

BAB 4

HASIL PENELITIAN

4.1 RINGKASAN HASIL PENELITIAN

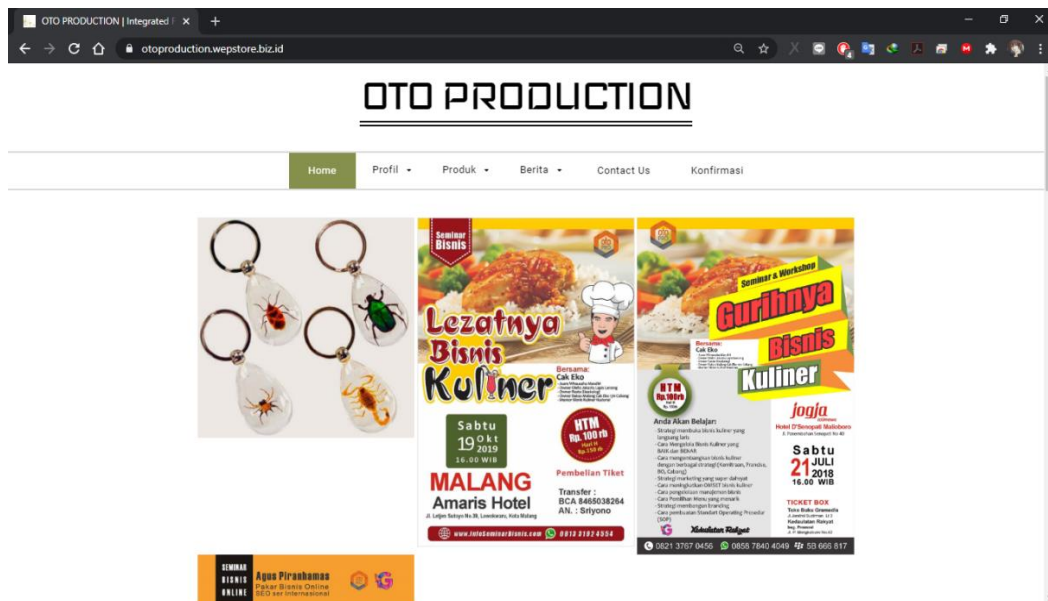
Sistem informasi manajemen ini merupakan sebuah sistem yang digunakan oleh pengunjung dan admin untuk melakukan pendaftaran seminar dan workshop. Sistem ini dapat membantu mempermudah penyampaian informasi seminar ke seluruh tempat dalam upaya meningkatkan jangkauan tersebarnya informasi. Pada sistem ini pengunjung dapat mengisi identitas diri meliputi : nama, alamat, nomer telepon dan lain-lain. Admin dapat melakukan konfirmasi pembayaran yang telah dilakukan oleh pengunjung. Admin juga dapat melakukan proses penambahan data baru, menampilkan data yang sudah ada, pencarian data, perubahan data, penghapusan data. Dalam pengembangan dilakukan beberapa kali pengujian sebelum benar-benar jadi sebuah sistem seperti saat ini..

4.2 IMPLEMENTASI DISAIN INTERFACE

Implementasi aplikasi merupakan langkah realisasi dari perancangan desain interface sistem. Setiap rancangan yang telah dibuat sebelumnya akan direalisasikan menggunakan bahasa pemrograman. Dalam pengembangan sistem informasi manajemen ini, bahasa pemrograman yang digunakan adalah PHP dengan memanfaatkan *framework* CodeIgniter dan pengolahan datanya yaitu menggunakan basis data MySQL. Berikut beberapa kode dan contoh halaman yang terdapat pada sistem informasi manajemen ini.

4.2.1 Halaman Awal

Halaman awal merupakan salah satu halaman default yang tampil pada saat halaman utama diakses. Halaman ini digunakan oleh pengunjung untuk melihat berbagai event, seminar dan workshop yang akan berlangsung. Tampilan halaman login dapat dilihat pada Gambar 4.1.



Gambar 4.1 Halaman Awal

Kode berikut berada pada *controller* digunakan untuk menampilkan halaman utama website, function ini berfungsi untuk menampilkan sesuai dengan menu yang tersedia (kode pada baris 10-23).

```

1. public function __construct(){
2.     parent::__construct();
3.     $this->load->model('konfigurasi_model');
4.     $this->load->model('produk_model');
5.     $this->load->model('berita_model');
6.     $this->load->model('video_model');
7. }
8.
9. // Index
10. public function index() {
11.     $site = $this->konfigurasi_model->listing();
12.     $produk = $this->produk_model->home();
13.     $berita = $this->berita_model->home();
14.     $video = $this->video_model->home();
15.
16. $data = array( 'title' => $site['namaweb'].' | '.$site['tag
    line'],
17. 'keywords' => $site['namaweb'].'', '.$site['keywords'],
18. 'produk' => $produk,
19. 'berita' => $berita,
20. 'video' => $video,
21. 'isi' => 'home/list');
22. $this->load->view('layout/wrapper',$data);
23. }

```

4.2.2 Halaman Login

Login merupakan salah satu menu yang terdapat pada halaman utama saat akan masuk halaman admin. Halaman ini digunakan admin untuk masuk ke dasbor dengan cara memasukkan username, password secara benar sesuai dengan email yang telah terdaftar. Tampilan halaman login dapat dilihat pada Gambar 4.2.



Gambar 4.2 Halaman Login

Kode berikut berada pada *controller* digunakan untuk memvalidasi form login agar sesuai dengan aturan, tidak boleh kosong dan harus cocok dengan isian yang dimasukkan pada saat user dibuat. Pengecekan tersebut terjadi pada saat user menekan tombol login kemudian data yang telah dimasukkan user di validasi oleh sistem (kode pada baris 3-9). Setelah tervalidasi maka akan dialihkan ke halaman dashboard ketika inputan yang di masukkan berhasil (kode pada baris 10). Apabila ada yang tidak diisi akan tampil peringatan (kode pada baris 4-5) dan akan tampil pesan error ketika username/password yang di imputkan salah (kode pada baris 14).

