BAB IV
HASIL DAN PEMBAHASAN

4.1 Pengembangan Sistem

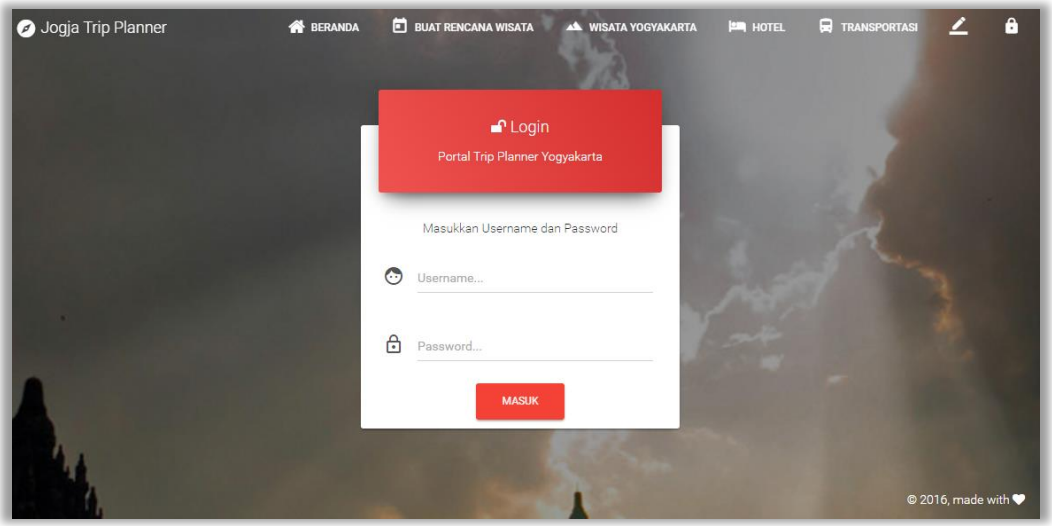
Pengembangan *bussiness logic* dari *website* program *tourism* berbasis *web* menggunakan Bahasa pemrograman PHP dan *framework CodeIgniter*. *CodeIgniter* menggunakan metode MVC di mana terdapat tiga komponen yaitu *folder Models* yang akan menyimpan *file* PHP yang akan digunakan untuk mengakses *database*, *folder* View yang akan menyimpan semua *file* yang berhubungan dengan *interface website*, dan *folder Controllers* yang akan menjadi penghubung antara *views* dan *models*.

4.2 Pembuatan Bussines Logic Rencana Berlibur

Berdasarkan *activity* diagram pada bab 3 maka dibuatlah *business logic* dari proses pembuatan rencana berlibur di mana *user* akan melakukan *login* terlebih dahulu dan kemudian membuat rencana berlibur sampai akhirnya mencetak rencana tersebut.

4.2.1 Halaman Login

Pada gambar 4.1 memperlihatkan halaman *login* yang diakses dari *file login_v.php* dari *folder views*. di halaman *login* terdapat *form* yang digunakan oleh *user* untuk memasukan *username* dan *password*. Setelah *user* memasukan *username* dan *password* kemudian menekan *button* MASUK sehingga sistem dapat melakukan validasi data yang telah di masukan untuk memproses perintah selanjutnya.



Gambar 4.1 Tampilan halaman *Login*

```
<form class="form" method="post" action="<?php echo base_url();?>loginAuth">
    <div class="header header-danger text-center">
        <h4><i class="fa fa-unlock"></i> Login</h4>
        <p>Portal Trip Planner Yogyakarta</p>
    </div>
    <p class="text-divider">Masukkan Username dan Password</p>
<?php
        if($this->session->flashdata('msg'))
          echo '<p><div class="alert alert-info"><button type="button" class="close"
      data-dismiss="alert">×</button>'.$this->session->flashdata('msg').'</div></p>';
        else if($this->session->flashdata('msg_error'))
          echo '<p><div class="alert alert-danger"><button type="button" class="close"
      data-dismiss="alert">×</button>'.$this->session->flashdata('msg_error').'</div></p>';
        else if(validation_errors())
          echo '<p><div class="alert alert-danger"><button type="button" class="close"
      data-dismiss="alert">×</button>'.validation_errors().'</div></p>';
      else if(isset($error_upload))
          echo '<p><div class="alert alert-danger"><button type="button" class="close"
      data-dismiss="alert">×</button>'.$error_upload.'</div></p>';
  ?>
    <div class="content">
        <div class="input-group">
            <span class="input-group-addon">
                <i class="material-icons">face</i>
            </span>
            <input name="username" type="text" class="form-control" placeholder="Username..." required
            value="<?php echo set_value('username');?>">
        </div>
        <div class="input-group">
            <span class="input-group-addon">
                <i class="material-icons">lock_outline</i>
            </span>
            <input name="password" type="password" placeholder="Password..." class="form-control" required >
        </div>
    </div>
    <div class="footer text-center">
        <button type="submit" class="btn btn-danger btn-raised btn-wd">masuk</button>
    </div>
</form>
```

Gambar 4.2 *Coding interface* pada halaman *Login*

Pada gambar 4.2 memperlihatkan *coding* yang digunakan untuk menampilkan *form login*. Pada awal *coding* terdapat *session flashdata* untuk

menampilkan pesan jika terjadi kesalahan saat *login*. ketika *user* selesai mengisi *username* dan *password* dan kemudian menekan *button* Masuk maka *form* di atas akan mengarahkan ke *function loginAuth* yang ada di *file website.php* pada *folder Controllers*.

Pada gambar 4.3 memperlihatkan *coding* fungsi *loginAuth* yang ada di *file website.php* pada *folder Controllers*. Fungsi ini akan mengakses *class auth_member* dan menjalankan fungsi *do_login* pada *folder libraries* untuk mengecek apakah *username* dan *password* yang di masukan *user* ada pada *database*. Jika validasi berhasil maka *coding* mengarahkan ke fungsi akun namun jika gagal akan tampil *message error* dan diarahkan ke fungsi *login*.

```
public function loginAuth()
{
  if(!$_POST)
  {
    redirect('auth');
  }


    $username = $this->input->post('username');
    $password = $this->input->post('password');
    $success = $this->auth_member->do_login($username,$password);
    if($success)
    {
      // masuk ke halaman index user
      redirect('akun');
    }
    else
    {

      $this->session->set_flashdata('msg_error','username atau password tidak cocok');
      redirect('login');
    }

}
```

Gambar 4.3 *Coding function loginAuth* pada *Controllers*

```php
public function akun()
 {
  $this->auth_member->hak_akses_member();
  $id_member = $this->session->userdata('user_id');

   // Some example data
    $data = array(
                  'page_title'     => 'akun',
                  'nama_header'    => 'docs-header',
                  //'logo'         => TRUE,
                  //'aktif_banner' => true,
                  'jenis'          => 'profil-page',
                  'header'         => "background-image: url('".get_media_url().
                  "theme2/img/banner/biru.jpg');
                  background-position: 0px -450px;  padding-top: 100px;",
                  'judul'          => '<i class="fa fa-user"></i> Halaman Akun',
                  'data'         => $this->model->get_('member',['id '=>$id_member]),

        );
    // Load the template from the views directory
    $this->load->view("akun_v", $data);
 }
```

Gambar 4.4 *Coding function* akun pada *folder Controllers*

Pada gambar 4.4 memperlihatkan *coding* fungsi akun yang ada di *file website.php* pada *folder Controller*. Fungsi akun akan membuat *session* untuk *user* dengan menyimpan sementara *id user* ke dalam *session*. fungsi akun juga akan membuat *array* penyimpanan sementara untuk *session id member* dan tampilan yang akan digunakan *user* selama *login*. Dan fungsi akun juga akan mengarahkan *user* ke halaman *akun_v* yang ada pada *folder views*.

```php
public function login()
{
  if($this->session->userdata('user_id')!='')
   {
     redirect('akun');
   }


  // Some example data
   $data = array(
                 'page_title'   => 'Login',
                 'nama_header'  => 'docs-header',
                 'logo'         => TRUE,
                 'aktif_banner' => true,
                 'halamanauth'  => true,
                 'jenis'        => 'signin-page',
                 'header'       => "background-image: url('".get_media_url().
                 "theme2/img/banner/borobudur.jpg');",
                 'judul'     => '<i class="material-icons">terrain</i> Wisata Yogyakarta.'
       );
   // Load the template from the views directory
   $this->load->view("login_v", $data);
}
```

Gambar 4.5 *Coding function login* pada *Controllers*

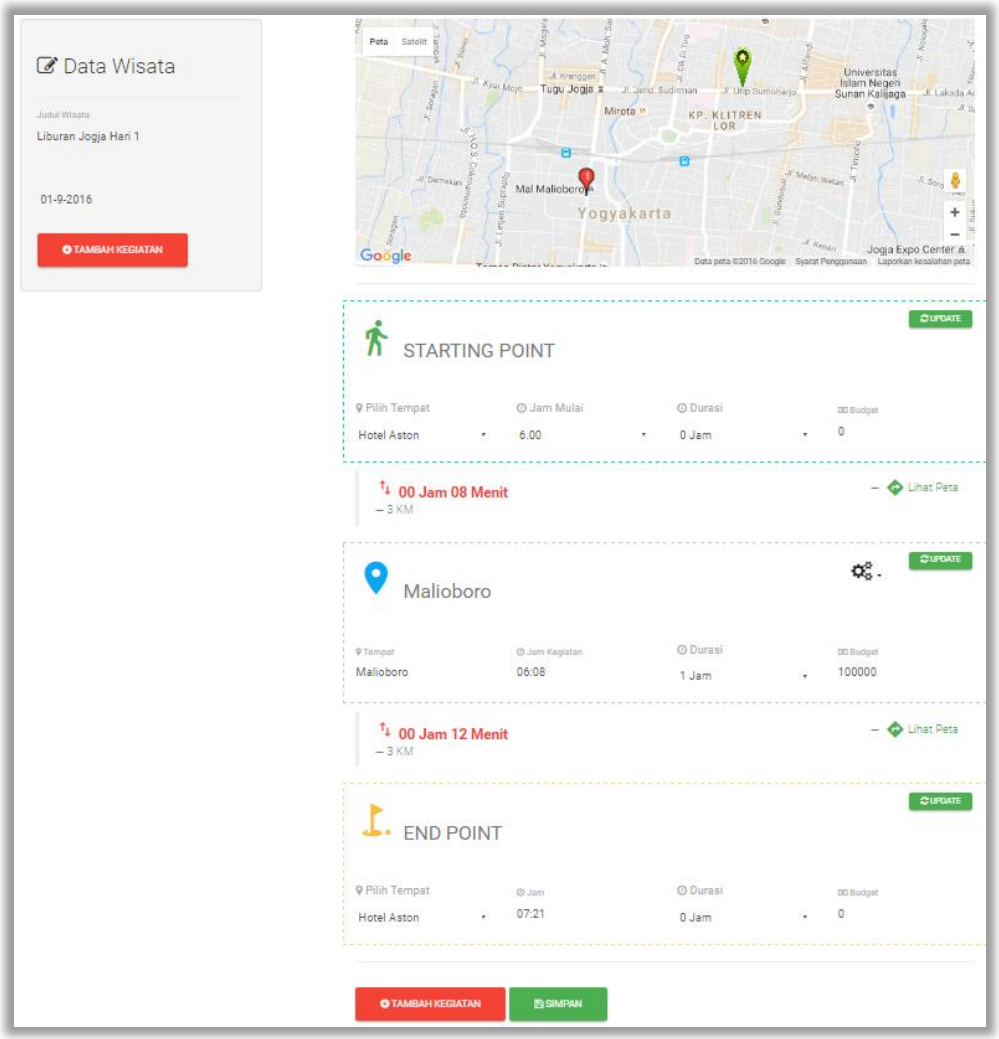
Pada gambar 4.5 memperlihatkan *coding* fungsi *login* yang ada di *file website.php* pada *folder Controller*. Pada awal *coding* di fungsi *login* akan memastikan terlebih dahulu apakah *id user* yang masih tersimpan di *session* jika masih ada maka akan menuju fungsi akun pada *folder controllers*. Kemudian fungsi akun akan membuat *user* kembali ke halaman *login* dan membuat *array* untuk menyimpan tampilan untuk *user* di halaman *login* dan registrasi.

