model* ide. Fungsi *add_ide* pada *model* ide dapat dilihat pada Gambar 4.65.



Gambar 4.65 Fungsi tambah ide

Pada Gambar 4.66, dapat dilihat *coding* fungsi *edit_ide* dalam *controller* IdeSkripsi memanggil fungsi *select* dalam *model* ide untuk mengambil data ide dengan parameter id ide. Dosen dapat mengubah data ide skripsi melalui fungsi *update_ide* yang ada pada *model* ide. Fungsi *select* dan *update_ide* dapat dilihat pada Gambar 4.67.

```
function edit_ide($id)
{
    if($_POST==NULL) {
        $data['data_ide'] = $this->ide->select($id);
        $this->load->view('dsn/v_editide',$data);
    }else {
        $data['IDE_SKRIPSI'] = $this->input->post('ideskripsi');
        $data['DESKRIPSI'] = $this->input->post('deskripsi');
        $data['KUOTA_AMBIL'] = $this->input->post('kuota_ambil');
        $data['MATKUL'] = $this->input->post('kuota_ambil');
        $data['MATKUL'] = $this->input->post('matkul');
        $id = $this->input->post('id');
        $this->ide->update_ide($id, $data);
        redirect('Dosen/data_ide');
        }
}
```

Gambar 4. 66 Coding edit ide

```
function select($id)
{
return $this->db->get_where('ideskripsi', array('ID_IDE'=>$id))->row();
}
//UPDATE
function update_ide($id, $data)
{
$this->db->where(array('ID_IDE' => $id, 'ID_IDE !=' => 0));
$this->db->update('ideskripsi', $data);
}
```



Pada Gambar 4.68, dapat dilihat *coding* fungsi *delete_ide* dalam *controller* IdeSkripsi memanggil fungsi *delete_ide* dalam *model* ide untuk menghapus data ide dengan parameter id ide. Fungsi *delete_ide* pada *model* ide dapat dilihat pada Gambar 4.69.

```
public function delete_ide($id)
{
    $this->ide->delete_ide($id);
$this->session->set_flashdata('message', 'Deleted Sucessfully');
redirect('Dosen/data_ide');
}
```

Gambar 4. 68 Coding hapus ide skripsi

```
public function delete_ide($id)
{
    $this->db->where('ID_IDE', $id);
    return $this->db->delete('ideskripsi');
}
```

Gambar 4. 69 Fungsi delete ide

Pada Gambar 4.70, dapat dilihat bahwa fungsi seleksi memanggil fungsi *get_pengambilan* yang ada pada *model* pengambilan. Fungsi *get_pengambilan* digunakan untuk mendapatkan data mahasiswa pengambil ide skripsi berdasarkan NIDN dosen yang mempunyai ide skripsi dari *database*. *Coding* fungsi *get pengambilan* dapat dilihat pada Gambar 4.71.

```
public function seleksi()
{
    //GET REQUIRED DATA FROM DB
    $data = array();
$data['title'] = 'seleksi';
$data['sort_cols'] = array(
    'IDE_SKRIPSI' => 'Ide Skripsi',
    'INU_MHS' => 'NIM',
    'NAM_MHS' => 'NAma',
    'IPK_MHS'=> 'IPK',
    'NILAI_MATKUL'=> 'Nilai Mata Kuliah'
);
$config = array();
$data['sort_by'] = $this->uri->segment(3, 'IDE_SKRIPSI');
$corderBy = $this->uri->segment(4, "desc");
if($orderBy == "asc") $data['sort_order'] = "desc"; else $data['sort_order'] = "asc";
$data["data"] = $this->pengambilan->get_pengambilan($data['sort_by'], $data['sort_order']);
    $this->load->view('dsn/v_seleksi', $data);
}
```

Gambar 4. 70 Coding data seleksi

```
function get_pengambilan(Ssortfield, SorderBy, Sid=0)
{
    if(empty(Sid)){
    Sthis->db->sorder_by("Ssortfield", "SorderBy");
    Sthis->db->select("pengambilanide.ID_PIS, pengambilanide.NIM_MHS, pengambilanide.NILAI_MATKUL, ideskripsi.IDE_SKRIPSI, mahasiswa.NAMA_MM
    Sthis->db->form("pengambilanide");
    Sthis->db->form("nahasiswa", "mahasiswa.NAMA_INE");
    Sarray = array("NIDW_DSN" => Sthis->session->userdata('id_user'), 'STATUS' => NULL);
    Sthis->db->get();
    if (Squery->num_rows() > 0) {
        foreach (Squery->result() as Srow) {
            i Sdata[] = Srow;
            }
        return false;
        } else {
        Squery = Sthis->db->get_where('pengambilanide', array('ID_AD' => $id));
        return Squery->row_array();
        }
    }
}
```