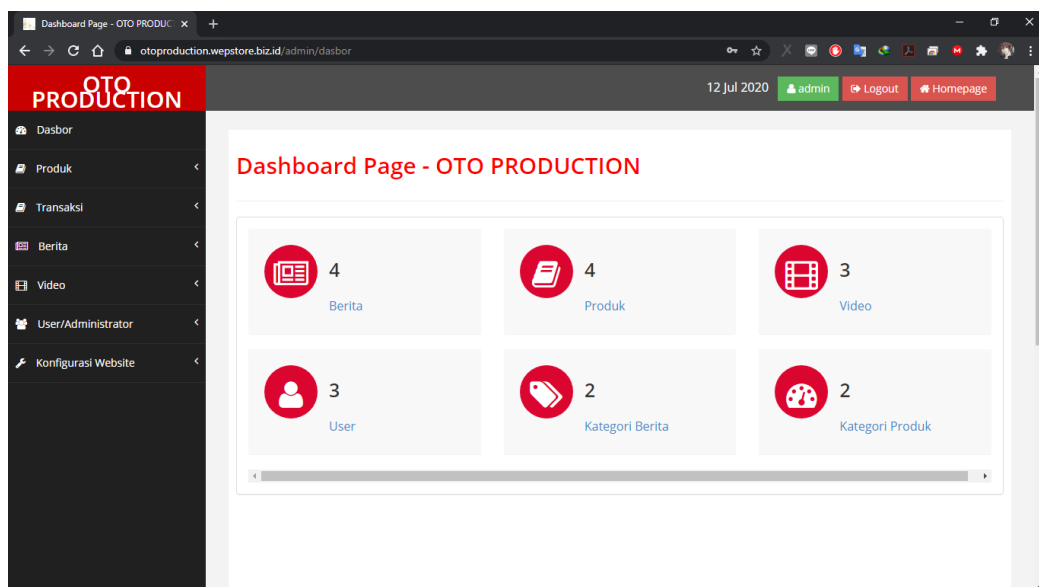
```

1. public function index() {
2. // Validasi
3. $valid = $this->form_validation;
4. $valid->set_rules('username','Username','required',
    array('required' => 'Username harus diisi'));
5. $valid->set_rules('password','Password','required',
    array('required' => 'Password harus diisi'));
6. $username = $this->input->post('username');
7. $password = $this->input->post('password');
8.
9. if($valid->run()) {
10.$this->simple_login->login($username,$password,
    base_url('admin/dasbor'), base_url('login'));
11.}
12.// End validasi
13.

```

4.2.3 Halaman Dashboard

Halaman *dashboard* merupakan halaman yang tampil terlebih dahulu ketika *user* telah *login* ke halaman admin. Halaman ini digunakan untuk melihat secara ringkas isi dari keseluruhan data didalam website. Tampilan halaman dashboard dapat dilihat pada Gambar 4.3.



