

**PERANCANGAN DAN PEMBUATAN BUSINESS LOGIC  
BERBASIS PHP MENGGUNAKAN FRAMEWORK  
CODEIGNITER PADA SITUS PARIWISATA PENDUKUNG  
PROSES PEMBUATAN RENCANA PROGRAM TOURISM**

Skripsi

untuk memenuhi sebagian persyaratan  
mencapai derajat Sarjana S-1



Diajukan oleh  
Mohammad Fikri  
NIM: 20120140030

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS TEKNIK  
UNIVERSITAS MUHAMMADIYAH YOGYAKARTA**

**2016**

**HALAMAN PENGESAHAN I**

**SKRIPSI**

**PERANCANGAN DAN PEMBUATAN BUSINESS LOGIC**

**BERBASIS PHP MENGGUNAKAN FRAMEWORK**

**CODEIGNITER PADA SITUS PARIWISATA PENDUKUNG**

**PROSES PEMBUATAN RENCANA PROGRAM TOURISM**



**Haris Setyawan, S.T., M.Eng.**

NIK: 19691111200004123043

**Reza Giga Isnanda, S.T., M.Sc.**

NIK: 19860603201504123071

## HALAMAN PENGESAHAN II

### SKRIPSI

PERANCANGAN DAN PEMBUATAN BUSINESS LOGIC  
BERBASIS PHP MENGGUNAKAN FRAMEWORK  
CODEIGNITER PADA SITUS PARIWISATA PENDUKUNG  
PROSES PEMBUATAN RENCANA PROGRAM TOURISM

Diajukan Oleh:

**MOHAMMAD FIKRI**  
**20120140030**

Skripsi telah Dipertahankan dan Disahkan di depan Dewan Penguji Program Studi  
Teknologi Informasi Fakultas Teknik Universitas Muhammadiyah Yogyakarta

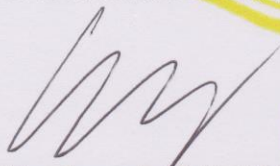
Tanggal 31 Agustus 2016

Dewan Penguji:



**Haris Setyawan, S.T., M.Eng.**

NIK: 19691111200004123043



**Reza Giga Isnanda, S.T., M.Sc.**

NIK: 19860603201504123071



**Cahya Damarjati, S.T., M.Eng.**

NIK: 19870315201507123077

## **PERNYATAAN**

Dengan ini saya menyatakan bahwa skripsi ini tidak mengandung karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, kecuali yang secara tertulis diacu dalam naskah dan disebutkan dalam daftar pustaka.

Yogyakarta, 31 Agustus 2016

Mohammad Fikri

## PRAKATA

Puji syukur kehadiran Allah SWT yang telah melimpahkan hidayah dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “Perancangan dan Pembuatan *Business Logic* Berbasis PHP Menggunakan *Framework Codeigniter* Pada Situs Pariwisata Pendukung Proses Pembuatan Rencana Program Tourism”. Laporan skripsi disusun untuk memenuhi salah satu syarat dalam memperoleh gelar Sarjana Teknik (S.T) pada Program Studi S1 Teknologi Informasi Universitas Muhammadiyah Yogyakarta.

Dalam melakukan penelitian dan penyusunan laporan skripsi ini, penulis mendapatkan banyak bantuan, bimbingan, petunjuk, saran, dan dukungan dari semua pihak yang terlibat. Untuk itu, Penulis mengucapkan banyak terima kasih kepada:

1. Haris Setyawan, S.T., M.Eng. selaku pembimbing utama yang telah membimbing penulis dengan kesabaran dan ketulusan.
2. Reza Giga Isnanda, S.T., M.Sc. selaku pembimbing pendamping yang telah meluangkan waktu dan sangat sabar membimbing penulis dalam pengembangan aplikasi.
3. Cahya Damarjati, S.T., M.Eng. selaku dosen penguji yang mau meluangkan waktu untuk memberikan kritik dan saran kepada penulis dalam pembuatan skripsi.
4. Muhammad Helmi Zain Nuri, S.T., M.T selaku Ketua Program Studi S1 Teknologi Informasi yang memberikan izin kepada penulis untuk belajar.
5. Para dosen Jurusan Teknologi Informasi Universitas Muhammadiyah Yogyakarta, Bapak Asroni, Bapak Eko, Bapak Joko, Bapak Slamet, dan Mas Okto yang telah memberikan tambahan pengetahuan dan mengajarkan ilmunya kepada penulis selama perkuliahan.
6. Para Staff Jurusan Teknologi Informasi UMY, Mas Reza, Mas Ronald, Mbak April, Mbak Lilis, Mas Andi, dan Mas Haris yang senantiasa membantu penulis dalam urusan administrasi.

7. Segenap Staff Tata Usaha Fakultas Teknik Universitas Muhammadiyah Yogyakarta yang telah memberikan pelayanan terbaiknya bagi penulis.
8. Ayah dan Ibu yang dari lahir sampai sekarang selalu memberikan kasih sayang dan cinta kasihnya kepada penulis sehingga penulis dapat mencapai ke tahap sekarang.
9. Teman-teman Teknologi Informasi, khususnya teman seperjuangan angkatan 2012, kalian merupakan sahabat dan keluarga yang senantiasa berbagi ilmu, suka cita, suka duka, dan pengalaman selama kuliah di Fakultas Teknik.
10. Terima kasih Keluarga Besar Kos Famili yang telah mendukung dan memberikan semangat kepada penulis.

Hanya doa yang dapat penulis panjatkan kepada Allah SWT, semoga Allah SWT berkenan membalas semua kebaikan Ayah, Ibu, Saudara dan teman-teman. Semoga penelitian ini dapat bermanfaat bagi pihak-pihak yang berkepentingan.

Yogyakarta, 31 Agustus 2016

Mohammad Fikri

## DAFTAR ISI

HALAMAN PENGESAHAN I .....	<b>Error! Bookmark not defined.</b>
HALAMAN PENGESAHAN II .....	<b>Error! Bookmark not defined.</b>
PERNYATAAN .....	iv
PRAKATA .....	v
DAFTAR ISI .....	vii
DAFTAR GAMBAR .....	ix
DAFTAR TABEL .....	xi
INTISARI .....	xii
ABSTRACT .....	xiii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah .....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan Penelitian .....	3
1.5 Manfaat Penelitian .....	4
1.6 Sistematika Penulisan .....	4
BAB II TINJAUAN PUSTAKA .....	6
2.1 Tinjauan Pustaka .....	6
2.2 Landasan Teori .....	10
2.2.1 Web Browser .....	10
2.2.2 Web server .....	10
2.2.3 HTML (Hypertext Markup Language) .....	10
2.2.4 CSS (Cascading Style Sheet) .....	11
2.2.5 Javascript .....	11
2.2.6 JQuery .....	11
2.2.7 PHP (PHP Hypertext Preprocessor) .....	12
2.2.8 CodeIgniter .....	13
2.2.9 MVC .....	13
2.2.10 XAMPP .....	14

2.2.11	Notepad++.....	14
2.2.12	Metode Penelitian .....	15
2.2.13	Flowchart .....	17
2.2.14	UML.....	18
<b>BAB III ERANCANGAN PENELITIAN .....</b>		<b>23</b>
3.1	Peralatan Pendukung .....	23
3.1.1	Perangkat Keras .....	23
3.1.2	Perangkat Lunak .....	23
3.2	Analisis Kebutuhan .....	24
3.3	Rancangan .....	25
3.3.1	Flowchart .....	25
3.3.2	Use Case Diagram.....	27
3.3.3	Activity Diagram .....	28
3.3.4	Class Diagram.....	32
3.4	Metode Pengujian.....	34
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>		<b>37</b>
4.1	Pengembangan Sistem.....	37
4.2	Pembuatan Bussines Logic Rencana Berlibur .....	37
4.2.1	Halaman Login.....	37
4.2.2	Halaman Buat Rencana .....	41
4.2.3	Halaman Detail Rencana.....	61
4.3	Pengujian sistem.....	67
4.3.1	Pengujian User Interface .....	67
4.3.2	Pengujian Fungsi Sistem.....	69
4.3.3	Pengujian Validasi .....	71
4.3.4	Pengujian Keamanan Sistem.....	72
4.4	Pembahasan .....	73
<b>BAB V KESIMPULAN DAN SARAN .....</b>		<b>75</b>
5.1	Kesimpulan.....	75
5.2	Saran.....	76
<b>DAFTAR PUSTAKA .....</b>		<b>77</b>
<b>LAMPIRAN.....</b>		<b>78</b>



## DAFTAR GAMBAR

<b>Gambar 2. 1</b> <i>Use Case Diagram</i> menurut Scott W. Ambler .....	20
<b>Gambar 2. 2</b> <i>Activity Diagram</i> menurut Scott W.Ambler .....	21
<b>Gambar 2. 3</b> <i>Class Diagram</i> menurut Scott W.Ambler .....	22
<b>Gambar 3.1</b> <i>Flowchart Member Aplikasi</i> .....	26
<b>Gambar 3.2</b> <i>Use Case Diagram Aplikasi</i> .....	27
<b>Gambar 3.3</b> <i>Activity Diagram Buat Rencana Member</i> .....	29
<b>Gambar 3.4</b> <i>Activity Diagram Buat Rencana Non Member</i> .....	31
<b>Gambar 3.5</b> <i>Class Diagram Aplikasi</i> .....	33
<b>Gambar 4.1</b> Tampilan halaman <i>Login</i> .....	38
<b>Gambar 4.2</b> <i>Coding interface</i> pada halaman <i>Login</i> .....	38
<b>Gambar 4.3</b> <i>Coding function loginAuth</i> pada <i>Controllers</i> .....	39
<b>Gambar 4.4</b> <i>Coding function akun</i> pada <i>folder Controllers</i> .....	40
<b>Gambar 4.5</b> <i>Coding function login</i> pada <i>Controllers</i> .....	40
<b>Gambar 4.6</b> Tampilan halaman rencana .....	42
<b>Gambar 4.7</b> <i>Coding session user</i> .....	42
<b>Gambar 4.8</b> <i>Coding Button Login</i> Disini.....	43
<b>Gambar 4.9</b> Halaman rencana <i>non member</i> .....	43
<b>Gambar 4.10</b> <i>Coding session starting point</i> .....	44
<b>Gambar 4.11</b> <i>Coding session end point</i> .....	44
<b>Gambar 4.12</b> <i>Coding form</i> judul dan tanggal .....	45
<b>Gambar 4.13</b> Hasil <i>coding form</i> judul dan tanggal.....	45
<b>Gambar 4.14</b> <i>Coding form starting point</i> bagian 1 .....	46
<b>Gambar 4.15</b> <i>Coding form starting point</i> bagian 2 .....	47
<b>Gambar 4.16</b> <i>Coding form starting point</i> bagian 3 .....	48
<b>Gambar 4.17</b> Hasil <i>coding form starting point</i> .....	48
<b>Gambar 4.18</b> Tampilan jarak dan waktu halaman rencana .....	48
<b>Gambar 4.19</b> <i>Coding tampilan jarak dan waktu</i> .....	49
<b>Gambar 4.20</b> <i>Coding form end point</i> bagian 1 .....	50
<b>Gambar 4.21</b> <i>Coding form end point</i> bagian 2 .....	51
<b>Gambar 4.22</b> <i>Coding form end point</i> bagian 3 .....	52

<b>Gambar 4.23</b> <i>Coding form end point</i> bagian 4 .....	53
<b>Gambar 4.24</b> <i>Coding button</i> tambah kegiatan dan simpan .....	53
<b>Gambar 4.25</b> <i>Coding dialog box</i> tempat wisata dan penginapan bagian 1 .....	54
<b>Gambar 4.26</b> <i>Coding dialog box</i> tempat wisata dan penginapan bagian 2 .....	55
<b>Gambar 4.27</b> <i>Coding dialog box</i> tempat wisata dan penginapan bagian 3 .....	56
<b>Gambar 4.28</b> Hasil <i>coding dialog box</i> tempat wisata dan penginapan.....	56
<b>Gambar 4.29</b> <i>Coding form</i> kegiatan bagian 1 .....	57
<b>Gambar 4.30</b> <i>Coding form</i> kegiatan bagian 2.....	58
<b>Gambar 4.31</b> <i>Coding form</i> kegiatan bagian 3.....	58
<b>Gambar 4.32</b> <i>Coding map</i> bagian 1 .....	59
<b>Gambar 4.33</b> <i>Coding map</i> bagian 2.....	60
<b>Gambar 4.34</b> <i>Coding map</i> bagian 3.....	60
<b>Gambar 4.35</b> Tampilan halaman detail rencana .....	61
<b>Gambar 4.36</b> <i>Coding</i> tampilan judul dan tanggal kegiatan .....	62
<b>Gambar 4.37</b> <i>Coding</i> tampilan tabel halaman detail rencana bagian 1 .....	62
<b>Gambar 4.38</b> <i>Coding</i> tampilan tabel halaman detail rencana bagian 2 .....	63
<b>Gambar 4.39</b> <i>Coding</i> tampilan tabel halaman detail rencana bagian 3 .....	64
<b>Gambar 4.40</b> <i>Coding</i> tampilan tabel halaman detail rencana bagian 4 .....	64
<b>Gambar 4.41</b> <i>Coding</i> tampilan tabel halaman detail rencana bagian 5 .....	65
<b>Gambar 4.42</b> <i>Coding</i> tampilan informasi .....	66
<b>Gambar 4.43</b> Tampilan halaman cetak rencana .....	66

## DAFTAR TABEL

<b>Tabel 2.1</b> Simbol <i>Flowchart</i> .....	17
<b>Tabel 4.1</b> Hasil Pengujian <i>Interface</i> .....	67
<b>Tabel 4.2</b> Hasil Pengujian Fungsi Sistem.....	69
<b>Tabel 4.3</b> Hasil Pengujian Validasi .....	71
<b>Tabel 4.4</b> Hasil Pengujian Keamanan Sistem .....	73

## INTISARI

Turis sebutan untuk para pelaku *tourism*, memanfaatkan jasa biro-biro *travel* (Perjalanan) yang sudah berpengalaman dalam mengelola dan menjalankan *tourism*. Biasanya, biro-biro ini sudah memiliki beberapa program *tourism* yang siap ditawarkan kepada para turis. Dari beberapa pilihan program yang ditawarkan, para turis cukup memilih program yang sesuai dengan kriteria mereka.