Gambar 4. 71 Fungsi get_pengambilan

Pada Gambar 4.72, dapat dilihat bahwa fungsi *seleksi_mhs* memanggil fungsi *select* dan *update_status* pada *model* pengambilan. Fungsi *select* digunakan untuk mengambil data pengangambil ide skripsi. Sedangkan, fungsi *update_status* untuk mengubah status pengambil ide skripsi. *Coding* fungsi *select* dan *update_status* pada *model* pengambilan dapat dilihat pada Gambar 4.73.

```
public function seleksi_mhs($id_p)
         if($_POST==NULL) {
            $data['data_seleksi'] = $this->pengambilan->select($id_p);
            $this->load->view('dsn/v_seleksimhs', $data);
         }
         else{
            $data['STATUS'] =
                               $this->input->post('status');
            $statuss = $this->input->post('status');
            $id p
                           = $this->input->post('id_p');
            $ID_IDE = $this->input->post('id_ide');
            $this->pengambilan->update_status($id_p, $data);
            if($statuss == 'N')
            {
                $this->pengambilan->add_kuota($ID_IDE);
            3
            redirect('Dosen/seleksi');
         }
}
```

Gambar 4. 72 Coding seleksi mahasiswa

```
function select($id)
{
    return $this->db->get_where('pengambilanide', array('ID_PIS'=>$id))->row();
}
function update_status($id, $data)
{
    $this->db->where(array('ID_PIS' => $id, 'ID_PIS !=' => 0));
    $this->db->update('pengambilanide', $data);
}
```

Gambar 4. 73 Fungsi select dan update_status

4.2.8 Halaman Ide Skripsi

Halaman ide skripsi menampilkan daftar ide skripsi dosen yang dapat diambil oleh mahasiswa. Pada halaman ini, dapat dilakukan pencarian ide skripsi dengan mengetikkan kata kunci atau mengurutkan data ide skripsi dengan mengklik *header cell* pada tabel. Mahasiswa dapat mengambil ide skripsi dengan mengklik pilih pada ide skripsi yang ingin diambil. Kemudian halaman baru tertampil untuk menanyakan kembali apakah mahasiswa yakin mengambil ide skripsi tersebut. Jika iya, mahasiswa mengklik *submit*. Lalu, muncul notifikasi bahwa mahasiswa telah mengambil ide skripsi tersebut dan dipersilakan menunggu pengumuman. Tampilan dapat dilihat pada Gambar 4.74, Gambar 4.75, Gambar 4.76, Gambar 4.77, Gambar 4.78.

ost 81/ideskripsi/IdeSkripsi	C Q, Search			+ 1
de SKRIPSI	Beranda Ide Skripsi Pengumuman	Tentang Kami	Dashboard	d Log
				1
	Daftar Ide Skripsi			
Cari ide skripsi			Q Sea	arch
Ide Skripsi	Deskripsi	Dosen PJ	Kuota	Action
Ide Skripsi Sistem Ide Skripsi	Deskripsi Membuat perancangan dan implementasi sistem ide skripsi - satu logic - satu database	Dosen Pj Dwijoko Purbohadi	Kuota 2	Action
Ide Skripsi Sistem Ide Skripsi Sistem Keamanan Jaringan UMY menggunakan Packet Tracer	Deskripsi Membuat perancangan dan implementasi sistem ide skripsi - satu logic - satu database Membuat perancangan dan implementasi sistem keamanan jaringan di UMY menggunakan Packet Tracer	Dosen PJ Dwijoko Purbohadi Eko Prasetyo	Kuota 2 1	Action Pilith Pilith
Ide Skripsi Sistem Ide Skripsi Sistem Keamanan Jaringan UMY menggunakan Packet Trocer Sistem pembuangan limbah	Deskripsi Membuat perancangan dan implementasi sistem ide skripsi - satu logic - satu database Membuat perancangan dan implementasi sistem keamanan jaringan di UMY menggunakan Packet Tracer website sistem pembuangan limbah	Dosen PJ Dwijoko Purbohadi Eko Prasetyo Haris Setyawan	Kuota 2 1 1	Action Pilih Pilih Pilih
Ide Skripsi Sistem Ide Skripsi Sistem Keamanan Jaringan UMY menggunakan Packet Tracer Sistem pembuangan limbah Sistem Penggajian Prodi Ti UMY	Deskripsi Membuat perancangan dan implementasi sistem ide skripsi - satu logic - satu database Membuat perancangan dan implementasi sistem keamanan jaringan di UMY menggunakan Packet Tracer website sistem pembuangan limbah Sistem yang memudahkan penggiaan Dosen dan karyawan di Prodi Ti.	Dosen PJ Dwijoko Purbohadi Eko Prasetyo Haris Setyawan Asroni	Kuota 2 1 1 1 1	Action Pilih Pilih Pilih