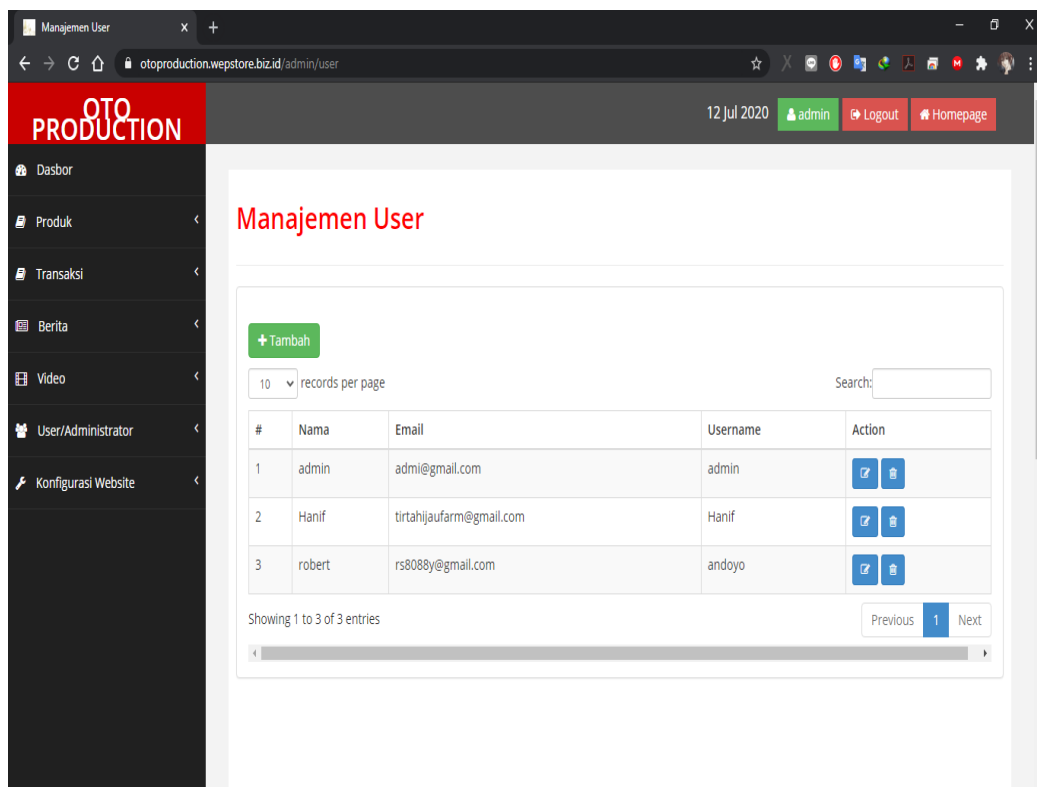
Gambar 4.3 Halaman Dashboard

Kode berikut berada pada *controller* digunakan untuk menampilkan halaman utama dashboard, function ini berfungsi untuk menampilkan isi sesuai dengan menu yang akan tersedia (kode pada baris 10-19). Kode pada baris 2-8 berfungsi untuk menghubungkan halaman dengan data yang tersimpan dalam table yang ada di database,

```
1. public function index() {
2.     $site= $this->konfigurasi_model->listing();
3.     $user= $this->user_model->listing();
4.     $video= $this->video_model->listing();
5.     $berita= $this->berita_model->listing();
6.     $produk= $this->produk_model->listing();
7.     $kategori_produk = $this->kategori_produk_model->listing();
8.     $kategori_berita = $this->kategori_berita_model->listing();
9.
10. $data = array('title'=> 'Dashboard Page - '.$site['namaweb'],
11.             'user' => $user,
12.             'video' => $video,
13.             'berita' => $berita,
14.             'produk' => $produk,
15.             'kategori_produk'=> $kategori_produk,
16.             'kategori_berita'=> $kategori_berita,
17.             'isi' => 'admin/dasbor/list');
18.     $this->load->view('admin/layout/wrapper',$data);
19. }
```

4.2.4 Halaman Manajemen Data User

Halaman pada Gambar 4.4 ini digunakan untuk menampilkan data *user* sistem. Pada halaman ini terdapat tombol untuk menambah user akun. Terdapat juga tombol edit dan delete pengguna, untuk merubah data user yang ada. Halaman ini menampilkan data pengguna sistem diambil dari Table data *user*.



Gambar 4.4 Halaman Data User

Kode berikut berada pada *view* digunakan untuk menampilkan halaman manajemen data user, kode tersebut untuk menampilkan data user yang disajikan dalam sebuah Table (kode pada baris 14-20). Kode pada baris 21-22 berfungsi untuk melakukan prosed edit dan delete pada data yang ditampilkan.

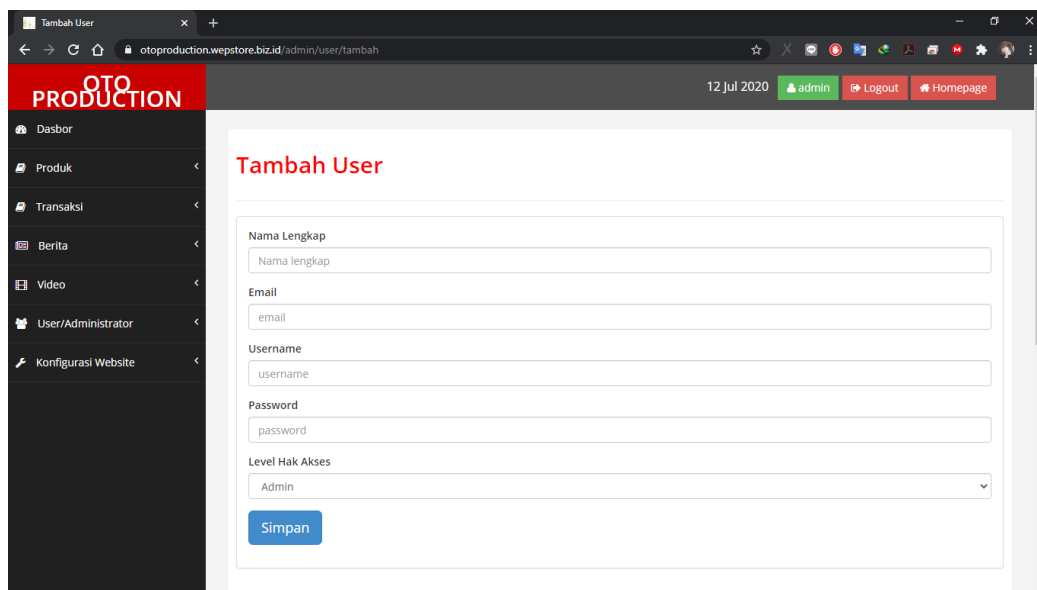
```

1. <p><a href="<?php echo base_url('admin/user/tambah') ?>" class
   ="btn btn-success">
2. <i class="fa fa-plus"></i> Tambah</a></p>
3. <table class="table table-striped table-bordered table-hover"
   id="dataTables-example">
4. <thead>
5.     <tr>
6.         <th>#</th>
7.         <th>Nama</th>
8.         <th>Email</th>
9.         <th>Username</th>
10.        <th>Action</th>
11.    </tr>
12.</thead>
13.<tbody>
14.<?php $i=1; foreach($user as $user) { ?>
15.    <tr class="odd gradeX">
16.        <td><?php echo $i ?></td>
17.        <td><?php echo $user->nama ?></td>
18.        <td><?php echo $user->email ?></td>
19.        <td><?php echo $user->username ?></td>
20.        <td>
21.<a href="<?php echo base_url('admin/user/edit/' . $user->id_user
   ) ?>"class="btn btn-primary btn-sm"><i class="fa fa-edit"></i>
   </a>
22.<a href="<?php echo base_url('admin/user/delete/' . $user->id_us
   er) ?>"class="btn btn-primary btn-sm" onClick="return confirm(
   'Yakin ingin menghapus user ini?')"><i class="fa fa-trash-o"><
   /i></a>
23.        </td>
24.    </tr>
25.<?php $i++; } ?>
26.</tbody>
27.</table>

```


4.2.5 Halaman Form Tambah User

Halaman form ini digunakan oleh admin untuk menambah data user. Pada form ini admin mengisikan data user yang akan ditambahkan. Seperti nama, email, password dan *role*(level hak akses). Tampilan halamanan ini dapat dilihat seperti pada Gambar 4.5.



