



**IMPLEMENTASI MODEL *PERSONAL EXTREME PROGRAMMING*
(PXP) DALAM PENGEMBANGAN SISTEM INFORMASI GEOGRAFIS
PEMETAAN FLORA DAN FAUNA DI TAMAN NASIONAL MERU
BETIRI**

SKRIPSI

Oleh

RIDLO PAMUNGKAS

NIM 132410101063

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS JEMBER**

2018



**IMPLEMENTASI MODEL *PERSONAL EXTREME PROGRAMMING*
(PXP) DALAM PENGEMBANGAN SISTEM INFORMASI GEOGRAFIS
PEMETAAN FLORA DAN FAUNA DI TAMAN NASIONAL MERU
BETIRI**

SKRIPSI

Diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan pendidikan Sarjana (S1) Program Studi Sistem Informasi
Universitas Jember dan mendapat gelar Sarjana Komputer

Oleh

RIDLO PAMUNGKAS

NIM 132410101063

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER

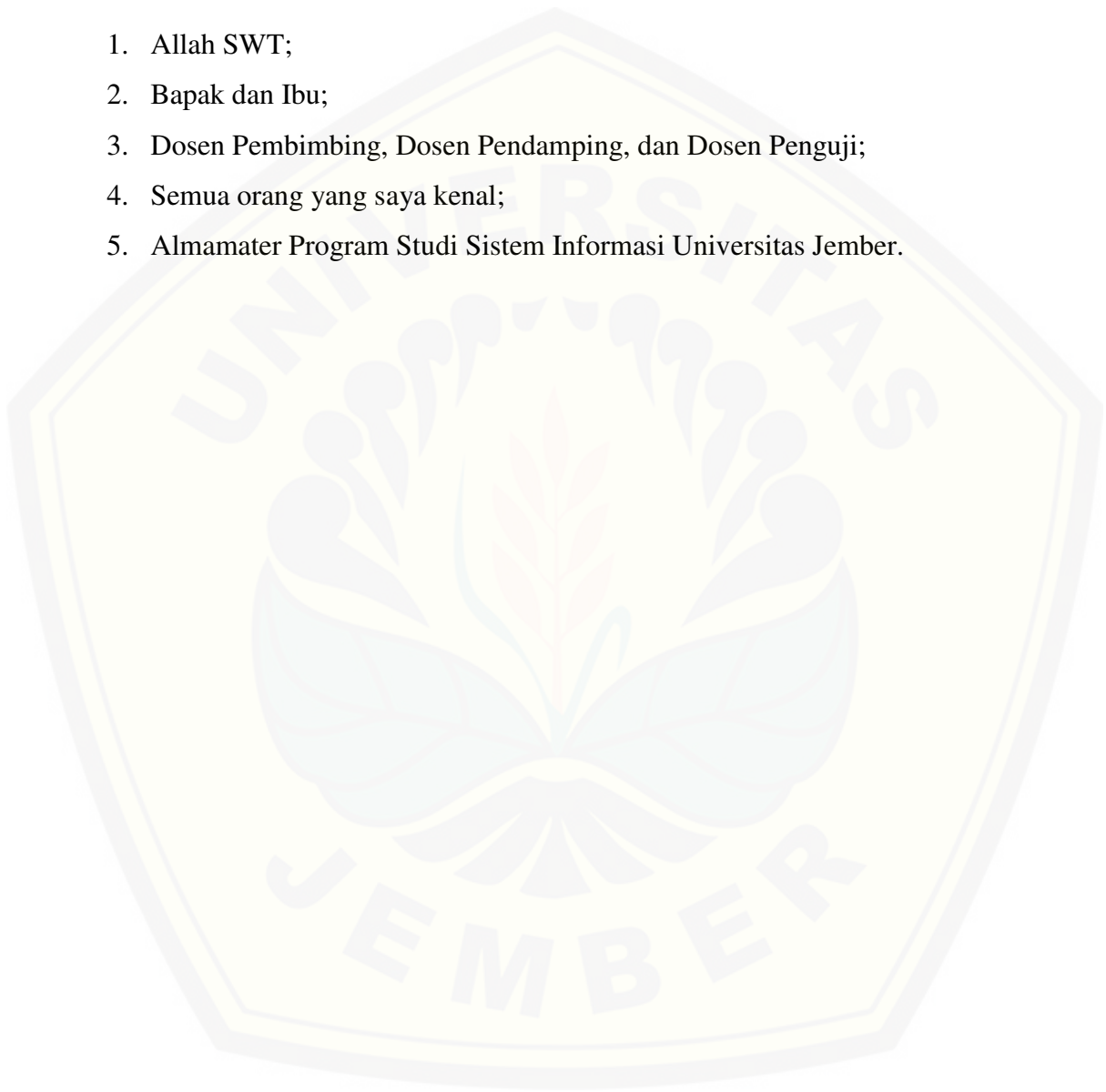
UNIVERSITAS JEMBER

2018

PERSEMBAHAN

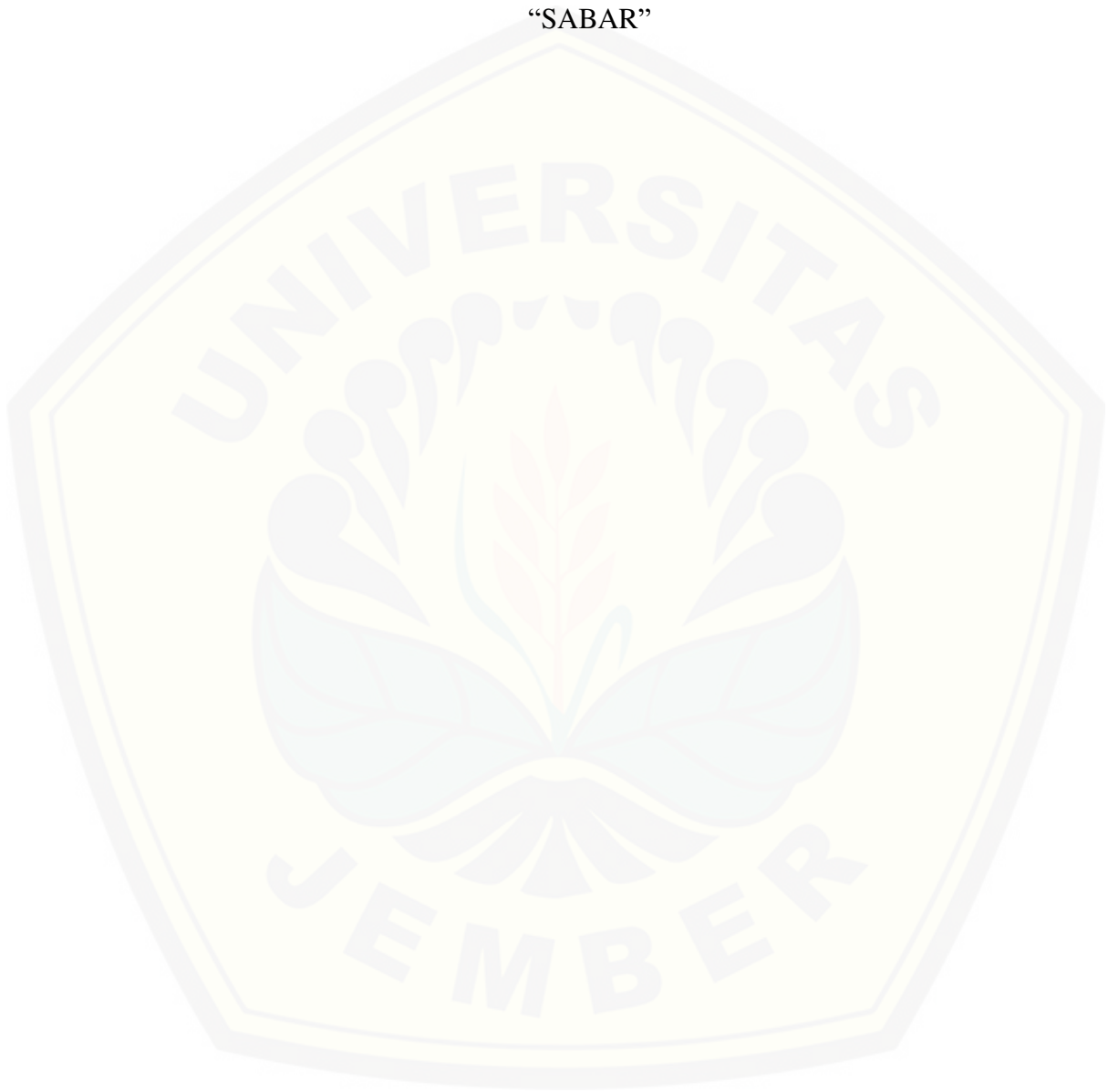
Skripsi ini saya persembahkan untuk :

1. Allah SWT;
2. Bapak dan Ibu;
3. Dosen Pembimbing, Dosen Pendamping, dan Dosen Penguji;
4. Semua orang yang saya kenal;
5. Almamater Program Studi Sistem Informasi Universitas Jember.



MOTO

“SABAR”



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Ridlo Pamungkas

NIM : 132410101063

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Implementasi Model *Personal Extreme Programming* (XP) dalam Pengembangan Sistem Informasi Geografis Pemetaan Flora dan Fauna di Taman Nasional Meru Betiri”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 13 Mei 2018

Yang menyatakan,

Ridlo Pamungkas

NIM 132410101063

SKRIPSI

**SISTEM INFORMASI GEOGRAFIS PEMETAAN FLORA DAN FAUNA
DI TAMAN NASIONAL MERU BETIRI**

Oleh :

Ridlo Pamungkas

NIM 132410101063

Pembimbing

Dosen Pembimbing Utama : Anang Andrianto., S.T., M.T.

Dosen Pembimbing Pendamping : Diah Ayu Retnani W., S.T., M.Eng.

PENGESAHAN PEMBIMBING

Skripsi berjudul “Implementasi Model *Personal Extreme Programming* (PXP) dalam Pengembangan Sistem Informasi Geografis Pemetaan Flora dan Fauna di Taman Nasional Meru Betiri”, telah diuji dan disahkan pada:

hari, tanggal : Senin, 16 Juli 2018

tempat : Fakultas Ilmu Komputer, Universitas Jember

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Anang Andrianto., S.T., M.T.
NIP. 196909281993021001

Diah Ayu Retnani W., S.T., M.Eng.
NIP 198603052014042001

PENGESAHAN PENGUJI

Skripsi berjudul “Implementasi Model *Personal Extreme Programming* (PXP) dalam Pengembangan Sistem Informasi Geografis Pemetaan Flora dan Fauna di Taman Nasional Meru Betiri”, telah diuji dan disahkan pada:

hari, tanggal : Senin, 16 Juli 2018

tempat : Fakultas Ilmu Komputer, Universitas Jember

Tim Penguji :

Penguji I,

Penguji II,

Achmad Maududie ST., M.Sc.

NIP. 19700422 1995121001

Januar Adi Putra, S.Kom., M.Kom

NIP. 760017015

Mengesahkan
Penjabat Dekan

Prof. Drs. Slamin, M.Comp.Sc.,Ph.D

NIP. 19670420 1992011001

RINGKASAN

Implementasi Model *Personal Extreme Programming* (PXP) dalam Pengembangan Sistem Informasi Geografis Pemetaan Flora dan Fauna di Taman Nasional Meru Betiri, Ridlo Pamungkas, 132410101063; 2018, 241 halaman; Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Jember.

Software Development Life Cycle (SDLC) adalah kumpulan dari berbagai langkah yang diikuti secara sistematis dalam pengembangan, perancangan dan pemeliharaan proyek perangkat lunak dan memastikan bahwa semua kebutuhan pengguna terpenuhi dengan sedikit jumlah konsumsi sumber daya. *Extreme Programming* adalah suatu bentuk pembangunan perangkat lunak yang berbasis nilai kemudahan, komunikasi, umpan balik, dan keberanian.

Extreme Programming digunakan dalam pengembangan sistem berskala kecil yang membutuhkan kecepatan dan ketepatan. Model pengembangan ini memakai desain yang sedikit dan menekankan dalam testing program atau aplikasi (Dzhurov, Krasteva, & Ilieva, 2009). *Practices XP* dapat diterapkan untuk menangani situasi proses pengembangan perangkat lunak dengan pemrogram tunggal, yang kemudian dinamakan dengan metodologi *Personal Extreme Programming* (PXP). Penerapan PXP dapat mengatasi masalah dalam pengembangan sistem skala kecil yang membutuhkan kecepatan dalam pembuatan, memperkecil resiko dan biaya sistem, contohnya dalam pengembangan Sistem Informasi Geografis. Penerapan metode mengikuti alur *sdlc* membuat pengembangan sistem menjadi tertata dan lebih efisien. Hasil perbandingan 12 *core practice* menunjukkan bahwa pada penelitian ini 2 *core* masih belum terpenuhi, sehingga belum maksimal. Proses penambahan dan pengurangan fitur tidak banyak dilakukan yang menyebabkan *Continuous Integration* tidak terpenuhi dan *Customer* tidak selalu berkomunikasi dengan *programmer* yang menyebabkan *On-site Customer* tidak terpenuhi. Terpenuhinya 10 dari 12 *core practice* dari PXP membuat metode PXP tidak cocok untuk pengembangan sistem pada penelitian ini.

PRAKATA

Puji syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Implementasi Model *Personal Extreme Programming* (PXP) dalam Pengembangan Sistem Informasi Geografis Pemetaan Flora dan Fauna di Taman Nasional Meru Betiri”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Jember.

Penyusunan skripsi ini tidak lepas dari dukungan berbagai pihak. Oleh karena itu penulis menyampaikan terima kasih kepada :

1. Anang Andrianto S.T., MT. Selaku Dosen Pembimbing Utama dan Diah Ayu Retnani W., S.T., M.Eng. Selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
2. selaku Dosen Penguji I dan selaku Dosen Penguji II yang telah memberikan saran dan arahan dalam penulisan skripsi ini;
3. Semua pihak yang tidak dapat disebutkan satu persatu.

Dengan harapan bahwa penelitian ini nantinya akan terus berlanjut dan berkembang kelak, penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, 13 Mei 2018

Penulis

DAFTAR ISI

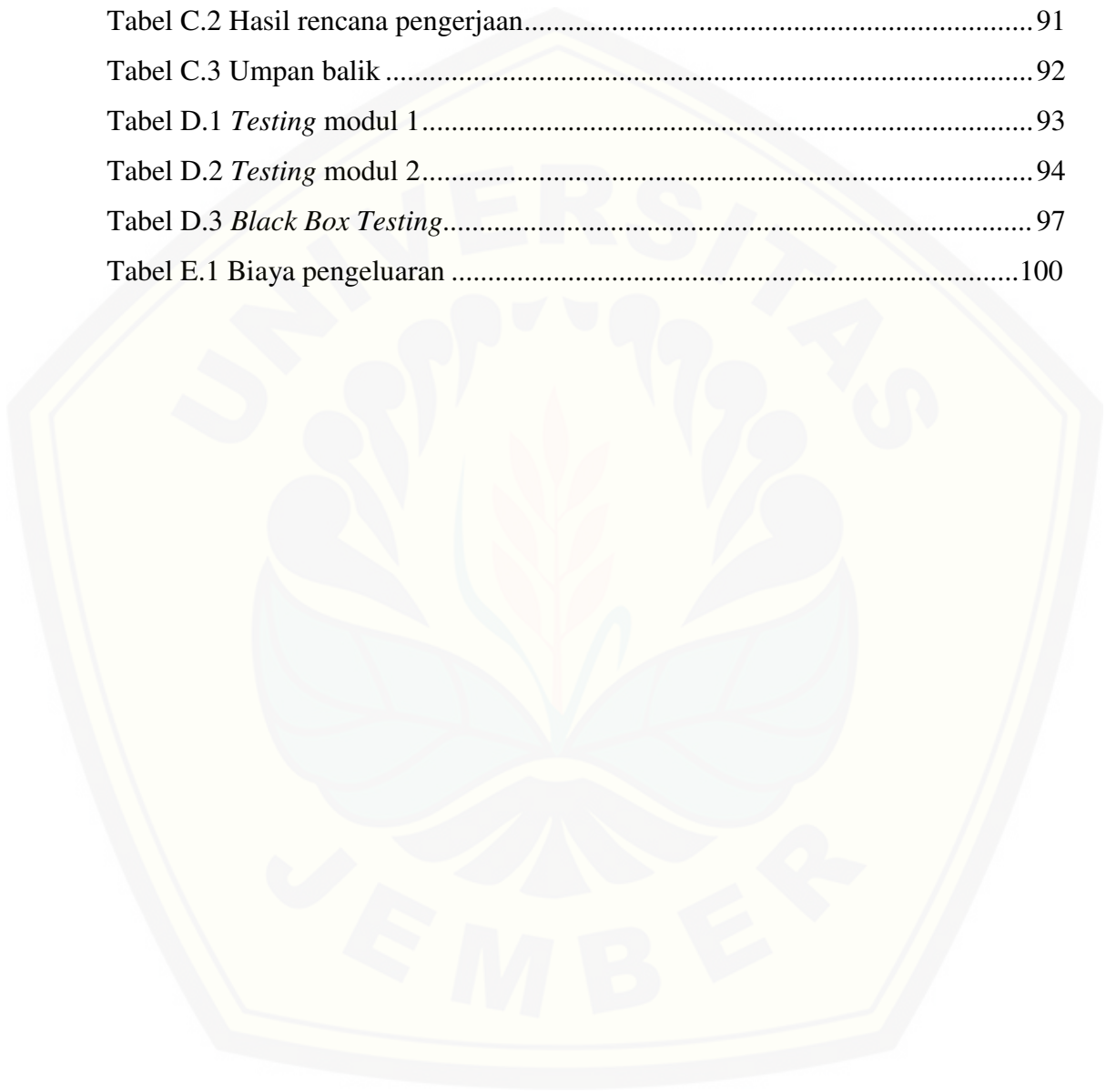
SKRIPSI.....	i
PERSEMBAHAN.....	ii
MOTO.....	iii
PERNYATAAN.....	iv
PENGESAHAN PEMBIMBING.....	vi
PENGESAHAN PENGUJI.....	vii
RINGKASAN.....	viii
PRAKATA.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	3
1.4 Batasan Masalah.....	3
1.5 Sitematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 <i>Personal Extreme Programming (PXP)</i>	5
2.2 Taman Nasional Meru Betiri.....	10
2.3 Sistem Informasi Geografis (SIG).....	13
2.4 <i>Google Maps API</i>	14
BAB III METODOLOGI PENELITIAN.....	16
3.1 Jenis Penelitian.....	16

3.2 Tahapan Penelitian	16
4.1 <i>Planning</i>	19
4.2 <i>Design</i>	25
4.3 <i>Implementation</i>	34
4.4 <i>System Testing</i>	39
BAB V HASIL DAN PEMBAHASAN.....	40
5.1 <i>Life Cycle</i> PXP dalam Pengembangan Sistem Informasi Geografis Pemetaan Flora dan Fauna	40
5.2 Penerapan Nilai Kemudahan, Komunikasi, Umpan Balik, dan Keberanian dalam <i>Personal Extreme Programming</i> (PXP).....	41
5.3 Penerapan 12 <i>core practice</i> dari <i>Personal Extreme Programming</i> (PXP) dalam pengembangan sistem informasi geografis pemetaan flora dan fauna di Taman Nasional Meru Betiri.....	43
BAB VI PENUTUP	46
6.1 Kesimpulan	46
6.2 Saran.....	47
DAFTAR PUSTAKA	48
LAMPIRAN.....	49