Gambar 4. 74 Tampilan daftar ide skripsi



Gambar 4. 75 Tampilan pencarian ide skripsi

st/81/ideskrips//IdeSkripsi/index/IDE_SKRIPSI/desc/1/	C Q Search	4	₫ ♥	∔ ☆)
Ide SKRIPSI	Beranda Ide Skripsi Pengumuman	Tentang Kami L	Dashboard	d Log Out
	Daftar Ide Skripsi			
Cari ide skripsi			Q Sea	arch
Ide Skripsi	Deskripsi	Dosen PJ	Kuota	Action
Sistem Ide Skripsi	Membuat perancangan dan implementasi sistem ide skripsi - satu logic - satu database	Dwijoko Purbohadi	2	Pilih
Sistem Ide Skripsi Sistem Keamanan Jaringan UMY menggunakan Packet Tracer	Membuat perancangan dan implementasi sistem ide skripsi - satu logic - satu database Membuat perancangan dan implementasi sistem keamanan jaringan di UMY menggunakan Packet Tracer	Dwijoko Purbohadi Eko Prasetyo	2	Pilih Pilih
Sistem Ide Skripsi Sistem Keamanan Jaringan UMY menggunakan Packet Tracer Sistem pembuangan limbah	Membuat prenncangan dan implementasi sistem ide skripai - satu database Membuat perancangan dan implementasi sistem keamanan jaringan di UMY menggunakan Packet Tracer website sistem pembuangan limbah	Dwijoko Purbohadi Eko Prasetyo Haris Setyawan	2	Pilih Pilih Pilih
Sistem Kde Skripsi Sistem Keamanan Jaringan UMY menggunakan Packet Tracer Sistem pembuangan limbah Sistem Penggajian Prodi Ti UMY	Membuat prenicangan dan implementasi sistem ide skripai - satu distabase distabase Membuat perancangan dan implementasi sistem keamanan jaringan di UMY menggunakan Packet Träcer website sistem pembuangan Imbah Sistem yang memudahkan penggalan Dosen dan karyawan di Prodi Ti	Dwijoko Purbohadi Eko Prasetyo Haris Setyawan Asroni	2 1 1 1	Pilih Pilih Pilih Pilih

Gambar 4. 76 Tampilan sorting ide skripsi berdasarkan judul

localhost:81/ideskripsi/Ide5kripsi/add_pengambilan/5		C	Q, Search	☆ 自 ♥ ↓	A X = 😣
Ide SKRIPSI		Beranda Ide Skripsi	Pengumuman Tentang	Kami Dashboard Lo	g Out
i i i i i i i i i i i i i i i i i i i					
	Pengamb Apakah Anda ya	ilan Ide Skri	psi		
Ide Skripsi	Sistem Perizinan Kuliah				
Deskripsi	Membutuhkan 3 Mahasiswa- Logic	- Database- HCI			
Mata Kuliah Pendukung	PKW, IBD, HCI				
Nilai Mata Kuliah Pendukung	cth: IBD=A, PAW=B				
		✓ Sub	mit		

Gambar 4. 77 Tampilan pengambilan ide skripsi

Skripsi	× 🛵 localhost:81 / 127.0.0.1 / ide: ×	+					
iost:81/ide	leskripsi/IdeSkripsi/index		C Q, Search	ŕ	n 🖻 🛡	+ 1	
Ide	SKRIPSI	Beranda	Ide Skripsi Pengumuman	Tentang Kami	Dashboard	Log	Du
						1	
		Daftar Ide S	kripsi				
		Anda telah mengambil ide skripsi. Silakan r	nenunggu pengumuman.				
	Cari ide skripsi				Q Sea	rch	
	Ide Skripsi	Deskrip	si	Dosen PJ	Kuota	Action	
Sis	stem ide Skripsi	Membuat perancangan dan implementasi database	sistem ide skripsi - satu logic - sat	u Dwijoko Purbohadi	2	Pilih	
Sis Pa	stem Keamanan Jaringan UMY menggunakan acket Tracer	Membuat perancangan dan implementasi menggunakan Packet Tracer	sistem keamanan jaringan di UM	Eko Prasetyo	1	Pilih	
Sis	stem pembuangan limbah	website sistem pembuangan limbah		Haris Setyawar	1	Pilih	
Sis	stem Penggajian Prodi TI UMY	Sistem yang memudahkan penggjalan Dos	en dan karyawan di Prodi Ti	Asroni	1	Pilih	

Gambar 4. 78 Tampilan notifikasi pengambilan ide skripsi

Pada Gambar 4.79, dapat dilihat bahwa fungsi *index* dalam *controller* IdeSkripsi memanggil fungsi *get_ide* dan *record_count* yang ada pada *model* ide. Fungsi *get_ide* digunakan untuk mendapatkan data ide skripsi dari *database*. Sedangkan fungsi *record_count* digunakan untuk menghitung jumlah data ide skripsi. *Coding* fungsi *get_ide* dan *record_count* dalam *model* ide dapat dilihat pada Gambar 4.80.

```
function index()
 $data = array();
$data['title'] = 'Ide Skripsi';
$data['sort_cols'] = array(
  'IDE_SKRIPSI' => 'Ide Skripsi',
  'DESKRIPSI' => 'Deskripsi',
  'NAMA_DSN' => 'Dosen P1',
  'KUOTA_AMBIL' => 'Kuota'
);
      $config = array();
      //base_url(). 'index.php/questions/page/'.$sortfield.'/'.$order.'/',
$search_string = $this->input->post('search');
      $config["per_page"] = 10;
      //max number of page links
$config['num_links'] = 2;
//use page number as param
                                                   eter
      $config['use_page_numbers'] = TRUE;
      $data['search_string']
      if(!empty($search_string)) {
        $this->uri->segment(6, $this->uri->segment(5, 1));
$data['search_string'] = $this->uri->segment(5, $search_string);
     } elseif($this->uri->segment(5) != null && !empty($this->uri->segment(5)) && $this->uri->segment(6) != null) {
    $data['search_string'] = $this->uri->segment(5);
     }
//set default page uri
      $page_uri = 5;
      if(!empty($data['search_string']))
$page_uri = 6;
      $config["uri_segment"] = $page_uri;
      $config["total_rows"] = $this->ide->record_count($data['search_string']);
      $data['page'] = $this->uri->segment($page_uri, 1);
     $data['sort_by'] = $this->uri->segment(3, 'IDE_SKRIPSI');
$orderBy = $this->uri->segment(4, "desc");
$offset = ($data['page']-1) * $config['per_page'];
$data['total_rows'] = $config["total_rows"];
if($orderBy == "asc") $data['sort_order'] = "desc"; else $data['sort_order'] = "asc";
      $config["base_url"] = base_url().'IdeSkripsi/index/'.$data['sort_by'].'/'.$orderBy.'/'.$data['search_string'];
$config['full_tag_open'] = '';
$config['full_tag_close'] = '';
$config['full_tag_open'] = '';
$config['first_tag_open'] = '';
$config['first_tag_close'] = '';
      $config['last_link'] = 'Last »';
$config['last_tag_open'] = '';
$config['last_tag_close'] = '';
      $config['next_link'] = 'Next →';
$config['next_tag_open'] = '';
$config['next_tag_close'] = '';
      $config['prev_link'] = '← Previous';
$config['prev_tag_open'] = '';
$config['prev_tag_close'] = '';
      $config['cur_tag_open'] = '<a href="">';
$config['cur_tag_close'] = '</a>';
     $config['num_tag_open'] = '';
$config['num_tag_close'] = '';
      $data["data"] = $this->ide->get_ide($config["per_page"], $offset, $data['sort_by'], $data['sort_order'], $data['search_string']);
     $this->pagination->initialize($config);
$data["links"] = $this->pagination->create_links();
```