Memanfaatkan jasa biro *travel* merupakan proses yang cepat dan mudah, namun terkadang para turis juga memutuskan untuk membuat sendiri rencana program *tourism* mereka dikarenakan program *tourism* yang ditawarkan oleh biro *travel* umumnya bersifat tidak fleksibel. Di mana para turis tidak punya keleluasaan atau kemampuan untuk mengubah program kegiatan *tourism* seperti memilih sendiri obyek wisata, jenis transportasi, tempat makan, atau tempat menginap yang diinginkan. Oleh karena itu, dengan membuat rencana program *tourism* sendiri, para turis dapat lebih fleksibel dalam melakukan *tourism*. Keuntungan lainnya, fleksibilitas dalam menentukan pilihan *tourism* memungkinkan para turis untuk lebih menghemat biaya perjalanan, makan, ataupun penginapan.

Walau membuat sendiri rencana program *tourism* dengan memanfaatkan situs pariwisata memberikan keuntungan dari segi fleksibilitas, namun proses ini sulit dilakukan karena informasi yang diberikan sebuah situs pariwisata belum tentu lengkap sehingga mereka harus mengunjungi beberapa situs yang menyebabkan lebih banyak lagi situs yang harus dikunjungi oleh para turis.

Melihat kondisi tersebut, Penelitian ini bertujuan untuk merancang dan membuat *business logic* sebuah situs pariwisata yang mendukung proses pembuatan rencana program *tourism*. Sistem ini dibuat menggunakan bahasa pemrograman PHP dan *framework codeigniter*. hasil dari pembuatan *Bussines logic* ini dapat mendukung pembangunan program aplikasi yang dapat digunakan oleh para turis dalam membuat sendiri rencana *tourism* seperti memilih obyek wisata yang ingin mereka kunjungi per harinya yang nantinya dapat disimpan di dalam akunnya dan dapat dicetak untuk dibawa ketika dalam perjalanan *tourism*.

**Kata Kunci :** Pariwisata, Aplikasi Web, Perencanaan Berlibur

## ABSTRACT

Tourist is called for people who do tourism, using by the services of travel agencies which is experienced in managing and organizing tourism. Typically, these agencies already have some tourism program that is ready to offer to the tourists. From the program options that offered, tourists can simply choose a program that fits of their criteria.

Using the services of a travel agency is a process that is fast and easy, but sometimes the tourists decided to make their own tourism program plans because of that offered by travel agents usually inflexible. Where the tourists do not have flexibility or ability to change the program like choosing their own activities such as tourism attractions, modes of transport, restaurant, or hotel that desired. Therefore, by making their own tourism program plan, the tourists can have more flexibility in doing tourism. Another advantage, also allows tourists to save money on travel, meals, or lodging.

Although making your own program of tourism plan by using the tourism sites provide will given benefits in terms of flexibility, but the process is difficult because the information that given by tourism website is not necessarily complete, so they have to visit multiple sites that leading to more sites to be visited by tourists.

Seeing these conditions, the research purpose to design and create business logic of tourism site for supporting tourism program planning process. This system is built using PHP and CodeIgniter framework. the Result of developing Business logic can support the development of application programs that can be used by tourists in making their own plans of tourism such as choosing the attractions that they want to visit per day, which will be saved in their account or can be printed to be carried while in tourism.

**Keyword :** Tourism, Website Application, Tourism Planning

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Kondisi manusia dewasa ini tidak bisa lepas dari konsep *leisure* (waktu luang). Dalam mengisi *leisure*, tiap orang memiliki cara yang berbeda. Salah satu yang umum dilakukan adalah mengisi *leisure* dengan *tourism*. *Tourism* (Pariwisata) merupakan sebuah kegiatan di mana orang melakukan perjalanan dan menetap di sebuah tempat di luar lingkungan biasa mereka selama tidak lebih dari satu tahun untuk bersantai, bisnis, dan tujuan lain (World Tourism Organization, 1995).

Turis sebutan untuk para pelaku *tourism*, memanfaatkan jasa biro-biro *travel* (Perjalanan) yang sudah berpengalaman dalam mengelola dan menjalankan *tourism*. Biasanya, biro-biro ini sudah memiliki beberapa program *tourism* yang siap ditawarkan kepada para turis. Setiap program yang diberikan dapat berisi informasi mengenai biaya perjalanan, jadwal perjalanan per hari, lama kunjungan di tiap objek wisata, informasi mengenai objek wisata, dan lain-lain. Dari beberapa pilihan program yang ditawarkan, para turis cukup memilih program yang sesuai dengan kriteria mereka.

Walau memanfaatkan jasa biro *travel* merupakan proses yang cepat dan mudah, terkadang para turis juga memutuskan untuk membuat sendiri rencana program *tourism* mereka dikarenakan program *tourism* yang ditawarkan oleh biro *travel* umumnya bersifat tidak fleksibel. Di mana para turis tidak punya keleluasaan atau kemampuan untuk mengubah program kegiatan *tourism* seperti memilih sendiri obyek wisata, jenis transportasi, tempat makan, atau tempat menginap yang

diinginkan. Oleh karena itu, dengan membuat rencana program *tourism* sendiri, para turis dapat lebih fleksibel dalam melakukan *tourism*. Keuntungan lainnya, fleksibilitas dalam menentukan pilihan *tourism* memungkinkan para turis untuk lebih menghemat biaya perjalanan, makan, ataupun penginapan. Dalam membantu membuat rencana program *tourism*, para turis umumnya memanfaatkan informasi-informasi yang tersedia di berbagai situs pariwisata yang memuat informasi-informasi yang berkaitan dengan obyek wisata.

Walau membuat sendiri rencana program *tourism* dengan memanfaatkan situs pariwisata memberikan keuntungan dari segi fleksibilitas, namun, proses ini sulit dilakukan karena informasi yang diberikan sebuah situs pariwisata belum tentu lengkap sehingga mereka harus mengunjungi beberapa situs (Putera, 2010). Hal ini menyebabkan lebih banyak lagi situs yang harus dikunjungi oleh para turis.

Melihat kondisi tersebut, Penelitian ini bertujuan untuk merancang dan membuat *business logic* sebuah situs pariwisata yang mendukung proses pembuatan rencana program *tourism*. *Bussines logic* ini dapat mendukung pembangunan program aplikasi yang dapat digunakan oleh *user* yaitu para turis dalam membuat sendiri rencana *tourism* seperti memilih obyek wisata yang ingin dikunjungi per harinya yang nantinya dapat disimpan di dalam akunnya atau dapat dicetak untuk di bawah ketika dalam perjalanan *tourism*.

Karena banyaknya tempat-tempat wisata yang ada di dunia, maka sebagai langkah awal, penelitian ini hanya memfokuskan pada obyek-obyek wisata besar yang ada di provinsi Daerah Istimewa Yogyakarta. Untuk jangka panjang, penelitian ini akan mengembangkan sebuah situs pariwisata yang terintegrasi

dengan situs-situs pendukung seperti situs penerbangan, pelayaran, dan lain-lain. Dengan integrasi tersebut, informasi lengkap mengenai tempat wisata beserta komponen pendukungnya bisa didapatkan turis melalui sebuah situs saja.

## **1.2 Perumusan Masalah**

Berdasarkan dari latar belakang tersebut maka rumusan masalah yang paling mendasar adalah “Bagaimana membantu turis dalam membuat program *tourism* yang detail dan jelas secara mandiri?”

## **1.3 Batasan Masalah**

Berdasarkan uraian di atas dan menghindari luasnya pokok permasalahan yang ada, maka batasan masalah yang meliputi antara lain :

1. Pengembangan *website* perencanaan program *tourism* hanya fokus pada bagian *business logic*.
2. Pengembangan *website* perencanaan program *tourism* menggunakan *framework codeigniter*.

## **1.4 Tujuan Penelitian**

Adapun tujuan dalam penelitian ini adalah:

1. Membangun *business logic* untuk pembuatan suatu *website* perencanaan program *tourism*.
2. Memanfaatkan bahasa pemrograman PHP untuk membangun *website* perencanaan *tourism*.
3. Memanfaatkan *Framework CodeIgniter* untuk membangun *website* perencanaan *tourism*.



## **1.5 Manfaat Penelitian**

Manfaat yang didapat dari penelitian adalah sebagai berikut:

1. Suatu *website* untuk para turis agar bisa membuat rencana berlibur dengan detail dan jelas.
2. Media promosi obyek-obyek wisata yang ada di Yogyakarta dan sekitarnya yang diharapkan dapat meningkatkan minat para turis untuk *tourism* ke Yogyakarta.

## **1.6 Sistematika Penulisan**

Adapun Sistematika penulisan laporan ini sebagai berikut:

### **BAB I PENDAHULUAN**

Pada bab PENDAHULUAN membahas mengenai latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

### **BAB II TINJAUAN PUSTAKA**

Pada bab TINJAUAN PUSTAKA membahas menjelaskan teori-teori yang menjadi acuan dalam penulis yaitu mulai dari penjelasan tentang teknologi, penjelasan tentang teori yang digunakan dalam membangun aplikasi hingga pada Bahasa pemrograman yang digunakan.

### BAB III PERANCANGAN PENELITIAN

Pada bab PERANCANGAN PENELITIAN menjelaskan rancangan sistem agar dapat diimplementasi di dalam sistem yang sesuai harapan mengacu pada teori-teori penunjang dan metode yang sudah dijelaskan pada bab sebelumnya.

### BAB IV HASIL DAN PEMBAHASAN

Pada bab ini menjelaskan mengenai implementasi sistem dan hasil pengujian sistem yang telah dibuat.

### BAB V KESIMPULAN DAN SARAN

Pada bab ini menjelaskan mengenai kesimpulan dan saran yang dibutuhkan dari hasil penelitian yang telah dibuat.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Tinjauan Pustaka**

Anwar Firdaus (2010). Dalam penelitiannya yang berjudul “Pembuatan *Website* Peta Wisata Kabupaten Karanganyar”. Selama ini promosi dan sosialisasinya masih menggunakan media konvensional seperti *booklet* atau brosur yang memiliki kekurangan yaitu keterbatasan ukuran dan jumlah. Sebuah sistem informasi peta wisata Kabupaten Karanganyar yang berbasis web telah dikembangkan sebagai alternatif media promosi dan sosialisasi. Tujuan dari pembuatan tugas akhir ini adalah untuk mempermudah promosi dan sosialisasi objek wisata yang ada di Kabupaten Karanganyar. Metode yang digunakan dalam pembuatan *website* ini adalah pencarian data berupa studi pustaka dan survei lapangan. Penentuan konsep, desain aplikasi dan pembuatan *website* ini menggunakan *Macromedia Dreamweaver 8* sebagai *software* utama dan *XAMPP 1.6.4*, *Navicat Lite 8.2*, *Adobe Photosop CS*, *CorelDraw 11* dan *Macromedia Flash 8* sebagai *software* pendukung.

Arsad (2011). Dalam penelitiannya yang berjudul “Perancangan Sistem Informasi Pariwisata Berbasis Web Di Dinas Kebudayaan Dan Pariwisata Kabupaten Muna”. Tujuan perancangan sistem informasi web ini adalah untuk mempromosikan wisata kabupaten Muna Sulawesi Tenggara. Sistem informasi berbasis web adalah media yang memiliki unsur teks, gambar animasi, dan suara. Unsur-unsur tersebut menjadikan media ini menarik. Selain itu sistem informasi berbasis web menjadi sebuah media yang memiliki informasi yang cukup lengkap

bagi wisatawan dan dapat dilihat setiap saat. Media ini tidak lepas dari aspek desain dan teknologi di dalam proses pembuatannya. Seperti media informasi lain, keberhasilan media ini juga ditentukan oleh keberhasilan penyampaian pesan yang dibawanya. Metode penelitian yang dilakukan adalah dengan pengumpulan data melalui studi literatur, wawancara, dan observasi, serta menganalisis kebutuhan perangkat lunak, membangun *database* dengan *mysql*, merancang antar muka menggunakan *Macromedia Dreamweaver* dan PHP, melakukan pengujian program sebagai tahap akhir dalam pembuatan sistem informasi pariwisata berbasis web.

Rizal Aiyubi (2012). Dalam penelitiannya yang berjudul “Perancangan *Website* Pariwisata Berbasis Visual Animasi Di Dinas Kebudayaan Dan Pariwisata Aceh Besar”. Saat ini, Dinas Kebudayaan, Pariwisata, Pemuda dan Olah Raga Kabupaten Aceh Besar masih menggunakan *Blogger* dalam mempublikasi Objek Wisata yang ada di kawasan Aceh Besar. Tugas akhir ini bertujuan untuk membuat suatu *website* Pariwisata Berbasis Visual Animasi di Dinas Kebudayaan dan Pariwisata Aceh Besar yang dinamis. pembuatan *website* Pariwisata menggunakan *software Adobe flash CS3* dan *Macromedia Dreamweaver*, maka *website* Pariwisata berbasis Visual Animasi akan lebih praktis dan mudah.