DAFTAR TABEL

Tabel 4.1 Kebutuhan fungsional	19
Tabel 4.2 Kebutuhan non-fungsional	20
Tabel 4.3 <i>Sampling</i> data flora	21
Tabel 4.4 <i>Sampling</i> data fauna	22
Tabel 4.5 <i>Sampling</i> data pos pantau	23
Tabel 4.6 <i>Sampling</i> data cabang	23
Tabel 4.7 <i>User Story</i>	24
Tabel 4.8 Definisi aktor sistem	25
Tabel 4.9 Definisi <i>usecase</i>	25
Tabel 4.10 Definisi <i>CRC Card</i>	31
Tabel 4.11 <i>Testing design</i>	34
Tabel 4.12 Kode program melihat peta flora	36
Tabel 4.13 Kode program melihat peta fauna	37
Tabel 5.1 Penerapan nilai-nilai PXP	41
Tabel 5.2 Penerapan 12 <i>core practice</i> PXP	43
Tabel A.1 Kode program <i>login</i>	50
Tabel A.2 Kode program menambah data pemetaan cabang	51
Tabel A.3 Kode program melihat data cabang	53
Tabel A.4 Kode program mengubah data cabang	54
Tabel A.5 Kode program menambah data pemetaan pos pantau	57
Tabel A.6 Kode program melihat data pos pantau	59
Tabel A.7 Kode program mengubah data pos pantau	60
Tabel A.8 Kode program melihat peta cabang	63
Tabel A.9 Kode program melihat rute peta cabang	64
Tabel A.10 Kode program melihat detail rute peta cabang	65
Tabel A.11 Kode program melihat peta pos pantau	69
Tabel A.12 Kode program melihat rute peta pos pantau	70
Tabel A.13 Kode program melihat detail rute peta pos pantau	74
Tabel A.14 Kode program melihat data flora	78

Tabel A.15 Kode program mengubah data flora.....	80
Tabel A.16 Kode program melihat data fauna	82
Tabel A.17 Kode program mengubah data fauna	84
Tabel C.1 Rencana Target Pengerjaan	91
Tabel C.2 Hasil rencana pengerjaan.....	91
Tabel C.3 Umpan balik	92
Tabel D.1 <i>Testing</i> modul 1	93
Tabel D.2 <i>Testing</i> modul 2.....	94
Tabel D.3 <i>Black Box Testing</i>	97
Tabel E.1 Biaya pengeluaran	100



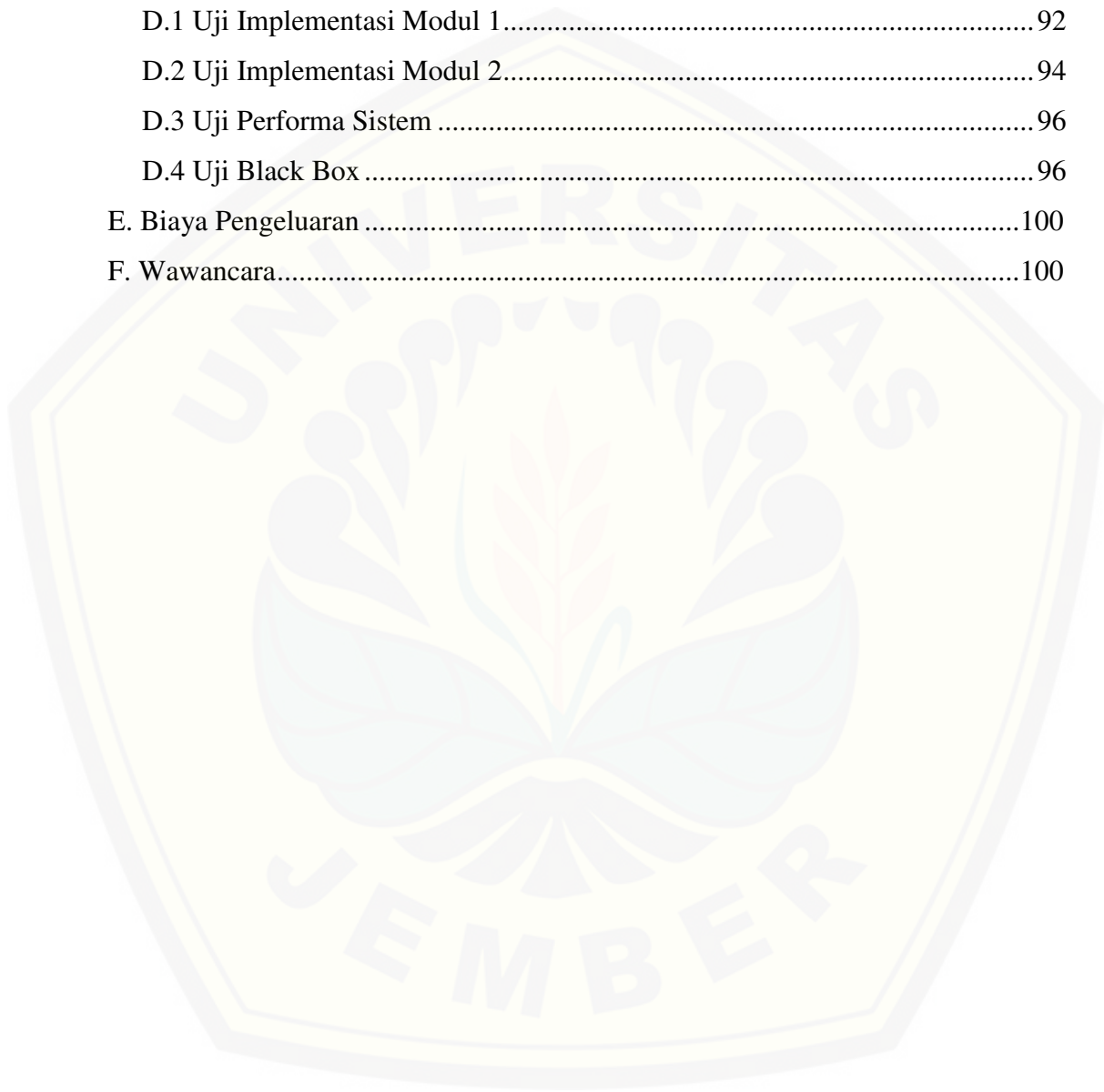
DAFTAR GAMBAR

Gambar 2.1 <i>Life Cycle PXP</i> (Aji, 2015)	6
Gambar 4.1 <i>Requirment</i> Kebutuhan Fungsional.....	19
Gambar 4.2 <i>Requirment</i> Kebutuhan non-fungsional	20
Gambar 2.3 <i>Usecase</i> Diagram.....	28
Gambar 4.4 <i>CRC Card</i>	29
Gambar 4.5 <i>Class diagram</i>	33
Gambar 4.6 Tampilan peta Flora	37
Gambar 4.7 Tampilan peta Fauna	39
Gambar B.1 Halaman <i>login</i>	86
Gambar B.2 Tampilan Data Fauna.....	87
Gambar B.3 Tampilan tambah data fauna.....	88
Gambar B.4 Tampilan ubah data fauna.....	88
Gambar B.5 Tampilan data flora.....	89
Gambar B.6 Tampilan tambah data flora.....	90
Gambar B.7 Tampilan ubah data flora.....	90
Gambar D.1 <i>Testing</i> performa sistem	96

DAFTAR LAMPIRAN

A. Kode Program	50
A.1 Kode program <i>login</i>	50
A.2 Kode program menambah data pemetaan cabang	51
A.3 Kode program melihat data cabang	53
A.4 Kode program mengubah data cabang	54
A.5 Kode program menambah data pemetaan pos pantau	57
A.6 Kode program melihat data pos pantau	59
A.7 Kode program mengubah data pos pantau	60
A.8 Kode program melihat peta cabang	62
A.9 Kode program melihat rute peta cabang	64
A.10 Kode program melihat detail rute peta cabang	65
A.11 Kode program melihat peta pos pantau	69
A.12 Kode program melihat rute peta pos pantau	70
A.13 Kode program melihat detail rute peta pos pantau	74
A.14 Kode program melihat data flora.....	78
A.15 Kode program mengubah data flora	80
A.16 Kode program melihat data fauna	82
A.17 Kode program mengubah data fauna.....	84
B. Tampilan Sistem.....	86
B.1 Halaman <i>login</i>	86
B.2 Fitur melihat data pemetaan fauna.....	86
B.3 Fitur menambah data pemetaan fauna	87
B.4 Fitur mengubah data pemetaan fauna	88
B.5 Fitur melihat data pemetaan flora	89
B.6 Fitur menambah data pemetaan flora	89
B.7 Fitur mengubah data pemetaan flora	90
C. Pengerjaan Sistem	91

C.1 Target pengerjaan	91
C.2 Laporan pengerjaan	91
C.3. <i>Feedback</i>	92
D. Testing	92
D.1 Uji Implementasi Modul 1.....	92
D.2 Uji Implementasi Modul 2.....	94
D.3 Uji Performa Sistem	96
D.4 Uji Black Box	96
E. Biaya Pengeluaran	100
F. Wawancara.....	100



BAB I PENDAHULUAN

Bab ini menjelaskan hal-hal yang menjadi dasar dalam penelitian dan pembuatan sistem informasi geografi pemetaan flora dan fauna di Taman Nasional Meru Betiri. Adapun yang akan dijelaskan antara lain adalah latar belakang penelitian, rumusan masalah, tujuan penelitian, batasan masalah, serta sistematika penulisan.

1.1 Latar Belakang

Taman Nasional Meru Betiri memiliki luas wilayah sekitar 58.000 ha, yang terbagi atas 57.155 ha daratan dan 845 ha perairan. Taman Nasional Meru Betiri terletak di wilayah pemerintahan Kabupaten Banyuwangi (20.415 ha) dan pemerintahan Kabupaten Jember (37.585 ha). Kawasan Taman Nasional Meru Betiri secara geografis terletak antara 113°38'48"- 113°58'30"BT dan 8°20'48"- 8°33'48"LS. Flora yang teridentifikasi di Taman Nasional Meru Betiri sebanyak 518 jenis, terdiri 15 jenis yang dilindungi dan 503 jenis yang tidak dilindungi. Sedangkan untuk fauna sebanyak 217 jenis, terdiri dari 92 jenis yang dilindungi dan 115 jenis yang tidak dilindungi. Jumlah sebanyak itu meliputi 25 jenis mamalia (18 diantaranya dilindungi), 8 reptilia (6 jenis diantaranya dilindungi), dan 184 jenis burung (68 jenis diantaranya dilindungi) (Siswoyo, 2002). Beberapa habitat yang dihuni oleh flora dan fauna diatas perlu dipetakan lokasinya untuk kepentingan penelitian lain yang berkaitan dengan flora dan fauna di Taman Nasional Meru Betiri.

Kebutuhan publikasi lokasi flora dan fauna untuk kepentingan pelayanan publik, yang biasanya digunakan oleh orang umum, seperti peneliti, wisatawan, serta pihak lainnya sangat dibutuhkan. Selama ini ketika orang ingin mengetahui lokasi dari flora dan fauna harus meminta data ke pihak TNMB kemudian memetakan sendiri di peta fisik atau menggunakan alat bantu GPS sehingga membutuhkan waktu lebih banyak dalam melakukan penelitian.

Pembuatan sistem yang sederhana dan cepat dapat membantu pihak TNMB dalam menyediakan informasi publik berupa peta lokasi flora dan fauna yang bisa diakses secara langsung. Terkait hal tersebut, pengurus TNMB tidak

menyediakan anggaran sehingga dalam pengembangannya diharuskan memakai biaya yang sedikit dan cepat.

Extreme Programming (XP) merupakan model *System Development Life Cycle* (SDLC) turunan *Agile* yang digunakan untuk pengembangan perangkat lunak berbasis kesederhanaan, komunikasi, umpan balik, dan keberanian. Implementasi XP memiliki *standard 12 core practice* sebagai parameter kesuksesan pengembangan sistem. *Personal Extreme Programming* (PXP) merupakan turunan dari XP yang menggunakan *programmer* tunggal. Penelitian sebelumnya yang berjudul “Aplikasi ISC (*Informatics Student Center*) Menggunakan Metode *Personal Extreme Programming* Berbasis Android” mengatakan, PXP memiliki kecenderungan dokumentasi yang sedikit dan sederhana sehingga proses yang digunakan dalam pengembangan sistem menjadi lebih cepat dibandingkan model lain dan penggunaan PXP berpengaruh besar pada kecepatan dan pengurangan biaya dalam proses pengembangan aplikasi ISC (Anjuliani & Astuti, 2015)

Penelitian ini mengimplementasikan model *Personal Extreme Programming* (PXP) dalam pengembangan sistem informasi geografis dalam memetakan lokasi flora dan fauna di Taman Nasional Meru Betiri mempercepat proses pengembangan sistem dan memperkecil biaya berdasarkan nilai dan *practice* yang ada di PXP.

1.2 Rumusan Masalah

Berdasarkan uraian diatas, permasalahan yang muncul adalah:

- 1) Bagaimana mengimplementasikan model *Personal Extreme Programming* (PXP) dalam pengembangan sistem informasi geografis pemetaan flora dan fauna di Taman Nasional Meru Betiri dengan standar *12 core practices extreme programming*?
- 2) Bagaimana implementasi nilai kesederhanaan, komunikasi, umpan balik, dan keberanian dalam proses pengembangan sistem informasi geografis pemetaan flora dan fauna di Taman Nasional Meru Betiri?

1.3 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini antara lain :

- 1) Membangun sistem informasi geografis pemetaan flora dan fauna di Taman Nasional Meru Betiri menggunakan model *personal extreme programming* (PXP) dengan standar 12 *core practice* PXP.
- 2) Mengetahui kesuaian nilai kemudahan, komunikasi, umpan balik, dan keberanian dalam proses pengembangan sistem informasi geografis pemetaan flora dan fauna di Taman Nasional Meru Betiri.

1.4 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

- 1) Fokus penelitian adalah implementasi *Personal Extreme Programming* dalam pembangunan sistem.
- 2) Flora dan fauna yang dipetakan adalah hasil inventarisasi dari pengelola Taman Nasional Meru Betiri.
- 3) Sistem khusus memetakan lokasi dari flora dan fauna menggunakan inputan data koordinat.

1.5 Sistematika Penulisan

Adapun sistematika penulisan skripsi ini adalah sebagai berikut:

- 1) Pendahuluan
Bab pendahuluan merupakan langkah awal dari penyusunan tugas akhir yang membahas latar belakang, rumusan masalah, tujuan, batasan masalah, dan sistematika penulisan.
- 2) Tinjauan Pustaka
Bab tinjauan pustaka merupakan bab yang menjelaskan teori-teori yang melandasi penelitian, tinjauan pustaka, dan studi terdahulu yang menjadi acuan dalam penelitian.
- 3) Metodologi Penelitian
Bab metodologi penelitian jenis penelitian dan tahapan penelitian sesuai dengan model pengembangan sistem yang dipakai.
- 4) Pengembangan Sistem

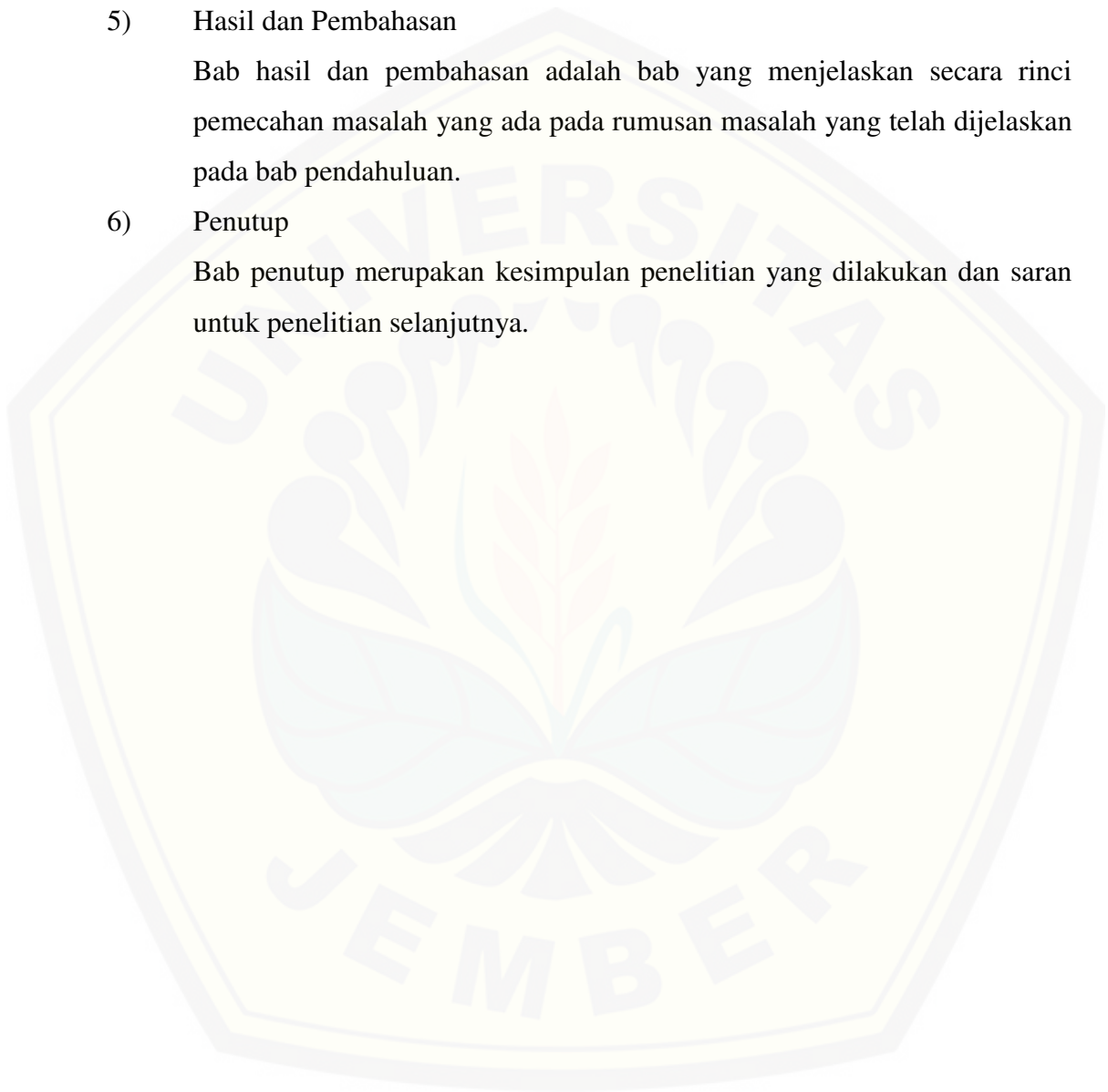
Bab pengembangan sistem adalah bab yang berisi proses pengembangan sistem yang dibangun dalam penelitian. Adapun tahapan-tahapan pengembangan sistem ini didasarkan pada metodologi penelitian yang telah ditulis pada bab sebelumnya.

5) Hasil dan Pembahasan

Bab hasil dan pembahasan adalah bab yang menjelaskan secara rinci pemecahan masalah yang ada pada rumusan masalah yang telah dijelaskan pada bab pendahuluan.