\$this->load->view('v ideskripsi', \$data);

}

£

Gambar 4. 79 Coding data ide skripsi

```
function get_ide($per_page, $offset, $sortfield, $orderBy, $search_string, $id=0)
ł
  if(empty($id)){
     //echo $per_page.'fff'.$offset.'fff'.$sortfield.'fff'.$orderBy;
    if(!empty($search_string)) {
      $this->db->like('IDE_SKRIPSI',$search_string);
      $this->db->or_like('DESKRIPSI',$search_string);
    $this->db->order_by("$sortfield", "$orderBy");
    $this->db->limit($per_page,$offset);
     $this->db->select("*");
    $this->db->from('ideskripsi');
    $this->db->join('dosen', 'dosen.NIDN_DSN = ideskripsi.NIDN_DSN');
$this->db->where('KUOTA>', '0');
     $query = $this->db->get();
     if ($query->num_rows() > 0) {
      foreach ($query->result() as $row) {
        $data[] = $row;
    return $data;
  }
  return false;
  } else {
    $query = $this->db->get where('ideskripsi', array('ID IDE' => $id));
  return $query->row_array();
  - }
}
public function record count($search string) {
if(!empty($search_string)) {
  $this->db->like('ideskripsi.IDE_SKRIPSI',$search_string);
  $this->db->or_like('ideskripsi.DESKRIPSI',$search_string);
}
  return $this->db->count_all_results("ideskripsi");
```

Gambar 4. 80 Coding model data ide skripsi

Pada Gambar 4.81, dapat dilihat bahwa fungsi add_pengambilan dalam *controller* IdeSkripsi memanggil *add_pengambilan, get_mhs, get_mhy,* dan *update_kuota* yang ada pada *model* pengambilan. Fungsi *add_pengambilan* digunakan untuk menambah data pengambil ide skripsi ke *database*. Fungsi *get_mhs* dan *get_mhy* digunakan untuk mendapatkan data pengambil ide skripsi dengan status NULL atau Y. Fungsi *update_kuota* digunakan untuk memperbarui kuota setelah ide skripsi diambil oleh mahasiswa. *Coding add_pengambilan, get_mhs, get_mhy,* dan *update_kuota* dalam *model* pengambilan dapat dilihat pada Gambar 4.82.

```
public function add_pengambilan($id)
          if($this->session->userdata('tipeuser')== '2') {
               $data["mhs"] = $this->pengambilan->get_mhs($this->session->userdata('id_user'));
$data["mhy"] = $this->pengambilan->get_mhy($this->session->userdata('id_user'));
                $this->form_validation->set_rules('nilai_matkul', 'NILAI MATKUL', 'required');
                if($this->form_validation->run() == FALSE) {
    $data['data_ide'] = $this->ide->select($id);
    $this->load->view('v_pengambilan', $data);
                 elsef
                      $ID_IDE = $this->input->post('id_ide');
                     $NIM_MHS = $this->input->post('nim');
$NILAI_MATKUL = $this->input->post('nilai_matkul');
$data = array(_'ID_IDE'=>$ID_IDE, 'NIM_MHS'=>$NIM_MHS, 'NILAI_MATKUL'=> $NILAI_MATKUL);
                      $this->pengambilan->add_pengambilan($data);
                      $this->pengambilan->update_kuota($ID_IDE);
                      $this->session->set_flashdata('message', 'Anda telah mengambil ide skripsi. Silakan menunggu pengumuman.');
                      redirect('IdeSkripsi/index');
               1
           }
           else{
               $this->session->set_flashdata('message', 'Maaf, Anda harus login sebagai mahasiswa untuk mengambil ide skripsi.');
               redirect('IdeSkripsi/index');
           3
     }
```