4.2.2 Halaman Buat Rencana

Pada gambar 4.6 memperlihatkan halaman rencana yang diakses dari *file rencana_v.php* dari *folder views*. Halaman rencana adalah halaman yang digunakan oleh *member* untuk membuat rencana berlibur. Ketika ingin membuat rencana berlibur, *member* diharuskan memasukan 3 *point* terlebih dahulu yaitu judul dan tanggal kegiatan, *starting point* atau tempat memulai kegiatan, dan *end point* atau tempat terakhir kegiatan.

Pada gambar 4.6 memperlihatkan *Member* dapat memilih tempat kegiatan dengan menekan *button* TAMBAH KEGIATAN atau menuju halaman wisata untuk memilih tempat wisata yang ingin ditambahkan. Setelah *member* memilih tempat wisata yang diinginkan *member* dapat mengatur jam mulainya kegiatan, durasi di tempat kegiatan, dan *budget* yang diinginkan ketika berada di tempat kegiatan. *member* juga dapat mengubah urutan tempat kegiatan dengan memilih fitur opsi dengan logo pengaturan. Setelah mengolah data kegiatan, *member* diharuskan menekan *button update* di *form* kegiatan yang di ubahnya.

Setelah *member* selesai membuat rencana berlibur *member* dapat menekan *button* SIMPAN untuk menyimpan rencana berlibur yang telah dibuat.



Gambar 4.6 Tampilan halaman rencana

```php
<?php if($this->session->userdata('user_id') != ""){?>
<?php
    if($this->session->flashdata('msg'))
      echo '<p><div class="alert alert-info"><button type="button" class="close"
  data-dismiss="alert">×</button>'.$this->session->flashdata('msg').'</div></p>';
    else if($this->session->flashdata('msg_error'))
      echo '<p><div class="alert alert-danger"><button type="button" class="close"
  data-dismiss="alert">×</button>'.$this->session->flashdata('msg_error').'</div></p>';
    else if(validation_errors())
      echo '<p><div class="alert alert-danger"><button type="button" class="close"
  data-dismiss="alert">×</button>'.validation_errors().'</div></p>';
  else if(isset($error_upload))
      echo '<p><div class="alert alert-danger"><button type="button" class="close"
  data-dismiss="alert">×</button>'.$error_upload.'</div></p>';
?>
```

Gambar 4.7 *Coding session user*

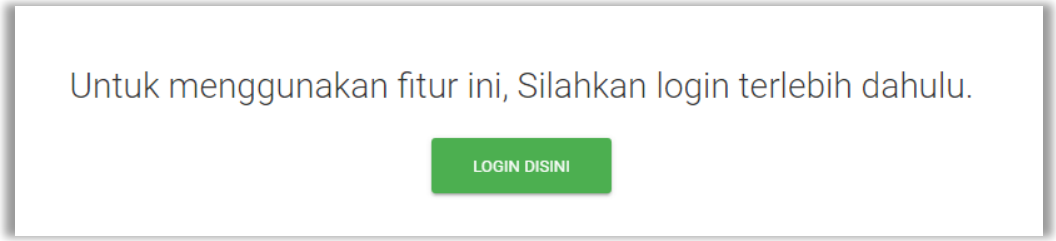
Pada gambar 4.7 adalah *coding session userdata* dan *flashdata*. *Userdata* digunakan untuk mengambil *session* dari *array* yang telah dibuat ketika melakukan

login. Jika *session* ada maka *user* akan dapat menggunakan *fitur* halaman buat rencana, tetapi ketika *session* tidak tersedia maka akan memproses *coding* pada gambar 4.8.

```
<?php } else { echo '<center><h3>Untuk menggunakan fitur ini, Silahkan login terlebih dahulu.
</h3> <a href="'.base_url().'login" class="btn btn-raised btn-success">Login Disini</a></center>';} ?>
```

Gambar 4.8 *Coding Button Login* Disini

Pada gambar 4.8 adalah *coding* untuk memproses pilihan jika tidak terdapat *user* di *session*. Sehingga *user* diminta untuk melakukan *login* terlebih dahulu agar dapat menggunakan halaman buat rencana. Hasil dari *coding* pada gambar 4.8 yaitu tampilan seperti pada gambar 4.9.



Gambar 4.9 Halaman rencana *non member*

Pada gambar 4.9 adalah tampilan ketika *user* membuka halaman buat rencana namun belum melakukan *login*. Untuk itu *user* harus menuju halaman *login* terlebih dengan menekan *button* LOGIN DISINI.

Pada gambar 4.10 adalah *coding session* yang digunakan untuk memastikan apakah sistem menyimpan *starting point* yang belum di simpan ke *database* ada di *session*, Jika iya sistem akan mengambil data *starting point* yang telah dibuat dari *array* dan menampilkan ke *form starting point*. Namun jika kosong makan *form starting point* akan kosong.

```php
if($this->session->userdata('session_start') !="")
{
    $where = [
        'id_member'=>$this->session->userdata('user_id'),
        'status' => 0
    ];
    $id_trip = tampilkan('trip',$where,'id');
    $judul = tampilkan('trip',$where,'judul');
    $tanggal = ind_format_dua(tampilkan('trip',$where,'tanggal'));

    //khusus starting poin
    $jenis = tampilkan('trip_detail',['id_trip'=>$id_trip,'starting_point'=>1],'jenis');
    $id_tempat = tampilkan('trip_detail',['id_trip'=>$id_trip,'starting_point'=>1],'id_tempat');
    $jam = tampilkan('trip_detail',['id_trip'=>$id_trip,'starting_point'=>1],'jam');
    $durasi = tampilkan('trip_detail',['id_trip'=>$id_trip,'starting_point'=>1],'durasi');
    $budget = tampilkan('trip_detail',['id_trip'=>$id_trip,'starting_point'=>1],'budget');

    $id_tempat = tampilkan('trip_detail',['id_trip'=>$id_trip,'starting_point'=>1],'id_tempat');
    $id_detail = tampilkan('trip_detail',['id_trip'=>$id_trip,'starting_point'=>1],'id');
}
else
{
    $judul = "";
    $tanggal = date("d-m-Y");
    $jenis = "";
    $id_tempat = "";
    $jam = "";
    $durasi = "";
    $budget = "";
    $id_trip = "";

}
```

Gambar 4.10 *Coding session starting point*

```php
if($this->session->userdata('session_end') !="")
{
    $where = [
        'id_member'=>$this->session->userdata('user_id'),
        'status' => 0
    ];
    $id_trip2 = tampilkan('trip',$where,'id');
    $judul = tampilkan('trip',$where,'judul');
    $tanggal = ind_format_dua(tampilkan('trip',$where,'tanggal'));

    $jenis2 = tampilkan('trip_detail',['id_trip'=>$id_trip2,'end_point'=>1],'jenis');
    $id_tempat2 = tampilkan('trip_detail',['id_trip'=>$id_trip2,'end_point'=>1],'id_tempat');
    $jam2 = tampilkan('trip_detail',['id_trip'=>$id_trip2,'end_point'=>1],'jam');
    $durasi2 = tampilkan('trip_detail',['id_trip'=>$id_trip2,'end_point'=>1],'durasi');
    $budget2 = tampilkan('trip_detail',['id_trip'=>$id_trip2,'end_point'=>1],'budget');
    $id_trip_detail = tampilkan('trip_detail',['id_trip'=>$id_trip2,'end_point'=>1],'id');
    $posisi_end = tampilkan('trip_detail',['id_trip'=>$id_trip2,'end_point'=>1],'posisi');
}
else
{

    $jenis2 = "";
    $id_tempat2 = "";
    $jam2 = "";
    $durasi2 = "";
    $budget2 = "";
    $id_trip_detail = tampilkan('trip_detail',['id_trip'=>$id_trip,'end_point'=>1],'id');
    $posisi_end = "";

}
```

Gambar 4.11 *Coding session end point*

Pada gambar 4.11 adalah *coding session* yang digunakan untuk memastikan apakah sistem menyimpan *end point* yang belum di simpan ke *database* ada di *session*, Jika iya sistem akan menampilkan *end point* yang telah dibuat dari *array*

dan menampilkan ke *form end point*. Namun jika kosong makan *form end point* akan kosong dan dapat diisi ketika *user* telah selesai mengisi *starting point*.

```
<div class="col-md-3 well">
    <h3><i class="fa fa-edit"></i> Data Wisata</h3>
    <hr>

    <div class="form-group label-floating">
        <label class="control-label">Judul Kegiatan</label>
        <input type="text" class="form-control" required name="judul" id="judul" value="<?php echo $judul;?>">
    </div>

    <div class="form-group">
        <input class="datepicker form-control" type="text" value="<?php echo $tanggal;?>" required
        data-date-format="dd-mm-yyyy" date-minDate="<?php echo date("d-m-Y");?>" id="tanggal">
        <span class="material-input"></span>
    </div>

    <?php if($this->session->userdata('session_start') !="" and $this->session->userdata('session_end') !=""){?>
    <button type="button" class="btn btn-danger btn-raised tombolTambah" data-toggle="modal"
    data-target="#myModal2"><i class="fa fa-plus-circle"></i> Tambah Kegiatan</button>
    <?php } else { ?>
    <p><b>PERHATIAN !!</b> Anda Harus Menentukan 3 point terlebih dahulu yaitu
    <b>Judul dan Tanggal Kegiatan</b>, <b>Starting Point</b>, <b>End Point</b></p>
    <?php } ?>
</div>
```