Fransiskus Lesomar (2015). Dalam penelitiannya yang berjudul “Rancang Bangun Portal Web Pariwisata Maluku Tenggara”. Penelitian ini dilakukan untuk mempromosikan kuliner dan tempat-tempat pariwisata baik alam maupun wisata rohani yang ada di Maluku Tenggara. *Variable* yang akan disoroti dalam penelitian ini adalah strategi promosi pariwisata. Strategi yang akan diterapkan dalam mempromosikan potensi pariwisata Maluku Tenggara adalah dengan merancang

sebuah Portal Web Pariwisata. Metode pengumpulan data dilakukan dengan wawancara, observasi lapangan, dan study pustaka. Hasil yang didapatkan berupa data-data yang dibutuhkan dalam merancang dan membangun sebuah Portal Web Pariwisata. Data yang diperoleh kemudian akan diolah menggunakan metode Rapid *Application Development* (RAD) dengan 3 tahapan yaitu Perencanaan syarat-syarat (*Requirements Planning*), *Workshop* Desain RAD (*RAD Design Workshop*), dan Implementasi (*Implementation*).

Berdasarkan empat penelitian yang menjadi tinjauan dalam melakukan penelitian. Peneliti akan membangun suatu sistem di mana aplikasi berbasis web yang nantinya tidak hanya menjadi media promosi dan sosialisasi namun juga menjadi media yang dapat membantu para turis untuk merancang rencana berlibur yang membuat para turis mudah dalam melakukan kegiatan *tourism*. Penggunaan Bahasa pemrograman PHP dan *xampp* sebagai web server yang lebih umum digunakan dapat membuat peneliti memiliki banyak referensi sehingga memutuskan untuk menggunakan Bahasa pemrograman PHP. Metode penelitian yang dilakukan oleh empat peneliti di atas menggunakan wawancara untuk memperoleh data dalam membangun aplikasi, untuk itu peneliti juga akan menggunakan metode wawancara namun ditambahkan dengan kuesioner atau angket untuk memperoleh data yang lebih banyak sehingga peneliti memiliki banyak informasi yang dapat dijadikan referensi nantinya.

Seiring perkembangan teknologi informasi dan komunikasi, para turis sebutan untuk pelaku *tourism* memanfaatkan internet untuk membuat rencana *tourism* sendiri. Penggunaan Internet sebagai media promosi tempat wisata juga

didukung oleh aktivitas para turis yang memanfaatkan Internet sebagai sumber informasi utama mengenai tempat tujuan wisata (Buhalis, 2011).

Kegiatan para turis yang memanfaatkan situs-situs pariwisata untuk kegiatan perencanaan *tourism* dinilai tidaklah mudah karena mereka harus mengkombinasikan informasi dari banyak sumber dan memilih informasi mana yang lebih terpercaya (Vansteenwegen & Souffriau, 2011).

Dalam menyiapkan diri sebelum *tourism*, turis membutuhkan informasi yang lengkap dan terpercaya mengenai tempat tujuan *tourism*. Tanpa informasi yang lengkap dan terpercaya, para turis tidak bisa benar-benar membuat rencana program *tourism* (Egger, 2008).

Selain itu, informasi dalam situs-situs pariwisata masih belum terintegrasi dengan informasi dari komponen-komponen pendukung pariwisata seperti perusahaan penerbangan, pelayaran, penginapan, asuransi, dan lain-lain (Putera, 2010).

Untuk mengatasi masalah tersebut maka penelitian ini bermaksud untuk membuat *business logic* suatu *website* menggunakan bahasa pemrograman web PHP dan *framework codeIgniter* yang dapat diimplementasikan untuk pembangunan suatu *website* perencanaan program *tourism*.

Dengan *website* tersebut, para turis akan lebih mudah memahami tempat tujuan dan merancang rencana program *tourism* mereka. Di samping itu, dengan menyajikan semua informasi seperti penginapan dan transportasi dalam satu situs, para turis tidak perlu repot mengunjungi banyak situs demi memperoleh informasi lengkap.

## **2.2 Landasan Teori**

Untuk mendukung hasil penelitian diperlukan suatu konsep dalam merumuskan definisi-definisi yang menunjang kegiatan penelitian baik teori dasar maupun teori umum.

### **2.2.1 Web Browser**

*Web Browser* adalah aplikasi yang mampu menjalankan dokumen-dokumen *web* dengan cara diterjemahkan. Prosesnya dilakukan oleh komponen yang terdapat di dalam aplikasi *browser* yang biasa disebut *Web Engine*. Semua dokumen *web* ditampilkan oleh *browser* dengan cara diterjemahkan (Arief, 2011). Beberapa jenis *browser* yang populer saat ini di antaranya adalah *Internet Explorer* yang diproduksi oleh *Microsoft*, *Mozilla Firefox*, *Opera*, dan *Safari* yang diproduksi oleh *Apple*.

### **2.2.2 Web server**

*Web server* adalah sebuah komputer yang terdiri dari perangkat keras dan perangkat lunak. Secara bentuk fisik dan cara kerjanya, perangkat keras *web server* dengan PC dibedakan oleh kapasitas dan kapabilitasnya. Perangkat lunak dalam *web server* memiliki karakteristik dan teknologi yang digunakan untuk mengatur kerja sistemnya (Sibero, 2011).

### **2.2.3 HTML (Hypertext Markup Language)**

HTML atau dengan singkatan *Hyper Text Markup Language* adalah bahasa pemrograman yang digunakan dalam *web* sebagai bahasa untuk pertukaran dokumen *web* (Sibero, 2011).

#### **2.2.4 CSS (Cascading Style Sheet)**

*Cascading Style Sheet* (CSS) merupakan salah satu bahasa pemrograman *web* untuk mengendalikan beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam (Slamet, 2009).

CSS adalah bahasa *style sheet* yang digunakan untuk memformat tampilan *web* yang dibuat dengan bahasa HTML dan XHTML. CSS mampu mengatur warna *body* teks, ukuran gambar, ukuran *border*, warna *mouse over*, warna tabel, warna *hyperlink*, margin, spasi *paragraph*, spasi teks dan parameter lainnya. Dengan adanya CSS memungkinkan menampilkan halaman yang sama dengan format yang berbeda (Slamet, 2009).

#### **2.2.5 Javascript**

*Javascript* adalah bahasa pemrograman yang sederhana karena bahasa ini tidak dapat digunakan untuk membuat aplikasi ataupun *applet*. Dengan *javascript* kita dapat dengan mudah membuat sebuah halaman *web* yang interaktif (Hardjono, 2006).

#### **2.2.6 JQuery**

*JQuery* adalah *Open Source add-on* pustaka *JavaScript* yang menekankan pada interaksi antara *JavaScript* dan HTML (Beighly, 2010). *JQuery* merupakan kode *JavaScript* yang telah ditulis, dan tinggal menambahkan satu atau dua *coding* untuk memanggil *coding JQuery*. *JQuery* banyak digunakan sebagai aplikasi untuk mempercantik tampilan suatu aplikasi *web*.

*JQuery* memiliki banyak keuntungan, di antara lain adalah :

1. Mudah digunakan dan dipelajari.



2. Ringkas dan ringan.
3. Tidak perlu me-*reload* 1 halaman untuk dijalankan.
4. *Reusability* yang tinggi.
5. Lisensi bebas.

### 2.2.7 PHP (PHP Hypertext Preprocessor)

*PHP (Perl Hypertext Preprocessor)* adalah bahasa server-side-scripting yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis (Arief, 2011). Dengan menggunakan program PHP, sebuah *website* akan lebih interaktif dan dinamis. PHP adalah singkatan dari *Hypertext Preprocessor*. PHP disebut sebagai *Server Side Programming* dikarenakan seluruh prosesnya dijalankan pada server. *Script* PHP yang dibuat harus tersimpan dalam sebuah server dan dieksekusi atau diproses dalam server tersebut. Dengan menggunakan PHP, sebuah halaman *web* akan lebih interaktif dan dinamis.

PHP tidak dapat dijalankan tanpa menggunakan *software* pendukung yang bekerja di komputer lokal. *Software* pendukung yang harus di instalasi, yaitu *Apache*, PHP, *phpMyAdmin*, dan *database* MySQL. *Software* pendukung ini terdapat dalam sebuah paket *software*, seperti *Xampp* dan *Appserver*.

Adapun kelebihan-kelebihan dari PHP yaitu:

1. PHP merupakan sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya. Tidak seperti halnya bahasa pemrograman aplikasi yang lainnya.
2. PHP dapat berjalan pada *web server* yang dirilis oleh Microsoft, seperti IIS atau PWS juga pada *apache* yang bersifat *open source*.

3. Karena sifatnya yang *open source*, maka perubahan dan perkembangan interpreter pada PHP lebih cepat dan mudah, karena banyak *developer* yang siap membantu pengembangannya.
4. Jika dilihat dari segi pemahaman, PHP memiliki referensi yang begitu banyak sehingga sangat mudah untuk dipahami.
5. PHP dapat berjalan pada 3 sistem operasi, yaitu: *Linux*, *unix*, dan *windows*, dan juga dapat dijalankan secara *runtime* pada suatu *console*.

### 2.2.8 CodeIgniter

Menurut situs resmi *CodeIgniter* (diakses pada tanggal 11 Februari 2016 di <https://www.codeigniter.com>) *CodeIgniter* adalah *framework* PHP yang baik dengan *footprint* yang sangat kecil, dibangun untuk pengembang yang membutuhkan *toolkit* sederhana dan elegan untuk membuat aplikasi *web* dengan fitur lengkap.

Fitur-fitur yang didukung oleh *CodeIgniter* antara lain :

- Sistem berbasis *Model View Controller*
- *Compatible* dengan PHP versi 4
- Ringan dan cepat
- Terdapat dukungan untuk berbagai *database*, dll

### 2.2.9 MVC

MVC merupakan metode pengembangan sistem yang dibagi atas komponen *Model*, *View*, *Controller* secara terstruktur. Salah satu penggunaan metode MVC adalah *framework CodeIgniter* pada PHP.

a. Model

*Model* merupakan komponen yang berinteraksi langsung dengan *database*. Sistem-sistem seperti tampilan, penambahan, perubahan, dan penghapusan data terletak dalam komponen model.

b. View

*View* merupakan komponen yang berinteraksi langsung dengan *user* atau pengguna. Sistem-sistem seperti menampilkan hasil dari permintaan pada komponen *controller* yang berinteraksi langsung dengan komponen model.

c. Controller

*Controller* merupakan komponen penghubung antara *view* dan model. *Controller* berfungsi sebagai pengambilan *input* dari *view* dan mengirim *input* tersebut ke dalam komponen model.

### **2.2.10 XAMPP**

*Xampp* merupakan salah satu paket *web server* yang umumnya digunakan oleh *web developer*. *Xampp* berisi antara lain server *web Apache*, *intrepreter PHP*, dan *database MySQL*.

### **2.2.11 Notepad++**

Menurut situs resmi *Notepad++* (diakses pada tanggal 11 Februari 2016 di <https://notepad-plus-plus.org>) *Notepad ++* adalah *source code editor* gratis dan pengganti *Notepad* yang mendukung beberapa bahasa . Berjalan di lingkungan MS *Windows*, penggunaannya diatur oleh *GPL License* .

Berdasarkan *powerful editing component Scintilla*, Notepad++ ditulis dalam C++ dan menggunakan API Win32 murni dan STL yang menjamin kecepatan eksekusi lebih tinggi dan ukuran program yang lebih kecil . Dengan mengoptimalkan sebagai rutinitas sebanyak mungkin tanpa kehilangan keramahan pengguna, penggunaan *Notepad++* dikarenakan *notepad++* menjamin kecepatan eksekusi lebih tinggi dan ukuran program yang lebih kecil dan juga menggunakan CPU *power* yang minim, sehingga mengurangi konsumsi daya berlebih.

## **2.2.12 Metode Penelitian**

### **A. Metode Pengumpulan Data**

Metode yang digunakan untuk pengumpulan data pada penelitian ini adalah:

#### 1. Kuesioner (Angket)

Kuesioner atau angket hanya berbeda dalam bentuknya. Pada kuesioner, pertanyaan disusun dalam bentuk kalimat tanya, sedangkan pada angket pertanyaan disusun dalam kalimat pernyataan dengan opsi jawaban yang tersedia. Kalau metode pengamatan dan metode wawancara menempatkan peneliti dalam hubungan langsung dengan responden, maka dalam metode angket hubungan itu dilakukan melalui media, yaitu daftar pertanyaan yang dikirim kepada responden.

Dalam penelitian ini kuesioner yang dilakukan yaitu lebih cenderung angket karena disajikan beberapa pertanyaan berikut jawaban yang dapat dipilih oleh responden. Adapun responden berasal dari masyarakat yang sudah terbiasa membuat rencana *tourism* melalui internet. Sehingga data yang diperoleh dapat membantu dalam merancang dan membuat *bussines logic* dari *website* perencanaan program *tourism*.

## 2. Wawancara

Wawancara dilakukan untuk mendapatkan data dan informasi dari orang-orang yang membuat rencana terlebih dahulu sebelum mereka pergi untuk *tourism*. Wawancara bermaksud untuk mengajukan beberapa pertanyaan langsung untuk mengetahui masalah-masalah yang diperoleh turis ketika ingin membuat rencana *tourism*.

### **B. Metode Pengembangan Sistem**

Dalam Metode pengembangan sistem penulis menggunakan metode SDLC (*Sistem Development Life Cycle*). Fase pengembangan sistem aplikasi ini disebut juga sebagai siklus hidup pengembangan sistem informasi, dengan tahapan-tahapan di antaranya :

#### a. Analisis

Fase analisis menanyakan dan menjawab pertanyaan-pertanyaan penting seperti siapa para pengguna sistem, apa yang akan dicapai oleh sistem, dan di mana serta kapan sistem akan dijalankan.

#### b. Perancangan (*Design*)

Fase perancangan menandai bagaimana sistem akan bekerja, mempertimbangkan semua detail perangkat keras, perangkat lunak, infrastruktur jaringan, antarmuka pengguna, *form*, *display*, program, laporan, *database*, dan *file* ditetapkan.

#### c. Implementasi Sistem

Tahapan-tahapan yang dilakukan dalam implementasi sistem yaitu tahap pemrograman (*coding*).

d. Pengujian (*Testing*)

Fase pengujian berhubungan dengan pemilihan perangkat keras, penyusunan perangkat lunak aplikasi (*coding*), serta pengujian apakah sistem yang dibuat sudah sesuai dengan kebutuhan pengguna atau belum.