6) Penutup

Bab penutup merupakan kesimpulan penelitian yang dilakukan dan saran untuk penelitian selanjutnya.



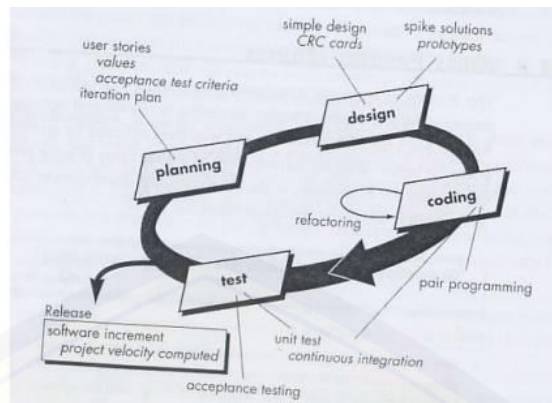
BAB II TINJAUAN PUSTAKA

Bab ini menguraikan teori serta konsep yang menjadi kerangka pemikiran dalam penelitian. Teori dan konsep tersebut didapatkan dari penelitian terdahulu serta kajian pustaka terkait penelitian.

2.1 *Extreme Programming (XP)*

Extreme Programming (XP) adalah proses pengembangan perangkat lunak dirancang untuk diterapkan oleh pengembang perangkat lunak secara individu. XP menjaga prinsip dasar mengurangi jumlah dokumentasi dan upaya pemeliharaan. Proses pengembangan XP bersifat fleksibel dan responsif untuk perubahan. Metodologi ini ditujukan untuk meningkatkan kinerja *programmer* dan mempersingkat pengkodean dan waktu yang digunakan untuk mendukung sistem perangkat lunak. Metodologi XP didasarkan pada prinsip-prinsip berikut (Rizal, 2014):

- 1) PXP memerlukan pendekatan disiplin, pengembang bertanggung jawab untuk mengikuti proses dan menerapkan praktik XP
- 2) Pengembang harus mengukur, melacak dan menganalisis pekerjaan sehari-hari mereka.
- 3) Pengembang harus belajar dari variasi kinerja mereka dan memperbaiki proses berdasarkan data proyek yang terkumpul
- 4) XP melibatkan pengujian terus menerus
- 5) Perbaikan cacat harus terjadi pada tahap pengembangan awal, bila biaya itu lebih rendah
- 6) Pengembang harus mencoba mengotomatisasi sebanyak mungkin hari mereka kerja



Gambar 2.1 Life Cycle PXP (Aji, 2015)

Gambar 2.1 merupakan alur dari metode *Personal Extreme Programming* dengan rincian sebagai berikut:

1) *Planning*

Pada tahap ini, perencanaan fungsionalitas keseluruhan yang akan dikembangkan dalam sistem. Seperti rencana pengolahan data, rencana fitur, rencana alur sistem, rencana pengerjaan sistem.

2) *Design*

Tahap ini merupakan pembuatan desain dari sistem. Desain yang dipakai dalam PXP adalah *Usecase Diagram*, *CRC Card*, dan *Class Diagram*.

Usecase diagram adalah *diagram* yang digunakan untuk menunjukkan hubungan antara fitur dan aktor yang dipakai pada sistem

CRC Card adalah kartu yang menunjukkan hubungan antara *class* dengan fungsi dan tanggungjawabnya serta menunjukkan kolaborasinya dengan *class* lain.

Class diagram adalah *diagram* yang menunjukkan hubungan antar *class*, dengan atribut dan *method* pada koding.

3) *Coding*

Pada tahapan ini desain sistem yang telah dibuat akan diimplementasikan secara langsung dalam sebuah sistem informasi geografis berbasis *website*. Pengkodean program menggunakan bahasa pemrograman PHP dengan *tool Sublime Text* serta mengatur manajemen basis data menggunakan *tool Xampp*. Apabila kode selesai kemudian kode tersebut diuji dalam unit *testing*, apabila ada kesalahan maka dilakukan koreksi ulang atau *refactor* pada tahap dimana

kesalahan tersebut bermula, apabila tidak ada kesalahan maka dilanjutkan ke unit selanjutnya.

4) *Test*

Pengujian yang dilakukan pada penelitian ini yaitu dengan menggunakan pengujian *black box*. *Black Box Testing* merupakan uji fungsionalitas apakah masih ada kekurangan atau sudah cukup sesuai.

Extreme programming memiliki 12 *core practices* (Ariaji et al., 2014) yaitu:

1) *Planning game*

Planning game merupakan *practice* yang digunakan untuk melakukan perencanaan dan melakukan prioritas terhadap fitur-fitur yang dituliskan pada tabel kebutuhan oleh *customer*.

2) *Small release*

Small releases adalah rilis yang dihasilkan untuk setiap iterasi sangat pendek dan dengan umpan balik terhadap perubahan dari *customer* juga sangat cepat.

3) *Metaphor*.

Metaphor adalah semacam *simple guidance* bagi proses pengembangan dari fase paling awal hingga terakhir.

4) *Simple design*.

Simple design merupakan rancangan yang sederhana untuk diproses pada setiap iterasi.

5) *Testing*.

Testing yang dilakukan adalah menguji performa sistem dan pengujian fitur.

6) *Refactoring*.

Refactoring adalah proses untuk memperbaiki *code* selain untuk menghindari berbagai redundansi yang mungkin terjadi.

7) *Pair programming*

Pair programming yaitu 2 orang *programmer* bekerja dalam 1 komputer. Model kerja ini dapat meningkatkan kualitas perangkat lunak tanpa mempengaruhi waktu pengerjaan. Dengan meningkatnya kualitas maka dapat meningkatkan penghematan proyek.

8) *Collective ownership*

Collective ownership adalah keadaan dimana semua anggota tim harus dapat menanggulangi semua hal yang berkaitan dengan proses pengembangan.

9) *Continuous integration*

Continuous integration bahwa proses pengembangan setiap hari bahkan setiap saat terdapat perubahan harus segera diintegrasikan.

10) *40-hour week*

40-hour week adalah jumlah jam kerja selama satu minggu. Hal ini sebenarnya tidak mutlak 40 jam, namun intinya adalah bahwa proses pengembangan tidak mengenal lembur, semua harus diselesaikan pada saat jam kerja. Semua masalah dan pekerjaan harus dioptimalkan pada waktu jam kerja tersebut.

11) *On-site customer*

On-site customer merupakan salah satu keunggulan bagi metodologi ini karena XP memerlukan satu orang dari pihak pemesan yang dibawa dalam proses pengembangan dari awal sampai berakhir. Kecepatan dalam mengatasi berbagai *requirements* yang mungkin berubah atau bertambah.

12) *Coding Standard*

Coding Standard adalah menstandarkan proses *coding* bagi *programmer* terutama karena hal ini berkaitan dengan *practice pair programming*

Nilai-nilai dalam XP

Kesederhaan

Kesederhanaan diterapkan dalam melakukan pengembangan sistem, terutama saat melakukan *coding*. Susunan *coding* yang sederhana dengan menggunakan method yang pendek namun efisien.

Komunikasi

XP sangat memperhatikan komunikasi antar setiap anggota tim sendiri maupun dengan pihak klien. Komunikasi yang baik antar tim akan mempermudah proses kerja proyek dimana setiap anggota tim bisa saling bantu dalam memecahkan sebuah masalah, terutama menghilangkan ego dari programmer yang cukup tinggi.

Umpan Balik

Umpan balik sangat diandalkan dalam proses model XP. Setiap anggota tim wajib memberikan setiap masalah kepada tim yang lainnya. Masukan, kritikan dan ide baru yang didapat bisa menjadi sebuah perbaikan maupun peningkatan dalam menyelesaikan setiap tugas.

Keberanian

Sebuah proyek yang memiliki tenggat waktu akan menghadapi tekanan dari berbagai pihak, terutama klien. XP menerapkan nilai keberanian untuk tetap memiliki integritas pada setiap anggota tim. Dengan integritas tersebut anggota-anggota tim akan memiliki keyakinan dapat menyelesaikan proyek sesuai dengan yang diharapkan. Kepercayaan itulah yang dibangun sehingga semangat untuk segera dapat menyelesaikan dapat terus terbentuk.

Model *Personal Extreme Programming* merupakan model *Extreme Programming* yang dilakukan oleh programmer tunggal, semua prinsip, nilai, dan *practice*-nya sama dengan XP.

Penelitian sebelumnya yang berjudul “Aplikasi ISC (*Informatics Student Center*) Menggunakan Metode *Personal Extreme Programming* Berbasis Android” Pengumpulan informasi saat ini dapat diklasifikasikan sebagai kebutuhan primer. Berbagai macam kegiatan bisa dilakukan dengan menggunakan teknologi, terutama *smartphone*, misalnya belajar, berinteraksi dengan orang lain, mencari informasi, atau hanya untuk hiburan. Banyak jenis kegiatan dan beragamnya untuk mengaksesnya. Agar bisa belajar dan mendapatkan informasi dapat dilakukan sekaligus dan mudah diakses, maka peneliti melakukan penelitian yang bertujuan untuk membangun aplikasi *Informatics Student Center* (ISC) yang dikembangkan dengan menggunakan *Personal Extreme Programming* (XP) terdiri dari beberapa tahap, seperti persyaratan, perencanaan, inisialisasi iterasi, desain, implementasi, dan pengujian sistem. Aplikasi ISC menggunakan sistem operasi *Android*, dibangun dengan kolaborasi bahasa Java, PHP, dan *database* MySQL. Selain kemudahan akses, ISC memiliki beberapa fitur yang menjadi informasi akademis, forum diskusi, hiburan, dan kesempatan kerja. Berdasarkan

hasil uji, disimpulkan bahwa aplikasi ISC bisa berjalan di perangkat *mobile* Android yang diuji. Hasil dari penelitian ini adalah tersedianya aplikasi pendukung kegiatan belajar dan akses informasi yang bisa diakses secara *online* melalui perangkat *mobile* Android. Penggunaan PXP berpengaruh besar pada kecepatan dan pengurangan biaya dalam proses pengembangan aplikasi *Informatics Student Center* (ISC) (Anjuliani & Astuti, 2015).

Penelitian Implementasi Model *Personal Extreme Programming* dalam Pengembangan SIG Pemetaan Flora dan Fauna di TNMB berfokus pada penerapan model PXP agar pengembangan SIG lebih maksimal, sedangkan dalam penelitian sebelumnya berfokus pada sistem yang dikembangkan.

2.2 Taman Nasional Meru Betiri

Taman Nasional Meru Betiri (TNMB) merupakan salah satu kawasan pelestarian alam yang memiliki potensi flora, fauna dan ekosistem serta gejala dan keunikan alam yang dapat dikembangkan sebagai obyek dan daya tarik wisata alam (ODTWA) (Siswoyo, 2002). Kawasan Taman Nasional Meru Betiri merupakan hutan hujan tropis dengan formasi hutan bervariasi yang terbagi ke dalam 5 tipe vegetasi yang tersebar di seluruh kawasan ini. Kondisi setiap tipe vegetasi di kawasan Taman Nasional Meru Betiri dapat dijelaskan sebagai berikut :

1. Tipe Vegetasi Hutan Pantai

Formasi vegetasi hutan pantai terdiri dari 2 tipe utama yaitu formasi ubi pantai (*Ipomea pescaprae*), dan formasi *Barringtonia* (25 - 50 m) pada daerah pantai yang landai dan akan berkurang luasnya jika pantainya terjal dan berbatu. Jenis yang paling banyak adalah ubi pantai (*Ipomoea pescaprae*) dan rumput lari (*Spinifex squarosus*). Formasi *Barringtonia* terdiri dari keben (*Barringtonia asiatica*), nyamplung (*Calophyllum inophyllum*), waru (*Hibiscus tiliaceus*), ketapang (*Terminalia catappa*), pandan (*Pandanus tectorius*) dan lain-lain.

2. Tipe Vegetasi Hutan Mangrove

Vegetasi ini dapat dijumpai di bagian timur Teluk Rajegwesi yang merupakan muara Sungai Lembu dan Karang Tambak, Teluk Meru dan Sukamade

merupakan vegetasi hutan yang tumbuh di garis pasang surut. Jenis-jenis yang mendominasi adalah Pedada (*Sonneratia caseolaris*), Tancang (*Bruguiera gymnorhiza*) dan Nipah (*Nypa fructicans*).

3. Tipe Vegetasi Hutan Rawa

Vegetasi ini dapat dijumpai di belakang hutan payau Sukamade. Jenis-jenis yang banyak dijumpai diantaranya mangga hutan (*Mangifera sp*), sawo kecil (*Manilkara kauki*), ingas/rengas (*Gluta renghas*), pulai (*Alstonia scholaris*), kepuh (*Sterculia foetida*), dan *Barringtonia spicota*.

4. Tipe Vegetasi Hutan *Rheophyt*

Tipe vegetasi ini terdapat pada daerah-daerah yang dibanjiri oleh aliran sungai dan jenis vegetasi yang tumbuh diduga dipengaruhi oleh derasnya arus sungai, seperti lembah Sungai Sukamade, Sungai Sanen, dan Sungai Bandalit. Jenis yang tumbuh antara lain glagah (*Saccharum spontanum*), rumput gajah (*Panisetum curcurium*) dan beberapa jenis herba berumur pendek serta rumput-rumputan.

5. Tipe Vegetasi Hutan Hujan Tropika Dataran Rendah

Sebagian besar kawasan hutan Taman Nasional Meru Betiri merupakan tipe vegetasi hutan hujan tropika dataran rendah. Pada tipe vegetasi ini juga tumbuh banyak jenis epifit, seperti anggrek dan paku-pakuan serta liana. Jenis tumbuhan yang banyak dijumpai diantaranya jenis walangan (*Pterospermum diversifolium*), winong (*Tetrameles nudiflora*), gondang (*Ficus variegata*), budengan (*Diospyros cauliflora*), pancal kidang (*Aglaia variegata*), rau (*Dracontomelon mangiferum*), glintungan (*Bischoffia javanica*), ledoyo (*Dysoxylum amoroides*), randu agung (*Gossampinus heptaphylla*), nyampuh (*Litsea sp*), bayur (*Pterospermum avanicum*), bungur (*Lagerstromia speciosa*), segawe (*Adenantha microsperma*), aren (*Arenga pinnata*), langsung (*Langsium domesticum*), bendo (*Artocarpus elasticus*), suren (*Toona sureni*), dan durian (*Durio sibethinus*). Terdapat pula vegetasi bambu seperti, bambu bubat (*Bambusa sp*), bambu wuluh (*Schizastychyum blumei*), dan bambu lamper (*Schizastychyum branchyladium*).

Kawasan Meru Betiri juga terdapat beberapa jenis rotan, diantaranya, rotan manis (*Daemonorops melanocaetes*), rotan slatung (*Plectmocomia longistigma*), rotan warak (*Plectmocomia elongata*) dan lain-lain. Hingga saat ini di kawasan Taman Nasional Meru Betiri telah teridentifikasi flora sebanyak 518 jenis, terdiri 15 jenis yang dilindungi dan 503 jenis yang tidak dilindungi. Contoh jenis yang dilindungi yaitu Balanopora (*Balanophora fungosa*) yaitu tumbuhan parasit yang hidup pada jenis pohon *Ficus sp.* dan Padmosari/Rafflesia (*Rafflesia zollingeriana*) yang hidupnya tergantung pada tumbuhan inang *Tetrastigma sp.* Selain itu terdapat pula jenis flora sebagai bahan baku obat/jamu tradisional, dimana berdasarkan hasil uji petik di lapangan telah teridentifikasi sebanyak 239 jenis yang dapat dikelompokkan dalam 7 habitus, yaitu bambu, memanjat, herba, liana, perdu, semak dan pohon.

Kawasan hutan Meru Betiri merupakan habitat terakhir harimau jawa (*Panthera tigris sondaica*). Pada tahun 1976 oleh WWF dilaporkan bahwa harimau jawa yang ada di Meru Betiri tinggal 5 ekor atau paling banyak 6 ekor. Perjumpaan secara langsung terhadap satwa ini tidak pernah ada, namun beberapa inventarisasi yang dilakukan menunjukkan adanya tanda-tanda harimau jawa di kawasan ini yaitu berupa cakaran dan kotoran. Jenis satwa lain yang potensial dan perlu mendapatkan perhatian khusus adalah populasi penyu yang sering bertelur di Pantai Sukamade. Pantai ini merupakan habitat bertelur bagi penyu hijau (*Chelonia mydas*) dan penyu sisik (*Eretmochelys imbricata*), serta jenis penyu lainnya seperti penyu slengkrah (*Lepidochelys olivacea*) dan penyu belimbing (*Dermochelys coriacea*).

Upaya pelestarian penyu yang dilakukan ditempuh melalui kegiatan pengamanan pantai, pengumpulan telur, pembuatan tempat penetasan semi permanen, pemeliharaan telur yang ditetaskan, pemeliharaan tukik yang baru menetas, pemeliharaan tukik di tempat penampungan, *tagging*, *sexing*, pencatatan data jumlah penyu, pencatatan data jumlah telur, penyuluhan, pelayanan penelitian, pelepasan tukik ke laut, pendidikan dan pelatihan untuk pelajar dan mahasiswa (Puroso, 2014). Beberapa jenis satwa yang terdapat di dalam kawasan Meru Betiri antara lain kijang (*Muntiacus muntjak*), banteng (*Bos javanicus*),

macan tutul (*Panthera pardus*), babi hutan (*Sus sp*), rusa (*Cervus timorensis*), kancil (*Tragulus javanicus*), musang luwak (*Phardoxorus hermiprodytus*), kukang (*Nycticebus caoncang*), landak (*Hystrix brachiura*), monyet ekor panjang (*Macaca fascicularis*), kera hitam/lutung budeng (*Trachypithecus auratus*), kera (*Macaca irus*), trenggiling (*Manis javanicus*).

Beberapa jenis burung seperti burung elang Jawa (*Spizateus bartelsi*), burung ular bodo (*Spilormis cheela*), burung laut perut putih (*Haliaeetus leucogaster*), burung elang hitam (*Ictinaetus malayensis*), burung elang bondol (*Haliastur indus*), burung elang brontok (*Spizaetus cirrhatus*), burung elang kelabu (*Butastur indicus*), burung sikep madu asia (*Pernis ptilorynchus*), burung kukuk beluk (*Strix leptogrammica*), burung alap-alap capung (*Microhierax fringillarius*), burung merak (*Pavo muticus*), burung rangkong (*Buceros rhinoceros*), serta beberapa jenis burung lainnya.

2.3 Sistem Informasi Geografis (SIG)

Sistem Informasi Geografis (SIG) merupakan sistem yang dirancang untuk bekerja dengan data yang terreferensi secara spasial atau koordinat-koordinat geografi. SIG memiliki kemampuan untuk melakukan pengolahan data dan melakukan operasi-operasi tertentu dengan menampilkan dan menganalisa data (Masykur, 2014).

Aplikasi GIS saat ini tumbuh tidak hanya secara jumlah aplikasi namun juga bertambah dari jenis keragaman aplikasinya. Sebagai contoh adalah adanya peta online sebuah daerah dimana pengguna dapat dengan mudah mencari lokasi yang diinginkan secara online melalui jaringan intranet/internet tanpa mengenal batas geografi penggunaannya. Secara umum Sistem Informasi Geografis dikembangkan berdasarkan pada prinsip input/masukan data, manajemen, analisis dan representasi data.