Gambar 4.81 Coding tambah pengambilan

```
function add_pengambilan($data)
 1
       $this->db->insert('pengambilanide',$data);
  3
function get_mhs($id_user)
{
     return $this->db->get_where('pengambilanide', array('NIM_MHS'=>$id_user, 'STATUS'=> NULL))->row();
function get_mhy($id_user)
{
     return $this->db->get_where('pengambilanide', array('NIM_MHS'=>$id_user, 'STATUS'=> 'Y'))->row();
3
function update kuota($id)
{
 $this->db->set('KUOTA-', 'KUOTA-1', false);
$this->db->where(array('ID_IDE' => $id, 'ID_IDE !=' => 0));
 $this->db->update('ideskripsi');
}
```



4.2.9 Halaman Pengumuman

Halaman pengumuman menampilkan hasil seleksi mahasiswa pengambil ide skripsi. Pada halaman ini mahasiswa dapat melihat apakah mereka diterima atau tidak untuk mengambil ide skripsi dosen.Tampilan halaman penguuman dapat dilihat pada Gambar 4.83.



Gambar 4.83 Tampilan pengumuman

Pada Gambar 4.84, dapat dilihat bahwa fungsi *index* dalam *controller* Pengumuman memanggil fungsi *get_pengambilan* yang ada pada *model* pengambilan. Fungsi *get_pengumuman* digunakan untuk mendapatkan data hasil seleksi mahasiswa pengambil ide skripsi dari *database*. *Coding* fungsi *get pengumuman* dapat dilihat pada Gambar 4.85.

```
public function index()
{
    $data = array();
    $data['title'] = 'Pengumuman';
    $data['sort_cols'] = array(
        'NIM_MHS' => 'NIM',
        'NAMA_MHS' => 'Nama',
        'IDE_SKRIPSI' => 'Ide Skripsi',
        'NAMA_DSN' => 'Dosen PJ',
        'STATUS' => 'Status'
        ;
        $config = array();

    $data['sort_by'] = $this->uri->segment(3, 'NIM_MHS');
        $conderBy = $this->uri->segment(4, "desc");

        if($orderBy == "asc") $data['sort_order'] = "desc"; else $data['sort_order'] = "asc";
        $data["data"] = $this->pengumuman->get_pengumuman($data['sort_by'], $data['sort_order']);
        $this->load->view('v_pengumuman', $data);
    }
}
```

Gambar 4. 84 Coding controller pengumuman



Gambar 4. 85 Coding model pengumuman

4.2.10 Halaman Tentang Kami

Halaman tentang kami adalah halaman yang menampilkan penjelasan singkat mengenai *website* ide skripsi. Tampilan halaman tentang kami dapat dilihat pada Gambar 4.86.



Gambar 4.86 Tampilan halaman tentang kami

Gambar 4.87 memperlihatkan *coding controller* Tentang yang memanggil tampilan tentang dalam folder *views*. *Coding* tampilan tentang dapat dilihat pada Gambar 4.88.

```
class Tentang extends CI_Controller {
    public function index()
    {
        $this->load->view('tentang');
    }
}
```

Gambar 4. 87 Coding controller tentang



Gambar 4.88 Tampilan halaman tentang kami

4.3 Pengujian Sistem

Pengujian sistem dilakukan untuk mengetahui apakan sistem yang telah dibuat sudah sesuai dengan yang diharapkan. Pengujian juga dilakukan untuk menghindari kesalahan-kesalahan yang mungkin terjadi ketika digunakan oleh pengguna.

4.3.1 Pengujian User Interface

Pengujian *user interface* ditujukan untuk mengetahui fungsionalitas tiap elemen *interface* yang terdapat dalam halaman sistem. Elemen yang diujikan adalah elemen *button* pada setiap halaman *website*. Hasil pengujian dapat dilihat pada Tabel 4.1.

No.	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapatkan	Status
1.	Tombol daftar pada halaman registrasi mahasiswa	Daftar	Sistem menampilkan halaman berhasil mendaftarkan akun.	Pendaftaran berhasil Selamat, Anda sudah terdaftar.	Berhasil
2.	Tombol daftar pada halaman registrasi dosen	Daftar	Sistem menampilkan halaman berhasil mendaftarkan akun.	Pendaftaran berhasil Selamat, Anda sudah terdaftar.	Berhasil
3.	Tombol <i>login</i>	Login	Sistem dapat menampilkan halaman <i>dashboard</i> .	Dashboard Selamat datang di dashboard, Fathaniaffl	Berhasil
4.	Tombol edit pada manajemen akun dalam <i>dashboard</i> administrator.		Sistem dapat menampilkan halaman edit data.	Ubah Data NDN 0515038702 Nama Catiya Damarjati Fakutas Tetrak Prodi Tetrak Tetrak Prodi Tetrak 100matha Password 213650011885e10208344e000:73240073	Berhasil

Tabel 4. 1 Hasil pengujian user interface

No	Kagug yang diuli	Test Case	Hasil yang	Hasil yong didenation	Status
110.	Kasus yang diuji	Test Case	diharapkan	Hash yang didapatkan	Status
5.	Tombol hapus pada manajemen akun dalam <i>dashboard</i> administrator.		Sistem dapat menampilkan kotak dialog penghapusan.	Apakah Anda yakin menghapus akun pengguna ini? OK Cancel	Berhasil
6.	Tombol aktivasi pada manajemen akun dalam <i>dashboard</i> administrator	۹,	Sistem dapat menampilkan notifikasi aktivasi akun.	Activated Successfully	Berhasil
7.	Tombol <i>reset</i> pada halaman ubah data	ී Reset	Sistem dapat mengembalikan nilai pada input.	Password 1fd5cd9766249f170035b7251e2c6b61	Berhasil