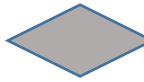
e. Perawatan




Fase perawatan adalah tahap melakukan operasi dan pemeliharaan aplikasi yang telah dibuat dengan melakukan perbaikan-perbaikan terhadap masalah yang ada.

### 2.2.13 Flowchart

Bagan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika (Jogiyanto, 2005). Bagan alir program (*program flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. Simbol *flowchart* yang umum digunakan dapat dilihat pada tabel 2.1.

**Tabel 2.1** Simbol *Flowchart*

Gambar	Simbol Untuk	Keterangan
	Menunjukkan awal atau akhir sebuah proses	Digunakan untuk menunjukkan awal dan juga akhir sebuah proses.
	Proses oleh komputer	Menyatakan kegiatan pengolahan yang dilakukan oleh komputer.
	Proses tanpa komputer	Menyatakan kegiatan pengolahan yang tidak dilakukan oleh komputer.
	Menunjukkan keberadaan dokumen	Digunakan untuk menunjukkan proses atau keberadaan dokumen.
		Proses / Langkah di mana perlu adanya keputusan atau adanya kondisi tertentu. Di titik ini selalu

Gambar	Simbol Untuk	Keterangan
	Titik keputusan	ada dua keluaran untuk melanjutkan aliran kondisi yang berbeda.
	Masukan / Keluaran Data	Digunakan untuk mewakili data masuk, atau data keluar ( <i>Input</i> dan <i>Output</i> ).
	Tampilan di komputer	Digunakan untuk menampilkan informasi yang ke layar komputer.
	Garis Alir	Digunakan untuk menunjukkan arah aliran dari satu proses ke proses yang lain.

### 2.2.14 UML

*Unified Modeling Language* (UML) adalah sebuah gambar yang berisikan data-data yang dapat membantu menjelaskan atau mendesain sebuah aplikasi yang sedang dibuat (Fowler, 2004).

*Models* digunakan oleh *developer* profesional untuk berkomunikasi antara pekerjaan mereka dengan *project stakeholders* dan dengan *developer* lainnya (Ambler, 2005). *Unified Modeling Language* (UML) menjadi bagian penting dalam pembuatan *software* sejak diperkenalkan pada tahun 1997.

UML termasuk sebuah set diagram konsisten yang digunakan untuk menjelaskan dan komunikasi kebutuhan sistem *software*, desain, dan kode. UML yang biasa digunakan selama kegiatan antara lain:

#### A. Use Case Diagram

*Use case* diagram memperlihatkan hubungan antara aktor dan *use case* dalam sebuah sistem (Ambler, 2005).

*Use case* sering digunakan untuk :

- a. Memberikan gambaran tentang semua atau sebagian dari persyaratan penggunaan untuk sistem atau organisasi dalam bentuk model penting.
- b. Mengkomunikasikan lingkup sebuah proyek yang dibangun.

*Use case* model terdiri dari satu atau banyak *use case* diagram dan didukung oleh dokumentasi seperti spesifikasi *use case* dan definisi aktor.

*Use case* memiliki dua tingkat *use case* diagram, yaitu :

- a. *User level*, yang menggambarkan bagaimana *user* berinteraksi dengan sistem.
- b. *Developer level*, yang menggambarkan bagaimana komponen sistem akan berinteraksi

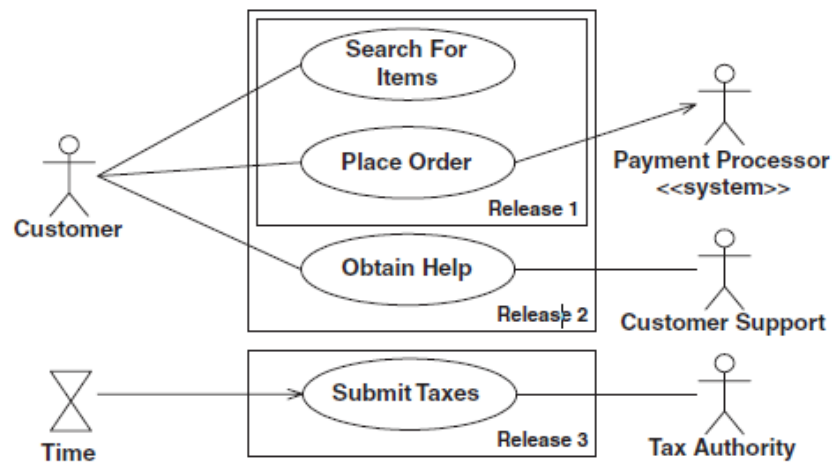
*User level* dalam *use case* ditempatkan pada kebutuhan sistem sedangkan *developer level* menyediakan kebutuhan untuk sub sistem.

*Use case* digunakan untuk memodelkan sistem/sub sistem dari aplikasi. Sebuah *use case* diagram menangkap fungsi tertentu dari sistem.

*Use case* diagram membutuhkan :

- a. Aktor, yang berfungsi sebagai yang menjalankan sistem.
- b. *Use case*, yang berfungsi sebagai kegiatan yang akan dijalankan oleh *actor*.
- c. Garis penghubung yang menghubungkan aktor dengan *use case* yang dijalankan.





**Gambar 2. 1** Use Case Diagram menurut Scott W. Ambler

## B. Activity Diagram

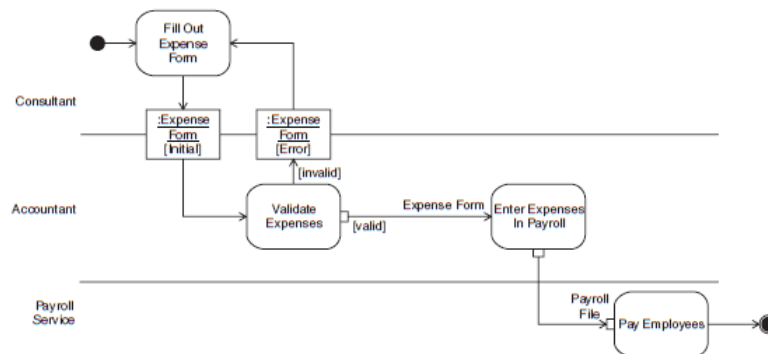
Activity diagram setara dengan berorientasi objek seperti *flowchart* dan diagram aliran data (Ambler, 2005).

Activity diagram digunakan untuk mengeksplorasi logika seperti:

- d. Operasi yang kompleks.
- e. Aturan bisnis yang kompleks.
- f. Use case tunggal.
- g. Use case yang banyak.
- h. Proses bisnis.
- i. Proses yang bersamaan.
- j. Proses perangkat lunak.

Activity diagram pada dasarnya adalah *flowchart* untuk mewakili bentuk aliran satu aktivitas dengan aktivitas lainnya.

Activity diagram dapat digambarkan sebagai operasi dari sistem.



**Gambar 2. 2** Activity Diagram menurut Scott W.Ambler

### C. Class Diagram

Class diagram menampilkan semua kelas yang terdapat dalam sistem, hubungan mereka, dan operasi dan atribut dari class (Ambler, 2005).

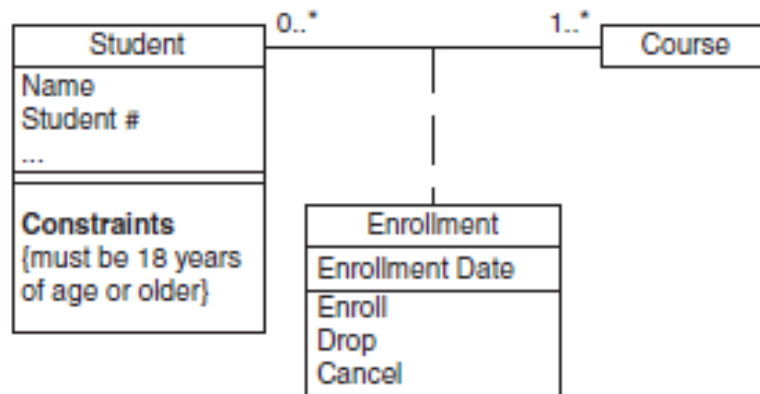
Class diagram sering digunakan untuk :

- Menjelaskan konsep domain dalam bentuk model.
- Menganalisis persyaratan dalam bentuk model analisis dan konseptual.
- Menggambarkan desain secara rinci berorientasi objek atau objek berbasis *software*.

Class diagram merupakan diagram statis. Class diagram mewakili pandangan statis dari aplikasi. Class diagram tidak hanya digunakan untuk memvisualisasikan, menggambarkan, dan mendokumentasikan perbedaan aspek dari sistem, tetapi juga untuk membangun kode *executable* dari aplikasi *software*.

Class diagram menampilkan koleksi dari class, interface, associations, collaborations, dan constraints. Hal ini juga dikenal sebagai diagram struktural.

*Class* diagram memberikan pandangan dari kelas desain dari sistem. *Class* diagram mengandung representasi dari kelas dan paket dan bagaimana mereka berhubungan.



**Gambar 2. 3** *Class Diagram* menurut Scott W.Ambler

## **BAB III**

### **PERANCANGAN PENELITIAN**

#### **3.1 Peralatan Pendukung**

Peralatan pendukung dalam pembuatan aplikasi berbasis *website* terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*) untuk mendukung berjalannya perancangan dan pembuatan.

##### **3.1.1 Perangkat Keras**

Adapun perangkat keras yang dibutuhkan dalam merancang dan membuat *bussines logic* dari *website* perencanaan program *tourism*:

1. *Processor intel core i5*
2. *Memory 8GB.*
3. *Hardisk minimal 250 GB.*
4. *Monitor, Keyboard dan Mouse.*

##### **3.1.2 Perangkat Lunak**

Adapun perangkat lunak yang dibutuhkan dalam merancang dan membuat *bussines logic* dari *website* perencanaan program *tourism*:

1. *Sistem operasi windows 10*
2. *Web Server : XAMPP*
3. *Web Browser.*
4. *Bahasa Pemrograman : PHP, Javascript, HTML, CSS*
5. *Database Server : MySQL*

### 3.2 Analisis Kebutuhan

Analisis kebutuhan didapat dari Angket dan wawancara. Berikut merupakan analisis kebutuhan pada aplikasi:

- a. Halaman Akun untuk *member*.
- b. Halaman Wisata untuk melihat daftar tempat wisata.
- c. Halaman Detail Wisata untuk melihat informasi detail tempat wisata.
- d. Halaman Hotel untuk melihat daftar hotel.
- e. Halaman Detail Hotel untuk melihat informasi detail hotel.
- f. Halaman Transportasi untuk melihat daftar Transportasi.
- g. Halaman *Login* untuk *member*.
- h. Halaman Beranda.
- i. Halaman Registrasi untuk non *member* melakukan registrasi.
- j. Halaman Buat Rencana untuk membuat rencana berlibur.
- k. Halaman Detail Rencana untuk menampilkan rencana berlibur yang telah dibuat.
- l. Halaman Riwayat Rencana untuk *member* mengelola rencana berlibur yang telah dibuat.
- m. Fitur *Map* untuk menampilkan letak koordinat dari tempat wisata dan juga hotel.
- n. Fitur *Print* untuk mencetak rencana berlibur yang telah dibuat.
- o. Fitur *Likes* untuk mengetahui tempat wisata paling favorit.

### 3.3 Rancangan

Dalam perancangan *Logic* Aplikasi ini menggunakan metode Bagan Alir atau *Flowchart* dan *United Markup Language* (UML). Model UML yang digunakan dalam pengembangan aplikasi yaitu *Use case* diagram, *Activity* diagram, dan *Class* Diagram.

#### 3.3.1 Flowchart

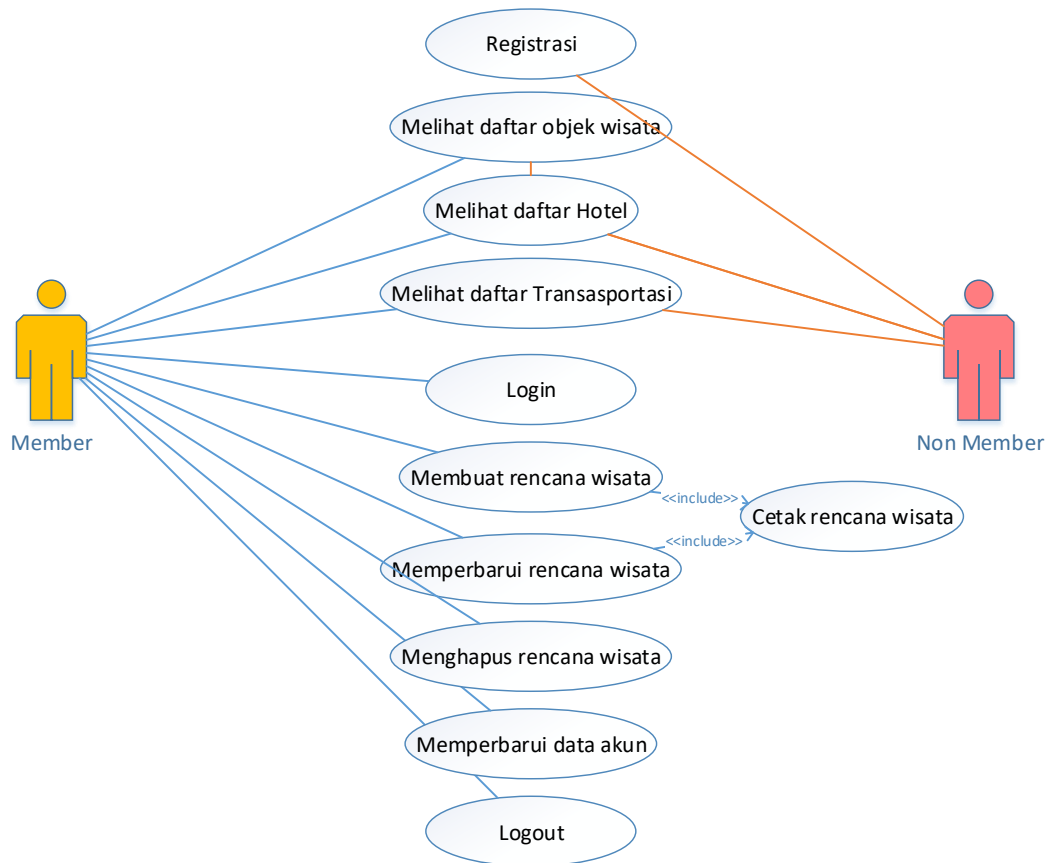
Berdasarkan analisis kebutuhan dibuatlah *flowchart* untuk membantu perancangan *logic* aplikasi. Berikut penjelasan *Flowchart* tentang Gambar 3.6:

1. Member harus *login* terlebih dahulu agar dapat menggunakan fitur halaman buat rencana. Jika *login* sukses maka akan muncul halaman akun. Namun jika terjadi kesalahan dalam proses *login*, maka akan kembali ke halaman *login*.
2. Ketika *member* selesai *login* maka akan tampil halaman akun. Di halaman ini *member* dapat melanjutkan ke halaman yang sesuai dengan kebutuhannya.
3. Jika *member* memutuskan untuk membuat rencana berlibur maka *member* dapat menuju halaman buat rencana, kemudian memilih tempat wisata, setelah itu akan tampil daftar tempat wisata dan juga koordinat tempat wisata yang telah dipilih. Ketika *member* menyimpan rencana berliburnya maka akan tampil detail rencana berlibur yang telah dibuat. Kemudian *member* dapat memutuskan untuk mencetak atau tidak rencana berlibur tersebut.
4. Jika *member* memutuskan untuk memperbarui rencana berlibur maka *member* dapat langsung menuju ke halaman riwayat wisata, kemudian memilih rencana berlibur yang ingin diperbarui. kemudian tampil halaman detail rencana. proses selanjutnya sama ketika *user* akan membuat rencana baru.



### 3.3.2 Use Case Diagram

Berdasarkan analisis kebutuhan dibuatlah *use case* diagram untuk membantu perancangan *logic* aplikasi. Terdapat 2 aktor yaitu *member* dan *non member*. Gambaran *Use Case* diagram yang digunakan dalam aplikasi dapat dilihat pada gambar 3.2.



**Gambar 3.2** *Use Case Diagram* Aplikasi

Berikut penjelasan tentang *use case* diagram pada Gambar 3.2:

- Registrasi : Memungkinkan *non member* untuk melakukan registrasi agar menjadi *member*.
- Melihat Daftar objek wisata : Memungkinkan *member* untuk melihat kumpulan daftar objek wisata yang dapat dikunjungi.



- c. Melihat Daftar hotel : Memungkinkan *member* untuk melihat kumpulan daftar hotel.
- d. Melihat Daftar transportasi : Memungkinkan *member* untuk melihat kumpulan daftar transportasi yang dapat di pesan.
- e. *Login* : Memungkinkan *member* untuk dapat menggunakan halaman buat rencana.
- f. Membuat rencana wisata : Memungkinkan *member* untuk membuat rencana berlibur.
- g. Memperbarui rencana wisata : Memungkinkan *member* untuk memperbarui rencana berlibur yang sebelumnya telah dibuat.
- h. Menghapus rencana wisata : Memungkinkan *member* untuk menghapus rencana berlibur yang sebelumnya telah dibuat.
- i. Cetak rencana wisata : Memungkinkan *member* untuk mencetak rencana berlibur.
- j. Memperbarui data akun : Memungkinkan *member* untuk memperbarui data akun seperti data diri, *username*, dan *password*.
- k. *Logout* : Memungkinkan *member* untuk tidak lagi menggunakan fitur halaman buat rencana.

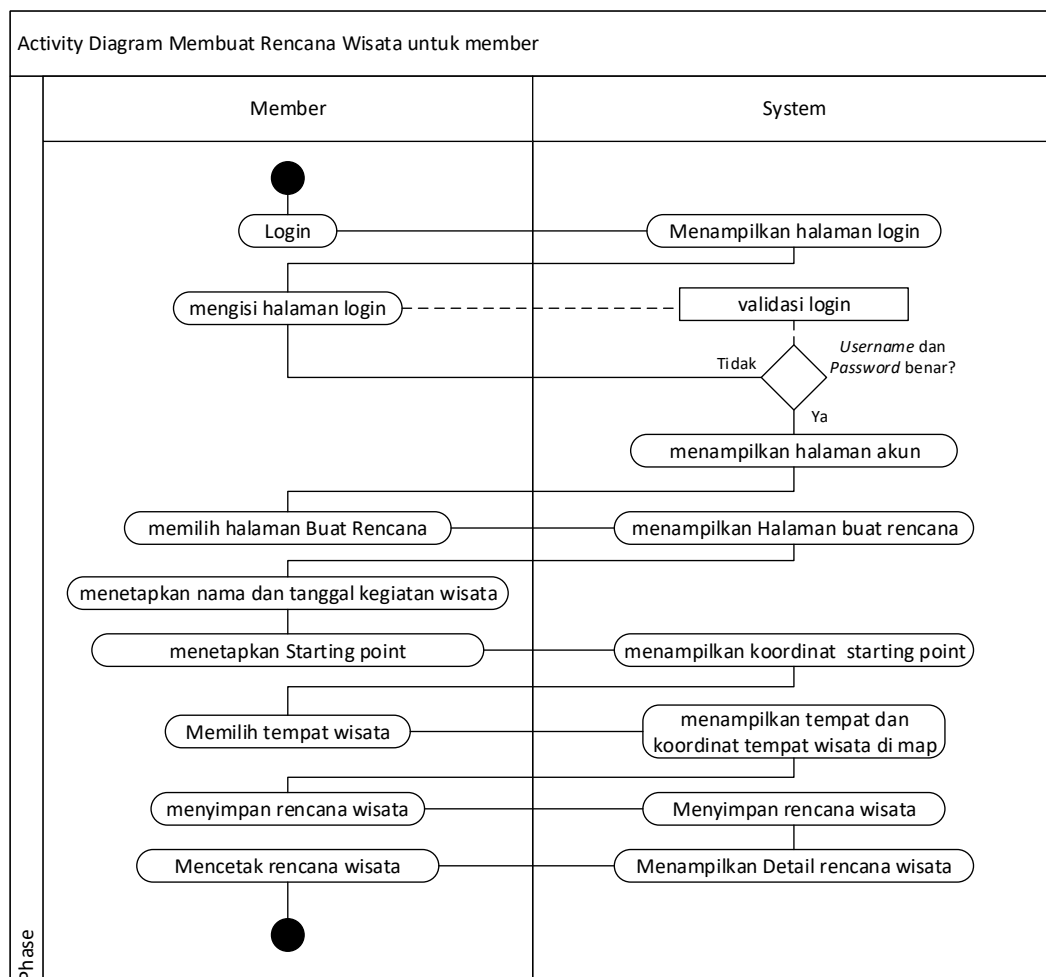
### 3.3.3 Activity Diagram

Berdasarkan *use case* yang telah dibuat sebelumnya maka dapat diperoleh *activity diagram* berdasarkan aktor yang terlibat dalam *usecase diagram*. *Activity diagram* dalam aplikasi dibagi menjadi dua bagian yaitu *activity diagram* buat

rencana wisata oleh *member* dan *activity diagram* buat rencana wisata oleh non *member*.

### A. Activity Diagram Buat Rencana Wisata Member

Gambaran *Activity Diagram* buat rencana *member* yang digunakan dalam aplikasi dapat dilihat pada Gambar 3.3.



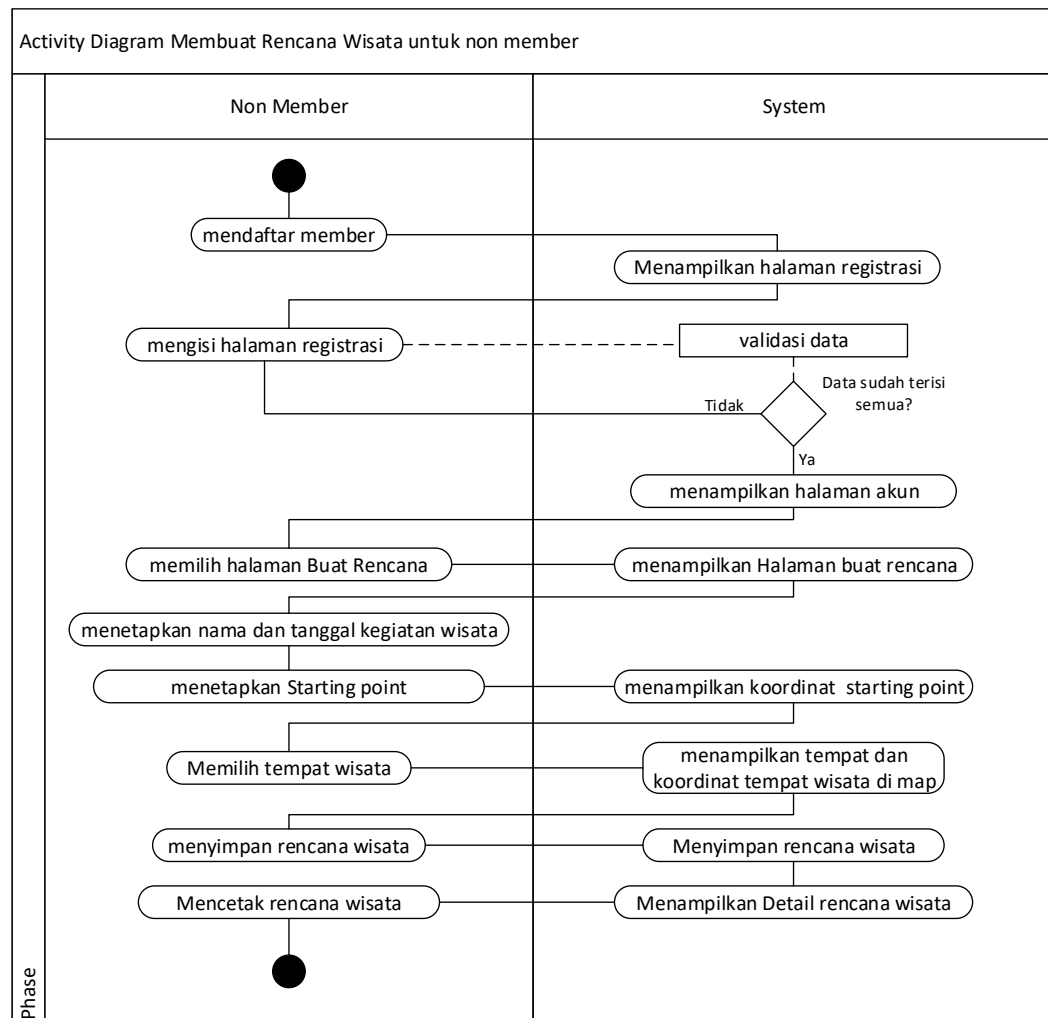
**Gambar 3.3** *Activity Diagram* Buat Rencana *Member*

Berikut Penjelasan *Activity Diagram* pada Gambar 3.3:

1. Alur dari kegiatan membuat rencana yang dilakukan oleh *member*, yaitu masuk ke halaman *login* terlebih dahulu, kemudian *system* akan menampilkan halaman *login*.
2. Member mengisi *username* dan *password*, kemudian *system* akan melakukan validasi *username* dan *password*. Ketika data yang di masukan benar maka akan muncul halaman akun. Namun apabila *login* gagal maka *member* akan diarahkan kembali ke halaman *login*.
3. Member memilih halaman buat rencana. Kemudian menetapkan nama dan tanggal kegiatan wisata. Member harus terlebih dahulu memilih *starting point* dari rencana berliburnya, kemudian *system* akan menampilkan koordinat *starting point*.
4. Member memilih objek-objek wisata yang ingin dikunjungi. Kemudian *system* akan menampilkan objek wisata yang telah dipilih beserta koordinatnya di peta.
5. Member kemudian menyimpan rencana berlibur ke *database*. Kemudian *system* akan menampilkan detail perjalanan wisata yang telah dipilih oleh *member*.
6. Member dapat mencetak rencana berlibur.

## B. Activity Diagram Buat Rencana Wisata Non Member

Gambaran *Activity Diagram* buat rencana *member* yang digunakan dalam aplikasi dapat dilihat pada Gambar 3.4.



**Gambar 3.4** *Activity Diagram* Buat Rencana *Non Member*

Berikut Penjelasan *Activity Diagram* pada Gambar 3.4:

1. Alur dari kegiatan membuat rencana yang dilakukan oleh non *member*, yaitu harus mendaftar *member* dengan masuk ke halaman registrasi. Kemudian mengisi data diri lengkap di halaman registrasi. Jika semua data yang di masukan sudah lengkap *system* akan mengarahkan ke halaman akun.

2. Kemudian memilih halaman buat rencana. menetapkan nama dan tanggal kegiatan wisata. terlebih dahulu memilih *starting point* dari rencana berliburnya, kemudian *system* akan menampilkan koordinat *starting point*.
3. Kemudian memilih objek-objek wisata yang ingin dikunjungi. *system* akan menampilkan objek wisata yang telah dipilih beserta koordinatnya di peta.
4. Kemudian menyimpan rencana berlibur ke *database*. *system* akan menampilkan detail perjalanan wisata yang telah dipilih oleh.
5. Mencetak rencana berlibur.