Karakteristik SIG

1. Merupakan suatu sistem hasil pengembangan perangkat keras dan perangkat lunak untuk tujuan pemetaan, sehingga fakta wilayah dapat disajikan dalam satu sistem berbasis komputer.
2. Melibatkan ahli geografi, informatika dan komputer, serta aplikasi terkait.

3. Masalah dalam pengembangan meliputi: cakupan, kualitas dan standar data, struktur, model dan visualisasi data, koordinasi kelembagaan dan etika, pendidikan, *expert system* dan *decision support system* serta penerapannya
4. Bukan hanya sekedar merupakan perubahan peta konvensional (tradisional) ke bentuk peta digital untuk kemudian disajikan (dicetak / diperbanyak) kembali
5. Mampu mengumpulkan, menyimpan, mentransformasikan, menampilkan, memanipulasi, memadukan dan menganalisis data spasial dari fenomena geografis suatu wilayah.
6. Mampu menyimpan data dasar yang dibutuhkan untuk penyelesaian suatu masalah.

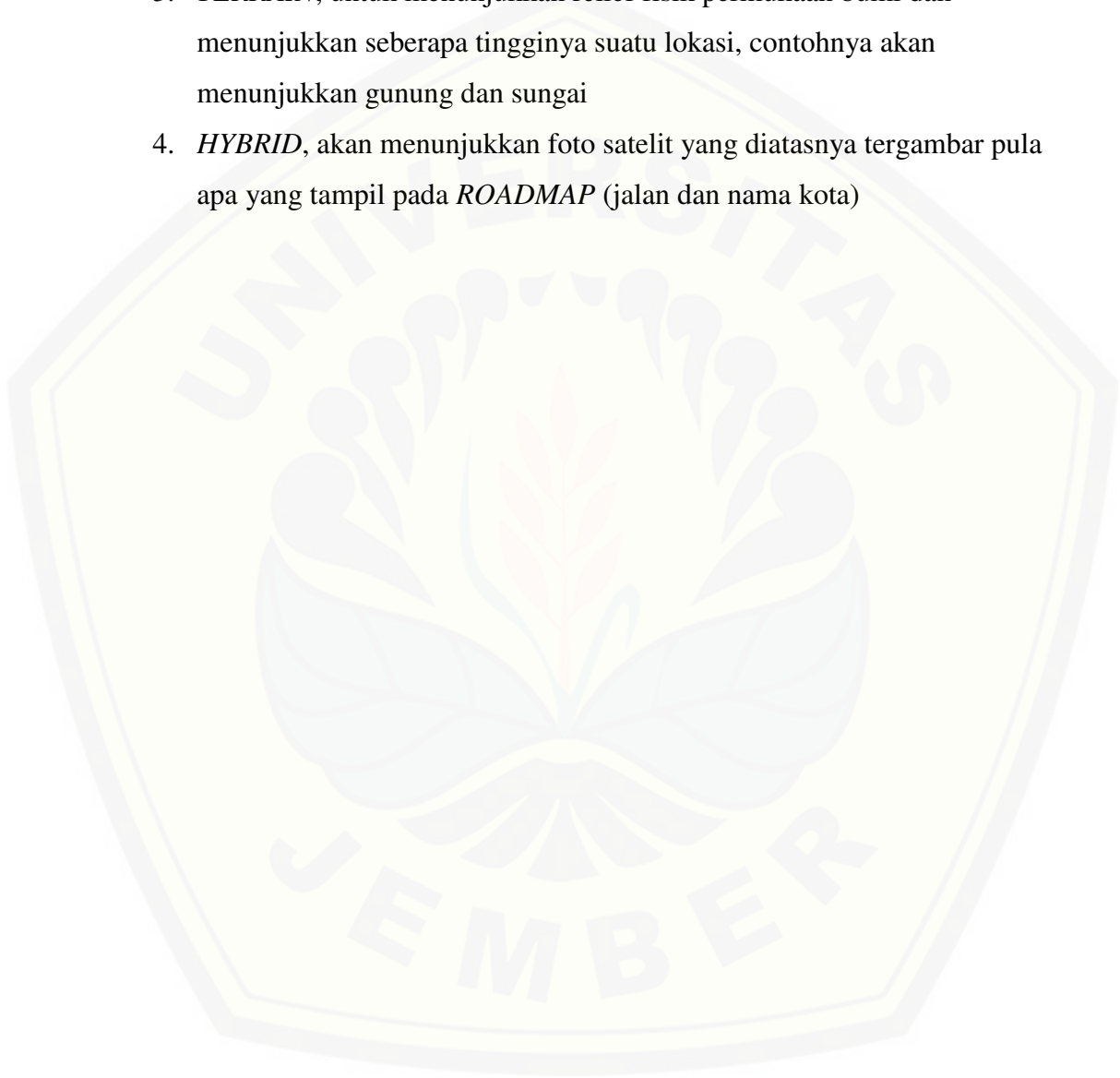
2.4 *Google Maps API*

Google Maps API adalah suatu *library* yang berbentuk *JavaScript*. Cara membuat *Google Maps* untuk ditampilkan pada suatu *web* membutuhkan pengetahuan mengenai HTML serta *JavaScript*, serta koneksi Internet yang sangat stabil. Dengan menggunakan *Google Maps API*, kita dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal, sehingga kita dapat fokus hanya pada data-data yang ditampilkan (Masykur, 2014). Dengan kata lain, kita hanya membuat suatu data sedangkan peta yang ditampilkan adalah milik *Google* sehingga kita tidak dipusingkan dengan membuat peta suatu lokasi, bahkan dunia. Dalam pembuatan program *Google Map API* menggunakan urutan sebagai berikut:

1. Memasukkan *Maps API JavaScript* ke dalam HTML.
2. Membuat *element div* dengan *map_canvas* untuk menampilkan peta.
3. Membuat beberapa objek literal untuk menyimpan properti-properti pada peta.
4. Menuliskan fungsi *JavaScript* untuk membuat objek peta.
5. Menginisiasi peta dalam *tag body* HTML dengan *event onload*.

Pada *Google Maps API* terdapat 4 jenis pilihan model peta yang disediakan oleh *Google* (Ichtiara, 2008), diantaranya adalah:

1. *ROADMAP*, ini yang saya pilih, untuk menampilkan peta biasa 2 dimensi
2. *SATELLITE*, untuk menampilkan foto satelit
3. *TERRAIN*, untuk menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi, contohnya akan menunjukkan gunung dan sungai
4. *HYBRID*, akan menunjukkan foto satelit yang di atasnya tergambar pula apa yang tampil pada *ROADMAP* (jalan dan nama kota)



BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan tentang sekumpulan metode yang digunakan dalam penelitian pengembangan sistem. Pada bab ini juga dijelaskan mengenai langkah dan prosedur yang dilakukan dalam pengumpulan data atau informasi guna memecahkan permasalahan dalam penelitian.

3.1 Jenis Penelitian

Penelitian ini menggunakan jenis penelitian kualitatif. Jenis penelitian kualitatif digunakan untuk mendapatkan kebutuhan sistem melalui wawancara dan studi literatur, yang kemudian dianalisa secara kualitatif untuk mendapatkan data flora dan fauna yang selanjutnya digunakan dalam proses pemetaan dan pengembangan sistem.

3.2 Tahapan Penelitian

Tahapan penelitian pengembangan sistem ini dilakukan dengan menggunakan model *Personal Extreme Programming* (PXP), model PXP dipilih karena pada penelitian terdahulu model ini cocok digunakan untuk pengembangan sistem yang membutuhkan waktu yang cepat dalam pengembangan untuk mengetahui kekurangan sistem agar memudahkan memperbaiki sistem ketika ada kesalahan seperti pada penelitian Anjuliani yang mengembangkan aplikasi ISC (*Informatics Student Center*) menggunakan metode *Personal Extreme Programming* berbasis *android*.

Adapun model PXP memiliki tahapan-tahapan seperti yang dapat dilihat pada bab dua. Alur dari metode *Personal Extreme Programming* yang akan digunakan pada penelitian ini dengan rincian sebagai berikut:

3.2.1 *Planning*

Pada tahap ini bertujuan untuk memperoleh data atau informasi yang dibutuhkan dalam pengembangan sistem. Pengumpulan data (*requirement*) pada penelitian ini dilakukan melalui beberapa cara yaitu:

- a. Studi literatur, jurnal, serta mencari informasi menggunakan media internet mengenai pemetaan, flora, fauna, dan penggunaan *Google Maps API*
- b. Wawancara kepada kepala Balai Taman Nasional Meru Betiri guna mengetahui data inventarisasi flora dan fauna di Taman Nasional Meru Betiri yang nantinya akan diolah dalam penelitian.

Adapun data dan informasi yang didapatkan dari tahap ini akan digunakan dalam penelitian ini.

Tahap selanjutnya yaitu melakukan perencanaan fungsionalitas keseluruhan yang akan dikembangkan dalam sistem. Beberapa perencanaan yang dibuat meliputi:

- 1) Rencana pengolahan data: data yang didapat dijadikan satu untuk kemudian dianalisa sebagai bahan penyusun sistem.
- 2) Rencana fitur: melakukan perencanaan fitur yang diperoleh dari wawancara.
- 3) Rencana alur sistem: merencanakan alur sistem yang akan dibuat.
- 4) Rencana pengerjaan sistem: merencanakan pengerjaan sistem yang telah diketahui fitur dan alurnya.

3.2.2 Design

Tahap ini merupakan pembuatan desain dari sistem. Desain yang dipakai dalam PXP adalah *Usecase Diagram*, *CRC Card*, dan *Class Diagram*.

3.2.3 Implementation

Pada tahapan ini desain sistem yang telah dibuat akan diimplementasikan secara langsung dalam sebuah sistem informasi geografis berbasis *web*. Pengkodean program menggunakan bahasa pemrograman PHP dengan *tool Sublime Text* serta mengatur manajemen basis data menggunakan *tool Xampp*. Apabila kode selesai kemudian kode tersebut diuji dalam *unit testing*, apabila ada kesalahan maka dilakukan koreksi ulang atau *refactor* pada tahap dimana kesalahan tersebut bermula, apabila tidak ada kesalahan maka dilanjutkan ke unit selanjutnya.

3.2.4 *System Testing*

Pengujian yang dilakukan pada penelitian ini yaitu dengan menggunakan pengujian *black box*. *Black Box Testing* merupakan uji fungsionalitas apakah masih ada kekurangan atau sudah cukup sesuai.

Testing juga dilakukan pada tahapan *design* dan *implementation*.



BAB IV PENGEMBANGAN SISTEM

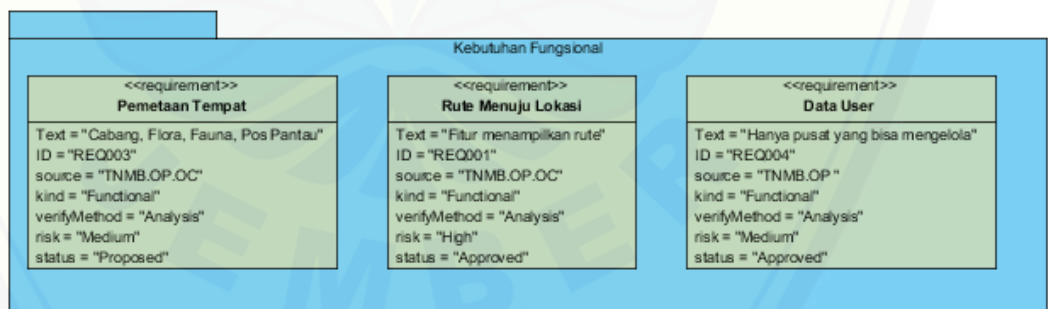
Bab ini menjelaskan tahapan pengembangan sistem informasi geografis pemetaan flora dan fauna. Tahapan-tahapan perancangan dilaksanakan berdasarkan model *Personal Extreme Programming* seperti yang telah dijelaskan pada bab tiga.

4.1 *Planning*

Tahap ini menggunakan hasil dari pengumpulan data yang telah dijabarkan pada bab sebelumnya sebagai dasar penentuan kebutuhan sistem.

4.1.1 *Requirement*

Kebutuhan sistem pada tahapan ini didefinisikan menjadi kebutuhan fungsional dan kebutuhan non-fungsional sistem. Kebutuhan sistem didefinisikan melalui proses pengolahan hasil wawancara dengan pihak Taman Nasional Meru Betiri yang bertujuan untuk menentukan apa saja fungsi dari setiap fitur sistem yang dikembangkan, sehingga menghasilkan kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional adalah kebutuhan yang menggambarkan proses yang mampu dilakukan oleh sistem. Kebutuhan fungsional dari sistem ini bisa dilihat di Gambar 2 :



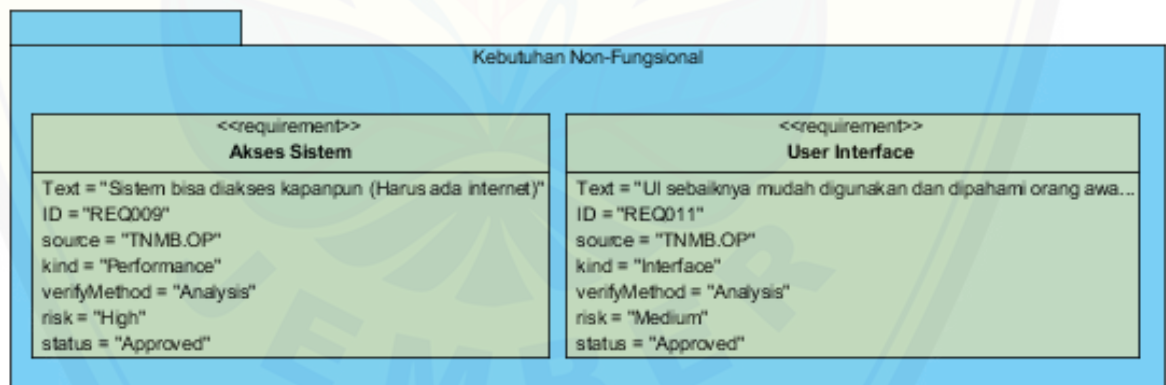
Gambar 4.1 *Requirement* Kebutuhan Fungsional

Kebutuhan fungsional didapatkan dari hasil requirement seperti yang ditampilkan di tabel kebutuhan fungsional.

Tabel 4.1 Kebutuhan fungsional

Requirement	ID	Fitur
Pemetaan Tempat	RQ003	Mengelola data peta cabang
		Mengelola data peta pos pantau
		Mengelola data peta flora
		Mengelola data peta fauna
Rute Menuju Tempat	RQ001	Melihat detail rute cabang
		Melihat detail rute pos pantau
		Melihat peta flora
		Melihat peta fauna
Data User	RQ004	Mengelola data user

Kebutuhan non-fungsional adalah kebutuhan yang tidak berkaitan dengan fungsi sistem tetapi diperlukan untuk mendukung aktivitas sistem dalam memenuhi kebutuhan fungsionalnya. Hasil dari kebutuhan non-fungsional dari sistem ini bisa dilihat di Gambar 4.2.



Gambar 4.2 Requirement Kebutuhan non-fungsional

Kebutuhan non-fungsional didapatkan dari requirement yang bisa dilihat pada tabel kebutuhan non-fungsional

Tabel 4.2 Kebutuhan non-fungsional

Requirement	ID	Fitur
Akses Sistem	RQ009	Sistem bisa diakses kapanpun selama

Requirment	ID	Fitur
		ada koneksi internet
		Sistem bisa diakses dimanapun selama ada koneksi internet
User Interface	RQ011	Sistem memiliki UI yang mudah dipahami orang awam

Sistem informasi geografis pemetaan flora dan fauna harus bisa diakses dimana kapan saja selama ada koneksi internet, dan tampilan dari sistem harus mudah dipahami oleh orang awam.

4.1.2 Perencanaan sistem

Tahapan perencanaan sistem digunakan untuk menyusun perencanaan yang dibutuhkan dalam pengembangan sistem informasi geografis pemetaan flora dan fauna. Setelah mendapatkan kebutuhan fungsional dan non-fungsional selanjutnya membuat beberapa perencanaan yang digunakan dalam PXP, yaitu perencanaan pengolahan data, perencanaan fitur, dan perencanaan pengerjaan sistem :

1) Perencanaan pengolahan data

Perencanaan pengolahan data dari wawancara adalah data *sampling*. Data *sampling* yang dimaksud adalah data contoh yang digunakan untuk mengembangkan sistem informasi geografis pemetaan flora dan fauna. Data yang diperoleh dari wawancara adalah data flora, data fauna, data pos pantau, dan data cabang yang berisikan nama, lokasi, koordinat, dan untuk data flora fauna memiliki entitas populasi. Proses pengerjaan sistem membutuhkan data ini untuk melakukan pemetaan, data ini merupakan inventaris di Taman Nasional Meru Betiri. Data flora akan digunakan untuk pemetaan tumbuhan, data fauna akan digunakan untuk pemetaan hewan, data pos pantau akan digunakan untuk pemetaan pos pantau, dan data cabang akan digunakan untuk pemetaan cabang dan pengelolaan akun *user*.

Data *sampling* yang didapat yaitu data flora, data fauna, data pos pantau, dan data cabang.