No.	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapatkan	Status
8.	Tombol pilih pada halaman ide skripsi	Pilih	Sistem menampilkan halaman pengambilan ide skripsi.	Constraints and a second a	Berhasil
9.	Tombol tambah ide skripsi pada halaman data ide skripsi dalam <i>dashboard</i> dosen	< Ternstall No Skopps	Sistem menampilkan halaman tambah ide skripsi	Image: Contract of the contra	Berhasil
10.	Tombol edit pada halaman data ide skripsi dalam <i>dashboard</i> dosen	2	Sistem dapat menampilkan halaman edit ide skripsi	Induction (Induction (In	Berhasil

No.	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapatkan	Status
11.	Tombol hapus pada halaman data ide skripsi dalam <i>dashboard</i> dosen	<u>ال</u>	Sistem dapat menampilkan notifikasi penghapusan.	Deleted Sucessfully	Berhasil
12.	Tombol pilih pada halaman seleksi mahasiswa skripsi dalam <i>dashboard</i> dosen	Pilih	Sistem menampilkan halaman seleksi mahasiswa	Internet (1992/n 2) Internet (1992/n 2) Idea Internet (1992/n 2) Idea Internet (1992/n 2)	Berhasil

4.3.2 Pengujian Fungsi Sistem

Pengujian fungsi sistem ditujukan untuk mengetahui apakah fungsi-fungsi yang ada dalam sistem sudah bekerja seperti yang seharusnya. Hasil pengujian dapat dilihat pada tabel 4.2.

No.	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapatkan	Status
1.	Pengujian fungsi register mahasiswa		Sistem dapat menambah mahasiswa dalam basis data.		Berhasil
2.	Pengujian fungsi register dosen		Sistem dapat menambah dosen dalam basis data.		Berhasil

Tabel 4. 2 Hasil pengujian fungsi sistem

No.	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapatkan	Status
3.	Mahasiswa melakukan <i>login</i>	netro:	Mahasiswa dapat masuk ke dalam dashboard.		Berhasil
4.	Dosen melakukan <i>login</i>		Dosen dapat masuk ke dalam dashboard dosen.	And the set of th	Berhasil
5.	Administrator melakukan <i>login</i>	Inter O O O O O O O O O O O O O O O O O O O	Administrator dapat masuk ke dalam dashboard administrator.	Administrator	Berhasil

No.	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapatkan	Status
6.	Fungsi ubah password dalam manajemen akun		Sistem dapat mengubah <i>password</i> akun dan menyimmpannya dalam enkripsi md5.		Berhasil
7.	Fungsi aktivasi akun dalam manajemen akum		Sistem dapat mengubah status akun pengguna menjadi aktif.	Control (Control (Contro) (Control (Control (Control (Control (Contro) (Control (Contro)	Berhasil
8.	Fungsi hapus dalam manajemen akun		Sistem dapat menghapus data akun.		Berhasil

No.	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapatkan	Status
9.	Fungsi pencarian dalam manajemen akun		Sistem dapat menampilkan hasil pencarian sesuai kata kunci pencarian.		Berhasil
10.	Fungsi <i>sorting</i> dalam manajemen akun	Control of the second of	Sistem dapat menampilkan data urut dengan pilihan <i>sorting</i> .		Berhasil
11.	Fungsi ubah data pada profil pengguna	Image: Strategy of the	Sistem dapat mengubah data profil pengguna.	Non- Non- Non- Non-	Berhasil

No.	Kasus yang	Test Case	Hasil yang	Hasil vang didapatkan	Status
	diuji		diharapkan		
12.	Fungsi		Sistem dapat		Berhasil
	tambah ide		menambah data ide		
	skripsi pada	∫ π. τα-μ. 2 ((0.00007/2000.0) (0.	skripsi dosen.	T before X second (the Q ₀ ×) Φ <thφ< td=""><td></td></thφ<>	
	menu data ide			Ido SUBIPSI Bende Michael Preprinten Teringkeit Galavet lig bit Data Ide Skripsi	
	skripsi dalam	Tambah Ide Skripsi Menu Manuari Sana Manuari San		Gendulapan Banan San Malanan	
	dashboard	Katagon (* 1970) - 197		Like Stepper Description Kinder Permanance Million Datar marking update address 1 1 2 2 Datar marking marking in the product and interpretation in th	
	dosen	view.		foder f	
13	Fungsi ubah		Sistem dapat		Berhasil
	ide skrinsi	() Sector C () Sector () (0 () × (+ − − − − − − − − − − − − − − − − − −	menguhah data ide	/ 1 May 1 (2 And 1 May 2 +	
	nada menu	Ide SIGNES anato inclano regenerar instance operation	skrinsi dosen	(€ Canal Balance Canal	
	data ida	Edit Ide Skripsi	skiipsi doseli.	Data Ide Skripsi Peri Grithwead State Manager State Manager	
		Lines weeks Description Codings Spatial and ending symmetry Lines 4		Marine Instance Marine Instance Advisings credupt credupt Marine Instance Salarn andere cylical laborgannian f PEC/Mar Imm	
	skripsi dalam	Marcada (M. No.) Indelay Vision		Mitrophicupa (2020.200.ybiophicupa et al. 1 974, 96 20) Mitro Twave Mark Residence Statuso Largo 1 973.18 90 20	
	dashboard				
	dosen				