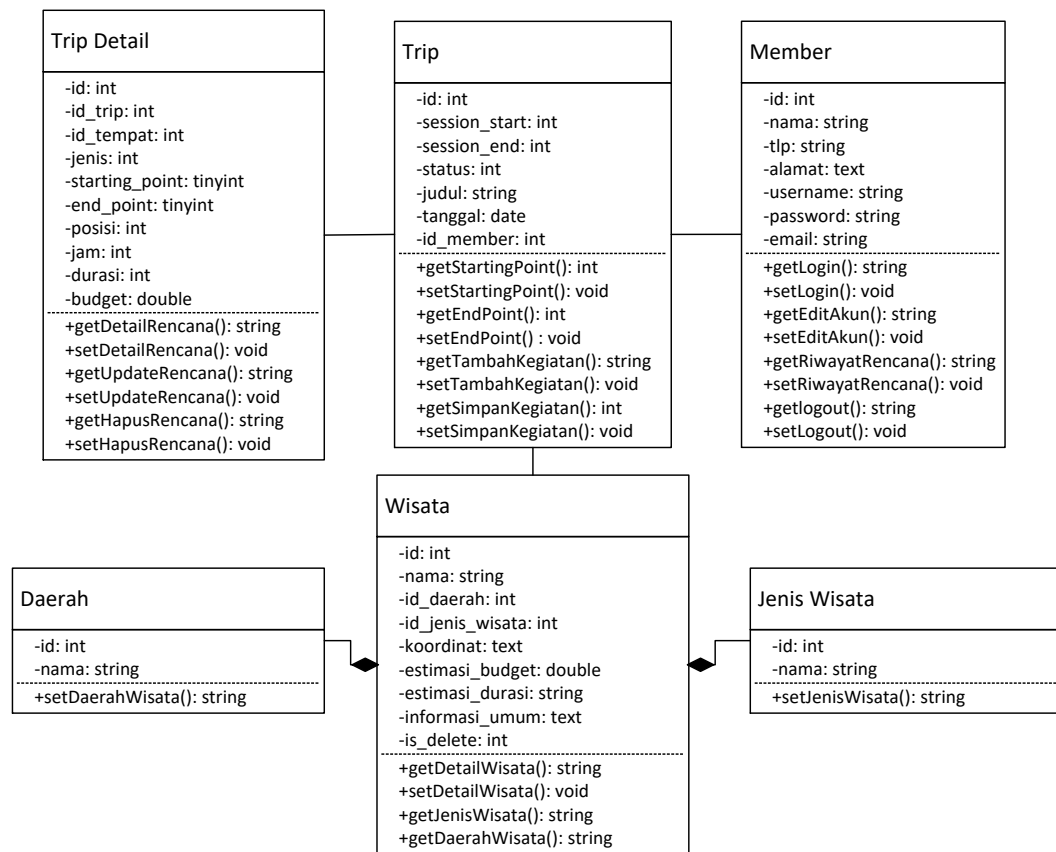
### 3.3.4 Class Diagram

*Class Diagram* yang digunakan dapat dilihat pada Gambar 3.5.

Berikut adalah penjelasan *class diagram* pada gambar 3.5:

1. Pada *class* Member terdapat fungsi *login* untuk *member* sehingga dapat menggunakan fitur halaman buat rencana. Terdapat fungsi edit akun untuk memperbarui data. Member juga dapat melihat riwayat rencana berlibur yang telah dibuat. Dan terakhir fungsi *logout*.
2. Pada *class* Trip terdapat fungsi untuk menetapkan *starting point* serta menambahkan tempat kegiatan wisata. Kemudian fungsi untuk menyimpan rencana berlibur.
3. Pada *class* Trip Detail, setelah menyimpan rencana berlibur *member* dapat melihat detail informasi dari rencana berlibur yang telah dibuat. Pada *class* ini juga terdapat fungsi untuk melakukan *update* rencana dan hapus rencana.

4. Pada *class* Wisata terdapat pilihan objek-objek wisata. Dan juga fungsi untuk menampilkan detail informasi dari objek wisata. *Class* ini juga mengambil data dari *class* daerah dan jenis wisata.
5. Pada *class* Daerah berfungsi menyimpan data daerah-daerah tempat wisata.
6. Pada *class* Jenis wisata berfungsi menyimpan data jenis-jenis wisata.



**Gambar 3.5** *Class Diagram* Aplikasi

Kelas Member memiliki *association* dengan kelas Trip. Member dapat membuat rencana berlibur dengan cara menetapkan terlebih dahulu *starting point* dari kegiatan wisata dengan menggunakan *method* *getStartingPoint()*. Kemudian memilih objek-objek wisata dengan *method* *getTambahKegiatan()*. Kegiatan yang terakhir dipilih akan otomatis menjadi *end point* dari kegiatan wisata dengan

*method getEndPoint()*. Untuk menyimpan rencana berlibur yaitu menggunakan *method getSimpanKegiatan()*.

Kelas Trip memiliki *association* dengan kelas Trip Detail. Rencana berlibur yang telah disimpan dapat dilihat detailnya pada kelas trip detail dengan *method getDetailRencana()*. pada kelas trip detail juga terdapat *method getUpdateRencana()* untuk memperbarui rencana dan *method getUpdateRencana()* untuk menghapus rencana yang telah dibuat.

Kelas Trip memiliki *association* dengan kelas Wisata di mana *member* dapat memilih objek wisata yang ingin ditambahkan ke bagian dari rencana berlibur. Pada kelas wisata terdapat *method getDetailWisata()* untuk menampilkan detail informasi objek wisata.

Kelas Wisata memiliki *composition* dengan kelas Daerah dan Jenis wisata, artinya kelas Daerah dan Jenis wisata merupakan bagian dari kelas Wisata. Kelas Daerah dan Jenis wisata tidak dapat berdiri sendiri apabila kelas Wisata tidak ada.

### **3.4 Metode Pengujian**

Pengujian perangkat lunak merupakan suatu kegiatan yang dilakukan untuk memperoleh informasi serta mengevaluasi kualitas dari produk atau layanan yang sedang diuji. Tujuan pengujian dalam pengembangan aplikasi adalah untuk mengetahui apakah aplikasi yang diuji dapat memenuhi kebutuhan *user* dengan mendasari pada rancangan dan pengembangan perangkat lunak.

Metode pengujian yang dipakai dalam pengembangan aplikasi adalah *black box testing*. *Black box testing* atau tes fungsional adalah pengujian yang dilakukan

hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak yang dikembangkan.

*Black box testing* berfokus pada persyaratan fungsional perangkat lunak. Disebut juga pengujian *behavioral* atau pengujian partisi. *Black box testing* memungkinkan perancang perangkat lunak mendapatkan serangkaian *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

Beberapa hal yang diujikan dalam sistem aplikasi yaitu apakah sesuai dengan tujuan awal dikembangkan sistem, tujuan pengembangan sistem adalah:

1. Aplikasi dapat membuat rencana berlibur sesuai dengan keinginan dan kebutuhan *user*.
2. Sistem dapat memperlihatkan letak koordinat dari tempat-tempat wisata yang akan dikunjungi.
3. Sistem dapat menghitung jarak dan waktu dari satu tempat kegiatan ke tempat kegiatan berikutnya.
4. Sistem dapat menyimpan semua *record* dari rencana berlibur yang telah dibuat oleh *user*.
5. Sistem dapat membuat *user* mudah untuk memperbarui rencana berliburnya.
6. Sistem dapat mencetak rencana berlibur.
7. Aplikasi dapat memberikan daftar objek-objek wisata yang dapat dikunjungi oleh *user*.
8. Aplikasi dapat memberikan informasi secara detail dari objek wisata.



9. Sistem dapat memberikan informasi mengenai objek-objek wisata yang paling banyak disukai berdasarkan *user* lainnya..
10. Aplikasi dapat memberikan daftar hotel dan penginapan sehingga memudahkan *user* untuk mencari tempat menginap.
11. Aplikasi dapat memberikan informasi detail dari penyedia layanan hotel atau penginapan.
12. Sistem dapat membuat *user* bisa menuju situs utama dari penyedia layanan hotel atau penginapan sehingga *user* dapat melakukan reservasi.
13. Aplikasi dapat memberikan daftar dan informasi singkat tentang layanan transportasi sehingga memudahkan *user* untuk mencari kendaraan yang dapat digunakan.
14. Sistem dapat membuat *user* bisa menghubungi pihak penyedia layanan transportasi sehingga *user* dapat memesan kendaraan.
15. Aplikasi dapat membuat *user* bisa memperbarui data akunnya.
16. Aplikasi dapat menyediakan layanan untuk *user* mengirimkan pesan kepada *admin*.

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Pengembangan Sistem**

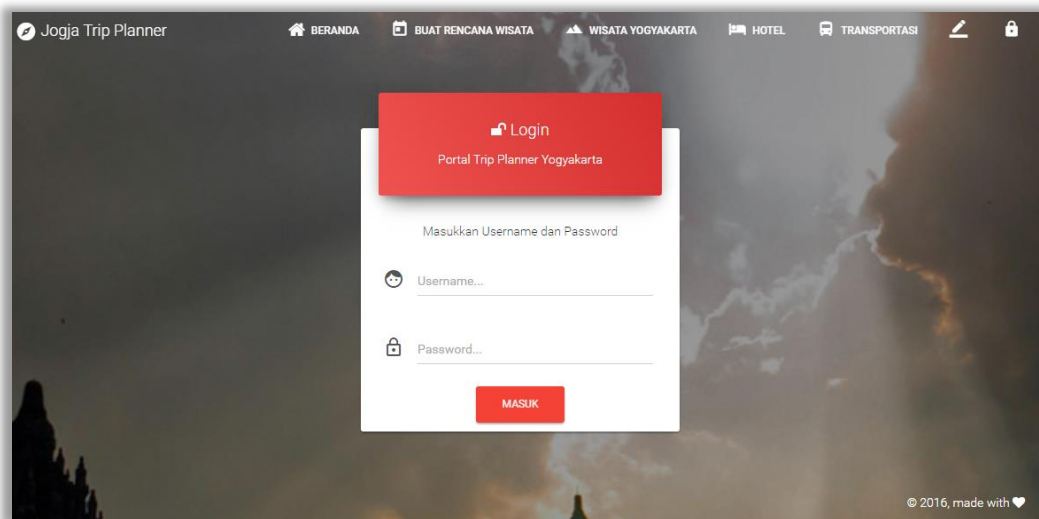
Pengembangan *bussiness logic* dari *website* program *tourism* berbasis *web* menggunakan Bahasa pemrograman PHP dan *framework CodeIgniter*. *CodeIgniter* menggunakan metode MVC di mana terdapat tiga komponen yaitu *folder Models* yang akan menyimpan *file* PHP yang akan digunakan untuk mengakses *database*, *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*

```

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.

```

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

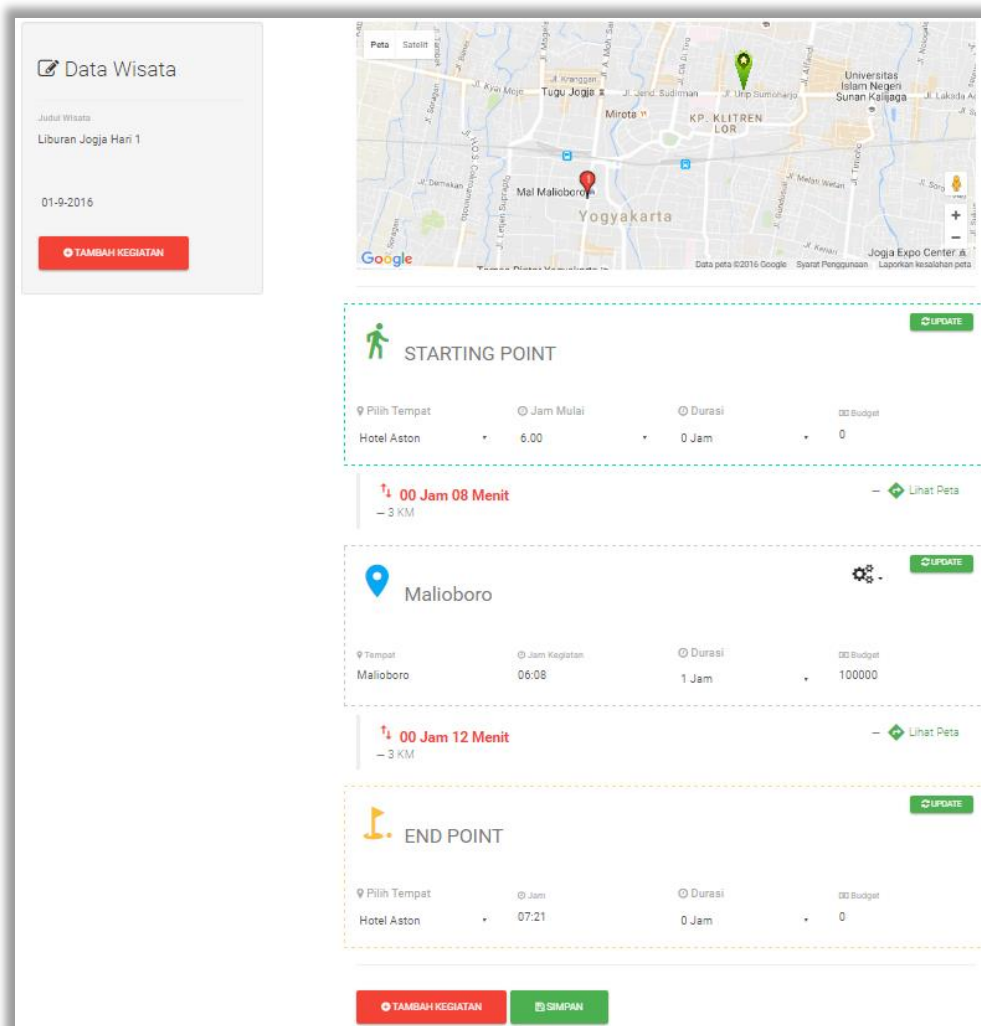
Pada gambar 4.5 memperlihatkan *coding* fungsi *login* yang ada di *file website.php* pada *folder Controller*. Pada awal *coding* di fungsi *login* akan memastikan terlebih dahulu apakah *id user* yang masih tersimpan di *session* jika masih ada maka akan menuju fungsi akun pada *folder controllers*. 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 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