Tabel 4.3 *Sampling* data flora

No	Nama	Lokasi	Koordinat
1	Mangrove Pedada	Andongrejo	-8.488238499156186, 113.72526174609379
2	Mangrove Tancang	Andongrejo	-8.50182074115527, 113.77195364062504
3	Sawo Kecil	Andongrejo	-8.408093514907762, 113.85847097460942
4	Rafflesia	Andongrejo	-8.445246, 113.769321
5	Bayur	Andongrejo	-8.48416373281834, 113.74311452929692
6	Aren	Andongrejo	-8.455639159200263, 113.68543630664067
7	Bendo	Sukamade	-8.505895319939267, 113.92438894335942
8	Mangga Hutan	Sukamade	-8.482805467765198, 113.90241628710942

Tabel 4.4 *Sampling* data fauna

No	Nama	Lokasi	Koordinat
1	Kijang	Andongrejo	-8.39450797241986, 113.87083059375004
2	Banteng	Andongrejo	-8.466505912757164, 113.80216604296879
3	Monyet Ekor Panjang	Andongrejo	-8.450205667499695, 113.84885793750004
4	Harimau Jawa	Andongrejo	-8.425754008218009, 113.88319021289067
5	Macan Tutul	Andongrejo	-8.433904733282901, 113.87083059375004
6	Kucing Hutan	Andongrejo	-8.423037061687097, 113.77607351367192
7	Ajag	Sukamade	-8.488238499156186, 113.91614919726567
8	Rusa	Sukamade	-8.533058070067284, 113.87220388476567
9	Bajing Terbang	Andongrejo	-8.428470935664931,

No	Nama	Lokasi	Koordinat
	Ekor Merah		113.88593679492192
10	Merak Hijau	Andongrejo	-8.456997520158705, 113.73487478320317
11	Penyu Hijau	Sukamade	-8.560218888210782, 113.88559347216801
12	Babi Hutan	Andongrejo	-8.444772099248075, 113.86671072070317

Tabel 4.5 *Sampling* data pos pantau

No	Nama	Lokasi	Koordinat
1	Pos Andongrejo	Andongrejo, Kab. Jember	-8.468438, 113.739765
2	Pos Sukamade	Sukamade, Kab. Banyuwangi	-8.55819, 113.887322

Tabel 4.6 *Sampling* data cabang

No	Nama	Lokasi	Koordinat
1	Balai TNMB Kalibaru	Jl. Raya Jember, Kalibaruwetan, Kalibaru, Kabupaten Banyuwangi, Jawa Timur 68467	-8.296561, 113.989352
2	Kantor Seksi SPTN 2 Ambulu Taman Nasional Meru Betiri	Jalan Ahmad Yani, Permai, Ambulu, Jember, Kabupaten Jember, Jawa Timur 68172	-8.349099, 113.612721
3	Balai Taman Nasional Meru Betiri Jember	Jl. Sriwijaya No.53, Kranjingan, Sumpersari, Kabupaten Jember, Jawa Timur 68124	-8.193707, 113.720412

2) Perencanaan fitur

Perencanaan kedua adalah perencanaan fitur yang digunakan untuk menentukan fitur yang dibutuhkan. Pada *user story* dibagi menjadi 2 modul untuk menunjukkan pembagian pengerjaan fitur agar mempermudah pengerjaan sistem. *Velocity* merupakan ketepatan pengerjaan sistem. Fitur akan dibuat menggunakan tabel *user story* sesuai dengan kebutuhan fungsional yang diberikan oleh *customer*:

Tabel 4.7 *User Story*

No	Kode <i>Story</i>	Deskripsi	Entitas	Estimasi hari (prioritas)
Modul-1				
1	ST01	Melakukan <i>Login</i> Sistem	<i>Username, password</i>	1 (<i>low</i>)
2	ST02	Menampilkan, Menambah, dan Mengubah Akun	<i>Username, password, cabang, nama, kontak</i>	2 (<i>low</i>)
3	ST03	Menampilkan, Menambah, dan Mengubah Data Flora	Nama, gambar, koordinat, lokasi, keterangan	2 (<i>high</i>)
4	ST04	Menampilkan, Menambah, dan Mengubah Data Fauna	Nama, gambar, koordinat, lokasi, keterangan	2 (<i>high</i>)
5	ST05	Menampilkan, Menambah, dan Mengubah Data Cabang	Nama, gambar, koordinat, lokasi	2 (<i>low</i>)
6	ST06	Menampilkan, Menambah, dan Mengubah Pos Pantau	Nama, gambar, koordinat, lokasi	2 (<i>low</i>)
<i>Velocity</i>				11
Modul-2				
7	ST07	Melihat Peta	Titik lokasi, radius, label lokasi	2 (<i>high</i>)
8	ST08	Melihat Detail Peta	Keterangan, peta statis, label lokasi	2 (<i>high</i>)
9	ST09	Melihat Rute	Garis rute	2 (<i>low</i>)
10	ST10	Melihat Detail Rute	Garis rute, detail jalan	2 (<i>low</i>)
<i>Velocity</i>				8

3) Perencanaan pengerjaan sistem

Pengerjaan sistem dilakukan selama 3 minggu. Minggu pertama digunakan untuk membuat *design*, desain berupa *usecase, colaboration responsibility class* (CRC Card), *class diagram*, dan *user interface*. Minggu kedua digunakan untuk

implementasi *coding*. Minggu ketiga digunakan untuk melakukan *testing*. Antara minggu pertama dan kedua sudah mulai melakukan implementasi koding sesuai dengan *user story*.

4.2 Design

Seperti yang telah dijelaskan pada bab sebelumnya, desain sistem pada penelitian ini meliputi *usecase diagram* untuk membuat fitur pada sistem, *collaboration responsibility class* (CRC Card) untuk menentukan tanggung jawab kelas yang dibuat, dan *class diagram* untuk menentukan hubungan antara kelas yang akan dibuat.

4.2.1 Usecase Diagram

Hasil dari tahapan *planning* sistem diperoleh keputusan bahwa sistem ini memiliki 2 aktor yang bisa mengakses karena merupakan sistem yang ditujukan untuk umum. Penentuan hak akses diperoleh dari proses *planning*. *Usecase diagram* digunakan untuk menggambarkan interaksi antara aktor-aktor tersebut dengan fitur sistem yang dibuat untuk memenuhi kebutuhan fungsional dan non-fungsional sistem. Selain menyajikan *usecase diagram* pada Gambar 4, sub bab ini akan memberikan penjelasan tentang setiap aktor dan *usecase* yang ada.

1. Definisi Aktor

Berikut merupakan penjelasan aktor atau pengguna pada sistem. Terdapat dua aktor yang dijelaskan pada Tabel 4.8 berikut ini.

Tabel 4.8 Definisi aktor sistem

No.	Aktor	Deskripsi
1.	AdminPusat	Aktor AdminPusat memiliki hak akses untuk menambah, melihat, dan mengubah akun user. Memiliki hak akses menambah, mengubah, dan melihat pemetaan dan juga melihat rute, detail rute.
2.	User	Aktor User memiliki hak akses melihat hasil pemetaan, rute, dan detail rute user merupakan semua orang yang bisa mengakses sistem, baik itu peneliti, wisatawan, atau orang yang ingin mengetahui informasi.

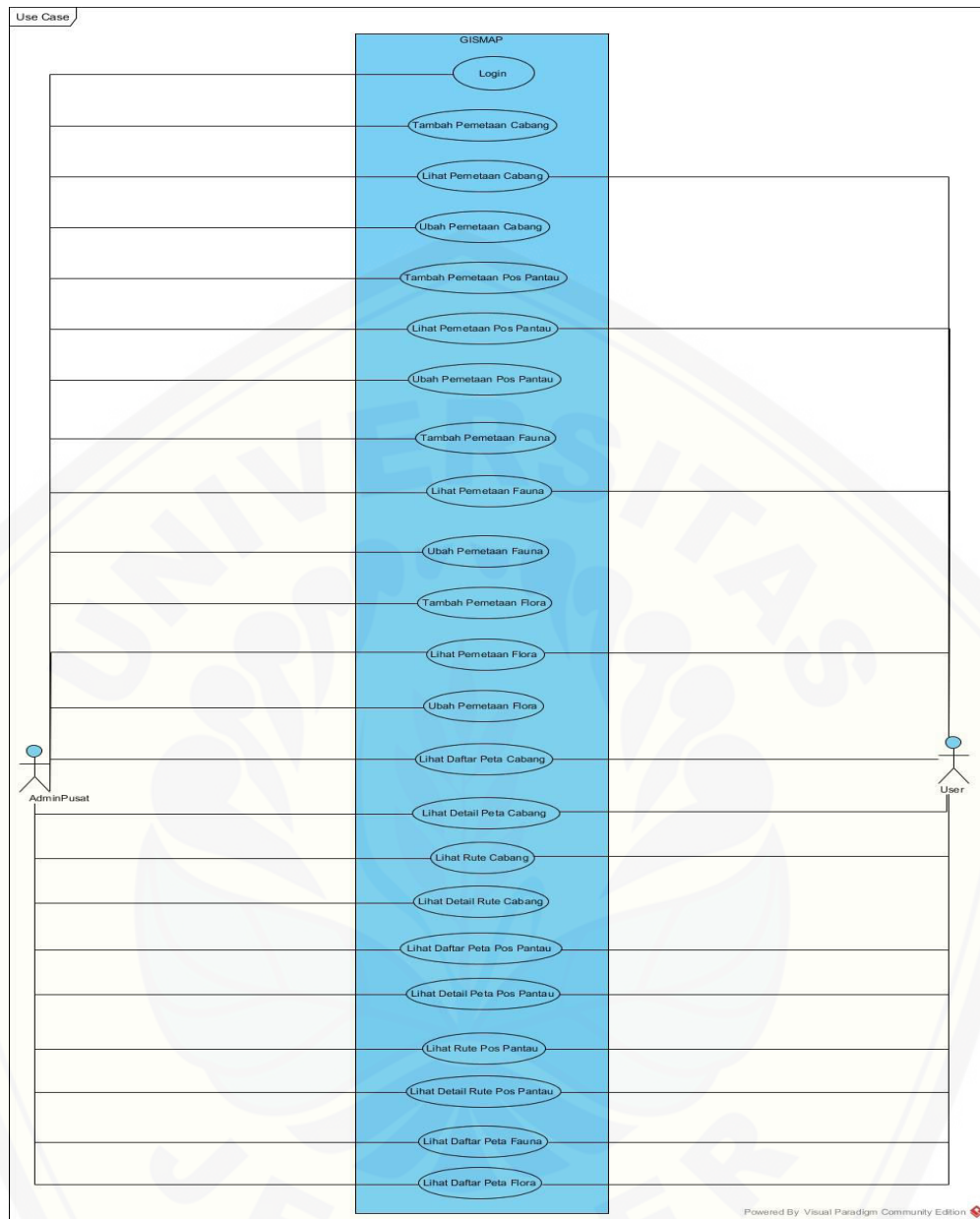
2. Definisi Usecase

Definisi *usecase* menjelaskan setiap *usecase* dalam sistem informasi pemetaan flora dan fauna. Definisi *usecase* ini dapat dilihat pada Tabel 4.9.

Tabel 4.9 Definisi *usecase*

No.	<i>Usecase</i>	Deskripsi
1.	<i>Login</i>	Menggambarkan proses autentifikasi Admin untuk masuk ke dalam sistem
2.	Tambah Pemetaan Cabang	Menggambarkan proses menambah pemetaan cabang
3.	Lihat Pemetaan Cabang	Menggambarkan proses melihat pemetaan cabang
4.	Ubah Pemetaan Cabang	Menggambarkan proses mengubah pemetaan cabang
5.	Tambah Pemetaan Pos Pantau	Menggambarkan proses menambah pemetaan pos pantau
6.	Lihat Pemetaan Pos Pantau	Menggambarkan proses melihat pemetaan pos pantau
7.	Ubah Pemetaan Pos Pantau	Menggambarkan proses mengubah pemetaan pos pantau
8.	Tambah Pemetaan Fauna	Menggambarkan proses menambah pemetaan fauna
9.	Lihat Pemetaan Fauna	Menggambarkan proses melihat pemetaan fauna
10.	Ubah Pemetaan Fauna	Menggambarkan proses mengubah pemetaan fauna
11.	Tambah Pemetaan Flora	Menggambarkan proses menambah pemetaan flora
12.	Lihat Pemetaan Flora	Menggambarkan proses melihat pemetaan flora
13.	Ubah Pemetaan Flora	Menggambarkan proses mengubah pemetaan flora
14.	Lihat Daftar Peta	Menggambarkan proses melihat daftar pemetaan

No.	<i>Usecase</i>	Deskripsi
	Cabang	cabang
15.	Lihat Detail Peta Cabang	Menggambarkan proses melihat detail masing-masing pemetaan cabang
16.	Lihat Rute Cabang	Menggambarkan proses melihat rute pemetaan cabang
17	Lihat Detail Rute Cabang	Menggambarkan proses melihat detail rute pemetaan cabang
18	Lihat Daftar Peta Pos Pantau	Menggambarkan proses melihat daftar pemetaan pos pantau
19	Lihat Detail Peta Pos Pantau	Menggambarkan proses melihat detail masing-masing pemetaan pos pantau
20	Lihat Rute Pos Pantau	Menggambarkan proses melihat rute pemetaan pos pantau
21	Lihat Detail Rute Pos Pantau	Menggambarkan proses melihat detail rute pemetaan pos pantau
22	Lihat Daftar Peta Fauna	Menggambarkan proses melihat daftar pemetaan fauna
23	Lihat Daftar Peta Flora	Menggambarkan proses melihat daftar pemetaan flora



Gambar 4.3 Usecase Diagram

4.2.2 Collaboration Responsibility Class (CRC Card)

CRC Card diperoleh dari penjabaran hubungan *responsibility* antar *class* di sistem sesuai dengan yang telah digambarkan dalam *Usecase Diagram*. *Class* pada sistem dibuat sesuai fitur yaitu fitur cabang meliputi cabang_tambah, cabang_ubah, dan cabang. Kemudian fitur pos pantau meliputi pos_tambah, pos_ubah, dan pos, fitur flora meliputi flora_tambah, flora_ubah, dan flora, fitur

fauna meliputi fauna_tambah, fauna_ubah, dan fauna, fitur peta meliputi fauna_list, flora_list, pos_detail, pos_list, cabang_detail, dan cabang_list.

Kelas cabang_tambah berkolaborasi dengan kelas cabang, kelas cabang_ubah berkolaborasi dengan kelas cabang, sedangkan kelas cabang berkolaborasi dengan kelas cabang_tambah dan cabang_ubah.

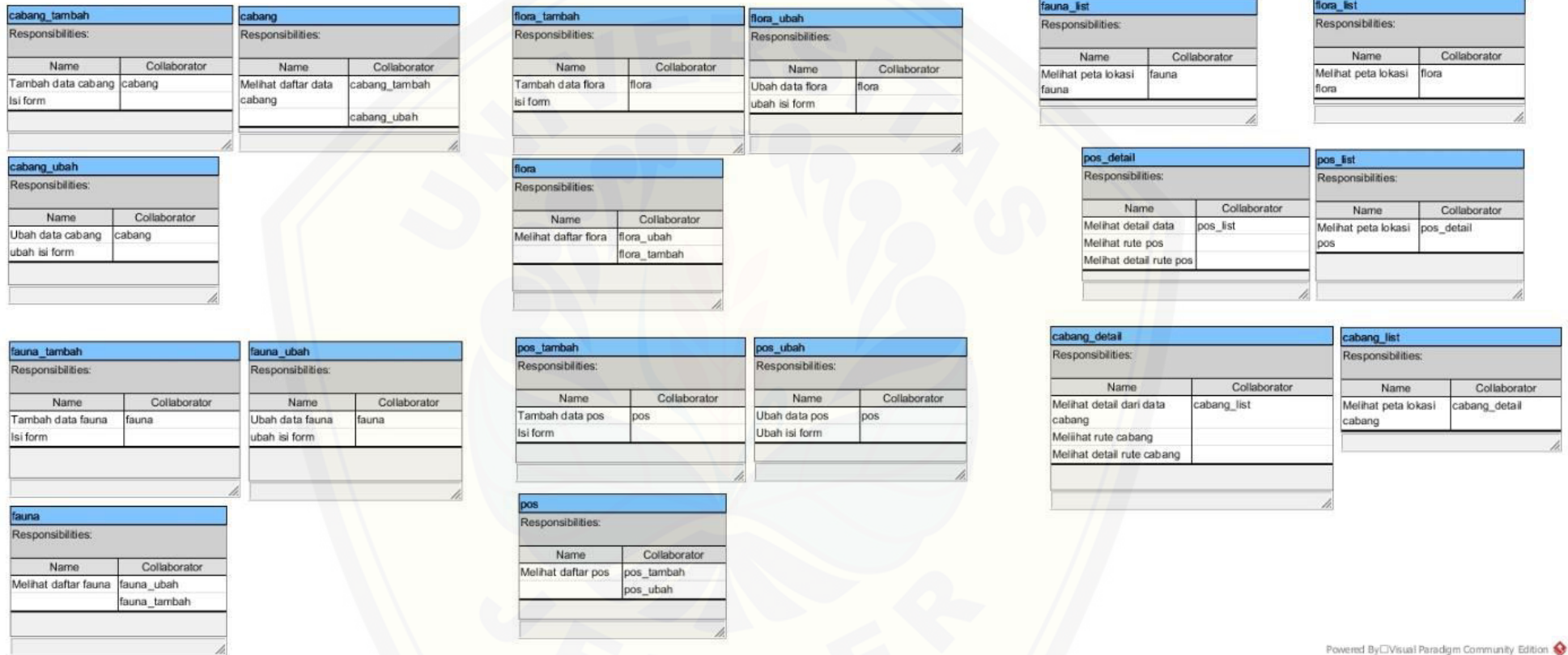
Kelas pos_tambah berkolaborasi dengan kelas pos, kelas pos_ubah berkolaborasi dengan kelas pos, sedangkan kelas pos berkolaborasi dengan kelas pos_tambah dan pos_ubah.

Kelas flora_tambah berkolaborasi dengan kelas flora, kelas flora_ubah berkolaborasi dengan kelas flora, sedangkan kelas flora berkolaborasi dengan kelas flora_tambah dan flora_ubah.

Kelas fauna_tambah berkolaborasi dengan kelas fauna, kelas fauna_ubah berkolaborasi dengan kelas fauna, sedangkan kelas fauna berkolaborasi dengan kelas fauna_tambah dan fauna_ubah.

Kelas fauna_list berkolaborasi dengan kelas fauna, kelas flora list berkolaborasi dengan kelas flora, kelas pos_detail berkolaborasi dengan kelas pos_list, kelas cabang_detail berkolaborasi dengan cabang_list

Berikut CRC Card dari sistem pada gambar 4.4.