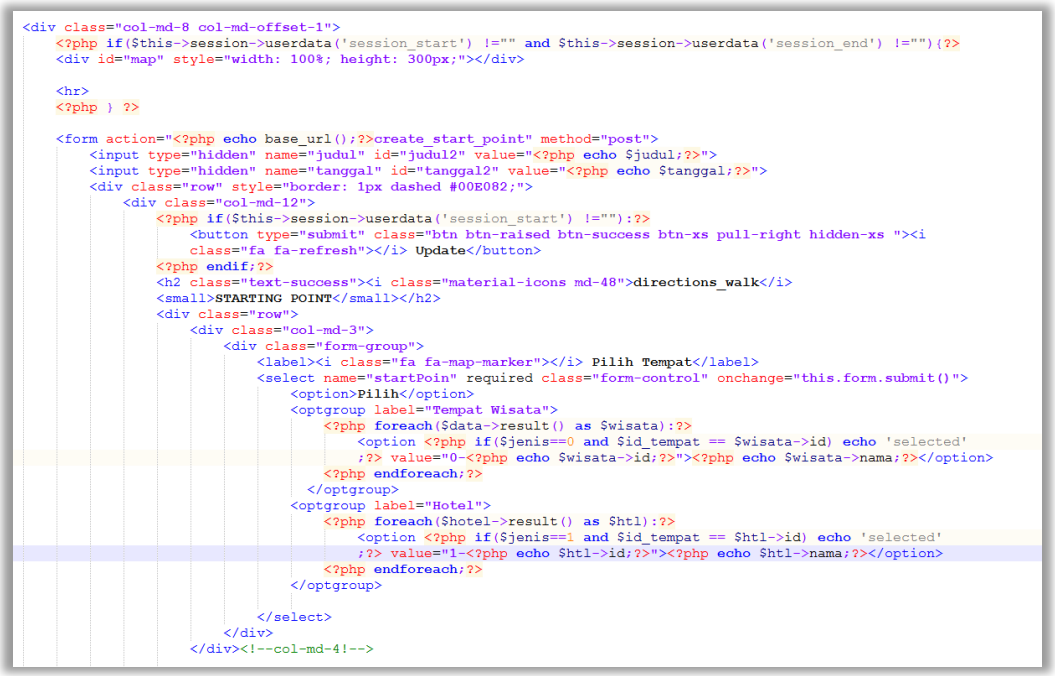
Gambar 4.12 *Coding form* judul dan tanggal

Pada gambar 4.12 adalah *coding* yang digunakan untuk tampilan *form* judul dan tanggal wisata. Pada gambar 4.12 juga terdapat *coding session* yang digunakan untuk memastikan apakah *user* sebelumnya telah mengisi *starting point* dan *end point* jika iya maka akan tampil *button* Tambah Kegiatan, namun jika belum maka akan tampil *text* berupa pesan yang harus dilakukan *user* ketika ingin membuat rencana.

Gambar 4.13 Hasil *coding form* judul dan tanggal

Pada gambar 4.13 adalah tampilan hasil *coding* dari gambar 4.12 di mana *form* ini digunakan untuk mengisi judul dan tanggal kegiatan yang akan dibuat. Dan juga pesan tata cara sebelum membuat rencana

Pada gambar 4.14, 4.15, dan 4.16 adalah *coding* yang digunakan untuk tampilan *form starting point* di halaman rencana. Di *form* inilah *user* menetapkan *starting point*.



```
<div class="col-md-8 col-md-offset-1">
    <?php if($this->session->userdata('session_start') !="" and $this->session->userdata('session_end') !=""){?>
    <div id="map" style="width: 100%; height: 300px;"></div>

    <hr>
    <?php } ?>

    <form action="<?php echo base_url();?>create_start_point" method="post">
        <input type="hidden" name="judul" id="judul2" value="<?php echo $judul;?>">
        <input type="hidden" name="tanggal" id="tanggal2" value="<?php echo $tanggal;?>">
        <div class="row" style="border: 1px dashed #00E082;">
            <div class="col-md-12">
                <?php if($this->session->userdata('session_start') !=""):?>
                    <button type="submit" class="btn btn-raised btn-success btn-xs pull-right hidden-xs "><i
                    class="fa fa-refresh"></i> Update</button>
                <?php endif;?>
                <h2 class="text-success"><i class="material-icons md-48">directions_walk</i>
                <small>STARTING POINT</small></h2>
                <div class="row">
                    <div class="col-md-3">
                        <div class="form-group">
                            <label><i class="fa fa-map-marker"></i> Pilih Tempat</label>
                            <select name="startPoin" required class="form-control" onchange="this.form.submit()">
                                <option>Pilih</option>
                                <optgroup label="Tempat Wisata">
                                    <?php foreach($data->result() as $wisata):?>
                                        <option <?php if($jenis==0 and $id_tempat == $wisata->id) echo 'selected'
                                        ;?> value="0-<?php echo $wisata->id;?>"><?php echo $wisata->nama;?></option>
                                    <?php endforeach;?>
                                </optgroup>
                                <optgroup label="Hotel">
                                    <?php foreach($hotel->result() as $htl):?>
                                        <option <?php if($jenis==1 and $id_tempat == $htl->id) echo 'selected'
                                        ;?> value="1-<?php echo $htl->id;?>"><?php echo $htl->nama;?></option>
                                    <?php endforeach;?>
                                </optgroup>

                            </select>
                        </div>
                    </div><!--col-md-4!-->
```

Gambar 4.14 *Coding form starting point* bagian 1

Pada gambar 4.14 terdapat *session* di awal *coding* untuk melihat apakah ada *session starting point* dan *end point* yang tersimpan di *array*, jika iya maka akan menampilkan *map* dengan ukuran yang ditentukan. *Coding form* pada gambar 4.14 digunakan untuk menampilkan *form starting point*. Dalam *form* terdapat *coding session* untuk melihat apakah ada *starting point* yang tersimpan di *session* yaitu *session_start*. Jika iya *button update* akan tampil, Namun jika tidak terdapat data *starting point* di *array* maka tampilan *form starting point* akan kosong dan *button*

update tidak muncul. Pada gambar 4.14 terdapat *coding* yang akan *submit* secara otomatis melalui fungsi *create_start_point* ketika *member* selesai memilih *starting point*.

```
<div class="col-md-3">
    <div class="form-group">
        <label><i class="fa fa-clock-o"></i> Jam Mulai</label>
        <select name="jamStart" required class="form-control">

            <?php
            for($jams=1;$jams<=24;$jams++):
            ?>
            <option <?php if($jams=="6") echo 'selected';?> <?php if($jam == $jams) echo 'selected'
            ;?> value="<?php echo $jams;?>"><?php echo $jams;?>.00</option>
            <?php
            endfor;
            ?>

        </select>
    </div>
</div><!--col-md-4!-->

<div class="col-md-3">
    <div class="form-group">
        <label><i class="fa fa-clock-o"></i> Durasi</label>
        <select name="durasiStart" required class="form-control">
            <?php
            for($jams=0;$jams<=10;$jams++):
            ?>
            <option value="<?php echo $jams;?>" <?php if($durasi == $jams) echo 'selected';?>><?php
            echo $jams;?> Jam</option>
            <?php
            endfor;
            ?>

        </select>
    </div>
</div><!--col-md-4!-->
```

Gambar 4.15 *Coding form starting point* bagian 2

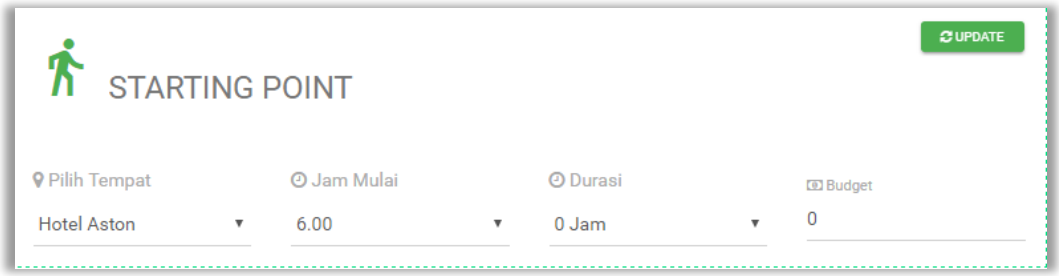
Pada gambar 4.15 terdapat coding untuk tampilan jam mulainya kegiatan dan durasi. jam dimulainya kegiatan di atur *default* ke jam 6. Namun *member* dapat mengubah jam sesuai dengan kebutuhannya. Untuk durasi diatur *default* ke angka nol, *member* dapat mengubah durasi sesuai dengan kebutuhannya juga namun hanya dibatasi sampai 10 jam tiap kegiatannya.

Pada gambar 4.16 terdapat coding untuk tampilan durasi. Untuk tampilan durasi langsung default ke nol, namun jika terdapat data di session maka akan langsung mengakses variable budget di session_start. Pada akhir coding terdapat coding untuk menampilkan button update ketika starting point telah ditentukan, Coding akan mengakses class create_start_point di file website.php pada folder

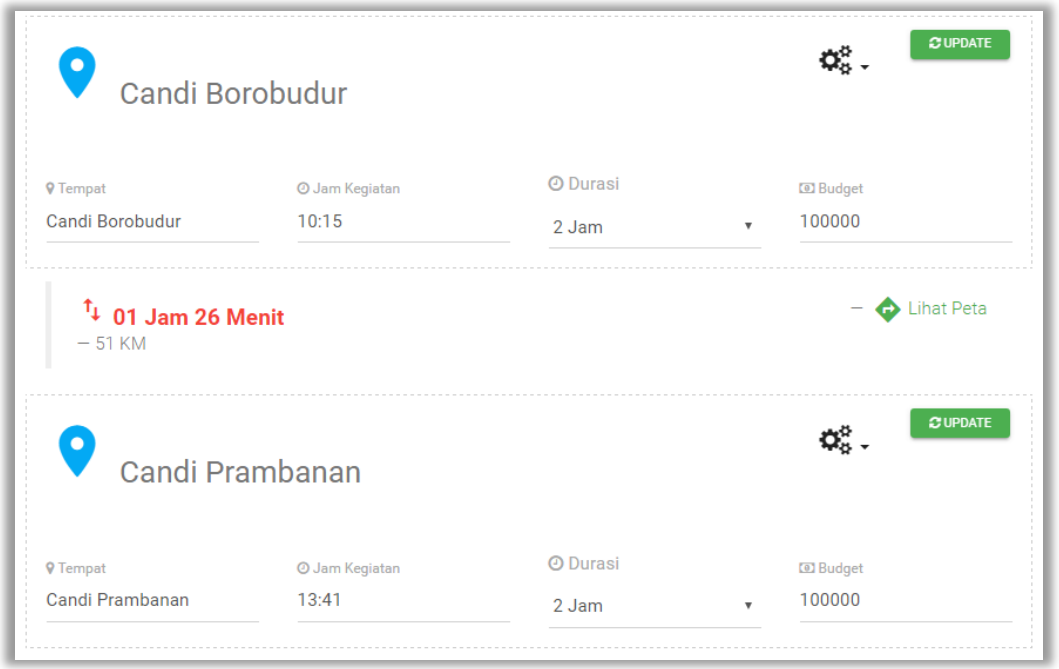
Controllers. Untuk hasil dari coding pada gambar 4.14, gambar 4.15, dan gambar 4.16 dapat dilihat pada gambar 4.17.

```
                    <div class="col-md-3">
                        <br>
                        <div class="form-group label-floating">
                            <label class="control-label"><i class="fa fa-money"></i> Budget</label>
                            <input type="text" class="form-control" name="budgetStart" value="<?php echo $budget;?>">
                        </div>
                    </div><!--col-md-4!-->

                    <?php if($this->session->userdata('session_start') !=""):?>
                    <button type="submit" class="btn btn-raised btn-success  pull-right visible-xs "><i class="fa
                    fa-refresh"></i> update</button>
                    <?php endif;?>

                </div>
            </div>
        </div>
    </form>
```