No.	Kasus yang	Test Case	Hasil yang	Hasil yang didapatkan	Status
	diuji		diharapkan		
14.	Fungsi hapus		Sistem dapat		Berhasil
	ide skripsi		menghapus data		
	pada menu	/ resta // balanderstation (rest / e / e / e / e / e / e / e / e / e /	ide skripsi dosen.	2 * 1000 × 2 ≤ 0.0488 (2018) × 4	
	data ide	Data Ide Skripsi		100 2001 In the second state of the second sta	
	skripsi dalam	Contrarege Contrarege Section 2012		Data Ide Skripsi	
	dashboard	Strategy Privategy Non March Sch and march Sch 4 And Sch 4 And Sch balan same schwarger Langer L		Therefore and the second secon	
	dosen	Solar Novar Skala Solada Salakovar igir 1 (K.K.K.K. 🚺		And the second s	
15.	Fitur seleksi		Sistem dapat		Berhasil
	mahasiswa	(1997) • • • • • • • • • • • • • • • • • • •	menampilkan hasil		
	pada menu	Construction and the second se	seleksi pada	Constant/Southers + Constant/Southers + Constant/Southers + Constant/Southers Constant/S	
	seleksi		halaman		
	mahasiswa	Seleksi Mahasiswa Meru wa sewara Sawaranga	pengumuman	Pengumuman Militarya Susan Sus	
	dalam	Same I a sum I de fordance	r8	international applications, Science Francescon Avenue (SAUSEURIS) NECESSION Status Vision International Avenue (SAUSEURIS) Anterescon Avenue (SAUSEURIS) Anterescon Avenue (SAUSEURIS) Anterescon Avenue (SAUSEURIS) Anterescon Avenue (SAUSEURIS) Avenue (SAUSEURIS	
	dashboard				
	uusnoouru				
	dosen				

No	Kasus yang	Tost Case	Hasil yang	Hasil yang didapatkan	Status
INU.	diuji	Test Case	diharapkan	nasii yang uluapatkan	Status
16.	Fitur		Sistem dapat		Berhasil
	pencarian		menampilkan hasil		
	pada halaman	r intrice x - x = 0 c = 0 c = 0 c = 0 c = 0 c = 0 c = 0 c = 0 c = 0 c = 0 c = 0 c = 0 c = 0 c c = 0 c c = 0 c <td>pencarian sesuai</td> <td>€ 11 Non • • ● • ●</td> <td></td>	pencarian sesuai	€ 11 Non • • ● • ●	
	ide skripsi	Daftar Ide Skripsi	kata kunci		
		Abbigue Desprey Montp	pencarian.	Image: Control of the Storight Image: Control of the Storight Interface: Control of the Storight Interface: Control of the Storight Interface: Control of the Storight Interface: Control of the Storight Interface: Control of the Storight Interface: Control of the Storight Interface: Control of the Storight Interface: Control of the Storight	
17.	Fitur sorting		Sistem dapat		Berhasil
	pada halaman	() 0	menampilkan data	• • • • • • • • • • • • • • • • • • •	
	ide skripsi	Daftar Ide Skripsi	urut dengan pilihan	000 SNB SN week week week week week week week wee	
		Control control Description Description <thdescription< th=""></thdescription<>	sorting.	Data briefs Same of the second s	

No	Kasus yang	Tost Casa	Hasil yang	Hasil yang didapatkan	Status
110.	diuji	diharapkan		masn yang uluapatkan	Status
18.	Fitur ambil ide skripsi oleh mahasiswa	Contract Particular Particular Contract Particular Particular Contract Particular Particular Contract Particular Particular Particular Contract Particular Particular Contract Particular Contract Particular Co	Sistem dapat menerima pengambilan ide skripsi oleh mahasiswa dan menampilkan	Daftar Ide Skripsi Bernarden de se osk wergenderen Bernarden de	Berhasil
		La daga Kana Kana Pang Kana Kana Pang Kana Kana Kana Kana Kana Kana Kana K	notifikasi. Kuota yang tersedia pun berkurang setelah ide skripsi diambil oleh mahasiswa.	Andream Landpoint Landpoint <thlandpoint< th=""> <thlandpoint< th=""> <thlan< td=""><td></td></thlan<></thlandpoint<></thlandpoint<>	

4.3.3 Pengujian Validasi

Pengujian validasi bertujuan untuk mengetahui apakah validasi-validasi yang ada dalam sistem sudah berjalan dengan baik. Hasil pengujian validasi dapat dilihat pada tabel 4.3.