The screenshot shows a web browser window with the URL `otoproduction.wepstore.biz.id/admin/user/tambah`. The page title is "Tambah User". On the left, there is a dark sidebar menu with the "OTO PRODUCTION" logo at the top and several menu items: "Dasbor", "Produk", "Transaksi", "Berita", "Video", "User/Administrator", and "Konfigurasi Website". The main content area is white and contains the following form fields:

- Nama Lengkap**: A text input field with the placeholder "Nama lengkap".
- Email**: A text input field with the placeholder "email".
- Username**: A text input field with the placeholder "username".
- Password**: A text input field with the placeholder "password".
- Level Hak Akses**: A dropdown menu currently showing "Admin".
- Simpan**: A blue button to submit the form.

At the top right of the page, there is a navigation bar showing the date "12 Jul 2020", the user "admin", and buttons for "Logout" and "Homepage".

Gambar 4.5 Halaman Tambah Data User

Kode berikut berada pada *view* digunakan untuk menambah data user, kode tersebut untuk memvalidasi form agar harus diisi (kode pada baris 3-8). Kode pada baris 15-17 berfungsi untuk menyimpan data user ke database setelah divalidasi dan akan kembali ke halaman data user ketika berhasil, kode pada baris 8-11 berfungsi untuk mengembalikan ke halaman tambah user ketika validasi gagal.

```

1. public function tambah() {
2.     // Validasi
3.     $valid = $this->form_validation;
4.     $valid->set_rules('nama','Nama','required',array( 'required' =
    > 'Nama harus diisi'));
5.     $valid->set_rules('email','Email','required',array( 'required'
    => 'email harus diisi'));
6.     $valid->set_rules('username','Username','required|is_unique[us
    ers.username]',array( 'required' => 'Username harus
    diisi','is_unique' => 'Username: <strong>'.$this->input->post('
    username').'</strong> sudah digunakan. Buat username baru!'));
7.     $valid->set_rules('password','Password','required',array( 'req
    uired' => 'Password harus diisi'));
8.     if($valid->run()===FALSE) {
9.         // End validasi
10.    $data = array( 'title' => 'Tambah User','isi' => 'admin/user/
    tambah');
11.    $this->load->view('admin/layout/wrapper',$data);
12.        // masuk database
13.    }else{
14.        $i = $this->input;
15.    $data = array('nama'=> $i->post('nama'),'email' => $i
    ->post('email'),'username' => $i->post('username'),'passw
    ord' => $i->post('password'),'akses_level' =>
    $i->post('akses_level'));
16.    $this->user_model->tambah($data);
17.    $this->session->set_flashdata('sukses','user telah ditambah');
18.    redirect(base_url('admin/user'));
19.    }
20.        // End masuk database
21.    }

```

4.2.6 Halaman Manajemen Data Produk

Halaman pada Gambar 4.6 ini digunakan untuk menampilkan data produk di sistem. Pada halaman ini terdapat tombol untuk menambah produk. Terdapat juga tombol edit dan delete produk, untuk merubah data produk yang ada. Halaman ini menampilkan data produk yang ada di sistem diambil dari Table data produk.

#	Gambar	Nama Produk	Kategori	Jadwal	Harga - Stok	Status	Action
1		pelatihan andorid	Workshop	2020-10-10 08:00:00	Rp. 50.000 - 50 Orang	Publish	
2		Seminar Bisnis Kuliner2	Workshop	2020-01-01 08:00:00	Rp. 100.000 - 100 orang	Publish	
3		Seminar Bisnis Kuliner	Workshop	2020-03-27 06:02:47	Rp. 24.000 - 100 orang	Publish	
4		BANJIR ORDER DARI INTERNET	Workshop	2020-03-27 06:02:47	Rp. 12.000 - 12 orang	Publish	

Gambar 4.6 Halaman Data Produk

Kode berikut berada pada controller yang digunakan untuk menampilkan halaman manajemen data produk, kode tersebut menghubungkan model dengan view agar data yang ada di database bisa tampil pada view (kode pada baris 1-7), kode tersebut untuk menampilkan data produk yang disajikan dalam sebuah Table (kode pada baris 9-32).

```

1. public function index() {
2.     $produk = $this->produk_model->listing();
3.     $data = array(      'title'           => 'Data produk',
4.     'produk'   => $produk,
5.     'isi'       => 'admin/produk/list');
6.     $this->load->view('admin/layout/wrapper', $data);
7. }
8. //view
9. <table class="table table-striped table-bordered table-hover"
   id="dataTables-example">
10.<thead><tr>
11.<th>#</th>
12.<th>Gambar</th>
13.<th>Nama Produk</th>
14.<th>Kategori</th>
15.<th>Jadwal</th>
16.<th>Harga - Stok</th>
17.<th>Status</th>
18.<th>Action</th>
19.</tr></thead><tbody>
20.<?php $i=1; foreach($produk as $produk) { ?>
21.<tr class="odd gradeX">
22.<td><?php echo $i ?></td>
23.<td><
   /td>
24.<td><?php echo $produk->nama_produk ?></td>
25.<td><?php echo $produk->nama_kategori_produk ?></td>
26.<td><?php echo $produk->jadwal_produk ?></td>
27.<td>Rp. <?php echo number_format($produk->harga,'0',' ','.') ?
   > -
28.<?php echo number_format($produk->stok,'0',' ','.') . ' ' . $produ
   k->satuan ?> </td>
29.<td><?php echo $produk->status_produk ?></td>
30.<td><a href="<?php echo base_url('admin/produk/edit/'.$produk-
   >id_produk) ?>" class="btn btn-primary btn-sm"><i class="fa fa-
   edit"></i></a>
31.<?php include('delete.php') ?>
32.</td></tr><?php $i++; } ?></tbody></table>

```

4.2.7 Halaman Form Tambah Produk

Halaman form ini digunakan oleh admin untuk menambah data produk. Pada form ini admin mengisi detail produk yang akan ditambahkan. Seperti nama, status, kategori, jadwal, harga, dan lain-lain. Tampilan halamanan ini dapat dilihat seperti pada Gambar 4.7.