Gambar 4.16 *Coding form starting point* bagian 3



Gambar 4.17 Hasil *coding form starting point*



Gambar 4.18 Tampilan jarak dan waktu halaman rencana

Pada gambar 4.18 menampilkan informasi jarak dan waktu yang dibutuhkan oleh *user* ketika bergerak dari tempat kegiatan pertama ke tempat kegiatan kedua. Tampilan ini akan otomatis muncul ketika *user* telah memilih *starting point*, *end point*, dan tempat kegiatan. *User* juga dapat melihat rute yang dapat digunakan untuk menuju tempat berikutnya dengan menekan *button* Lihat Peta yang nantinya sistem akan menampilkan halaman peta yang ada pada *folder views*.

```
<?php if($this->session->userdata('session_start') !="" and
$this->session->userdata('session_end') !=""){?>
<script type="text/javascript">
</script>
<blockquote style="margin-top: 10px;">
 <p>
    <small class="pull-right">
     <a class="text-success window-opener"
        href="javascript: void(0)"
        onclick="popup('<?php echo base_url();?>peta/<?php echo $id_detail;?>')">
        <i class="material-icons">directions</i> Lihat Peta
    </a> </small>
    <font class="text-danger"><i class="material-icons">swap_vert</i>
        <b class="durasi">
            <?php
            $ab = get_dari('trip_detail',['id_trip'=>$id_trip,'posisi'=>0],'id');
            $k1 = $ab->row();

            if($k1->jenis == 0)
            {
                $from = 'tempat_wisata';
            }
            else
            {
                $from = 'hotel';
            }
            $cd = get_dari('trip_detail',['id_trip'=>$id_trip,'posisi'=>1],'id');
            $k2 = $cd->row();

            if($k2->jenis == 0)
            {
                $from2 = 'tempat_wisata';
            }
            else
            {
                $from2 = 'hotel';
            }
            $koordinat1 = tampilkan($from,['id'=>$k1->id_tempat],'koordinat');
            $koordinat2 = tampilkan($from2,['id'=>$k2->id_tempat],'koordinat');
            echo get_jarak($koordinat1,$koordinat2,true);
            ?>
        </b> <small><?php echo get_jarak($koordinat1,$koordinat2,'',true);?> KM</small>
    </font>
 </p>
</blockquote>
<?php } ?>
```

Gambar 4.19 *Coding* tampilan jarak dan waktu

Pada gambar 4.19 terdapat *coding* yang digunakan untuk menghitung jarak dan waktu dari tempat kegiatan pertama ke tempat kegiatan kedua. Terdapat *session* untuk memastikan *user* sudah memilih *starting point* dan *end point* sehingga *coding*

di atas akan otomatis memproses posisi koordinat dari kegiatan pertama dengan posisi koordinat yang berikutnya. Untuk mengambil koordinat membutuhkan data posisi kegiatan dari tabel *trip_detail* yang ada di *database*. Kemudian sistem akan menghitung secara otomatis jarak dan waktu menggunakan fungsi *get_jarak* di *file apphelper.php* pada *folder helpers*. Untuk tampilan hasil dari *coding* gambar 4.19 dapat dilihat pada gambar 4.18.

```
<?php if($this->session->userdata('session_start') !=""):?>
<form action="<?php echo base_url();?>create_end_point" method="post">
    <div class="row" style="border: 1px dashed #FFDB85;">
        <div class="col-md-12">
            <?php if($this->session->userdata('session_end') !=""):?>
                <button type="submit" class="btn btn-raised btn-success btn-xs
                pull-right hidden-xs "><i class="fa fa-refresh"></i> Update</button>
            <?php endif;?>
            <h2 class="text-warning"><i class="material-icons md-48">golf_course</i>
            <small>END POINT</small></h2>
            <div class="row">
                <div class="col-md-3">
                    <div class="form-group">
                        <label><i class="fa fa-map-marker"></i> Pilih Tempat</label>
                        <select name="startPoin" required class="form-control"
                        onchange="this.form.submit()">
                            <option>Pilih</option>
                            <optgroup label="Tempat Wisata">
                                <?php foreach($data->result() as $wisata):?>
                                    <option <?php if($jenis2==0 and $id_tempat2 ==
                                    $wisata->id) echo 'selected';?> value="0-<?php echo
                                    $wisata->id;?>"><?php echo $wisata->nama;?></option>
                                <?php endforeach;?>
                            </optgroup>
                            <optgroup label="Hotel">
                                <?php foreach($hotel->result() as $htl):?>
                                    <option <?php if($jenis2==1 and $id_tempat2 ==
                                    $htl->id) echo 'selected';?> value="1-<?php echo
                                    $htl->id;?>"><?php echo $htl->nama;?></option>
                                <?php endforeach;?>
                            </optgroup>
                        </select>
                    </div>
                </div><!--col-md-4!-->
```

Gambar 4.20 *Coding form end point* bagian 1

Pada gambar 4.20 terdapat *coding* yang digunakan untuk memeriksa apakah *session starting point* telah dibuat. Jika iya, maka akan tampil *form end point*. *Form end point* menggunakan fungsi *create_end_point* di *file website.php* pada *folder controllers* ketika *member* selesai menentukan *end point* dari kegiatan.

Pada gambar 4.20 terdapat *coding session* untuk memeriksa apakah ada *end point* yang tersimpan di *session* pada *session_end*. Jika iya, maka akan menampilkan *button update* secara otomatis dan mengambil data dari *session*. Namun jika tidak maka akan tampil *form end point* yang kosong yang dapat diisi oleh *user*.

```
<div class="col-md-3">
<?php
error_reporting(0);
if($posisi_end == 1 )
{
    $jamnya = hitung_jam($id_trip_detail);
}
else
{
    if($before==0)
    {
        $dtk = $dtk;
    }
    else
    {
        $ambil_posisi_sebelumnya0  = get_dari('trip_detail',['id_trip'=>
        $id_trip,'posisi'=> $posisi_end-1],'id');
        $cc = $ambil_posisi_sebelumnya0->row();
        $drs = $cc->durasi*3600;
        $dtk = $before+$drs;
    }
```

Gambar 4.21 *Coding form end point* bagian 2

Pada gambar 4.21 dan 4.22 terdapat *coding* yang digunakan untuk menghitung jam kegiatan berdasarkan jam kegiatan yang dipilih *user* pertama kali saat *starting point* dan juga berdasarkan durasi tiap tempat kegiatan. *Coding* Pada gambar 4.21 dan 4.22 mengambil posisi dari tempat kegiatan di tabel *trip_detail* pada *database*, tujuannya untuk membedakan antara tempat wisata dan hotel karena wisata dan hotel memiliki tabel masing-masing tabel di *database*. pada gambar 4.22 terdapat *variable* yang mengakses fungsi tampilkan, *get_jarak*, dan *konversi_jam* yang mengakses *file apphelper.php* pada *folder controllers*. *Coding* tersebut akan menghitung jarak dan waktu dari *starting point* dengan *end point*.

```
    $ambil_posisi_sebelumnya  = get_dari('trip_detail',['id_trip'=>$id_trip,
    'posisi'=> $posisi_end-1],'id');
    $aa = $ambil_posisi_sebelumnya->row();
    if($aa->jenis == 0)
    {
        $dr = 'tempat_wisata';
    }
    else
    {
        $dr = 'hotel';
    }
    if($jenis2==0)
    {
        $dr2= 'tempat_wisata';
    }
    else
    {
        $dr2 = 'hotel';
    }
    $koordinat1 = tampilkan($dr,['id'=>$aa->id_tempat],'koordinat');
    $koordinat2 = tampilkan($dr2,['id'=>$id_tempat2],'koordinat');
    $jarak = get_jarak($koordinat1,$koordinat2,'','','',true);
    $before =+ $dtk+$jarak;
    $jamnya= konversi_jam($before);
}

?>
```

Gambar 4.22 *Coding form end point* bagian 3

Pada gambar 4.23 terdapat *coding* untuk tampilan jam mulainya kegiatan dan durasi. *member* tidak dapat mengubah jam *end point* karena jam untuk *end point* adalah hasil dari jam kegiatan sebelumnya di jumlahkan dengan jarak waktu yang diperlukan untuk sampai ke *end point*. Untuk durasi diatur *default* ke angka nol, *member* dapat mengubah durasi sesuai dengan kebutuhannya namun hanya dibatasi sampai 10 jam tiap kegiatannya.

Pada gambar 4.23 terdapat *coding* untuk menampilkan *button update* ketika *starting point* telah ditentukan, *Coding* akan mengakses *class create_end_point* di *file website.php* pada *folder Controllers.*

```
                    <br>
                    <div class="form-group label-floating">
                        <label class="control-label"><i class="fa
                        fa-clock-o"></i> Jam </label>
                        <input type="text" readonly class="form-control"
                        name="budgetStart" value="<?php echo $jamnya;?>">
                    </div>
                </div><!--col-md-4!-->
                <div class="col-md-3">
                    <div class="form-group">
                        <label><i class="fa fa-clock-o"></i> Durasi</label>
                        <select name="durasiStart" required class="form-control">
                            <?php
                            for($jams=0;$jams<=10;$jams++):
                            ?>
                            <option value="<?php echo $jams;?>" <?php if($durasi2
                            == $jams) echo 'selected';?>><?php echo $jams;?> Jam</option>
                            <?php
                            endfor;
                            ?>
                        </select>
                    </div>
                </div><!--col-md-4!-->
                <div class="col-md-3">
                    <br>
                    <div class="form-group label-floating">
                        <label class="control-label"><i
                        class="fa fa-money"></i> Budget</label>
                        <input type="text" class="form-control"
                        name="budgetStart" value="<?php echo $budget2;?>">
                    </div>
                </div><!--col-md-4!-->
                <?php if($this->session->userdata('session_end') !=""):?>
                <button type="submit" class="btn btn-raised btn-success
                pull-right visible-xs "><i class="fa fa-refresh"></i> update</button>
                <?php endif;?>
            </div>
        </div>
    </div>
</form>
<?php endif;?>
```

Gambar 4.23 *Coding form end point* bagian 4

```
<?php if($this->session->userdata('session_start') !="" and
$this->session->userdata('session_end') !=""):?>
    <button type="button" class="btn btn-danger btn-raised tombolTambah"
    data-toggle="modal" data-target="#myModal2"><i class="fa fa-plus-circle">
    </i> Tambah Kegiatan</button>
    <a class="btn btn-success btn-raised tombolTambah"
    href="<?php echo base_url();?>save_trip/<?php echo $id_trip;?>"><i
    class="fa fa-save"></i> Simpan</a>
<?php endif;?>
```