Tabel 4. 3 Hasil pengujian validasi

No.	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapatkan	Status
3.	Validasi jika nomor hp sudah terdaftar		Sistem dapat menampilkan peringatan bahwa nomor hp sudah ada dalam sistem.		Berhasil
4.	Validasi jika <i>username</i> sudah terdaftar		Sistem dapat menampilkan peringatan bahwa <i>username</i> sudah ada dalam sistem.		Berhasil

No.	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapatkan	Status
5.	Validasi jika <i>email</i> sudah terdaftar		Sistem dapat menampilkan peringatan bahwa <i>email</i> sudah ada dalam sistem.		Berhasil
6.	Validasi <i>email</i> jika tidak valid		Sistem dapat menampilkan peringatan bahwa <i>email</i> tidak valid.		Berhasil

No.	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapatkan	Status
7.	Validasi confirm password jika tidak sama dengan password		Sistem dapat menampilkan peringatan bahwa <i>confirm</i> <i>password</i> tidak sama dengan <i>password</i> .		Berhasil
8.	Validasi login jika <i>username</i> atau <i>password</i> salah		Sistem dapat menampilkan peringatan bahwa <i>username</i> atau <i>password</i> salah.		Berhasil

No.	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapatkan	Status
9.	Validasi jika input ide skripsi masih kosong	Instrum Image: Control Image: Contro Image: Contro	Sistem dapat menampilkan peringatan bahwa input harus diisi.	Intermining Intermining Intermining Intermining	Berhasil

4.3.4 Pengujian Keamanan Sistem

Pengujian keamanan sistem ditujukan untuk mengetahui apakah keamanan yang diterapkan dalam sistem sudah berjalan dengan baik. Hasil pengujian keamanan sistem dapat dilihat pada Tabel 4.4.

No	Kasus yang	Tost Casa	Hasil yang	Hasil yang didapatkan	Status
110.	diuji	Test Case	diharapkan	masn yang uluapatkan	Status
1.	Pengguna		Pengguna		Berhasil
	dengan level	(*1830) Mahada (Mé) (or 2) =	dengan level		
	administrator	Traja Traja	administrator	A Streams for A	
			hanya dapat		
			masuk ke		
			dashboard		
			administrator.		
2.	Pengguna		Dashboard		Berhasil
	dengan level	1	mahasiswa		
	mahasiswa	Cajtr	hanya bisa	Ido SidilPSI Bende Istanov cejor	
			diakses oleh	DASHBOARD	
		ing an annual The second se	pengguna	a construction of the second sec	
			dengan level	Dathboard Menu truit date diversity between	
			mahasiswa.		

Tabel 4. 4 Hasil pengujian keamanan sistem

No.	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapatkan	Status
3.	Pengguna dengan level dosen		Dashboard dosen hanya bisa diakses oleh pengguna dengan level dosen.		Berhasil
4.	Pengguna yang tidak <i>login</i> atau level dosen mengambil ide skripsi.	Note: Note: <t< td=""><td>Muncul notifikasi bahwa pengguna harus login sebagai mahasiswa untuk dapat mengambil ide skripsi.</td><td>Index Operation O I and <t< td=""><td>Berhasil</td></t<></td></t<>	Muncul notifikasi bahwa pengguna harus login sebagai mahasiswa untuk dapat mengambil ide skripsi.	Index Operation O I and O I and <t< td=""><td>Berhasil</td></t<>	Berhasil

No.	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapatkan	Status
5.	Mahasiswa yang sudah diterima atau sudah mengambil ide skripsi mengambil ide skripsi untuk kedua kalinya	Image: Section of the section of th	Muncul peringatan bahwa mahasiswa hanya dapat mengambil satu ide skripsi.	Very Alexandree Alexan	Berhasil

4.4 Pembahasan

Website menyediakan fasilitas *login* untuk tiga level pengguna, yaitu administrator, Dosen, dan Mahasiswa. Pada tiap level pengguna, sistem menyediakan fungsi-fungsi yang hanya dapat diakses oleh pengguna dengan levelnya masing-masing.

Pada *login* level administrator, sistem menyediakan menu manajemen akun mahasiswa dan dosen. Dalam manajemen akun mahasiswa dan dosen, Administrator dapat melihat, mencari dan mengurutkan data akun. Administrator juga dapat melakukan reset password akun mahasiswa dan dosen, serta menghapus akun mahasiswa dan dosen.

Pada *login* level dosen, sistem menyediakan menu profil, data ide skripsi, dan seleksi mahasiswa. Pada menu profil, dosen dapat mengubah data diri. Menu data ide skripsi menampilkan data ide skripsi dosen yang bersangkutan. Pada menu ide skripsi, dosen dapat melakukan pencarian, pengurutan, penambahan, pengubahan, dan penghapusan data ide skripsi. Menu seleksi mahasiswa menampilkan data mahasiswa yang mengambil ide skripsi. Pada menu seleksi mahasiswa, dosen dapat memilih mahasiswa untuk kemudian memberikan keputusan diterima atau tidak diterima.

Pada *login* level mahasiswa, sistem menyediakan menu profil, dimana mahasiswa dapat mengubah data diri. Selain itu, terdapat fasilitas pengambilan ide skripsi. Mahasiswa dapat melihat daftar ide skripsi pada halaman ide skripsi, kemudian mengambil ide skripsi yang diinginkan. Jika mahasiswa sudah mengambil ide skripsi, mahasiswa bisa melihat pengumuman untuk mengetahui hasil seleksi.

Secara keseluruhan, sistem dapat berjalan dengan baik sesuai dengan tujuan penulis dalam membuat *Website* Pengambilan Ide Skripsi. Sistem dapat membantu mahasiswa dalam mendapatkan informasi ide skripsi dari dosen. Mahasiswa juga bisa mengambil ide skripsi melalui sistem. Dosen pun dapat mengumumkan hasil seleksi mahasiswa pengambil ide skripsi.