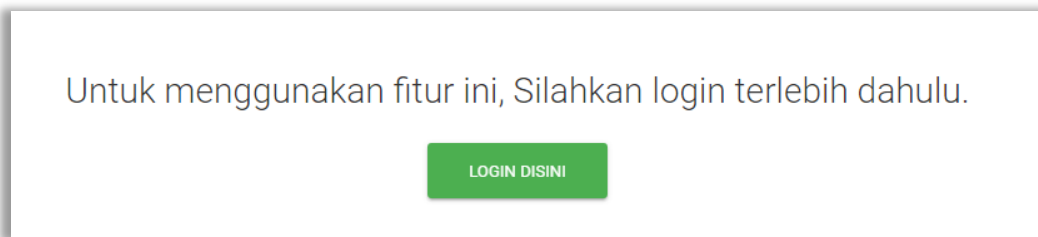
Pada gambar 4.7 adalah coding session userdata dan flashdata. Userdata digunakan untuk mengambil session dari array yang telah dibuat ketika melakukan

*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.



```

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

```

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 maka *form end point* akan kosong dan dapat diisi ketika *user* telah selesai mengisi *starting point*.

```
<div class="ool-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



*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>
</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 tampilan, *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>
</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;">
                      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>
                          <br>
                          <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>
                    </div>
                </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </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 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">
            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>
            </p>
            <p 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>

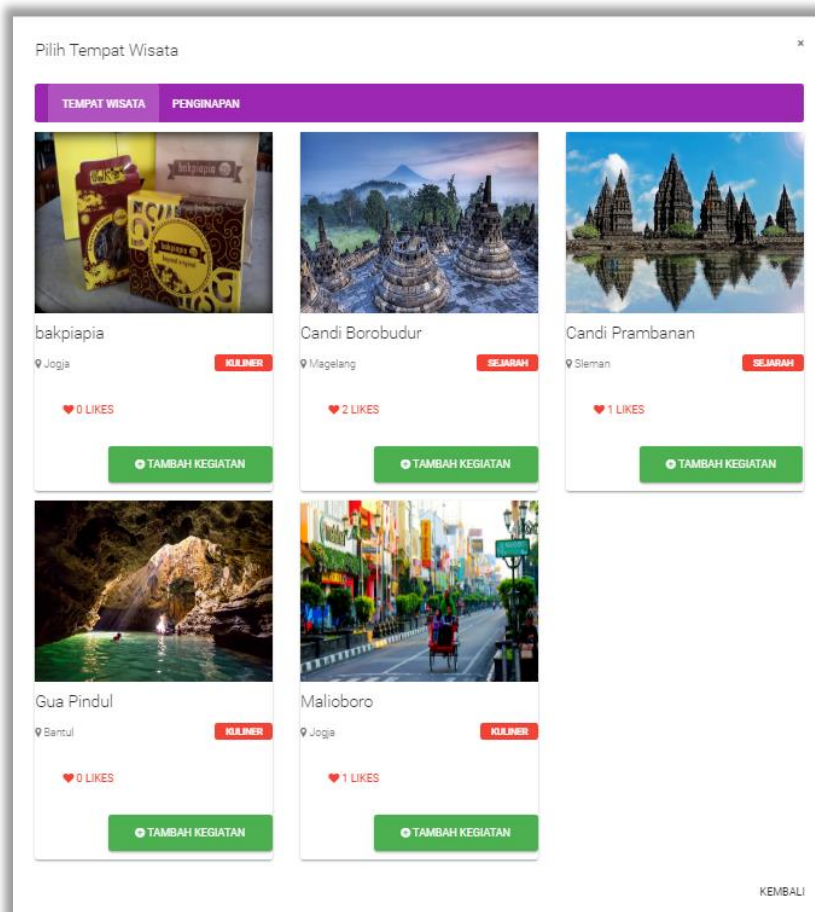
```

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

```
</div>
</div>
<div 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>
    <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);
        }
      </div>
    </div>
  </div>
</form>

```

**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>
</form>

```

**Gambar 4.31** Coding form kegiatan bagian 3

```

<?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);
```

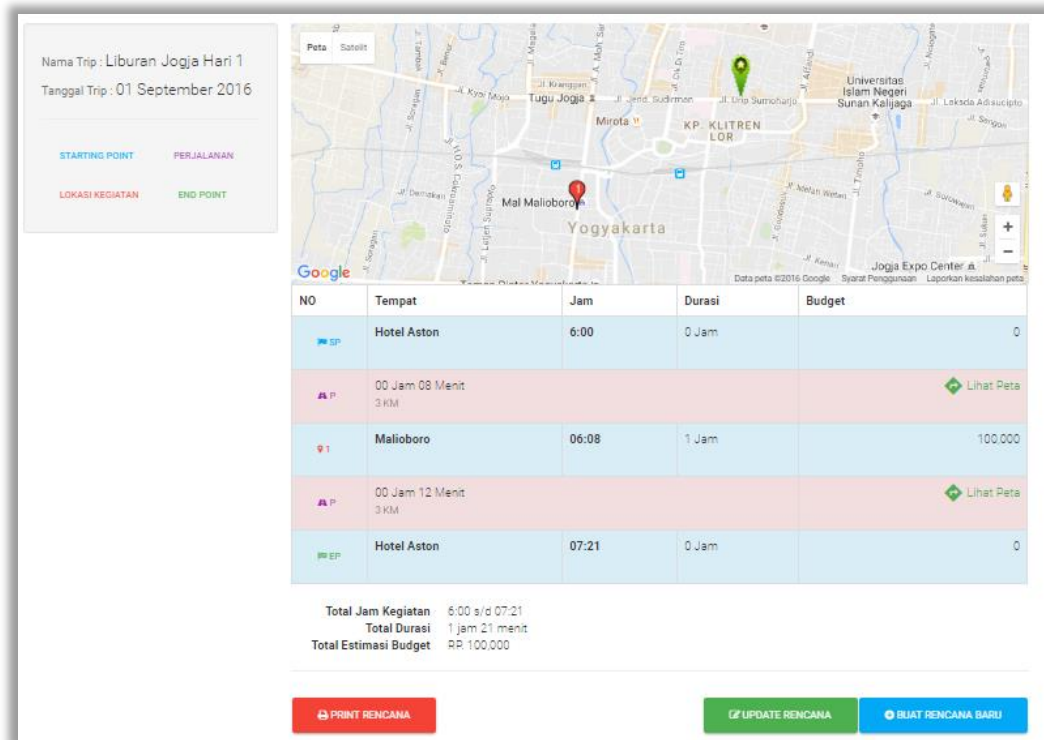
**Gambar 4.33** Coding map bagian 2

```
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 $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.

```

<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');
        </?php
        ?>

```

**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*.

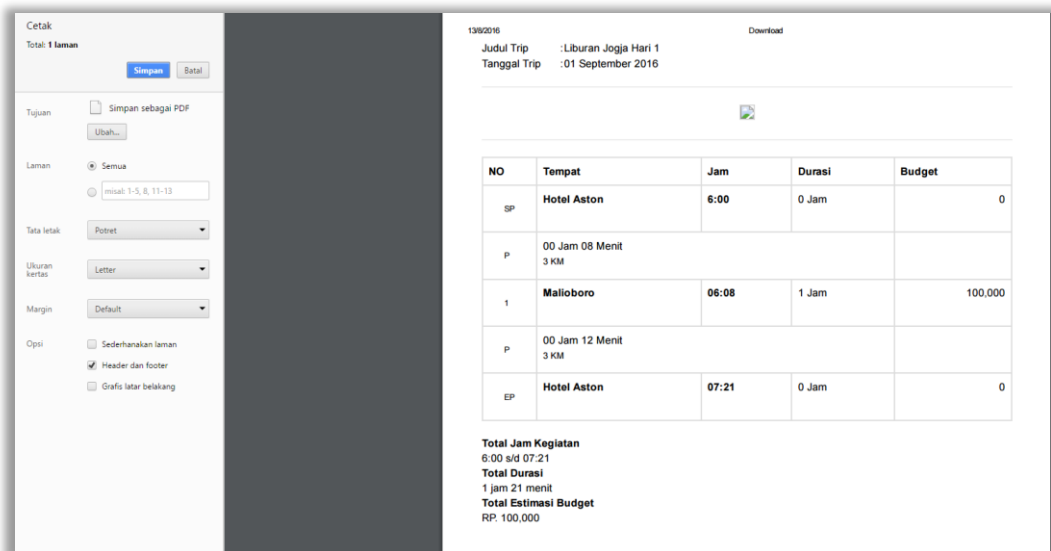
```

<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	<i>Button</i> MASUK di halaman <i>login</i>	Sistem dapat masuk ke halaman akun	Setelah <i>button</i> Masuk ditekan, sistem akan mengarahkan <i>user</i> ke halaman akun	Berhasil
2	<i>Button</i> Menu di <i>header</i> halaman <i>website</i>	Sistem dapat menampilkan halaman sesuai	Setelah <i>button</i> ditekan, muncul halaman sesuai dengan nama tombol	Berhasil



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

No	Kasus Yang diuji	Hasil Yang Diharapkan	Hasil Yang Didapatkan	Status
		yang ingin di <i>update</i>	beserta data yang ingin diperbarui	
8	<i>Button</i> BUAT RENCANA BARU di halaman detail rencana	Sistem dapat menampilkan halaman rencana	Setelah <i>button</i> 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 <i>budget</i> kegiatan	Sistem dapat menghitung total <i>budget</i> kegiatan secara otomatis	Sistem otomatis menghitung total <i>budget</i> kegiatan	Berhasil
5	Pengujian fungsi menampilkan Map	Sistem dapat menampilkan <i>map</i> dan koordinat dari tiap tempat kegiatan	Sistem menampilkan <i>map</i> dan koordinat tempat kegiatan	Berhasil
6	Pengujian fungsi Menyimpan rencana berlibur	Sistem dapat menyimpan rencana berlibur ke <i>database</i>	Rencana yang di simpan sudah tersimpan di <i>database</i>	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 <i>database</i>		
8	Pengujian fungsi Menghapus rencana berlibur	Sistem dapat menghapus rencana berlibur yang terdapat di <i>database</i>	Rencana berlibur yang tersimpan di <i>database</i> 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 <i>username</i> dan <i>password</i> tidak sesuai	Sistem dapat menampilkan pesan peringatan bahwa <i>username</i> atau <i>password</i> yang di <i>input</i> salah	Muncul pesan peringatan yang memberitahukan bahwa <i>username</i> atau <i>password</i> salah	Berhasil

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

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Dari hasil dan pembahasan yang telah dilakukan dalam pengembangan *bussines logic website* program *tourism* berbasis *web*, maka diperoleh kesimpulan

Sebagai berikut:

1. Aplikasi dapat membuat rencana berlibur secara detail dan jelas yang membuat rencana berlibur menjadi fleksibel sesuai dengan kebutuhan para turis.
2. Aplikasi dapat membantu dalam mencari referensi objek-objek wisata yang dapat dikunjungi di Yogyakarta dan sekitarnya.
3. Sistem dapat menampilkan koordinat tempat wisata yang memudahkan *user* dalam mengetahui letak posisi dari tempat wisata yang akan dikunjungi.
4. Sistem dapat menghitung jarak dan waktu antar tempat kegiatan yang membuat para turis dapat mengetahui informasi tentang jarak dan waktu yang akan ditempuh selama kegiatan *tourism*.
5. Sistem dapat mencetak rencana berlibur yang membuat para turis memiliki *hardcopy* dari rencana berlibur yang telah dibuat.
6. Aplikasi dapat membantu dalam mencari referensi hotel dan penginapan yang ada di Yogyakarta.
7. Aplikasi dapat membantu dalam mencari referensi transportasi yang dapat digunakan dalam kegiatan *tourism*.
8. Sistem dapat membantu para turis dalam memesan transportasi.



## 5.2 Saran

Berikut beberapa saran yang selanjutnya digunakan untuk menambah fitur-fitur tambahan dalam pengembangan aplikasi selanjutnya:

1. Aplikasi dapat dikembangkan menjadi sistem yang terintegrasi dengan pihak hotel sehingga para turis dapat langsung melakukan reservasi hotel tanpa harus menuju situs utama hotel terlebih dahulu.
2. Aplikasi dapat menampilkan suatu map di mana nantinya akan menampilkan semua tempat wisata yang terdapat di Yogyakarta, sehingga menjadi referensi bagi turis yang sebelumnya belum pernah mengunjungi Yogyakarta.
3. Aplikasi dapat mencakup banyak objek wisata tidak hanya Yogyakarta tapi juga daerah lain sehingga aplikasi dapat membantu para turis untuk membuat rencana berlibur.
4. Aplikasi dapat dikembangkan di sistem operasi *android* dan *ios* sehingga pengguna *android* ataupun *ios* dapat langsung mengakses aplikasi langsung dari gadget tanpa melalui *web browser*.

## DAFTAR PUSTAKA

- Ambler, S. W. (2005). *The Elements of UML 2.0 Style*. s.l. Cambridge University Press.
- Arief, M. (2011). *Pemrograman Web Dinamis Menggunakan PHP dan Mysql*. Yogyakarta: ANDI.
- Beighley. (2010). *jQuery For Dummies*. Indianapolis, Indiana: Wiley Publishing.
- Buhalis, D. (2011). *E-Tourism*. In *Contemporary Tourism Reviews*. Oxford: Goodfellow Publishers Limited.
- CodeIgniter. (2016, Februari 11). *Pengertian CodeIgniter*. Diambil kembali dari CodeIgniter Site: <https://www.codeigniter.com>
- Egger, R. (2016, Februari 11 ). *Restructuring The Destination Management System Paradigm*. Diambil kembali dari EuroChire Dubai Conference 2008: <http://pc.parnu.ee/~htooman/EuroChire>
- Fowler, M. (2004). *UML Distilled (Panduan Singkat Bahasa Pemodelan Objek Standar)*. Yogyakarta: ANDI.
- Hardjono, D. (2006). *Seri Panduan Lengkap Menguasai Pemrograman Web dengan PHP 5*. Yogyakarta: ANDI .
- Jogiyanto, H. (2005). *Analisis & Desain Sistem Informasi Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: ANDI.
- Notepad++. (2016, Februari 11). *Pengertian Notepad++*. Diambil kembali dari Notepad++: <https://notepad-plus-plus.org>
- Putera, P. B., & Oktavianti, D. (2010). Perbandingan Pencapaian Jejaring Informasi Pariwisata Terpadu Berbasis Web (Electronic Tourism). *Prosiding Seminar Nasional* , hal. C-45 – C.52.
- Sibero, A. (2011). *Kitab Suci Web Programing*. Yogyakarta: MediaKom.
- Slamet, R. (2009). *Membuat Web Portal Multi Bahasa Joomla*. Elex Media Komputindo.
- Vansteenwegen, P., & Souffriau, W. (2011). Information Technology & Tourism. *TRIP PLANNING FUNCTIONALITIES: STATE OF THE ART AND FUTURE*, 305-315.
- World Tourism Organization. (1995). *UNWTO technical manual: Collection of Tourism Expenditure Statistics*. Diambil kembali dari World Tourism Organization: <http://pub.unwto.org/>

## LAMPIRAN

### Source Code halaman Login