Gambar 4.24 *Coding button* tambah kegiatan dan simpan

Pada gambar 4.24 terdapat *coding session* yang digunakan untuk memeriksa apakah ada *starting point* dan *end point* yang tersimpan di *array*. Jika iya, maka akan tampil *button* Tambah Kegiatan dan *button* Simpan. Ketika *button* tambah kegiatan di tekan maka sistem akan menampilkan dialog *box* dengan *id* myModal2

yang nantinya dapat digunakan untuk memilih tempat wisata yang ingin dikunjungi.

```
<div class="modal fade" id="myModal2" tabindex="0" role="dialog" aria-labelledby="myModalLabel"
data-backdrop="false"  aria-hidden="true" style="z-index: 9999;">
  <div class="modal-dialog modal-lg" style="border: 1px solid #ccc;">
    <div class="modal-content">

      <div class="modal-header" style="">
        <button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>
        <h4 class="modal-title" id="myModalLabel">Pilih Tempat Wisata</h4>
      </div>
      <div class="modal-body">
       <!-- Nav tabs -->
      <ul class="nav nav-tabs" role="tablist">
        <li role="presentation" class="active"><a href="#tempatwisata" aria-controls="home"
        role="tab" data-toggle="tab">Tempat Wisata</a></li>
        <li role="presentation" class=""><a href="#penginapan" aria-controls="home"
        role="tab" data-toggle="tab">Penginapan</a></li>
      </ul>
       <div class="tab-content">
            <div role="tabpanel" class="tab-pane active" id="tempatwisata">
                <div class="row">
                    <div class="card-deck">
                        <?php foreach($data->result() as $row):?>
                            <div class="col-md-4">
                                <div class="card" style="margin-top: 10px;">
                                  <img src="<?php echo get_media_url();?>uploads/tempat_wisata/
                                  <?php echo tampilkan('gambar_wisata',['id_tempat_wisata'=>
                                  $row->id],'gambar');?>" class="card-img-top img-responsive"
                                  data-src="assets/img/banner/kuda.jpg"  alt="<?php echo $row->nama;
                                  ?>" style="height: 200px; width: 100%; display: block;">
                                  <div class="card-block">
                                    <h4 class="card-title"><?php echo $row->nama;?></h4>
                                    <p class="card-text">
                                        <small class="text-muted"><i class="fa fa-map-marker"></i>
                                        <?php echo tampilkan('daerah',['id'=>$row->id_daerah],'nama')
                                        ;?></small>

                                        <span class="label label-danger pull-right"><?php echo tampilkan
                                        ('jenis_wisata',['id'=>$row->id_jenis_wisata],'nama');?></span>
                                    </p>
                                     <a href="#" class="btn btn-danger "><i class="fa fa-heart heart">
                                    </i>  <?php echo likes($row->id);?> Likes</small></a>
                                    <a href="<?php echo base_url();?>tambah_kegiatan/<?php echo $row->id;
                                    ?>/0/<?php echo $id_trip;?>" class="btn btn-success btn-raised pull-right">
                                    <i class="fa fa-plus-circle"></i> Tambah Kegiatan</a>
                                  </div>
```

Gambar 4.25 *Coding dialog box* tempat wisata dan penginapan bagian 1

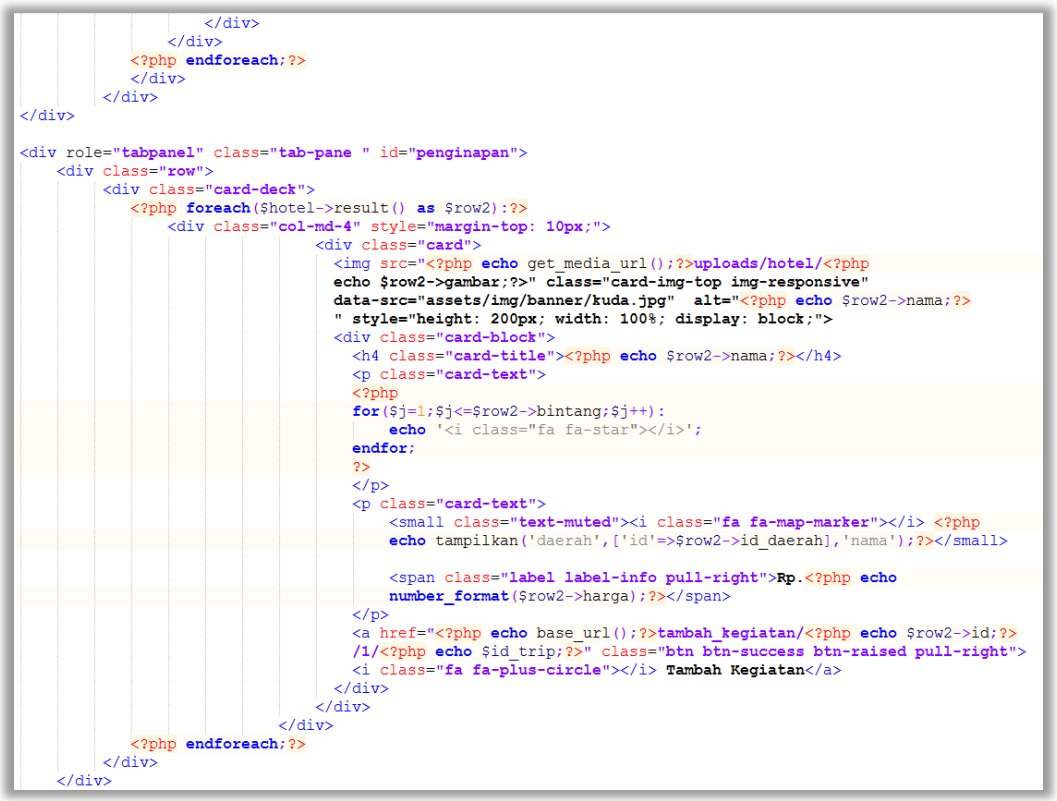
Pada gambar 4.25 terdapat *coding* yang digunakan untuk membuat *dialog box*. *Dialog box* digunakan oleh *user* untuk memilih tempat kegiatan. dalam *dialog box* terbagi menjadi 2 navigasi tab yaitu Tempat Wisata dan Penginapan.

Pada gambar 4.25 terdapat *coding* di tab Tempat wisata mengambil gambar dari *folder uploads/tempat_wisata/* dan data dari *database* dengan tabel *gambar_wisata* untuk gambar dan nama dari wisata, tabel daerah untuk posisi kota tempat wisata, tabel *jenis_wisata* untuk jenis wisata. Semua tabel yang diakses menggunakan fungsi tampilkan di *apphelper.php* pada *folder helpers*.

Pada bagian terakhir gambar 4.25 terdapat *coding* untuk menampilkan jumlah *Likes* dari tempat wisata dan *coding* untuk *button* Tambah Kegiatan untuk menambahkan kegiatan ke halaman rencana.

Pada gambar 4.26 terdapat *coding* di tab Penginapan mengambil gambar dari *uploads/hotel/* dan data dari *database* dengan tabel daerah untuk posisi kota dari hotel atau penginapan yang diakses menggunakan fungsi tampilkan di *apphelper.php* pada *folder helpers*.

Pada bagian terakhir gambar 4.26 terdapat *coding* untuk *button* Tambah Kegiatan untuk menambahkan Hotel ke halaman rencana.



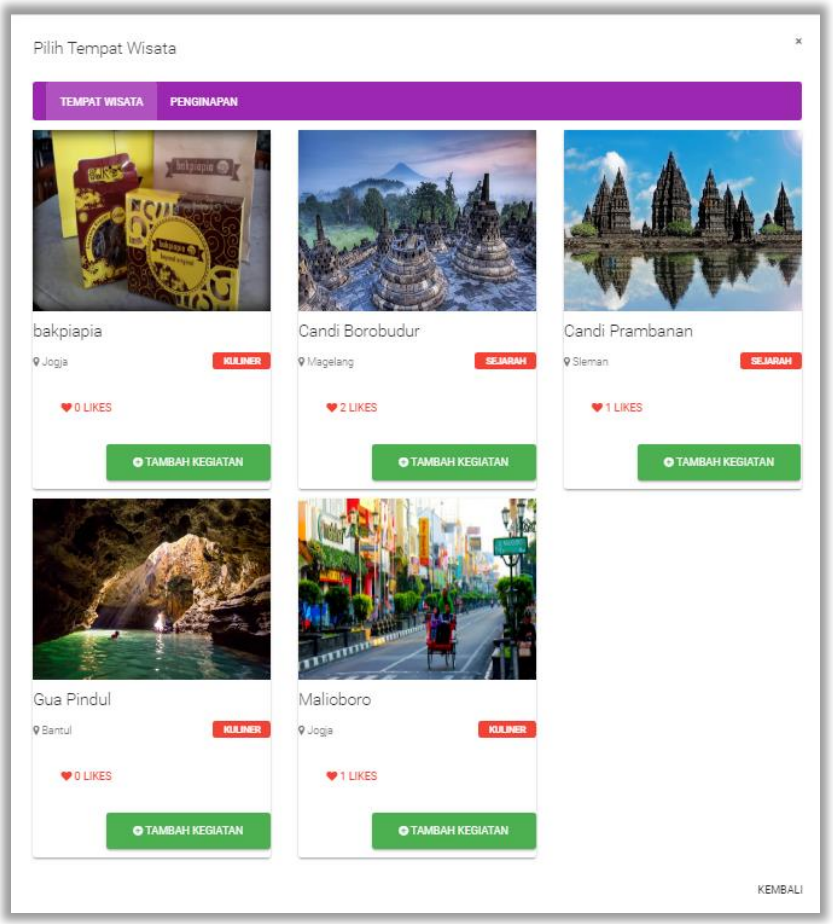
```
                    </div>
                </div>
            <?php endforeach;?>
            </div>
        </div>
</div>

<div role="tabpanel" class="tab-pane " id="penginapan">
    <div class="row">
        <div class="card-deck">
            <?php foreach($hotel->result() as $row2):?>
                <div class="col-md-4" style="margin-top: 10px;">
                    <div class="card">
                      <img src="<?php echo get_media_url();?>uploads/hotel/<?php
                      echo $row2->gambar;?>" class="card-img-top img-responsive"
                      data-src="assets/img/banner/kuda.jpg"  alt="<?php echo $row2->nama;?>
                      " style="height: 200px; width: 100%; display: block;">
                      <div class="card-block">
                        <h4 class="card-title"><?php echo $row2->nama;?></h4>
                        <p class="card-text">
                        <?php
                        for($j=1;$j<=$row2->bintang;$j++):
                            echo '<i class="fa fa-star"></i>';
                        endfor;
                        ?>
                        </p>
                        <p class="card-text">
                            <small class="text-muted"><i class="fa fa-map-marker"></i> <?php
                            echo tampilkan('daerah',['id'=>$row2->id_daerah],'nama');?></small>

                            <span class="label label-info pull-right">Rp.<?php echo
                            number_format($row2->harga);?></span>
                        </p>
                        <a href="<?php echo base_url();?>tambah_kegiatan/<?php echo $row2->id;?>
                        /1/<?php echo $id_trip;?>" class="btn btn-success btn-raised pull-right">
                        <i class="fa fa-plus-circle"></i> Tambah Kegiatan</a>
                      </div>
                    </div>
                </div>
            <?php endforeach;?>
        </div>
    </div>
```

Gambar 4.26 *Coding dialog box* tempat wisata dan penginapan bagian 2

```
            </div>
          </div>

        </div>
        <div class="modal-footer">
          <button type="button" class="btn btn-default" data-dismiss="modal">Kembali</button>
        </div>
      </div>
    </div>
</div>
```

Gambar 4.27 *Coding dialog box* tempat wisata dan penginapan bagian 3

Pada gambar 4.27 terdapat *coding button* kembali untuk menutup *dialog box* dan kembali ke halaman rencana. Hasil dari *coding* gambar 4.25, gambar 4.26, dan gambar 4.27 dapat dilihat di pada gambar 4.28.