Gambar 4.7 Halaman Tambah Produk

Kode berikut berada pada *view* digunakan untuk menambah data produk, kode tersebut untuk memvalidasi form agar harus diisi (kode pada baris 4-14). Kode pada baris 32-49 berfungsi untuk menyimpan data produk ke database setelah divalidasi dan akan kembali ke halaman data produk ketika berhasil, kode pada baris 17-28 berfungsi untuk mengembalikan ke halaman tambah produk ketika validasi gagal.

```

1. public function tambah() {
2.     $kategori          = $this->kategori_produk_model->listing();
3.     // Validasi
4.     $v = $this->form_validation;
5.     $v->set_rules('nama_produk','Nama produk','required|is_unique[produk.nama
   _produk]',
6.     array(          'required'          => 'Nama produk harus diisi',
7.     'is_unique'      => 'Nama produk: <strong>'.$this->input->post('na
   ma_produk').
8.     '</strong> sudah ada. Buat nama yang berbeda'));
9.     $v->set_rules('harga','Harga produk','required',
10.    array(          'required'          => 'Harga produk harus diisi'
   ));
11.    $v->set_rules('stok','Stok produk','required',
12.    array(          'required'          => 'Stok produk harus diisi'))
   ;
13.    $v->set_rules('keterangan','Keterangan produk','required',
14.    array(          'required'          => 'Keterangan produk harus di
   isi'));
15.
16.    if($v->run()) {
17.        $config['upload_path']          = './assets/upload/image/';
18.        $config['allowed_types']        = 'gif|jpg|png|svg';
19.        $config['max_size']              = '12000'; // KB
20.        $this->load->library('upload', $config);
21.        if(! $this->upload->do_upload('gambar')) {
22.            // End validasi
23.
24.            $data = array( 'title'          => 'Tambah produk',
25.            'kategori'      => $kategori,
26.            'error'         => $this->upload->display_errors(),
27.            'isi'           => 'admin/produk/tambah');
28.            $this->load->view('admin/layout/wrapper', $data);
29.            // Masuk database
30.        }else{
31.            // Proses ke database
32.            $i = $this->input;
33.            $data = array( 'id_user'          => $this->session->us
   erdata('id'),
34.            'id_kategori_produk'      => $i->post('id_kategori_produk'),
35.            'slug_produk'              => url_title($i->post('nama_produk'),'da
   sh',TRUE),
36.            'nama_produk'              => $i->post('nama_produk'),
37.            'keterangan'               => $i->post('keterangan'),
38.            'jadwal_produk'            => $i->post('jadwal_produk'),
39.            'alamat_produk'            => $i->post('alamat_produk'),
40.            'harga'                    => $i->post('harga'),
41.            'stok'                      => $i->post('stok'),
42.            'satuan'                    => $i->post('satuan'),
43.            'status_produk'            => $i->post('status_produk'),
44.            'gambar'                   => $upload_data['uploads']['fi
   le_name'],
45.            'tanggal_post'              => date('Y-m-d H:i:s')
46.        );
47.        $this->produk_model->tambah($data);
48.        $this->session->set_flashdata('sukses','Produk telah ditambah');
49.        redirect(base_url('admin/produk'));
50.    }}
51.    // End masuk database
52.    $data = array( 'title'          => 'Tambah produk',
53.    'kategori'      => $kategori,
54.    'isi'           => 'admin/produk/tambah');
55.    $this->load->view('admin/layout/wrapper', $data);
56. }

```

4.2.8 Halaman Form Pemesanan

Form ini digunakan oleh pengunjung untuk memesan produk. Pada form ini pengunjung mengisi data diri yang akan ditambahkan dan pilihan pembayarannya. Tampilan halaman form ini dapat dilihat seperti pada Gambar 4.8.

The image shows a web browser window with the URL `otoproduction.wepstore.biz.id/pesan/register/seminar-bisnis-kuliner2`. The page content is split into two main sections. On the left is a promotional banner for a seminar. The banner features a chef character and text: "Seminar Bisnis", "Lezatnya Bisnis Kuliner", "Bersama: Cak Eko", "Sabtu 19 Okt 2019 16.00 WIB", and "HTM Rp. 100 rb Hari H Rp. 150 rb". On the right is a registration form titled "Register Pemesanan Tiket Seminar Bisnis Kuliner2". The form contains the following fields and controls:

- Nama:** Input field with placeholder "Nama Lengkap".
- No.Telepon:** Input field with placeholder "Nomor Telepon".
- E-mail:** Input field with placeholder "E-mail".
- Alamat:** Input field with placeholder "ALamat".
- Jumlah Tiket:** Input field.
- Harga Tiket:** Input field with value "Rp. 100.000".
- Total Pembayaran:** Input field.
- Jenis Pembayaran:** Dropdown menu with selected option "Bayar ditempat".
- Buttons:** "Kembali" (red) and "Lanjut Pemesanan" (blue).