```
<?php echo get_header('theme_website');?>
<div class="header header-filter" style="background-image: url('<?php echo
get_media_url();?>theme2/img/banner/prambanan.jpg'); background-size:
cover; background-position: top center;">
    <div class="container">
        <div class="row">
            <div class="col-md-4 col-md-offset-4 col-sm-6 col-sm-
offset-3">
                <div class="card card-signup">
                    <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">
```

```

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">
<div class="material-icons">lock_outline</i>
</span>
<input
name="password" type="password" placeholder="Password..." class="form-
control" required >
</div>
</div> -->
</div>
<div class="footer text-center">
<button type="submit"
class="btn btn-danger btn-raised btn-wd">masuk</button>
</div>
</form>a
</div>
</div>
</div>
</div>
</div>
<?php echo get_footer('theme_website');?>

```

*Source Code* halaman Rencana

```

<?php echo get_header('theme_website');?>
<script type="text/javascript"
src="http://maps.google.com/maps/api/js?key=<?php echo
api_key();?>&sensor=true"></script>
<script src="http://ajax.aspnetcdn.com/ajax/jquery/jquery-
1.10.1.min.js"></script>

<script type="text/javascript">
var rad = function(x) {
return x * Math.PI / 180;
};

function popup(url)
{

```

```

        params = 'width='+screen.width;
        params += ', height='+screen.height;
        params += ', top=0, left=0'
        params += ', fullscreen=yes';

        newwin=window.open(url,'Peta Perjalanan', params);
        if (window.focus) {newwin.focus()}
        return false;
    }
}

</script>

<?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>';
?>

    <div class="row">
    <?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 = "";

    }
    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 = "";
    }
?>
<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
Wisata</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>Anda Harus Menentukan <b>Starting Point</b> dan
<b>End Point</b> Terlebih Dahulu</p>
        <?php } ?>
    </div>
    <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!-->

<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!-->
        <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($this->session-
>userdata('session_start') !=''):?>
class="btn btn-raised btn-success pull-right visible-xs "><i class="fa fa-
refresh"></i> update</button>
<?php endif;?>
</div>
</div>
</div>
</form>
<?php if($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');
$kl = $ab->row();
if($kl->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 } ?>
<?php if($this->session->userdata('session_start') != "" and $this-
>session->userdata('session_end') != ""){?>
<?php
    $dtk = 0;
    $dtk1 =0;
    $before = 0;
    $times = 0;
    $data_list =
get_dari('trip_detail',['id_trip'=>$id_trip,'starting_point'=>0,'end_point'=>0],'po
sisi','asc');
    $total_list = $data_list->num_rows();
    foreach($data_list->result() as $list):
        if($list->jenis == 0)
        {
            $dari = 'tempat_wisata';
        }
        else
        {
            $dari = 'hotel';

```

```

    }

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

```

```



```

```

'tempat_wisata';
}
else
{
    $dr = 'hotel';
}

$koordinat1 =
$koordinat2 =
$jarak =

get_jarak($koordinat1,$koordinat2,",",true);

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

$jamnya=
konversi_jam($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 $jamnya;?>" >
    </div>
</div><!--col-md-4!-->
<div class="col-md-3">
    <div class="form-group">
        <label><i class="fa
fa-clock-o"></i> Durasi</label>
        <select
name="durasi" required class="form-control">
            <?php
for($jams=0;$jams<=10;$jams++):
            ?>

```

```

value="<?php echo $jams;?>" <?php if($list->durasi == $jams) echo
'selected';?><?php echo $jams;?> Jam</option>
</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>
<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 $list->id;?>')">
=
<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

```



```

        $sebelum =
get_dari('trip_detail',['id_trip'=>$list->id_trip,'posisi'=>$list->posisi+1],'posisi');
        $rw = $sebelum->row();

        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');

        echo get_jarak($koordinat1,$koordinat2,true);
        ?>
        </b> <small><?php echo
get_jarak($koordinat1,$koordinat2,"true");?> KM</small>
        </font>

    </p>

</blockquote>
    <?php endforeach;?>

<!--batas!-->
<?php } ?>

    <?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;?>
            </div>
        </div>
    </form>
    </if?>

```

```

                                <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!-->
                                    <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();
                //durasi
                sebelumnya
                $drs = $cc-
>durasi*3600;
                $dtk =
$before+$drs;
            }
            //
            // $jamnya = 0;
            //get sebelumnya

            $ambil_posisi_sebelumnya =
get_dari('trip_detail',['id_trip'=>$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)
            {

```

```

'tempat_wisata';
}
else
{
$dr2 =
}
$koordinat1 =
$koordinat2 =
$jarak =
get_jarak($koordinat1,$koordinat2,",",",true);

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

//echo
konversi_jam($dtk);

$jamnya=
konversi_jam($before);
}
?>
<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">

```

```

class="fa fa-clock-o"></i> Durasi</label>
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;
group label-floating">
class="control-label"><i class="fa fa-money"></i> Budget</label>
type="text" class="form-control" name="budgetStart" value="<?php echo
$budget2;?>">
>userdata('session_end') !='')?:?>
class="btn btn-raised btn-success pull-right visible-xs "><i class="fa fa-
refresh"></i> update</button>
</div>
</div>
</div>
</form>
<?php endif;?>

```

```

<hr>
<?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;?>
</div>
</div>
<!-- Modal Core -->
<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;">
                                            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
$id_trip;?>" class="btn btn-success btn-raised pull-right"><i class="fa fa-plus-
circle"></i> Tambah Kegiatan</a>
</div>
</div>
</div>
<?php endforeach;?>
</div>
</div>
<div role="tabpanel" class="tab-pane "
id="penginapan">
<div class="row">
<div class="card-deck">
<?php foreach($hotel->result() as
$row2):?>
<div class="col-md-4"
style="margin-top: 10px;">
<div
class="card">
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>

</div>

</div>

```



```

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

<!-- Modal Core -->
<div class="modal fade" id="myModal" tabindex="-1" 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">Peta Perjalanan</h4>
            </div>
            <div class="modal-body">

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

                </div>
            </div>
        </div>
    </div>

    </div>
    </div>
    </div>
</div>
<?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++) {
    if(i == 0)
    {
        var icon_image = new google.maps.MarkerImage(
            "<?php echo get_media_url();?>theme2/img/start.png",
            /* 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",
            /* 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);

    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();
</script>
<?php } ?>

<?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>';} ?>
<?php echo get_footer('theme_website');?>

```

## Source Code halaman Detail Rencana

```
<?php echo get_header('theme_website');?>
<script type="text/javascript"
src="http://maps.google.com/maps/api/js?key=?php echo
api_key();?>&sensor=true"></script>
<script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-
1.10.1.min.js"></script>
<script type="text/javascript">
    function popup(url)
    {
        params = 'width='+screen.width;
        params += ', height='+screen.height;
        params += ', top=0, left=0'
        params += ', fullscreen=yes';

        newwin=window.open(url,'Peta Perjalanan', params);
        if (window.focus) {newwin.focus()}
        return false;
    }
</script>

<style type="text/css">
    .btn-raised{
        box-shadow: none;
    }
</style>
<?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>
```

```

<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'=>$list-
>id_tempat], 'nama');
            ?>
            <tr class="bg-info">
              <td>
                <?php
                  if($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
$nama_tempat;?></strong></td>
                                <td>
                                <?php
                                if($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';
                                                                    }

                                                                    $koordinat1
= tampilkan($dr,['id'=>$aa->id_tempat],'koordinat');

```

= tampilkan(\$dari,['id'=>\$list->id_tempat],'koordinat');	\$koordinat2
get_jarak(\$koordinat1,\$koordinat2,",",",true);	\$jarak =
hitung_detik(\$list->id);	//\$before =
\$dtk;	//\$dtk =
\$dtk+\$jarak;	\$before +=
konversi_jam(\$before);	\$jamnya=
echo \$jamnya;?></strong>	} ?> <strong><?php
>durasi;?> Jam	</td> <td> <?php echo \$list-
number_format(\$list->budget);?>	</td> <td class="text-right"> <?php echo
\$total_list-1):?>	</td> </tr> <?php if(\$list->posisi !=
class="btn btn-primary btn-sm "><i class="fa fa-road"></i> P</button>	<tr class="bg-danger"> <td> <button
sebelumnya	</td> <td colspan="3"> <?php //tempat
get_dari('trip_detail',['id_trip'=>\$list->id_trip,'posisi'=>\$list->posisi+1],'posisi');	\$sebelum =
\$sebelum->row();	\$rw =



```

>jenis==0)
    $dari2='tempat_wisata';
    $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>
    </tr>
    <?php endif;?>
    <?php endforeach;?>
</tbody>

</table>
</div>

<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>
<script type="text/javascript">
var locations = [
<?php

$map_rencana = get_dari('trip_detail',['id_trip'=>$row->id],'posisi','asc');
foreach($map_rencana->result() as $j => $row_map):
    if($row_map->jenis == 0)
    {
        $map_from = 'tempat_wisata';
    }
    else
    {
        $map_from = 'hotel';
    }
</script>

```

```

    }

    $map_koordinat = tampilkan($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++) {
    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|F
FFFFF';
}
marker = new google.maps.Marker({
    position: new google.maps.LatLng(locations[i][1], locations[i][2]),
    map: map,
    icon: icon_image
});
markers.push(marker);

//deskripsi
//
var contentString = '<div id="content">'+
'<div id="siteNotice">'+
'</div>'+
'<h1 id="firstHeading" class="firstHeading">Uluru</h1>'+
'<div id="bodyContent">'+
'<p><b>Uluru</b>, also referred to as <b>Ayers Rock</b>, is a large ' +
'sandstone rock formation in the southern part of the ' +
'Northern Territory, central Australia. It lies 335&#160;km (208&#160;mi)
'+
'south west of the nearest large town, Alice Springs; 450&#160;km ' +
'(280&#160;mi) by road. Kata Tjuta and Uluru are the two major ' +
'features of the Uluru - Kata Tjuta National Park. Uluru is ' +
'sacred to the Pitjantjatjara and Yankunytjatjara, the ' +
'Aboriginal people of the area. It has many springs, waterholes, ' +
'rock caves and ancient paintings. Uluru is listed as a World ' +
'Heritage Site.</p>'+
'<p>Attribution: Uluru, <a
href="https://en.wikipedia.org/w/index.php?title=Uluru&oldid=297882194">'+
'https://en.wikipedia.org/w/index.php?title=Uluru</a> ' +
'(last visited June 22, 2009).</p>'+
'</div>'+
'</div>';

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();
</script>
<?php echo get_footer('theme_website');?>

```

*Source Code* halaman Cetak Kegiatan

```

<!DOCTYPE html>
<html>
<head>
  <title>Download </title>
  <link href="<?php echo get_media_url();?>theme2/css/bootstrap.min.css"
rel="stylesheet" />
  <link href="<?php echo get_media_url();?>theme2/css/material-kit.css"
rel="stylesheet"/>
  <style type="text/css">
    body{
      background-color: #fff;
    }
  </style>

  <script type="text/javascript">
    window.onload = function () {
      window.print();
      history.go(-1);
    }
  </script>
</head>
<body>

<?php $row = $data->row();?>

<div class="container">
<table width="40%">
<tr>

```

```

<td>Judul Trip</td>
<td>:</td>
<td><?php echo $row->judul;?></td>
</tr>
<tr>
<td>Tanggal Trip</td>
<td>:</td>
<td><?php echo ind_format($row->tanggal);?></td>
</tr>
</table>
<hr>
<center>

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

```

```

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');
$a = $ambil_posisi_sebelumnya->row();
if($a->jenis == 0)
{
    $dr = 'tempat_wisata';
}
else
{
    $dr = 'hotel';
}

$koordinat1 = tampilkan($dr,['id'=>$a->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-success btn-sm btn-raised"><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();

        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">

    </td>
</tr>
<?php endif;?>
<?php endforeach;?>
</tbody>

</table>

<dl class="dl-horizontal">
<dt>Total Durasi</dt>
<dd><?php echo $awal_durasi;?> s/d <?php echo $jamnya;?></dd>
<dt>Total Jam Kegiatan</dt>
<dd><?php echo selisih_jam($awal_durasi.':00',$jamnya.':00');?></dd>
<dt>Total Estimasi Budget</dt>
<dd>RP. <?php echo number_format($total_budget);?></dd>
</dl>
</div>

</body>
</html>
```