Gambar 4.4 CRC Card

Penjelasan CRC Card pada tabel 4.10

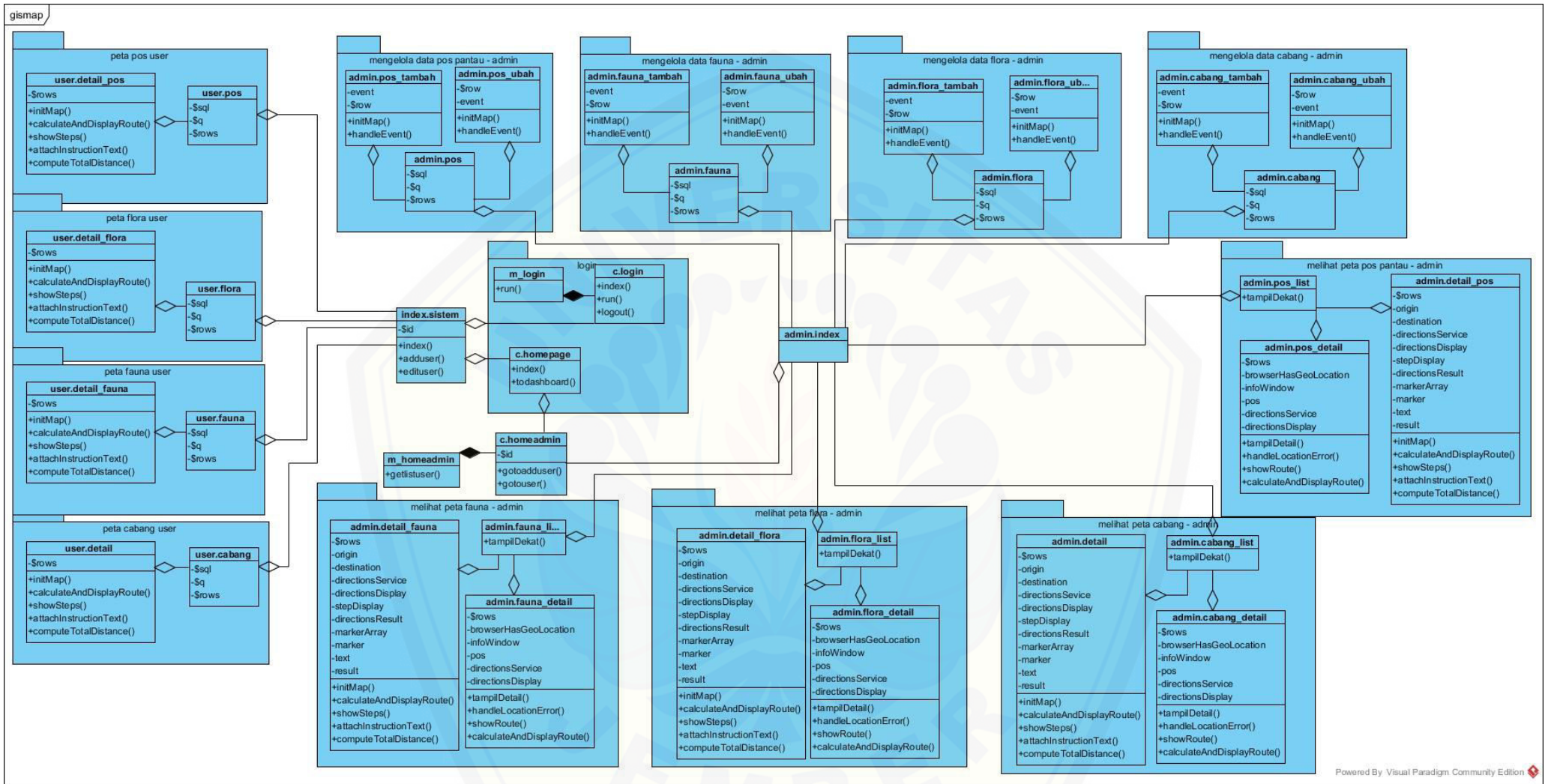
Tabel 4.10 Definisi CRC Card

No.	Nama Class	Deskripsi
1.	cabang_list	Class ini memiliki <i>responsibility</i> untuk melihat peta lokasi cabang. Class ini berkolaborasi dengan cabang_detail
2.	fauna_list	Class ini memiliki <i>responsibility</i> untuk melihat peta lokasi fauna. Class ini berkolaborasi dengan fauna_detail
3.	flora_list	Class ini memiliki <i>responsibility</i> untuk melihat peta lokasi flora. Class ini berkolaborasi dengan flora_detail
4.	cabang_tambah	Class ini memiliki <i>responsibility</i> untuk menambah data cabang dan mengisi form. Class ini berkolaborasi dengan cabang
5.	Cabang	Class ini memiliki <i>responsibility</i> untuk melihat daftar data cabang. Class ini berkolaborasi dengan cabang_tambah dan cabang_ubah
6.	cabang_ubah	Class ini memiliki <i>responsibility</i> untuk mengubah data cabang dan mengisi form. Class ini berkolaborasi dengan cabang
7.	fauna_tambah	Class ini memiliki <i>responsibility</i> untuk menambah data fauna dan mengisi form. Class ini berkolaborasi dengan fauna
8.	fauna_ubah	Class ini memiliki <i>responsibility</i> untuk mengubah data fauna dan mengisi form. Class ini berkolaborasi dengan fauna
9.	Fauna	Class ini memiliki <i>responsibility</i> untuk melihat data fauna. Class ini berkolaborasi dengan fauna_tambah dan fauna_ubah
10.	flora_tambah	Class ini memiliki <i>responsibility</i> untuk menambah data flora dan mengisi form. Class ini berkolaborasi dengan flora
11.	flora_ubah	Class ini memiliki <i>responsibility</i> untuk mengubah data flora dan mengisi form. Class ini berkolaborasi dengan flora
12.	Flora	Class ini memiliki <i>responsibility</i> untuk melihat data flora. Class ini berkolaborasi dengan flora_tambah dan flora_ubah

No.	Nama <i>Class</i>	Deskripsi
13.	pos_tambah	<i>Class</i> ini memiliki <i>responsibility</i> untuk menambah data pos dan mengisi form. <i>Class</i> ini berkolaborasi dengan pos
14.	pos_ubah	<i>Class</i> ini memiliki <i>responsibility</i> untuk mengubah data pos dan mengisi form. <i>Class</i> ini berkolaborasi dengan pos
15.	Pos	<i>Class</i> ini memiliki <i>responsibility</i> untuk melihat data pos. <i>Class</i> ini berkolaborasi dengan pos_tambah dan pos_ubah
16	fauna_detail	<i>Class</i> ini memiliki <i>responsibility</i> untuk menampilkan rute dan detail rute. <i>Class</i> ini berkolaborasi dengan fauna_list
17	flora_detail	<i>Class</i> ini memiliki <i>responsibility</i> untuk menampilkan rute dan detail rute. <i>Class</i> ini berkolaborasi dengan flora_list
18	pos_detail	<i>Class</i> ini memiliki <i>responsibility</i> untuk menampilkan rute dan detail rute. <i>Class</i> ini berkolaborasi dengan pos_list
19	cabang_detail	<i>Class</i> ini memiliki <i>responsibility</i> untuk menampilkan rute dan detail rute. <i>Class</i> ini berkolaborasi dengan cabang_list
20	pos_list	<i>Class</i> ini memiliki <i>responsibility</i> untuk melihat peta lokasi pos. <i>Class</i> ini berkolaborasi dengan pos_detail

4.2.3 *Class Diagram*

Class Diagram menggambarkan hubungan antarkelas yang ada dalam suatu sistem. *Class Diagram* sistem informasi geografis pemetaan flora dan fauna dapat dilihat pada Gambar 4.5.



Gambar 4.5 Class diagram

4.2.4 *Testing design*Tabel 4.11 *Testing design*

Fitur	<i>Usecase</i>	<i>CRC Card</i>	<i>Class Diagram</i>	Status
Mengelola data peta cabang	Tambah Pemetaan Cabang	cabang_tambah	admin.cabang_tambah	Sesuai
	Ubah Pemetaan Cabang	cabang_ubah	admin.cabang_tambah	Sesuai
	Lihat Pemetaan Cabang	cabang	admin.cabang user.cabang	Sesuai
Mengelola data peta pos pantau	Tambah Pemetaan Pos Pantau	pos_tambah	admin.pos_tambah	Sesuai
	Ubah Pemetaan Pos Pantau	pos_ubah	admin.pos_ubah	Sesuai
	Lihat Pemetaan Pos Pantau	pos	admin.pos user.pos	Sesuai
Mengelola data peta flora	Tambah Pemetaan Flora	flora_tambah	admin.flora_tambah	Sesuai
	Ubah Pemetaan Flora	flora_ubah	admin.flora_ubah	Sesuai
	Lihat Pemetaan Flora	flora	admin.flora user.flora	Sesuai

Fitur	Usecase	CRC Card	Class Diagram	Status
Mengelola data peta fauna	Tambah Pemetaan Fauna	fauna_tambah	admin.fauna_tambah	Sesuai
	Ubah Pemetaan Fauna	fauna_ubah	admin.fauna_ubah	Sesuai
	Lihat Pemetaan Fauna	fauna	admin.fauna user.fauna	Sesuai
Melihat detail rute cabang	Lihat daftar peta cabang	cabang_list	admin.cabang_list user.cabang_list	Sesuai
	Lihat detail rute cabang	cabang_detail	admin.cabang_detail user.cabang_detail	Sesuai
Melihat detail rute pos pantau	Lihat daftar peta pos pantau	pos_list	admin.pos_list user.pos_list	Sesuai
	Lihat detail rute pos pantau	pos_detail	admin.pos_detail user.pos_detail	Sesuai
Melihat peta flora	Lihat peta flora	flora_list	admin.flora_list user.flora_list	Sesuai
Melihat peta fauna	Lihat peta fauna	fauna_list	admin.fauna_list user.fauna_list	Sesuai

4.3 Implementation

Penulisan kode program merupakan tahap pengimplementasian desain yang telah dibuat sebelumnya. Setiap fitur yang dimiliki oleh sistem dituliskan dalam beberapa *class*. Tahap implementasi dibagi menjadi 2 modul dan 2 testing yang bisa dilihat pada lampiran.

4.3.1 Kode program melihat peta flora

Kode program melihat peta flora terletak pada class flora_list. Penulisan kode program ini dapat dilihat pada tabel . kesederhanaan koding bisa dilihat pada tabel yang ditandai abu-abu, function tampilDekat() berisi fungsi *Google Maps* API yang memanggil lokasi dari flora yang diambil dari data koordinat yang tersimpan di *database* sistem.

Tabel 4.12 Kode program melihat peta flora

```

<div class="page-header">
  <h1>Tempat</h1>
</div>
<div id="map" style="height: 500px;"></div>
<script>
function tampilDekat(){
  getCurLocation();

  map_dekat = new google.maps.Map(document.getElementById('map'), {
    zoom: <?=get_option('default_zoom')?>,
    center: {
      lat : default_lat,
      lng : default_lng
    }
  });

  var data = <?=json_encode($db->get_results("SELECT * FROM
tb_flora"))?>;
  $.each(data, function(k, v){
    var pos = {
      lat : parseFloat(v.lat),
      lng : parseFloat(v.lng)
    };
    var contentString = '<h3>' + v.nama_tempat + '</h3>' +
      '<p align="center"><a href="?m=flora_detail&ID=' + v.id_tempat + '"
class="link_detail btn btn-primary">Lihat Detail</a>';
    var infowindow = new google.maps.InfoWindow({
      content: contentString
    });
    var marker = new google.maps.Marker({
      position: pos,
      map: map_dekat,
      animation: google.maps.Animation.DROP
    });
    marker.addListener('click', function() {
      infowindow.open(map_dekat, marker);

```

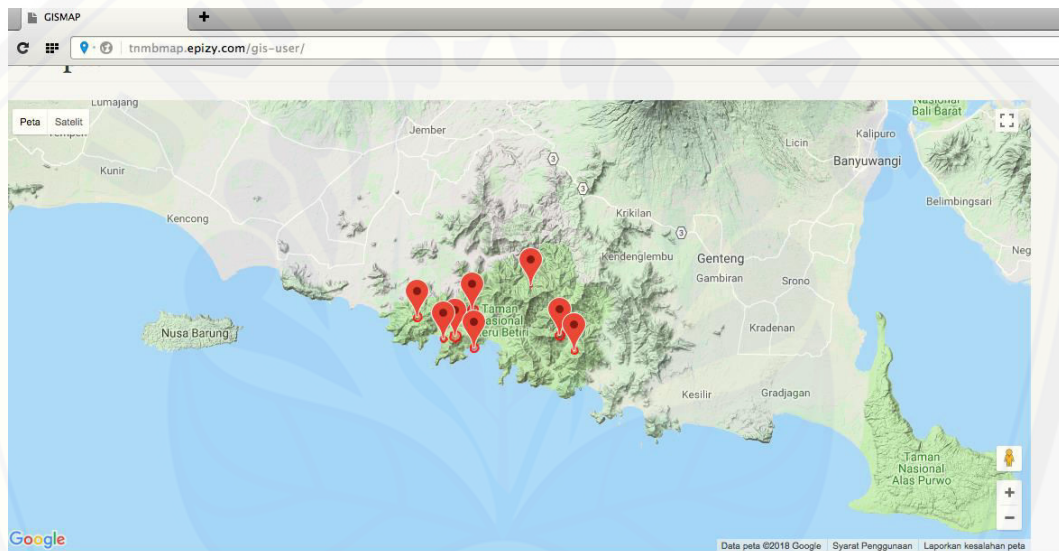
```

});
});
}

$(function(){
    tampilDekat();
})
</script>

```

Fitur ini dapat diakses oleh admin dan *user* dengan menekan menu peta kemudian memilih submenu flora. Fitur ini menampilkan ikon lokasi dari data pemetaan, yang ketika ditekan akan muncul tombol “Lihat Detail” seperti pada Gambar 4.6



Gambar 4.6 Tampilan peta Flora

4.3.2 Kode program melihat peta fauna

Kode program melihat peta fauna terletak pada class *fauna_list*.

Penulisan kode program ini dapat dilihat pada tabel 4.13

Tabel 4.13 Kode program melihat peta fauna

```

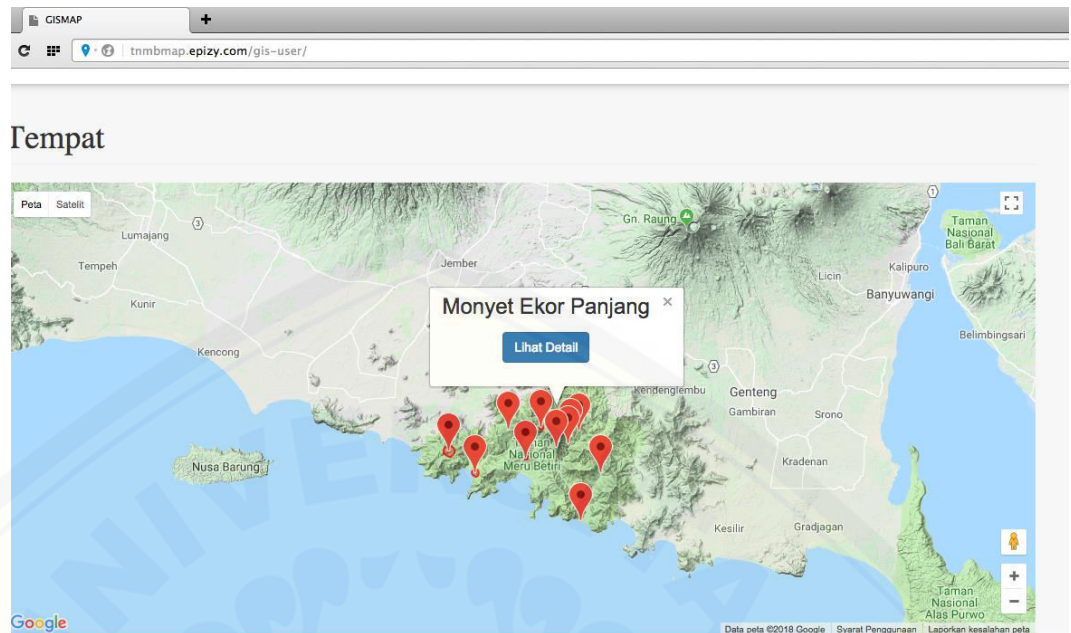
<div class="page-header">
    <h1>Tempat</h1>
</div>
<div id="map" style="height: 500px;"></div>
<script>
function tampilDekat(){
    getCurLocation();

    map_dekat = new google.maps.Map(document.getElementById('map'), {

```

```
zoom: <?=get_option('default_zoom')?>,
center: {
  lat : default_lat,
  lng : default_lng
}
});
var data = <?=json_encode($db->get_results("SELECT * FROM
tb_tempat"))?>;
$.each(data, function(k, v){
  var pos = {
    lat : parseFloat(v.lat),
    lng : parseFloat(v.lng)
  };
  var contentString = '<h3>' + v.nama_tempat + '</h3>' +
  '<p align="center"><a href="?m=tempat_detail&ID=' + v.id_tempat + '"
class="link_detail btn btn-primary">Lihat Detail</a>';
  var infowindow = new google.maps.InfoWindow({
    content: contentString
  });
  var marker = new google.maps.Marker({
    position: pos,
    map: map_dekat,
    animation: google.maps.Animation.DROP
  });
  marker.addListener('click', function() {
    infowindow.open(map_dekat, marker);
  });
});
}
$(function(){
  tampilDekat();})
</script>
```

Fitur ini dapat diakses oleh admin dan *user* dengan menekan menu peta kemudian memilih submenu fauna. Fitur ini menampilkan ikon lokasi dari data pemetaan, yang ketika ditekan akan muncul tombol “Lihat Detail” seperti pada Gambar 4.7.



Gambar 4.7 Tampilan peta Fauna

Kode program lainnya ditunjukkan pada lampiran.

4.4 System Testing

Tahap pengujian dilakukan untuk mengevaluasi sistem informasi yang telah selesai dan telah diimplementasikan. Seperti yang telah dijelaskan pada bab sebelumnya, tahapan ini dilakukan metode pengujian *black* yang dapat dilihat pada lampiran. Sedangkan untuk pengujian non fungsional dilakukan menggunakan bantuan website <https://www.webpagetest.org> yang hasilnya bisa dilihat di lampiran.

BAB VI PENUTUP

Bab ini berisi kesimpulan dan saran dari penelitian yang telah dilakukan. Kesimpulan dan saran ini diharapkan mampu menjadi acuan untuk melakukan penelitian selanjutnya.

6.1 Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah sebagai berikut:

- 1) Implementasi dari 12 *core practice* ditujukan pada *life cycle* dari PXP pada penelitian ini. Mulai dari *planning game* yang diimplementasikan pada tahapan *planning*, *practice small release* pada tahapan *implementation*, *metaphore* pada tahapan *planning*, *practice simple design* pada tahapan *design*, *practice testing* pada tahapan *testing*, *practice refactoring* pada tahapan *implementation*, *practice pair programmer* pada tahapan *implementation*, *practice collective ownership* pada tahap *implementation*, *continous integritation* pada tahapan *implementation*, *40-hour week* pada tahapan *planning*, *design*, dan *implementation*, *on-site customer* pada tahapan *planning*, dan *implementation*, *coding standart* pada tahapan *implementation*. Hasil perbandingan 12 *core practice* menunjukkan bahwa pada penelitian ini 2 *core* masih belum terpenuhi, sehingga belum maksimal. Proses penambahan dan pengurangan fitur tidak banyak dilakukan yang menyebabkan *Continous Integration* tidak terpenuhi dan *Customer* tidak selalu berkomunikasi dengan *programmer* yang menyebabkan *On-site Customer* tidak terpenuhi. 2 dari 12 *core* tidak terpenuhi dalam penelitian ini, sehingga dapat disimpulkan penerapan *practice* model PXP pada penelitian ini tidak cocok.
- 2) Penerapan model PXP berdasarkan nilai kemudahan, komunikasi, umpan balik, dan keberanian untuk *programmer* atau pengembang. Nilai kemudahan dalam pengembangan sistem informasi geografis pemetaan flora dan fauna di TNMB menggunakan desain yang sederhana dan mudah dipahami oleh *programmer*, memiliki desain yang sedikit namun bisa memenuhi kebutuhan dalam ilmpementasi kode. *Usecase* untuk fitur yang

dipakai, *CRC card* untuk hubungan kelas per fitur, dan *class diagram* hubungan kelas dari seluruh sistem. Kemudian dalam implementasi kode, menggunakan *Google Maps API* yang memudahkan dalam pembuatan sistem karena penggunaan API hanya tinggal memanggil kode yang disediakan oleh pengembang API seperti yang telah dijabarkan pada proses *implementation*, dalam penelitian ini adalah *Google Maps API*. Komunikasi dan umpan balik diterapkan secara bersamaan, komunikasi dengan user dan mendapatkan umpan balik untuk menyelesaikan pengembangan sistem informasi geografis seperti yang telah dijabarkan pada proses *planning*. Nilai keberanian diterapkan oleh pengembang dalam kreatifitas dan pengambilan resiko adanya inovasi baru dari user. Kecepatan pembuatan sistem merupakan faktor penting yang ada pada penelitian ini. Hasilnya penerapan nilai PXP sesuai dan menunjang dalam pengembangan sistem informasi geografis pemetaan flora dan fauna di Taman Nasional Meru Betiri

- 3) Mengacu pada 12 *core practice* dan nilai-nilai dari PXP menghasilkan pengerjaan sistem yang cepat dan biaya pembuatan sistem yang rendah, meskipun pada penelitian ini ada beberapa *practice* yang belum terpenuhi merupakan kekurangan yang besar dalam penerapan model PXP. Seperti yang ada di lampiran bahwa rincian biaya terpangkas karena tim hanya 1 orang, hal ini menunjukkan bahwa penerapan metode PXP memang bisa untuk mengurangi biaya pengembangan sistem.