Gambar 4.28 Hasil *coding dialog box* tempat wisata dan penginapan

Pada gambar 4.28 memperlihatkan *dialog box* yang akan tampil ketika *user* selesai menekan tombol Tambah Kegiatan di halaman buat rencana. Di dialog *box*

ini *user* terdapat 2 navigasi yaitu Tempat Wisata dan Penginapan sehingga *user* dapat memilih tempat kegiatan yang akan ditambahkan ke rencana berlibur yang dibuat.

```
<form action="<?php echo base_url();?>update_kegiatan" method="post">
<input type="hidden" name="id" value="<?php echo $list->id;?>">
<div class="row" style="border: 1px dashed #ccc;">
    <div class="col-md-12">
        <button type="submit" class="btn btn-raised btn-success btn-xs pull-right hidden-xs ">
        <i class="fa fa-refresh"></i> Update</button>
        <div class="pull-right dropdown">
            <a href="#" class="btn btn-simple dropdown-toggle" data-toggle="dropdown" aria-expanded="false">
                <i class="fa fa-cogs fa-fw fa-2x"></i>
                <b class="caret"></b>
            <div class="ripple-container"></div></a>
            <ul class="dropdown-menu">
                <?php if($list->posisi != 1):?>
                    <li><a href="<?php echo base_url();?>up_posisi/<?php echo $list->id;?>">Pindah Posisi
                    <i class="fa fa-chevron-up fa-fw"></i> </a></li>
                <?php endif;?>
                <?php if($list->posisi != $total_list):?>
                    <li><a href="<?php echo base_url();?>down_posisi/<?php echo $list->id;?>">Pindah Posisi
                    <i class="fa fa-chevron-down fa-fw"></i></a></li>
                <?php endif;?>
                <li class="divider"></li>
                <li><a href="<?php echo base_url();?>hapus_kegiatan/<?php echo $list->id;?>">Hapus
                <i class="fa fa-trash"></i></a></li>
            </ul>
        </div>
        <h2 class="text-info"><i class="material-icons md-48">place</i> <small><?php echo $nama_tempat;?>
        </small></h2>
        <div class="row">
            <div class="col-md-3">
                <br>
                <div class="form-group label-floating">
                    <label class="control-label"><i class="fa fa-map-marker"></i> Tempat </label>
                    <input type="text" readonly class="form-control" name="budgetStart" value="
                    <?php echo $nama_tempat;?>">
                </div>
            </div><!--col-md-4!-->
            <div class="col-md-3">
            <?php
                if($list->posisi == 1)
                {
                    $jamnya = hitung_jam($list->id);
                    $dtk = hitung_detik($list->id)+($list->durasi * 3600);
                }
```

Gambar 4.29 *Coding form* kegiatan bagian 1

Pada gambar 4.29, gambar 4.30, dan gambar 4.31 adalah *coding* untuk *form* kegiatan. *Form* ini akan tampil ketika *member* selesai menambahkan kegiatan. di *form* ini *member* dapat mengubah posisi urutan kegiatan dan menghapus *form* kegiatan. data dalam *form* semuanya akan mengambil data berdasarkan tempat kegiatan yang dipilih namun *member* tetap dapat mengubah data sesuai dengan kebutuhan. Pada gambar 4.30 terdapat *coding* untuk menghitung jarak dan waktu antar kegiatan.

```
    else
    {
        if($before==0)
        {
            $dtk = $dtk;
        }
        else
        {
            $ambil_posisi_sebelumnya0  = get_dari('trip_detail',['id_trip'=>$list->id_trip,
            'posisi'=> $list->posisi-1],'id');
            $cc = $ambil_posisi_sebelumnya0->row();
            //durasi sebelumnya
            $drs = $cc->durasi*3600;
            $dtk = $before+$drs;
        }
        $ambil_posisi_sebelumnya  = get_dari('trip_detail',['id_trip'=>$list->id_trip,
        'posisi'=> $list->posisi-1],'id');
        $aa = $ambil_posisi_sebelumnya->row();
        if($aa->jenis == 0)
        {
            $dr = 'tempat_wisata';
        }
        else
        {
            $dr = 'hotel';
        }
        $koordinat1 = tampilkan($dr,['id'=>$aa->id_tempat],'koordinat');
        $koordinat2 = tampilkan($dari,['id'=>$list->id_tempat],'koordinat');
        $jarak = get_jarak($koordinat1,$koordinat2,'','','',true);
        $before =+ $dtk+$jarak;
        $jamnya= konversi_jam($before);
    }
    ?>
    <br>
    <div class="form-group label-floating">
        <label class="control-label"><i class="fa fa-clock-o"></i> Jam Kegiatan</label>
        <input type="text" class="form-control" readonly value="<?php echo $jamnya;?>" >
    </div>
</div><!--col-md-4!-->
```

Gambar 4.30 *Coding form* kegiatan bagian 2

```
            <div class="col-md-3">
                <div class="form-group">
                    <label><i class="fa fa-clock-o"></i> Durasi</label>
                    <select name="durasi" required class="form-control">
                        <?php
                        for($jams=0;$jams<=10;$jams++):
                        ?>
                        <option value="<?php echo $jams;?>" <?php if($list->durasi == $jams) echo
                        'selected';?>><?php echo $jams;?> Jam</option>
                        <?php
                        endfor;
                        ?>
                    </select>
                </div>
            </div><!--col-md-4!-->
            <div class="col-md-3">
                <br>
                <div class="form-group label-floating">
                    <label class="control-label"><i class="fa fa-money"></i> Budget</label>
                    <input type="text" class="form-control" name="budget"  value="<?php echo
                    $list->budget;?>">
                </div>
            </div><!--col-md-4!-->
        </div>
        <button type="submit" class="btn btn-raised btn-success  pull-right visible-xs ">
        <i class="fa fa-refresh"></i> update</button>
    </div>
</div>
</form>
```

Gambar 4.31 *Coding form* kegiatan bagian 3

```php
<?php if($this->session->userdata('session_start') !=""
and $this->session->userdata('session_end') !=""){?>
<script type="text/javascript">
    var locations = [
    <?php
    //membedakan antara tempat wisata dan hotel (hanya untuk keterangan point)
    $map_rencana = get_dari('trip_detail',['id_trip'=>$id_trip],'posisi','asc');
    foreach($map_rencana->result() as $j => $row_map):
        if($row_map->jenis == 0)
        {
            $map_from = 'tempat_wisata';
        }
        else
        {
            $map_from = 'hotel';
        }

        $map_koordinat = tampilkan($map_from,['id'=>$row_map->id_tempat],'koordinat');
        $map_nama = tampilkan($map_from,['id'=>$row_map->id_tempat],'nama');


    ?>

      ['<?php echo $map_nama;?>', <?php echo $map_koordinat;?>],

    <?php endforeach;?>

    ];
    var map = new google.maps.Map(document.getElementById('map'), {
      zoom: 10,
      center: new google.maps.LatLng(-39.92, 151.25),
      mapTypeId: google.maps.MapTypeId.ROADMAP
    });
    var infowindow = new google.maps.InfoWindow();
    var marker, i;
    var markers = new Array();
    for (i = 0; i < locations.length; i++) {
```

Gambar 4.32 *Coding map* bagian 1

Pada gambar 4.32, 4.33, dan 4.34 terdapat *coding* untuk memproses *map* yang akan di tampilkan. Pada bagian awal gambar 4.32 terdapat *coding session* untuk memeriksa apakah *starting point* dan *end point* terdapat di *array*. Jika iya, maka sistem akan memproses *coding* berikutnya. *Coding* selanjutnya pada gambar 4.32 yaitu mengambil posisi dari tempat kegiatan di tabel *trip_detail* pada *database*, tujuannya untuk membedakan antara tempat wisata dan hotel karena wisata dan hotel memiliki tabel masing-masing di *database*.

Pada gambar 4.33 terdapat *coding* untuk memproses *map* dengan gambar start.png sebagai *icon starting point* dan gambar finish.png sebagai *icon end point*

di *map* dan juga *icon chart* dengan nomor urutan kegiatan. Sehingga nantinya *user* akan mudah melihat posisi dari tempat-tempat kegiatan yang telah dipilih.

```
if(i == 0)
{
  var icon_image = new google.maps.MarkerImage(
      "<?php echo get_media_url();?>theme2/img/start.png",
      null, /* size is determined at runtime */
      null, /* origin is 0,0 */
      null, /* anchor is bottom center of the scaled image */
      new google.maps.Size(42, 50)
  );
  //icon_image = '<?php echo get_media_url();?>theme2/img/start.png';
}
else if(i == <?php echo $j;?>)
{
  var icon_image = new google.maps.MarkerImage(
      "<?php echo get_media_url();?>theme2/img/finish.png",
      null, /* size is determined at runtime */
      null, /* origin is 0,0 */
      null, /* anchor is bottom center of the scaled image */
      new google.maps.Size(42, 50)
  );

}
else
{

  icon_image = 'http://chart.apis.google.com/chart?chst=d_map_pin_letter&chld='+i+'|f44336|FFFFFF';
}
marker = new google.maps.Marker({
  position: new google.maps.LatLng(locations[i][1], locations[i][2]),
  map: map,
  icon: icon_image
});
markers.push(marker);
```

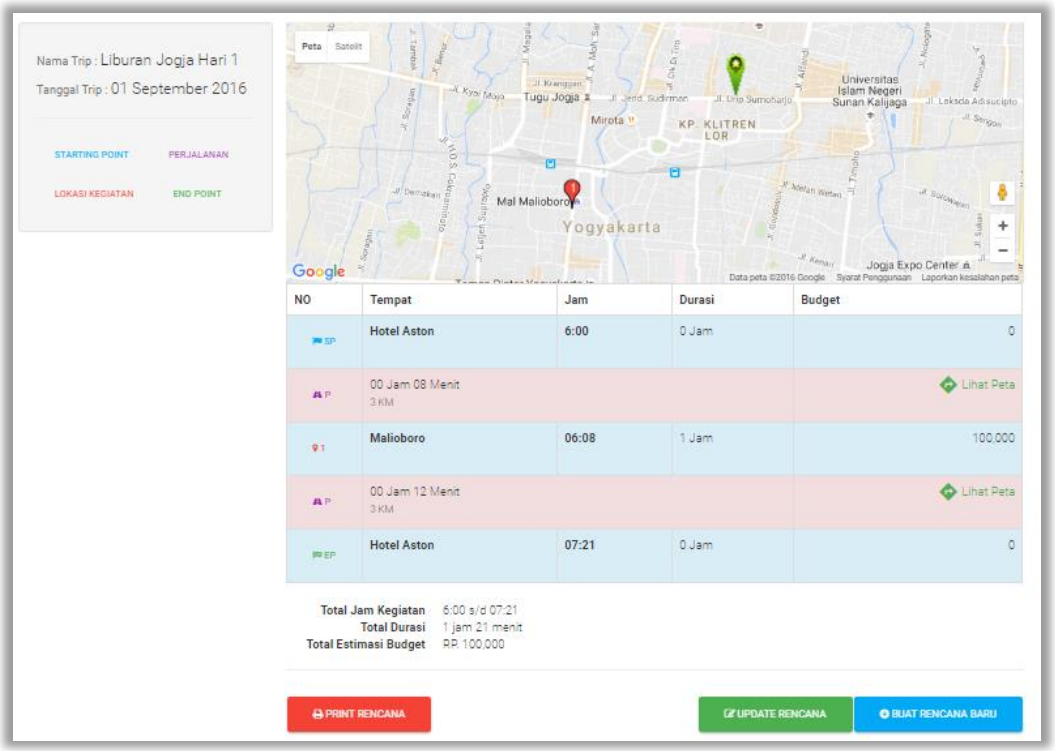
Gambar 4.33 *Coding map* bagian 2

```
  google.maps.event.addListener(marker, 'click', (function(marker, i) {
    return function() {
      infowindow.setContent('<h3>'+locations[i][0]+'</h3>');
      infowindow.open(map, marker);
    }
  })(marker, i));
}
function AutoCenter() {
  //  Create a new viewpoint bound
  var bounds = new google.maps.LatLngBounds();
  //  Go through each...
  $.each(markers, function (index, marker) {
  bounds.extend(marker.position);
  });
  //  Fit these bounds to the map
  map.fitBounds(bounds);
}
AutoCenter();
```

Gambar 4.34 *Coding map* bagian 3

Pada gambar 4.34 terdapat *coding* yang membuat *map* akan muncul tepat di tengah dari koordinat tempat-tempat wisata yang telah dipilih.