Gambar 4.8 Halaman Form Pemesanan

Kode berikut berada pada controller yang digunakan untuk memproses penyimpanan pesanan, kode tersebut menghubungkan model dengan view agar data yang ada di database bisa tampil pada view (kode pada baris 1-5), kode tersebut untuk mengirimkan pesan kepada pemesan sebagai bukti pemesanan dan pilihan pembayaran yang akan digunakan (kode pada baris 7-43).

```

1. // Register
2. public function register($slug_produk) {
3. $site      = $this->konfigurasi_model->listing();
4. $produk    = $this->produk_model->home();
5. $read      = $this->produk_model->read($slug_produk);
6.
7. $data      = array( 'title' => $read->nama_produk,
8. 'keywords' => $read->nama_produk,
9. 'produk'   => $produk,
10. 'read'     => $read,
11. 'slug_produk' => $slug_produk,
12. 'isi'      => 'pesan/register');
13. $this->load->view('layout/wrapper',$data); }
14. $this->email->subject('Notifikasi Pembayaran | otoproduction')
    ;
15. $body = '
16. OTO PRODUCTION                Notifikasi Pembayaran
17. Atas Nama : '.$i->post('nama').'
18. Slug Produk : '.$i->post('slug_produk').'
19. Total Harga : Rp. '.$i->post('total_bayar').'
20. Metode Pembayaran : '.$i->post('jenis_pembayaran').'
21. Berita Acara : '.$berita_acara.'
22. Masukkan Berita Acara pada saat transfer
23. dan saat konfirmasi bukti pembayaran !!!
24. Detail Nomor Rekening :
25. Rekening : BRI
26. No. Rekening : 098997***
27. Atas Nama : Nevinda
28. Rekening : BNI
29. No. Rekening : 435467***
30. Atas Nama : Nevinda
31. Rekening : BCA
32. No. Rekening : 786545***
33. Atas Nama : Nevinda
34. Rekening : Mandiri
35. No. Rekening : 7654768****
36. Atas Nama : Nevinda
37. ';
38. // Isi email
39. $this->email->message($body);
40. $this->email->send();
41. // akhir email
42. redirect(base_url('produk/read/'.$i->post('slug_produk')));
43. }

```


4.2.9 Halaman Form Konfirmasi

Form ini digunakan oleh pengunjung untuk mengkonfirmasi pembayaran. Pada form ini pengunjung mengisi kode transaksi dan upload bukti pembayaran yang telah dilakukan. Kemudian pengunjung tinggal menunggu validasi dari admin dan akan mendapatkan notifikasi di email. Tampilan halaman form ini dapat dilihat seperti pada Gambar 4.9.

Seminar Bisnis Kuliner2	
Berita Acara	<input type="text" value="Berita Acara"/>
Nama Workshop	<input type="text" value="seminar-bisnis-kuliner2"/>
Upload Bukti Pembayaran	<input type="button" value="Pilih File"/> Tidak ada file yang dipilih
<input type="button" value="Kembali"/>	<input type="button" value="Lanjut Konfirmasi"/>

Gambar 4.9

Gambar 4.9 Halaman Form Konfirmasi

Kode berikut berada pada controller yang digunakan untuk memproses penyimpanan konfirmasi pembayaran, kode tersebut untuk proses menyimpan data konfirmasi kedalam database sesuai produk yang di pesan (kode pada baris 1-23), kode tersebut untuk mengirimkan pesan kepada pemesan sebagai bukti bahwa konfirmasi pembayaran telah diterima (kode pada baris 26-59).

```

1. public function tambah() {
2. $kategori = $this->kategori_produk_model->listing();
3. $config['upload_path'] = './assets/upload/image/';
4. $config['allowed_types'] = 'gif|jpg|png|svg';
5. $config['max_size'] = '12000'; // KB
6. $this->load->library('upload', $config);
7. if(! $this->upload->do_upload('gambar')) {
8. // End validasi
9. $data = array( 'title' => 'Tambah produk',
10. 'kategori' => $kategori,
11. 'error' => $this->upload->display_errors(),
12. 'isi' => 'konfirmasi/tambah');
13. $this->load->view('layout/wrapper',$data);
14. // Masuk database
15. }else{
16. // Proses ke database
17. $i = $this->input;
18. $data = array( 'berita_acara' => $i->post(
19. 'berita_acara'),
20. 'slug_produk' => $i->post('slug_produk'),
21. 'gambar' => $upload_data['uploads']['file_name'],
22. 'tanggal_konfirmasi' => date('Y-m-d H:i:s')
23. );
24. $this->produk_model->tambah_konfirmasi($data);
25. $berita_acara = $i->post('berita_acara');
26. $this->produk_model->update_status_pesanan($berita_acara);
27. $email = $this->produk_model->read_email($berita_acara);
28. $this->load->library('email');
29. // Konfigurasi email
30. $config = [
31. 'mailtype' => 'html',
32. 'charset' => 'utf-8',
33. 'protocol' => 'smtp',
34. 'smtp_host' => 'smtp.gmail.com',
35. 'smtp_user' => 'otoproduction07@gmail.com', // Email gmail
36. 'smtp_pass' => 'omset100juta', // Password gmail
37. 'smtp_crypto' => 'ssl',
38. 'smtp_port' => 465,
39. 'crlf' => "\r\n",
40. 'newline' => "\r\n"];
41. // Load library email dan konfigurasinya
42. $this->load->library('email', $config);
43. // Email dan nama pengirim
44. $this->email->from('otoproduction07@gmail.com', 'otoproduction');
45. // Email penerima
46. $this->email->to($email['email']); // Ganti dengan email tujuan
47. // Lampiran email, isi dengan url/path file
48. // Subject email
49. $this->email->subject('Konfirmasi Bukti Pembayaran | otoproduction');
50. $body = '
51. OTO PRODUCTION Konfirmasi Bukti Pembayaran
52. Atas Nama : '.$email['nama'].'
53. Slug Produk : '.$email['slug_produk'].'
54. Berita Acara : '.$berita_acara.'

```

4.3 PENGUJIAN SISTEM

Pengujian yang akan dilakukan dengan metode *black box*. Pengujian *black box* menitik beratkan pada fungsionalitas sistem. Metode ini digunakan untuk mengetahui apakah sistem berfungsi dengan benar.

4.3.1 Pengujian *Form Login Admin*

Hasil pengujian pada halaman login admin website, dapat dilihat pada Table 4.1 berikut.

Table 4.1 Pengujian *Form Login Admin*

No	Fungsi yang diuji	Cara pengujian	Hasil yang diharapkan	Keterangan
1.	<i>Form Login Sistem</i>	Melakukan login dengan <i>username</i> dan password yang valid	Berhasil masuk kedalam sistem	OK berhasil login
2.	Link Logout	Mengklik Logout	Berhasil melakukan logout	OK berhasil logout
3.	<i>Form Login</i>	Melakukan login dengan <i>username</i> dan password yang tidak valid	Menampilkan peringatan password dan <i>username</i> tidak valid	OK berhasil menampilkan pesan

4.3.2 Pengujian *Dashboard Admin*

Hasil pengujian pada halaman dashboard admin website, dapat dilihat pada Table 4.2 berikut ini.