6.2 Saran

Saran yang ditujukan untuk memberikan masukan yang lebih baik yaitu :

- 1) Penelitian ini memiliki kekurangan pada komunikasi dengan customer, disarankan pengembangan selanjutnya sebaiknya menambah tingkat komunikasi berlanjut antara *customer* dengan *programmer*.
- 2) Penelitian ini membahas sebagian dari PXP, dsarankan untuk penelitian selanjutnya mengembangkan penelitian lebih ke pengimplementasian per *life cycle* dari PXP.

DAFTAR PUSTAKA

- Aji, A. B. S. (2015). *Implementasi Algoritma Backtracking untuk Pencarian Solusi pada Game Labirin Berbasis Android*. Semarang.
- Anjuliani, R., & Astuti, L. W. (2015). Aplikasi Isc (Informatics Student Center) Menggunakan Metode Personal Extreme Programming Berbasis Android. *DASI*, 6(1), 20–25.
- Ariaji, T., Utami, E., Sunyoto, A., Informatika, M. T., Sarjana, P. P., & Yogyakarta, S. A. (2014). Evaluasi Sistem Informasi yang Dikembangkan dengan Metodologi Extreme Programming. *Jurnal Ilmiah DASI*, 15(4), 53–62.
- Dewanto, I. J. (2004). System Development Life Cycle Dengan Beberapa Pendekatan. *System Development Life Cycle Dengan Beberapa Pendekatan*, 2(1), 39–47.
- Dzhurov, Y., Krasteva, I., & Ilieva, S. (2009). Personal Extreme Programming—An Agile Process for Autonomous Developers. *International Conference on Software, Services & Semantic Technologies*, (August 2016), 252–259.
- Ichtiara, C. (2008). *Implementasi Aplikasi Sistem Informasi Geografis (SIG) Universitas Indonesia (UI) Berbasis Web dengan Menggunakan Google Maps API*. Universitas Indonesia, Depok.
- Masykur, F. (2014). Implementasi Sistem Informasi Geografis Menggunakan Google Maps Api Dalam Pemetaan Asal Mahasiswa. *Jurnal SIMETRIS*, 5(2), 181–186.
- Puroso, P. (2014). Taman Nasional Meru Betiri., (September).
- Rizal, H. et al. (2014). *Perancangan dan Pembuatan Mobile Learning Interaktif Berbasis Android dengan Metode Personal Extreme Programming*.
- Siswoyo. (2002). *Buku Informasi : Obyek dan Daya Tarik Wisata Alam Taman. Balai Taman Nasional Meru*. Jember: Departemen Kehutanan, Direktorat Jendral.
- Stewart Baird. (2002). *Sams-Teach-Yourself-Extreme-Programming-in-24-Hours*. Sams Publishing.

LAMPIRAN

A. Kode Program

A.1 Kode program *login*

Kode program *login* terletak pada *class login*. Penulisan kode program ini dapat dilihat pada tabel A.1.

Tabel A.1 Kode program *login*

```

<?php
function run(){
    session::init();
    $username=$_POST['username'];
    $password=$_POST['password'];
    $statement=$this->db->prepare("SELECT * FROM user
WHERE username='".$username.'" AND password= '".$password.'" LIMIT
1");
    $statement->execute();
    $data=$statement->fetch();
    $count=$statement->rowCount();
    if(empty($password) && empty($username)){
        session::set('loginfail','Kolom Username dan Password
Kosong');
        header('location:../login/fail');
    } elseif(empty($password) && !empty($username)){
        session::set('loginfail','Kolom Password Kosong');
        header('location:../login/fail');
    } elseif(empty($username) && !empty($password)){
        session::set('loginfail','Kolom Username Kosong');
        header('location:../login/fail');
    } else{
        if($count>0){
            session::set('userlevel',$data['level']);
            session::set('loggedIn',true);
            session::set('username',$data['username']);
            session::set('id_member', $data['id']);
            header('location:'.URL.'dashboard');
        } else{
            session::set('loginfail','Username atau Password
Salah');
            header('location:../login/fail');
        }
    }
}
?>

```

A.2 Kode program menambah data pemetaan cabang

Kode program menambah peta cabang berada di class cabang_tambah. Penulisan kode program ini dapat dilihat pada tabel A.2.

Tabel A.2 Kode program menambah data pemetaan cabang

```

<div class="page-header">
  <h1>Tambah Cabang</h1>
</div>
<form method="post" action="?m=tempat_tambah" enctype="multipart/form-data">
<div class="container">
  <div class="wrapper1">
    <div class="col-sm-6">

      <?php if($_POST) include'aksi.php'?>
      <div class="form-horizontal">
        <label>Nama Cabang <span class="text-danger">*</span></label>
        <input class="form-control" type="text" name="nama_tempat"
value="<?=$_POST['nama_tempat']?>"/>
      </div>
      <div class="form-horizontal">
        <label>Gambar <span class="text-danger">*</span></label>
        <input class="form-control-file" type="file" name="gambar" />
      </div>
      <div class="form-horizontal">
        <label>Latitude <span class="text-danger">*</span></label>
        <input class="form-control" type="text" name="lat" id="lat"
value="<?=$_POST['lat']?>"/>
      </div>
      <div class="form-horizontal">
        <label>Longitude <span class="text-danger">*</span></label>
        <input class="form-control" type="text" id="lng" name="lng"
value="<?=$_POST['lng']?>"/>
      </div>
      <div class="form-horizontal">
        <label>Lokasi <span class="text-danger">*</span></label>
        <input class="form-control" type="text" name="lokasi"
value="<?=$_POST['lokasi']?>"/>
      </div>
      <div class="form-horizontal">
        <label>Keterangan</label>
        <textarea class="form-control" rows="3"
name="keterangan"><?=$_POST['keterangan']?></textarea>
      </div>
      <div class="form-horizontal">
        <button class="btn btn-primary"><span class="fa fa-save"></span>

```

```
Simpan</button>
    <a class="btn btn-danger" href="?m=tempat"><span class="fa fa-
arrow-left"></span> Kembali</a>
    </div>
</div>
<div class="col-sm-6">
    <div id="map" style="height: 400px;"></div>
</div>
</div>
</form>
<script>
var defaultCenter = {
    lat : <?=get_option('default_lat')?>,
    lng : <?=get_option('default_lng')?>
};
function initMap() {

var map = new google.maps.Map(document.getElementById('map'), {
    zoom: <?=get_option('default_zoom')?>,
    center: defaultCenter
});

var marker = new google.maps.Marker({
    position: defaultCenter,
    map: map,
    title: 'Click to zoom',
    draggable:true
});

marker.addListener('drag', handleEvent);
marker.addListener('dragend', handleEvent);

var infowindow = new google.maps.InfoWindow({
    content: '<h4>Drag untuk pindah lokasi</h4>'
});

infowindow.open(map, marker);
}

function handleEvent(event) {
    document.getElementById('lat').value = event.latLng.lat();
    document.getElementById('lng').value = event.latLng.lng();
}
```

```
$(function(){
  initMap();
})
</script>
```

A.3 Kode program melihat data cabang

Kode program melihat data cabang terletak pada class cabang.

Penulisan kode program ini dapat dilihat pada tabel A.3

Tabel A.3 Kode program melihat data cabang

```
<div class="page-header">
  <h1>Cabang</h1>
</div>
<div class="panel panel-default">
  <div class="panel-heading">
    <form class="form-inline">
      <input type="hidden" name="m" value="tempat" />
      <div class="form-horizontal">
        <input class="form-control" type="text" placeholder="Pencarian. . ."
name="q" value="<?=$_GET['q']?>" />
        <button class="btn btn-success">Refresh</button>
        <a class="btn bg-green" href="?m=tempat_tambah"> Tambah</a>
      </div>
    </form>
  </div>
</div>
<div class="container">
  <div class="wrapper1">
    <div class="row">
      <div class="col-sm-12">
        <table class="table table-responsive table-bordered">
          <thead>
            <tr class="nw">
              <th>No</th>
              <th>Gambar</th>
              <th>Nama Cabang</th>
              <th>Lat</th>
              <th>Lng</th>
              <th>Lokasi</th>
              <th>Aksi</th>
            </tr>
          </thead>
          <?php
          $q = esc_field($_GET['q']);

          $sql = "SELECT *
```

```

FROM tb_tempat p
WHERE nama_tempat LIKE '%$q%'
ORDER BY id_tempat";
$rows = $db->get_results($sql);

foreach($rows as $row):?>
<tr>
<td><?==+$no?></td>
<td></td>
<td><?=$row->nama_tempat?></td>
<td><?=$row->lat?></td>
<td><?=$row->lng?></td>
<td><?=$row->lokasi?></td>
<td class="nw">
<a class="btn btn-xs btn-warning"
href="?m=tempat_ubah&ID=<?=$row->id_tempat?>"><span class="fa fa-
edit"></span></a>
<a class="btn btn-xs btn-danger"
href="aksi.php?act=tempat_hapus&ID=<?=$row->id_tempat?>"
onclick="return confirm('Hapus data?')"><span class="fa fa-
trash"></span></a>
</td>
</tr>
<?php endforeach; ?>
</table>
</div>
</div>
</div>
</div>

```

A.4 Kode program mengubah data cabang

Kode program mengubah data cabang terletak pada class cabang_ubah. Penulisan kode program ini dapat dilihat pada tabel A.4

Tabel A.4 Kode program mengubah data cabang

```

<?php
$row = $db->get_row("SELECT * FROM tb_tempat WHERE
id_tempat='$_GET[ID]'");
?>
<div class="page-header">
<h1>Ubah Cabang</h1>
</div>
<div class="container">

```



```

<div class="wrapper1">
<div class="col-sm-6">

    <?php if($_POST) include'aksi.php'?>
    <form      method="post"      action="?m=tempat_ubah&ID=<?=$row-
>id_tempat?" enctype="multipart/form-data">
        <div class="form-horizontal">
            <label>Nama Cabang <span class="text-danger">*</span></label>
            <input class="form-control" type="text" name="nama_tempat"
value="<?=$row->nama_tempat?" />
        </div>
        <div class="form-horizontal">
            <label>Gambar <span class="text-danger">*</span></label>
            <input class="form-control-file" type="file" name="gambar" />
            <p class="help-block">Kosongkan jika tidak mengubah gambar</p>
            
        </div>

        <div class="form-horizontal">
            <label>Latitude <span class="text-danger">*</span></label>
            <input class="form-control" type="text" id="lat" name="lat"
value="<?=$row->lat?" />
        </div>
        <div class="form-horizontal">
            <label>Longitude <span class="text-danger">*</span></label>
            <input class="form-control" type="text" id="lng" name="lng"
value="<?=$row->lng?" />
        </div>
        <div class="form-horizontal">
            <label>Lokasi <span class="text-danger">*</span></label>
            <input class="form-control" type="text" name="lokasi"
value="<?=$row->lokasi?" />
        </div>
        <div class="form-horizontal">
            <label>Keterangan</label>
            <textarea class="form-control" rows="3"
name="keterangan"><?=$row->keterangan?</textarea>
        </div>
        <div class="form-horizontal">
            <button class="btn btn-primary"><span class="fa fa-save"></span>
Simpan</button>
            <a class="btn btn-danger" href="?m=tempat"><span class="fa fa-
arrow-left"></span> Kembali</a>
        </div>

```

```
</form>
</div>
</div>
<div class="col-md-6">
  <div id="map" style="height: 400px;"></div>
</div>
</div>
<script>
var defaultCenter = {
  lat : <?=$row->lat * 1?>,
  lng : <?=$row->lng * 1?>
};
function initMap() {

var map = new google.maps.Map(document.getElementById('map'), {
  zoom: <?=get_option('default_zoom')?>,
  center: defaultCenter
});

var marker = new google.maps.Marker({
  position: defaultCenter,
  map: map,
  title: 'Click to zoom',
  draggable:true
});

marker.addListener('drag', handleEvent);
marker.addListener('dragend', handleEvent);

var infowindow = new google.maps.InfoWindow({
  content: '<h4>Drag untuk pindah lokasi</h4>'
});

infowindow.open(map, marker);
}

function handleEvent(event) {
  document.getElementById('lat').value = event.latLng.lat();
  document.getElementById('lng').value = event.latLng.lng();
}

$(function(){
  initMap();
})
</script>
```

A.5 Kode program menambah data pemetaan pos pantau

Kode program menambah peta pos pantau berada di class `pos_tambah`. Penulisan kode program ini dapat dilihat pada tabel A.5.

Tabel A.5 Kode program menambah data pemetaan pos pantau

```

<div class="page-header">
  <h1>Tambah POS Pantau</h1>
</div>
<form method="post" action="?m=pos_tambah" enctype="multipart/form-data">
<div class="container">
  <div class="wrapper1">
    <div class="col-sm-6">
      <?php if($_POST) include'aksi.php'?>
      <div class="form-horizontal">
        <label>Nama POS <span class="text-danger">*</span></label>
        <input class="form-control" type="text" name="nama_tempat"
value="<?=$_POST['nama_tempat']?>"/>
      </div>
      <div class="form-horizontal">
        <label>Gambar <span class="text-danger">*</span></label>
        <input class="form-control-file" type="file" name="gambar" />
      </div>
      <div class="form-horizontal">
        <label>Latitude <span class="text-danger">*</span></label>
        <input class="form-control" type="text" name="lat" id="lat"
value="<?=$_POST['lat']?>"/>
      </div>
      <div class="form-horizontal">
        <label>Longitude <span class="text-danger">*</span></label>
        <input class="form-control" type="text" id="lng" name="lng"
value="<?=$_POST['lng']?>"/>
      </div>
      <div class="form-horizontal">
        <label>Lokasi <span class="text-danger">*</span></label>
        <input class="form-control" type="text" name="lokasi"
value="<?=$_POST['lokasi']?>"/>
      </div>
      <div class="form-horizontal">
        <label>Keterangan</label>
        <textarea class="form-control"
name="keterangan"><?=$_POST['keterangan']?></textarea>
      </div>
      <div class="form-horizontal">
        <button class="btn btn-primary"><span class="fa fa-save"></span>
Simpan</button>

```

```
        <a class="btn btn-danger" href="?m=tempat"><span class="fa fa-  
arrow-left"></span> Kembali</a>  
    </div>  
</div>  
<div class="col-sm-6">  
    <div id="map" style="height: 400px;"></div>  
</div>  
</div>  
</form>  
<script>  
var defaultCenter = {  
    lat : <?=get_option('default_lat')?>,  
    lng : <?=get_option('default_lng')?>  
};  
function initMap() {  
  
    var map = new google.maps.Map(document.getElementById('map'), {  
        zoom: <?=get_option('default_zoom')?>,  
        center: defaultCenter  
    });  
  
    var marker = new google.maps.Marker({  
        position: defaultCenter,  
        map: map,  
        title: 'Click to zoom',  
        draggable:true  
    });  
  
    marker.addListener('drag', handleEvent);  
    marker.addListener('dragend', handleEvent);  
  
    var infowindow = new google.maps.InfoWindow({  
        content: '<h4>Drag untuk pindah lokasi</h4>'  
    });  
  
    infowindow.open(map, marker);  
}  
  
function handleEvent(event) {  
    document.getElementById('lat').value = event.latLng.lat();  
    document.getElementById('lng').value = event.latLng.lng();  
}  
  
$(function){
```

```

    initMap();
  })
</script>

```

A.6 Kode program melihat data pos pantau

Kode program melihat data pos pantau terletak pada class pos.

Penulisan kode program ini dapat dilihat pada tabel A.6.

Tabel A.6 Kode program melihat data pos pantau

```

<div class="page-header">
  <h1>POS Pantau</h1>
</div>
<div class="panel panel-default">
  <div class="panel-heading">
    <form class="form-inline">
      <input type="hidden" name="m" value="pos" />
      <div class="form-horizontal">
        <input class="form-control" type="text" placeholder="Pencarian. . ."
name="q" value="<?=$_GET['q']?>" />
        <button class="btn btn-success"></span> Refresh</button>
        <a class="btn bg-green" href="?m=pos_tambah"></span>
Tambah</a>
      </div>
    </form>
  </div>
<div class="container">
  <div class="wrapper1">
    <div class="row">
      <div class="col-sm-12">
        <table class="table table-responsive table-bordered">
          <thead>
            <tr class="nw">
              <th>No</th>
              <th>Gambar</th>
              <th>Nama Tempat</th>
              <th>Lat</th>
              <th>Lng</th>
              <th>Lokasi</th>
              <th>Aksi</th>
            </tr>
          </thead>
          <?php
          $q = esc_field($_GET['q']);

          $sql = "SELECT *

```

```

FROM tb_pos p
WHERE nama_tempat LIKE '%$q%'
ORDER BY id_tempat";
$rows = $db->get_results($sql);

foreach($rows as $row):?>
<tr>
<td><?==+$no?></td>
<td></td>
<td><?=$row->nama_tempat?></td>
<td><?=$row->lat?></td>
<td><?=$row->lng?></td>
<td><?=$row->lokasi?></td>
<td class="nw">
<a
class="btn
btn-xs
btn-warning"
href="?m=pos_ubah&ID=<?=$row->id_tempat?"><span
class="fa
fa-
edit"></span></a>
<a
class="btn
btn-xs
btn-danger"
href="aksi.php?act=pos_hapus&ID=<?=$row->id_tempat?"
onclick="return
confirm('Hapus data?')"><span class="fa fa-trash"></span></a>
</td>
</tr>
<?php endforeach; ?>
</table>
</div>
</div>
</div>
</div>
</div>

```

A.7 Kode program mengubah data pos pantau

Kode program mengubah data pos pantau terletak pada class pos_ubah. Penulisan kode program ini dapat dilihat pada tabel A.7

Tabel A.7 Kode program mengubah data pos pantau

```

<?php
$row = $db->get_row("SELECT * FROM tb_pos WHERE
id_tempat='$_GET[ID]'");
?>
<div class="page-header">
<h1>Ubah POS Pantau</h1>
</div>
<div class="container">

```

```

<div class="wrapper1">
<div class="col-sm-6">
  <?php if($_POST) include'aksi.php'?>
  <form      method="post"      action="?m=pos_ubah&ID=<?=$row-
>id_tempat?" enctype="multipart/form-data">
    <div class="form-horizontal">
      <label>Nama POS <span class="text-danger">*</span></label>
      <input  class="form-control"  type="text"  name="nama_tempat"
value="<?=$row->nama_tempat?" />
    </div>
    <div class="form-horizontal">
      <label>Gambar <span class="text-danger">*</span></label>
      <input class="form-control-file" type="file" name="gambar" />
      <p class="help-block">Kosongkan jika tidak mengubah gambar</p>
      
    </div>
    <div class="form-horizontal">
      <label>Latitude <span class="text-danger">*</span></label>
      <input  class="form-control"  type="text"  id="lat"  name="lat"
value="<?=$row->lat?" />
    </div>
    <div class="form-horizontal">
      <label>Longitude <span class="text-danger">*</span></label>
      <input  class="form-control"  type="text"  id="lng"  name="lng"
value="<?=$row->lng?" />
    </div>
    <div class="form-horizontal">
      <label>Lokasi <span class="text-danger">*</span></label>
      <input  class="form-control"  type="text"  name="lokasi"
value="<?=$row->lokasi?" />
    </div>
    <div class="form-horizontal">
      <label>Keterangan</label>
      <textarea  class="form-control"  name="keterangan"><?=$row-
>keterangan?</textarea>
    </div>
    <div class="form-horizontal">
      <button class="btn btn-primary"><span class="fa fa-save"></span>
Simpan</button>
      <a class="btn btn-danger" href="?m=pos"><span class="fa fa-arrow-
left"></span> Kembali</a>
    </div>
  </form>
</div>
<div class="col-md-6">

```

```
<div id="map" style="height: 400px;"></div>
</div>
</div>
</div>
<script>
var defaultCenter = {
  lat : <?=$row->lat * 1?>,
  lng : <?=$row->lng * 1?>
};
function initMap() {

  var map = new google.maps.Map(document.getElementById('map'), {
    zoom: <?=get_option('default_zoom')?>,
    center: defaultCenter
  });

  var marker = new google.maps.Marker({
    position: defaultCenter,
    map: map,
    title: 'Click to zoom',
    draggable:true
  });

  marker.addListener('drag', handleEvent);
  marker.addListener('dragend', handleEvent);

  var infowindow = new google.maps.InfoWindow({
    content: '<h4>Drag untuk pindah lokasi</h4>'
  });

  infowindow.open(map, marker);
}
function handleEvent(event) {
  document.getElementById('lat').value = event.latLng.lat();
  document.getElementById('lng').value = event.latLng.lng();
}

$(function(){
  initMap();
})
</script>
```

A.8 Kode program melihat peta cabang

Kode program melihat peta cabang terletak pada class cabang_list.