4.2.3 Halaman Detail Rencana



Gambar 4.35 Tampilan halaman detail rencana

Pada gambar 4.35 terdapat tampilan halaman detail yang digunakan untuk memperlihatkan kepada *user* hasil rencana berlibur yang telah dibuat. Sehingga *user* mengetahui total jam kegiatan, total durasi, dan total *budget* yang *user* gunakan ketika berlibur.

Pada bagian bawah gambar 4.35 terdapat tiga *button* yang dapat digunakan oleh *user* yaitu PRINT RENCANA untuk mencetak rencana berlibur, UPDATE untuk memperbarui rencana yang telah kita buat, dan BUAT RENCANA BARU untuk membuat rencana baru.

```php
<?php $row = $data->row();?>
    <div class="row">
        <div class="col-md-3 well" >

            <h4><small>Nama Trip : </small><?php echo $row->judul;?></h4>
            <h4><small>Tanggal Trip : </small><?php echo ind_format($row->tanggal);?></h4>
            <hr>
            <button class="btn btn-sm  btn-info halah">Starting Point</button>
            <button class="btn btn-sm  btn-primary halah">Perjalanan</button>
            <button class="btn btn-sm  btn-danger halah">Lokasi Kegiatan</button>
            <button class="btn btn-sm  btn-success halah">End Point</button>
        </div>
```

Gambar 4.36 *Coding* tampilan judul dan tanggal kegiatan

Pada gambar 4.36 terdapat *coding* untuk menampilkan informasi judul dan tanggal kegiatan. dan juga informasi dari warna *icon* di *map* dan tabel.

```php
<div class="col-md-9">
    <div id="map" style="width: 100%; height: 300px;"></div>
    <div class="table-responsive">
        <table class="table table-hover table-bordered">
            <thead>
                <tr>
                    <th width="30">NO</th>
                    <th>Tempat</th>
                    <th>Jam</th>
                    <th>Durasi</th>
                    <th>Budget</th>
                </tr>
            </thead>
            <tbody>
            <?php
            $awal_durasi = "";
            $total_budget = 0;
            $dtk = 0;
            $dtk1 =0;
            $before = 0;
            $times = 0;
            $data_list = get_dari('trip_detail',['id_trip'=>$row->id],'posisi','asc');
            $total_list = $data_list->num_rows();
            foreach($data_list->result() as $x => $list):
                $total_budget += $list->budget;
                if($list->jenis == 0)
                {
                    $dari = 'tempat_wisata';
                }
                else
                {
                    $dari = 'hotel';
                }

                $nama_tempat = tampilkan($dari,['id'=>$list->id_tempat],'nama');
            ?>
```

Gambar 4.37 *Coding* tampilan tabel halaman detail rencana bagian 1

Pada gambar 4.37 terdapat *coding* untuk judul dari tiap baris tabel. Untuk urutan posisi dari kegiatan diambil dari tabel *trip_detail. budget* juga mengambil dari tabel *tempat_wisata* atau hotel di *database*. untuk nama tempat juga mengambil menggunakan fungsi tampilkan di *apphelper.php* pada *folder helpers*.

```
<tr class="bg-info">
    <td>
        <?php
        if($list->posisi==0)
        {
            echo '<button class="btn btn-info btn-sm  halah"><i class="fa fa-flag"></i> SP</button>';
        }
        else if($list->posisi == $total_list-1)
        {
            echo '<button class="btn btn-success btn-sm halah"><i class="fa fa-flag-checkered"></i> EP</button>';
        }
        else
        {
            echo '<button class="btn btn-danger btn-sm  halah"><i class="fa fa-map-marker"></i> '.$x.'</button>';
        }
        ?>
    </td>
    <td><strong><?php echo $nama_tempat;?></strong></td>
```

Gambar 4.38 *Coding* tampilan tabel halaman detail rencana bagian 2

Pada gambar 4.38 terdapat *coding* untuk tampilan *icon* dan nama di kolom tabel. *Coding* ini akan memperlihatkan *icon* dan nama berdasarkan posisi dari tempat kegiatan. Pada gambar 4.38 juga terdapat *coding* untuk menampilkan baris nama tempat kegiatan.

Pada gambar 4.39 terdapat *coding* untuk menampilkan baris jam kegiatan. *Coding* pada gambar untuk menghitung jam kegiatan berdasarkan durasi dari tempat kegiatan sebelumnya.

```
<td>
    <?php
    if($list->posisi == 0)
    {
        $jamnya = $list->jam.':00';
        $awal_durasi = $list->jam.':00';
    }
    else if($list->posisi == 1)
    {
        $jamnya = hitung_jam($list->id);
        $dtk = hitung_detik($list->id)+($list->durasi * 3600);
    }
    else
    {
        if($before==0)
        {
            $dtk = $dtk;
        }
        else
        {
            //sebelumnya
            //echo konversi_jam($before);
            //echo "<hr>";
            $ambil_posisi_sebelumnya0  = get_dari('trip_detail',['id_trip'=>
            $list->id_trip,'posisi'=> $list->posisi-1],'id');
            $cc = $ambil_posisi_sebelumnya0->row();
            //durasi sebelumnya
            $drs = $cc->durasi*3600;
            $dtk = $before+$drs;
        }

        //get sebelumnya
        $ambil_posisi_sebelumnya  = get_dari('trip_detail',['id_trip'=>$list->
        id_trip,'posisi'=> $list->posisi-1],'id');
        $aa = $ambil_posisi_sebelumnya->row();
        if($aa->jenis == 0)
        {
            $dr = 'tempat_wisata';
        }
        else
        {
            $dr = 'hotel';
        }
```

Gambar 4.39 *Coding* tampilan tabel halaman detail rencana bagian 3

```
            $koordinat1 = tampilkan($dr,['id'=>$aa->id_tempat],'koordinat');
            $koordinat2 = tampilkan($dari,['id'=>$list->id_tempat],'koordinat');
            $jarak = get_jarak($koordinat1,$koordinat2,'','','',true);

            //$before = hitung_detik($list->id);
            //$dtk = $dtk;
            $before =+ $dtk+$jarak;



            $jamnya= konversi_jam($before);
        }
        ?>
        <strong><?php echo $jamnya;?></strong>
    </td>
    <td>
        <?php echo $list->durasi;?> Jam
    </td>
    <td class="text-right">
        <?php echo number_format($list->budget);?>
    </td>
</tr>
<?php if($list->posisi != $total_list-1):?>
    <tr class="bg-danger">
        <td>
            <button class="btn btn-primary btn-sm "><i class="fa fa-road"></i> P</button>
        </td>
        <td colspan="3">
            <?php
            //tempat sebelumnya
            $sebelum = get_dari('trip_detail',['id_trip'=>$list->id_trip,'posisi'=>$list->posisi+1],'posisi');
            $rw = $sebelum->row();
```

Gambar 4.40 *Coding* tampilan tabel halaman detail rencana bagian 4

```
                    if($rw->jenis==0)
                    {
                        $dari2='tempat_wisata';
                    }
                    else
                    {
                        $dari2 = 'hotel';
                    }


                    $koordinat1 = tampilkan($dari,['id'=>$list->id_tempat],'koordinat');
                    $koordinat2 = tampilkan($dari2,['id'=>$rw->id_tempat],'koordinat');



                    //$detik = get_jarak($koordinat1,$koordinat2,true);

                     echo get_jarak($koordinat1,$koordinat2,true);
                     echo "<br>";

                    ?>
                    <small><?php echo get_jarak($koordinat1,$koordinat2,'',true);?> KM</small>
                </td>
                <td class="text-right">
                    <a class="text-success window-opener"
                        href="javascript: void(0)"
                        onclick="popup('<?php echo base_url();?>peta/<?php echo $list->id;?>')">

                        <i class="material-icons">directions</i> Lihat Peta
                    </a>
                </td>
            </tr>
        <?php endif;?>
    <?php endforeach;?>
    </tbody>

</table>
</div>
```

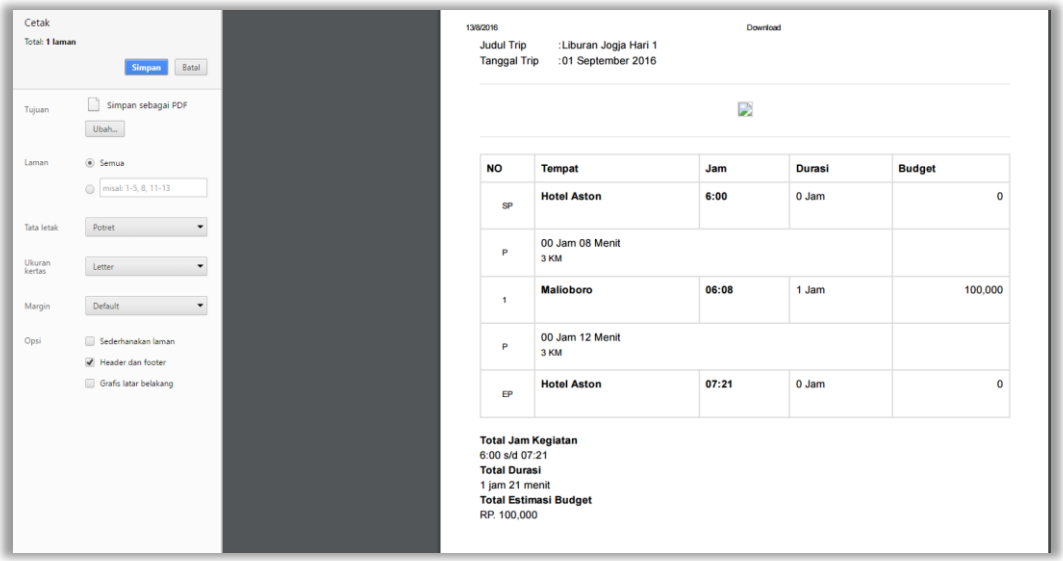
Gambar 4.41 *Coding* tampilan tabel halaman detail rencana bagian 5

Pada gambar 4.41 terdapat *coding* untuk menampilkan informasi jarak dan waktu antara tempat kegiatan dan juga *button* Lihat Peta di samping dari kolom perjalanan. Ketika *button* Lihat Peta ditekan, *coding* akan mengarahkan ke fungsi peta di *file website.php* pada *folder controllers*. Dari fungsi peta tersebut kemudian akan menampilkan halaman peta di *views*.

```
        <dl class="dl-horizontal">
          <dt>Total Jam Kegiatan</dt>
          <dd><?php echo $awal_durasi;?> s/d <?php echo $jamnya;?></dd>
          <dt>Total Durasi</dt>
          <dd><?php echo selisih_jam($awal_durasi.':00',$jamnya.':00');?></dd>
          <dt>Total Estimasi Budget</dt>
          <dd>RP. <?php echo number_format($total_budget);?></dd>
        </dl>
        <hr>
        <a href="<?php echo base_url();?>rencana" class="btn btn-info btn-raised pull-right">
        <i class="fa fa-plus-circle"></i> Buat Rencana Baru</a>
        <a href="<?php echo base_url();?>update_rencana/<?php echo $row->id;?>" class="btn
        btn-success btn-raised pull-right"><i class="fa fa-edit"></i> Update Rencana</a>
        <a href="<?php echo base_url();?>download/<?php echo $row->id;?>"  class="btn
        btn-danger btn-raised"><i class="fa fa-print"></i> Print Rencana</a>
    </div>
</div>
```

Gambar 4.42 *Coding* tampilan informasi

Pada gambar 4.42 terdapat *coding* untuk menampilkan total jam kegiatan, total durasi, dan total *budget*. Gambar 4.42 juga memperlihatkan *coding* yang digunakan untuk *button* Buat rencana baru yang mengakses fungsi rencana, *button Update* Rencana yang mengakses fungsi *update_rencana*, dan *button* Print Rencana yang mengakses fungsi *download*. Semua fungsi mengakses *file website.php* pada *folder controllers*.