Table 4.2 Pengujian *Dashboard Admin*

No.	Fungsi yang diuji	Cara pengujian	Hasil yang diharapkan	Keterangan
1.	Memastikan link pada tampilan dashboard sesuai	Mengklik satu per satu link pada dashboard	Menampilkan halaman sesuai link yang dipilih	Ok berhasil menampilkan halaman sesuai dengan link yang dipilih

4.3.3 Pengujian Data Users

Hasil pengujian pada halaman data *users*, dapat dilihat pada Table 4.3 berikut ini.

Table 4.3 Pengujian Data *Users*

No.	Fungsi yang diuji	Cara pengujian	Hasil yang diharapkan	Keterangan
1.	<i>Users</i>	Klik menu <i>users</i>	Menampilkan halaman <i>users</i>	OK
2.	Tambah <i>users</i> dan <i>form</i> tambah <i>users</i>	Masukan data <i>users</i> di <i>form</i> tambah data sebelah kiri halaman dan mengklik tombol simpan	Berhasil menyimpan	OK
3.	Edit data <i>users</i> dan <i>form</i> edit data <i>users</i>	Klik tombol edit pada Table data <i>users</i> dan menampilkan <i>form</i> edit data <i>users</i>	Berhasil melakukan edit data <i>users</i>	OK
5.	Hapus data <i>users</i>	Klik tombol hapus <i>users</i> pada Table data <i>users</i>	Menampilkan notifikasi persetujuan, jika klik ok maka berhasil menghapus data <i>users</i>	OK
6.	Mencari data <i>users</i>	Memasukan <i>keyword</i> yang berkaitan dengan data <i>users</i> yang dicari	Menampilkan data <i>users</i> yang dicari berdasarkan <i>keyword</i>	OK

4.3.4 Pengujian Data Produk

Hasil pengujian pada halaman data *produk*, dapat dilihat pada Table 4.4 berikut ini.

Table 4.4 Pengujian Data *Produk*

No.	Fungsi yang diuji	Cara pengujian	Hasil yang diharapkan	Keterangan
1.	<i>Produk</i>	Klik menu <i>produk</i>	Menampilkan halaman <i>produk</i>	OK
2.	Tambah <i>produk</i> dan <i>form</i> tambah <i>produk</i>	Masukan data <i>produk</i> di <i>form</i> tambah data sebelah kiri halaman dan mengklik tombol simpan	Berhasil menyimpan	OK
3.	Edit data <i>produk</i> dan <i>form</i> edit data <i>produk</i>	Klik tombol edit pada Table data <i>produk</i> dan menampilkan <i>form</i> edit data <i>produk</i>	Berhasil melakukan edit data <i>produk</i>	OK
5.	Hapus data <i>produk</i>	Klik tombol hapus <i>produk</i> pada Table data <i>produk</i>	Menampilkan notifikasi persetujuan, jika klik ok maka berhasil menghapus data <i>produk</i>	OK
6.	Mencari data <i>produk</i>	Memasukan <i>keyword</i> yang berkaitan dengan data <i>produk</i> yang dicari	Menampilkan data <i>produk</i> yang dicari berdasarkan <i>keyword</i>	OK

4.3.5 Pengujian Data Berita

Hasil pengujian pada halaman data *berita*, dapat dilihat pada Table 4.5 berikut ini.

Table 4.5 Pengujian Data *Berita*

No.	Fungsi yang diuji	Cara pengujian	Hasil yang diharapkan	Keterangan
1.	<i>Berita</i>	Klik menu <i>berita</i>	Menampilkan halaman <i>berita</i>	OK
2.	Tambah <i>berita</i> dan <i>form</i> tambah <i>berita</i>	Masukan data <i>berita</i> di <i>form</i> tambah data sebelah kiri halaman dan mengklik tombol simpan	Berhasil menyimpan	OK
3.	Edit data <i>berita</i> dan <i>form</i> edit data <i>berita</i>	Klik tombol edit pada Table data <i>berita</i> dan menampilkan <i>form</i> edit data <i>berita</i>	Berhasil melakukan edit data <i>berita</i>	OK
5.	Hapus data <i>berita</i>	Klik tombol hapus <i>berita</i> pada Table data <i>berita</i>	Menampilkan notifikasi persetujuan, jika klik ok maka berhasil menghapus data <i>berita</i>	OK
6.	Mencari data <i>berita</i>	Memasukan <i>keyword</i> yang berkaitan dengan data <i>berita</i> yang dicari	Menampilkan data <i>berita</i> yang dicari berdasarkan <i>keyword</i>	OK

4.3.6 Pengujian Data Video

Hasil pengujian pada halaman data *video*, dapat dilihat pada Table 4.6 berikut ini.

Table 4.6 Pengujian Data *Video*

No.	Fungsi yang diuji	Cara pengujian	Hasil yang diharapkan	Keterangan
1.	<i>Video</i>	Klik menu <i>video</i>	Menampilkan halaman <i>video</i>	OK
2.	Tambah <i>video</i> dan <i>form</i> tambah <i>video</i>	Masukan data <i>video</i> di <i>form</i> tambah data sebelah kiri halaman dan mengklik tombol simpan	Berhasil menyimpan	OK
3.	Edit data <i>video</i> dan <i>form</i> edit data <i>video</i>	Klik tombol edit pada Table data <i>video</i> dan menampilkan <i>form</i> edit data <i>video</i>	Berhasil melakukan edit data <i>video</i>	OK
5.	Hapus data <i>video</i>	Klik tombol hapus <i>video</i> pada Table data <i>video</i>	Menampilkan notifikasi persetujuan, jika klik ok maka berhasil menghapus data <i>video</i>	OK
6.	Mencari data <i>video</i>	Memasukan <i>keyword</i> yang berkaitan dengan data <i>video</i> yang dicari	Menampilkan data <i>video</i> yang dicari berdasarkan <i>keyword</i>	OK

4.3.7 Pengujian Data Transaksi

Hasil pengujian pada halaman data *transaksi*, dapat dilihat pada Table 4.7 berikut ini.