Gambar 4.43 Tampilan halaman cetak rencana

Pada gambar 4.43 terdapat tampilan halaman Cetak rencana. Halaman cetak rencana akan tampil setelah *user* menekan *button Print* rencana pada halaman detail

rencana. Halaman cetak rencana akan muncul ketika fungsi *download* di *controller* dijalankan. Halaman cetak rencana memanfaatkan *javascript* untuk mengolah tampilan.

4.3 Pengujian sistem

Pengujian sistem yang dilakukan bertujuan untuk mengetahui apakah sistem yang dibuat sudah sama dengan sistem yang diharapkan. Sehingga ketika *user* membuat rencana berlibur tidak terjadi kesalahan-kesalahan dalam kerja sistem yang akan mengganggu *user* dalam proses membuat rencana berlibur.

4.3.1 Pengujian User Interface

Pengujian *user interface* bertujuan untuk mengetahui fungsionalitas dari elemen-elemen *interface* yang terdapat di dalam halaman sistem. Elemen yang diujikan adalah elemen *button* di halaman *login*, Buat rencana, dan detail rencana pada aplikasi. Hasil pengujian dapat dilihat pada Tabel 4.1.

Tabel 4.1 Hasil Pengujian *Interface*

No	Kasus Yang diuji	Hasil Yang Diharapkan	Hasil Yang Didapatkan	Status
1	*Button* MASUK di halaman *login*	Sistem dapat masuk ke halaman akun	Setelah *button* Masuk ditekan, sistem akan mengarahkan *user* ke halaman akun	Berhasil
2	*Button* Menu di *header* halaman *website*	Sistem dapat menampilkan halaman sesuai	Setelah *button* ditekan, muncul halaman sesuai dengan nama tombol	Berhasil

No	Kasus Yang diuji	Hasil Yang Diharapkan	Hasil Yang Didapatkan	Status
		dengan nama *button*		
3	*Button* LOGIN DISINI di halaman Buat Rencana	Sistem dapat menampilkan halaman *login*	Setelah *button* LOGIN DISINI ditekan, muncul halaman *login*	Berhasil
4	*Button* TAMBAH KEGIATAN di halaman buat rencana	Sistem dapat menampilkan *dialog box* tempat *wisatan* dan penginapan	Setelah *button* TAMBAH KEGIATAN ditekan, muncul dialog *box* tempat *wisatan* dan penginapan	Berhasil
5	*Button* SIMPAN di halaman buat rencana	Sistem menyimpan data dan menampilkan halaman detail rencana	Setelah *button* SIMPAN ditekan, sistem menyimpan data dan menampilkan halaman detail rencana	Berhasil
6	*Button PRINT* RENCANA di halaman detail rencana	Sistem dapat menampilkan halaman cetak rencana	Setelah *button PRINT* RENCANA ditekan, sistem menampilkan halaman cetak data	Berhasil
7	*Button UPDATE* RENCANA di halaman detail rencana	Sistem dapat menampilkan halaman rencana dengan data	Setelah *button UPDATE* RENCANA ditekan, sistem menampilkan halaman rencana	Berhasil

No	Kasus Yang diuji	Hasil Yang Diharapkan	Hasil Yang Didapatkan	Status
		yang ingin di *update*	beserta data yang ingin diperbarui	
8	*Button* BUAT RENCANA BARU di halaman detail rencana	Sistem dapat menampilkan halaman rencana	Setelah *button* BUAT RENCANA BARU ditekan, sistem menampilkan halaman rencana.	Berhasil

4.3.2 Pengujian Fungsi Sistem

Pengujian fungsi sistem bertujuan untuk mengevaluasi apakah kerja fungsi-fungsi yang ada pada sistem sudah berjalan dengan baik. Adapun halaman yang akan diujikan yaitu halaman Buat Rencana dan Detail Rencana. Hasil pengujian dapat dilihat pada Tabel 4.2.

Tabel 4.2 Hasil Pengujian Fungsi Sistem

No	Kasus Yang diuji	Hasil Yang Diharapkan	Hasil Yang Didapatkan	Status
1	Pengujian fungsi hitung jarak dan waktu antar tempat kegiatan	Sistem dapat menampilkan dan menghitung jarak dan waktu antar tempat kegiatan secara otomatis	Sistem otomatis menampilkan dan menghitung jarak dan waktu antar tempat wisata	Berhasil

No	Kasus Yang diuji	Hasil Yang Diharapkan	Hasil Yang Didapatkan	Status
2	Pengujian fungsi hitung total jam kegiatan	Sistem dapat menghitung total jam kegiatan secara otomatis	Sistem otomatis menghitung total jam kegiatan	Berhasil
3	Pengujian fungsi hitung total durasi kegiatan	Sistem dapat menghitung total durasi kegiatan secara otomatis	Sistem otomatis menghitung total durasi kegiatan	Berhasil
4	Pengujian fungsi hitung total *budget* kegiatan	Sistem dapat menghitung total *budget* kegiatan secara otomatis	Sistem otomatis menghitung total *budget* kegiatan	Berhasil
5	Pengujian fungsi menampilkan Map	Sistem dapat menampilkan *map* dan koordinat dari tiap tempat kegiatan	Sistem menampilkan *map* dan koordinat tempat kegiatan	Berhasil
6	Pengujian fungsi Menyimpan rencana berlibur	Sistem dapat menyimpan rencana berlibur ke *database*	Rencana yang di simpan sudah tersimpan di *database*	Berhasil
7	Pengujian fungsi Memperbarui rencana berlibur	Sistem dapat memperbarui rencana berlibur yang sudah	Rencana berlibur yang tersimpan dapat diperbarui	Berhasil

No	Kasus Yang diuji	Hasil Yang Diharapkan	Hasil Yang Didapatkan	Status
		tersimpan di *database*		
8	Pengujian fungsi Menghapus rencana berlibur	Sistem dapat menghapus rencana berlibur yang terdapat di *database*	Rencana berlibur yang tersimpan di *database* sudah tidak ada	Berhasil
9	Pengujian fungsi Mencetak rencana berlibur	Sistem dapat mencetak rencana berlibur	Sistem mencetak rencana berlibur	Berhasil

4.3.3 Pengujian Validasi

Pengujian validasi bertujuan untuk mengevaluasi apakah validasi-validasi yang ada pada sistem sudah berjalan dengan baik. Adapun halaman yang akan diujikan yaitu halaman *Login*. Hasil pengujian dapat dilihat pada Tabel 4.3.

Tabel 4.3 Hasil Pengujian Validasi

No	Kasus Yang diuji	Hasil Yang Diharapkan	Hasil Yang Didapatkan	Status
1	Validasi jika *username* dan *password* tidak sesuai	Sistem dapat menampilkan pesan peringatan bahwa *username* atau *password* yang di *input* salah	Muncul pesan peringatan yang memberitahukan bahwa *username* atau *password* salah	Berhasil

No	Kasus Yang diuji	Hasil Yang Diharapkan	Hasil Yang Didapatkan	Status
2	Validasi jika *user* belum terdaftar dalam halaman *login*	Sistem dapat menampilkan pesan peringatan bahwa *user* yang dimasukkan belum terdaftar	Muncul pesan yang memberitahukan *user* belum terdaftar	Berhasil
3	Validasi jika salah satu *input* masih kosong	Sistem dapat menampilkan pesan bahwa data yang di *input* tidak boleh kosong	Muncul pesan peringatan yang memberitahukan salah satu data yang di *input* tidak boleh kosong	Berhasil
4	Validasi halaman buat rencana	Sistem dapat menampilkan pesan jika *user* harus *login* terlebih dahulu untuk menggunakan halaman buat rencana.	Muncul pesan untuk *non member* untuk melakukan *login* terlebih dahulu agar dapat menggunakan halaman buat rencana	Berhasil

4.3.4 Pengujian Keamanan Sistem

Pengujian keamanan sistem bertujuan untuk mengetahui keamanan yang sudah diterapkan ke dalam sistem. Hasil pengujian dapat dilihat pada Tabel 4.4.

Tabel 4.4 Hasil Pengujian Keamanan Sistem

No	Kasus Yang diuji	Hasil Yang Diharapkan	Hasil Yang Didapatkan	Status
1	Pengguna dengan level *non member*	Pengguna dengan level *non member* tidak dapat menggunakan fitur halaman buat rencana	Pengguna tidak dapat membuat rencana.	Berhasil

4.4 Pembahasan

Aplikasi dapat menyusun kegiatan berlibur sesuai dengan kebutuhan *user*, di mana *user* dapat menentukan pilihan tempat berlibur yang ingin dikunjungi, waktu berkunjung, durasi di tempat kegiatan, dan *budget* yang diperlukan ketika berada di tempat kegiatan. Sistem membuat *user* dapat mengurutkan tempat-tempat kegiatan berdasarkan kemauannya. Sistem juga dapat menampikan total dari jam kegiatan, durasi kegiatan, dan *budget* kegiatan sehingga *user* dapat menyesuaikan dengan kemampuan dan kebutuhan *user*.

Aplikasi dapat menampilkan *Map* dari tempat kegiatan. koordinat yang ada di *map* akan memudahkan *user* dalam menemukan posisi dari tempat kegiatan. Sistem juga dapat membuat *user* mengetahui jalur yang dapat dilalui dari tempat kegiatan pertama dan kedua yang membuat *user* tidak kesulitan dalam mencari jalan ketika pergi dari satu tempat ke tempat berikutnya.

Aplikasi dapat menyimpan rencana berlibur yang telah dibuat ke *database* sehingga *user* nantinya dapat mengakses kembali data dari rencana berlibur yang

dibuat. *User* juga dapat memperbarui dan menghapus data rencana berlibur yang telah dibuat di dalam *database*.

Aplikasi dapat membantu *user* dalam mencari referensi tentang tempat wisata. Di mana *user* akan dapat melihat informasi tentang tempat wisata berupa informasi daerah atau kota dari tempat wisata tersebut, jenis wisata, *budget* yang diperlukan ketika ingin mendatangi tempat wisata tersebut, dan informasi umum seputar tempat wisata tersebut.

Aplikasi dapat mencetak rencana berlibur sehingga memudahkan *user* jika membutuhkan *hardcopy* dari rencana yang telah dibuat.