Table 4.7 Pengujian Data *Transaksi*

No.	Fungsi yang diuji	Cara pengujian	Hasil yang diharapkan	Keterangan
1.	<i>Transaksi</i>	Klik menu <i>transaksi</i>	Menampilkan halaman <i>transaksi</i>	OK
2.	Tambah <i>transaksi</i> dan <i>form</i> tambah <i>transaksi</i>	Masukan data <i>transaksi</i> di <i>form</i> tambah data sebelah kiri halaman dan mengklik tombol simpan	Berhasil menyimpan	OK
3.	Edit data <i>transaksi</i> dan <i>form</i> edit data <i>transaksi</i>	Klik tombol edit pada Table data <i>transaksi</i> dan menampilkan <i>form</i> edit data <i>transaksi</i>	Berhasil melakukan edit data <i>transaksi</i>	OK
5.	Hapus data <i>transaksi</i>	Klik tombol hapus <i>transaksi</i> pada Table data <i>transaksi</i>	Menampilkan notifikasi persetujuan, jika klik ok maka berhasil menghapus data <i>transaksi</i>	OK
6.	Mencari data <i>transaksi</i>	Memasukan <i>keyword</i> yang berkaitan dengan data <i>transaksi</i> yang dicari	Menampilkan data <i>transaksi</i> yang dicari berdasarkan <i>keyword</i>	OK

4.3.8 Pengujian Pesan Tiket

Hasil pengujian pada halaman pesan tiket, dapat dilihat pada Table 4.8 berikut ini.

Table 4.8 Pengujian Pesan Tiket

No.	Fungsi yang diuji	Cara pengujian	Hasil yang diharapkan	Keterangan
1.	Pesan tiket	Klik menu <i>pesan tiket</i>	Menampilkan halaman <i>pesan tiket</i>	OK
2.	Input data	Masukan data pemesan di <i>form</i> pemesana sebelah kanan halaman dan mengklik tombol lanjut pemesanan	Berhasil menyimpan	OK
3	Input data	Memasukan data pemesan yang tidak sesuai, seperti nomor atau nama dengan karakter berbeda atau tidak valid	Menampilkan pesan error ketika data yang di inputkan tidak sesuai	OK

4.3.9 Pengujian Konfirmasi

Hasil pengujian pada halaman konfirmasi, dapat dilihat pada Table 4.9 berikut ini.

Table 4.9 Pengujian Pesan Tiket

No.	Fungsi yang diuji	Cara pengujian	Hasil yang diharapkan	Keterangan
1.	Konfirmasi	Klik menu <i>konfirmasi</i>	Menampilkan halaman <i>konfirmasi</i>	OK
2.	Input data	Masukan data pembayaran dan upload bukti pembayaran di <i>form</i> pemesanan sebelah kanan halaman dan mengklik tombol lanjut pemesanan	Berhasil menyimpan	OK
3	Input data	Memasukan data pemesan yang tidak sesuai, seperti gambar dengan ukuran besar atau video atau tidak valid	Menampilkan pesan error ketika data yang di inputkan tidak sesuai	OK

4.4 PEMBAHASAN

Sistem informasi manajemen ini digunakan untuk memberikan kemudahan pemasaran atau penyebaran informasi mengenai seminar dan workshop, serta memudahkan pengunjung untuk melakukan pemesanan. Penyebaran informasi yang sebelumnya hanya dengan menyebarkan brosur atau banner tidak mampu menjangkau ke banyak tempat, dengan menempatkan brosur atau banner dalam

sebuah *website*, penyebaran informasinya dapat lebih luas. Sistem informasi ini pun sudah didesain dengan baik sehingga sudah *user friendly* dan *responsive* sehingga dapat diakses dari berbagai jenis perangkat atau gadget.

Sistem informasi ini pun mendukung kemudahan bagi pengunjung untuk melakukan pemesanan tiket, pengunjung cukup memilih produk yang diinginkan, kemudian mengisi data diri dan melakukan pembayaran, kemudian mengkonfirmasi pembayarannya. Admin yang akan memvalidasi pembayaran dan pemesanan sudah sesuai atau belum dan kemudian akan memberikan pengingat detail informasi ketika seminar akan dilaksanakan.

Mengelola pesanan atau data peserta juga lebih mudah karena sudah tersimpan dalam sistem, sehingga admin lebih mudah untuk mengelola dan merekap peserta yang akan mengikuti seminar atau workshop. Tugas manajer pun menjadi lebih mudah, tanpa harus meminta data dari admin, manajer bisa selalu memantau perkembangan jumlah peserta seminar atau workshop kapanpun dan dimanapun.

4.5 KETERBATASAN SISTEM

Sistem ini belum memiliki fitur konfirmasi pembayaran otomatis, hal ini tidak menjadi masalah ketika jumlah pendaftar sedikit. Namun, ketika jumlah peserta banyak sekitar diatas 500 orang, akan lebih mudah dengan menggunakan sistem konfirmasi pembayaran otomatis. Sistem belum dapat menampilkan grafik penjualan atau peserta per produk, untuk mempermudah tugas manajer dalam memonitoring perkembangan seminar dan workshopnya. Sistem juga belum mampu untuk menyimpan history atau memisahkan peserta berdasarkan seminar atau workshop yang sudah selesai. Hal ini berguna untuk mengambil data peserta sebelumnya agar dapat memberi penawaran seminar berikutnya yang relevan atau sejenis dengan seminar yang pernah diikutinya.