Penulisan kode program ini dapat dilihat pada tabel A.8

Tabel A.8 Kode program melihat peta cabang

```

<div class="page-header">
  <h1>Cabang</h1>
</div>
<div id="map" style="height: 500px;"></div>
<script>
function tampilDekat(){
  getCurLocation();

  map_dekat = new google.maps.Map(document.getElementById('map'), {
    zoom: <?=get_option('default_zoom')?>,
    center: {
      lat : default_lat,
      lng : default_lng
    }
  });

  var data = <?=json_encode($db->get_results("SELECT * FROM
tb_tempat"))?>;
  $.each(data, function(k, v){
    var pos = {
      lat : parseFloat(v.lat),
      lng : parseFloat(v.lng)
    };
    var contentString = '<h3>' + v.nama_tempat + '</h3>' +
      '<p align="center"><a href="?m=tempat_detail&ID=' + v.id_tempat + '"
class="link_detail btn btn-primary">Lihat Detail</a>';
    var infowindow = new google.maps.InfoWindow({
      content: contentString
    });
    var marker = new google.maps.Marker({
      position: pos,
      map: map_dekat,
      animation: google.maps.Animation.DROP
    });
    marker.addListener('click', function() {
      infowindow.open(map_dekat, marker);
    });
  });
}

$(function(){
  tampilDekat();
})
</script>

```

A.9 Kode program melihat rute peta cabang

Kode program melihat rute peta cabang terletak pada class cabang_detail. Penulisan kode program ini dapat dilihat pada tabel A.9

Tabel A.9 Kode program melihat rute peta cabang

```

<div class="page-header">
  <h1>Cabang</h1>
</div>
<div id="map" style="height: 500px;"></div>
<script>
function tampilDekat(){
  getCurLocation();

  map_dekat = new google.maps.Map(document.getElementById('map'), {
    zoom: <?=get_option('default_zoom')?>,
    center: {
      lat : default_lat,
      lng : default_lng
    }
  });

  var data = <?=json_encode($db->get_results("SELECT * FROM
tb_tempat"))?>;
  $.each(data, function(k, v){
    var pos = {
      lat : parseFloat(v.lat),
      lng : parseFloat(v.lng)
    };
    var contentString = '<h3>' + v.nama_tempat + '</h3>' +
      '<p align="center"><a href="?"m=tempat_detail&ID=' + v.id_tempat + "'
class="link_detail btn btn-primary">Lihat Detail</a>';
    var infowindow = new google.maps.InfoWindow({
      content: contentString
    });
    var marker = new google.maps.Marker({
      position: pos,
      map: map_dekat,
      animation: google.maps.Animation.DROP
    });
    marker.addListener('click', function() {
      infowindow.open(map_dekat, marker);
    });
  });
}

$(function(){

```

```
tampilDekat();
})
</script>
```

A.10 Kode program melihat detail rute peta cabang

Kode program melihat detail rute peta cabang terletak pada class detail_cabang. Penulisan kode program ini dapat dilihat pada tabel A.10.

Tabel A.10 Kode program melihat detail rute peta cabang

```
<style>
#right-panel {
  font-family: 'Roboto','sans-serif';
  line-height: 30px;
  padding-left: 10px;
}

#right-panel select, #right-panel input {
  font-size: 15px;
}

#right-panel select {
  width: 100%;
}

#right-panel i {
  font-size: 12px;
}

#map {
  height: 500px;
  float: left;
  width: 63%;
}

#right-panel {
  float: right;
  width: 34%;
  height: 500px;
  overflow: auto;
}

.panel {
  height: 500px;
  overflow: auto;
}
</style>
```

```
<?php
$row = $db->get_row("SELECT * FROM tb_tempat WHERE
id_tempat='$_GET[ID]'");
?>
<div class="page-header">
  <h1>Rute Detail ke <?=$row->nama_tempat?></h1>
</div>
<div class="clearfix" style="background: white;">
  <div id="map"></div>
  <div id="right-panel">
    <p>Total Jarak: <span id="total"></span><br />
    Node Terdekat: <span id="terdekat"></span></p>
  </div>
</div>
<p class="help-block">Geser marker atau garis untuk mengubah rute.</p>
<script>

$(function(){
  initMap();
})

var markerArray = [];

function initMap() {
  var map = new google.maps.Map(document.getElementById('map'), {
    zoom: 4,
    center: {lat: default_lat, lng: default_lng} // Australia.
  });

  var directionsService = new google.maps.DirectionsService;
  var directionsDisplay = new google.maps.DirectionsRenderer({
    draggable: true,
    map: map,
    panel: document.getElementById('right-panel')
  });

  var stepDisplay = new google.maps.InfoWindow;

  directionsDisplay.addListener('directions_changed', function() {
    //calculateAndDisplayRoute()
    computeTotalDistance(directionsDisplay.getDirections());
    for (var i = 0; i < markerArray.length; i++) {
      markerArray[i].setMap(null);
    }
  })
}
```

```
        showSteps(directionsDisplay.getDirections(), markerArray, stepDisplay,
map);
        //calculateAndDisplayRoute(pos, {lat: <?=$row->lat?>, lng: <?=$row-
>lng?>}, directionsService, directionsDisplay, stepDisplay, map);
    });

    // Try HTML5 geolocation.
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(function(position) {
            var pos = {
                lat: position.coords.latitude,
                lng: position.coords.longitude
            };

            //infoWindow.setPosition(pos);
            //infoWindow.setContent('Location found. ');
            //infoWindow.open(map);
            //map.setCenter(pos);
            calculateAndDisplayRoute(pos, {lat: <?=$row->lat?>, lng: <?=$row-
>lng?>}, directionsService, directionsDisplay, stepDisplay, map);
        }, function() {
            calculateAndDisplayRoute(getCurLocation(), {lat: <?=$row->lat?>,
lng: <?=$row->lng?>}, directionsService, directionsDisplay, stepDisplay,
map);
        });
    } else {
        // Browser doesn't support Geolocation
        calculateAndDisplayRoute(getCurLocation(), {lat: <?=$row->lat?>, lng:
<?=$row->lng?>}, directionsService, directionsDisplay, stepDisplay, map);
    }
}

function calculateAndDisplayRoute(origin, destination, directionsService,
directionsDisplay, stepDisplay, map) {

    for (var i = 0; i < markerArray.length; i++) {
        markerArray[i].setMap(null);
    }

    directionsService.route({
        origin: origin,
        destination: destination,
        //waypoints: [{location: 'Adelaide, SA'}, {location: 'Broken Hill, NSW'}],
        travelMode: 'DRIVING',
        avoidTolls: true
```

```
}, function(response, status) {
  if (status === 'OK') {
    //console.log(response);
    directionsDisplay.setDirections(response);
    showSteps(response, markerArray, stepDisplay, map);
  } else {
    alert('Could not display directions due to: ' + status);
  }
});
}

function showSteps(directionResult, markerArray, stepDisplay, map) {
  // For each step, place a marker, and add the text to the marker's infowindow.
  // Also attach the marker to an array so we can keep track of it and remove it
  // when calculating new routes.
  var myRoute = directionResult.routes[0].legs[0];

  //console.log(directionResult.routes[0].legs[0]);

  for (var i = 0; i < myRoute.steps.length; i++) {
    var marker = markerArray[i] = markerArray[i] || new
google.maps.Marker();
    //marker.setMap(map);
    //marker.setPosition(myRoute.steps[i].start_location);
    //marker.setIcon('http://maps.google.com/mapfiles/ms/icons/blue-dot.png');
    attachInstructionText(
      stepDisplay, marker, myRoute.steps[i].instructions, map);
  }
}

function attachInstructionText(stepDisplay, marker, text, map) {
  google.maps.event.addListener(marker, 'click', function() {
    // Open an info window when the marker is clicked on, containing the
text
    // of the step.
    stepDisplay.setContent(text);
    stepDisplay.open(map, marker);
  });
}

function computeTotalDistance(result) {
  var total = 0;
  var myroute = result.routes[0];
  var terdekat = 0;

  terdekat = myroute.legs[0].steps[0].distance.value;
```

```

//console.log(result);
for (var i = 0; i < myroute.legs.length; i++) {
  total += myroute.legs[i].distance.value;
}
total = total / 1000;
document.getElementById('total').innerHTML = total + ' km';
document.getElementById('terdekat').innerHTML = (terdekat / 1000) + '
km';// + terdekat + ' m';
}
</script>

```

A.11 Kode program melihat peta pos pantau

Kode program melihat peta pos pantau terletak pada class pos_list.

Penulisan kode program ini dapat dilihat pada tabel A.11.

Tabel A.11 Kode program melihat peta pos pantau

```

<div class="page-header">
  <h1>Tempat</h1>
</div>
<div id="map" style="height: 500px;"></div>
<script>
function tampilDekat(){
  getCurLocation();

  map_dekat = new google.maps.Map(document.getElementById('map'), {
    zoom: <?=get_option('default_zoom')?>,
    center: {
      lat : default_lat,
      lng : default_lng
    }
  });

  var data = <?=json_encode($db->get_results("SELECT * FROM
tb_tempat"))?>;
  $.each(data, function(k, v){
    var pos = {
      lat : parseFloat(v.lat),
      lng : parseFloat(v.lng)
    };
    var contentString = '<h3>' + v.nama_tempat + '</h3>' +
      '<p align="center"><a href="?m=tempat_detail&ID=' + v.id_tempat + "'
class="link_detail btn btn-primary">Lihat Detail</a>';
    var infowindow = new google.maps.InfoWindow({
      content: contentString

```

```

    });
    var marker = new google.maps.Marker({
        position: pos,
        map: map_dekat,
        animation: google.maps.Animation.DROP
    });
    marker.addListener('click', function() {
        infowindow.open(map_dekat, marker);
    });
    });
}

$(function(){
    tampilDekat();
})
</script>

```

A.12 Kode program melihat rute peta pos pantau

Kode program melihat rute peta pos pantau terletak pada class pos_detail. Penulisan kode program ini dapat dilihat pada tabel A.12.

Tabel A.12 Kode program melihat rute peta pos pantau

```

<?php
$row = $db->get_row("SELECT * FROM tb_tempat WHERE
id_tempat='$_GET[ID]'");
?>
<div class="page-header">
    <h1><?=$row->nama_tempat?></h1>
</div>
<div class="row">
    <div class="col-md-6">
        <p>Lokasi: <?=$row->lokasi?></p>
        <div>
            <?=$row->keterangan?>
        </div>
    </div>
    <div class="col-md-6">
        <p>
            <a href="?m=tempat_list" class="btn btn-primary btn-sm"><span
class="glyphicon glyphicon-chevron-left"></span> Lihat semua tempat</a>
            <a href="javascript:void(0)" onclick="showRoute()" class="btn btn-info
btn-sm"> <span class="glyphicon glyphicon-search"></span> Tampilkan Rute
</a>
            <a href="?m=detail&ID=<?=$_GET['ID']?>" class="btn btn-primary
btn-sm"><span class="glyphicon glyphicon-list"></span> Rute Detail</a>

```



```

</p>
<div id="map" style="height: 500px;"></div>
<h3>Galeri</h3>
<div class="row">
  <?php
  foreach($rows as $r):?>
    <div class="col-lg-3 col-md-4 col-xs-6 thumb">
      <a class="thumbnail" href="#" data-image-id="" data-
toggle="modal" data-title="<?=$r->nama_galeri?>" data-
caption="<?=$r->strip_tags($r->ket_galeri?>" data-
image="assets/images/galeri/<?=$r->gambar?>" data-target="#image-gallery">
        nama_galeri?>" />
      </a>
    </div>
  <?php endforeach?>
</div>
</div>
</div>

<div class="modal fade" id="image-gallery" tabindex="-1" role="dialog" aria-
labelledby="myModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal"><span
aria-hidden="true">x</span><span class="sr-only">Close</span></button>
        <h4 class="modal-title" id="image-gallery-title"></h4>
      </div>
      <div class="modal-body">
        <img id="image-gallery-image" class="img-responsive" src="">
      </div>
      <div class="modal-footer">

        <div class="col-md-2">
          <button type="button" class="btn btn-primary" id="show-previous-
image">Previous</button>
        </div>

        <div class="col-md-8 text-justify" id="image-gallery-caption">
          This text will be overwritten by jQuery
        </div>

        <div class="col-md-2">
          <button type="button" id="show-next-image" class="btn btn-
default">Next</button>

```



```
        origin_pos = pos;

        infoWindow.setPosition(pos);
        infoWindow.setContent('Lokasi anda');
        infoWindow.open(map_detail);
        map_detail.setCenter(pos);
    }, function() {
        handleLocationError(true, infoWindow, map_detail.getCenter());
    });
} else {
    handleLocationError(false, infoWindow, map_detail.getCenter());
}
}

function handleLocationError(browserHasGeolocation, infoWindow, pos) {
    infoWindow.setPosition(pos);
    infoWindow.setContent(browserHasGeolocation ?
        'Error: The Geolocation service failed.' :
        'Error: Your browser doesn\'t support geolocation.');
```

infoWindow.open(map);

```
}

//menampilkan rute lokasi
function showRoute(){
    var directionsService = new google.maps.DirectionsService;
    var directionsDisplay = new google.maps.DirectionsRenderer;
    directionsDisplay.setMap(map_detail);
    calculateAndDisplayRoute(directionsService, directionsDisplay);
    console.log('Route displayed ' + ++routeDisplayed);
}

function calculateAndDisplayRoute(directionsService, directionsDisplay) {
    directionsService.route({
        origin: origin_pos,
        destination: dst_pos,
        travelMode: 'DRIVING'
    }, function(response, status) {
        if (status === 'OK') {
            directionsDisplay.setDirections(response);
        } else {
            window.alert('Directions request failed due to ' + status);
        }
    });
}

$(function(){
```

```

    tampilDetail();
  })

$(document).ready(function(){

  loadGallery(true, 'a.thumbnail');

  //This function disables buttons when needed
  function disableButtons(counter_max, counter_current){
    $('#show-previous-image, #show-next-image').show();
    if(counter_max == counter_current){
      $('#show-next-image').hide();
    } else if (counter_current == 1){
      $('#show-previous-image').hide();
    }
  }
});
</script>

```

A.13 Kode program melihat detail rute peta pos pantau

Kode program melihat detail rute peta pos pantau terletak pada class detail_pos. Penulisan kode program ini dapat dilihat pada tabel A.13.

Tabel A.13 Kode program melihat detail rute peta pos pantau

```

<style>
  #right-panel {
    font-family: 'Roboto','sans-serif';
    line-height: 30px;
    padding-left: 10px;
  }

  #right-panel select, #right-panel input {
    font-size: 15px;
  }

  #right-panel select {
    width: 100%;
  }

  #right-panel i {
    font-size: 12px;
  }

  #map {

```

```
height: 500px;
float: left;
width: 63%;
}
#right-panel {
float: right;
width: 34%;
height: 500px;
overflow: auto;
}
.panel {
height: 500px;
overflow: auto;
}
</style>

<?php
$row = $db->get_row("SELECT * FROM tb_tempat WHERE
id_tempat='$_GET[ID]'");
?>
<div class="page-header">
  <h1>Rute Detail ke <?=$row->nama_tempat?></h1>
</div>
<div class="clearfix" style="background: white;">
  <div id="map"></div>
  <div id="right-panel">
    <p>Total Jarak: <span id="total"></span><br />
    Node Terdekat: <span id="terdekat"></span></p>
  </div>
</div>
<p class="help-block">Geser marker atau garis untuk mengubah rute.</p>
<script>

$(function(){
  initMap();
})

var markerArray = [];

function initMap() {
  var map = new google.maps.Map(document.getElementById('map'), {
    zoom: 4,
    center: {lat: default_lat, lng: default_lng} // Australia.
  });
```

```
var directionsService = new google.maps.DirectionsService;
var directionsDisplay = new google.maps.DirectionsRenderer({
  draggable: true,
  map: map,
  panel: document.getElementById('right-panel')
});

var stepDisplay = new google.maps.InfoWindow;

directionsDisplay.addListener('directions_changed', function() {
  //calculateAndDisplayRoute()
  computeTotalDistance(directionsDisplay.getDirections());
  for (var i = 0; i < markerArray.length; i++) {
    markerArray[i].setMap(null);
  }

  showSteps(directionsDisplay.getDirections(), markerArray, stepDisplay,
map);
  //calculateAndDisplayRoute(pos, {lat: <?=$row->lat?>, lng: <?=$row-
>lng?>}, directionsService, directionsDisplay, stepDisplay, map);
});

// Try HTML5 geolocation.
if (navigator.geolocation) {
  navigator.geolocation.getCurrentPosition(function(position) {
    var pos = {
      lat: position.coords.latitude,
      lng: position.coords.longitude
    };

    //infoWindow.setPosition(pos);
    //infoWindow.setContent('Location found.');
```



```
    //infoWindow.open(map);
    //map.setCenter(pos);
    calculateAndDisplayRoute(pos, {lat: <?=$row->lat?>, lng: <?=$row-
>lng?>}, directionsService, directionsDisplay, stepDisplay, map);
  }, function() {
    calculateAndDisplayRoute(getCurLocation(), {lat: <?=$row->lat?>,
lng: <?=$row->lng?>}, directionsService, directionsDisplay, stepDisplay,
map);
  });
} else {
  // Browser doesn't support Geolocation
  calculateAndDisplayRoute(getCurLocation(), {lat: <?=$row->lat?>, lng:
<?=$row->lng?>}, directionsService, directionsDisplay, stepDisplay, map);
}
```

```
    }  
  }  
  
  function calculateAndDisplayRoute(origin, destination, directionsService,  
directionsDisplay, stepDisplay, map) {  
  
    for (var i = 0; i < markerArray.length; i++) {  
      markerArray[i].setMap(null);  
    }  
  
    directionsService.route({  
      origin: origin,  
      destination: destination,  
      //waypoints: [{location: 'Adelaide, SA'}, {location: 'Broken Hill, NSW'}],  
      travelMode: 'DRIVING',  
      avoidTolls: true  
    }, function(response, status) {  
      if (status === 'OK') {  
        //console.log(response);  
        directionsDisplay.setDirections(response);  
        showSteps(response, markerArray, stepDisplay, map);  
      } else {  
        alert('Could not display directions due to: ' + status);  
      }  
    });  
  }  
  
  function showSteps(directionResult, markerArray, stepDisplay, map) {  
    // For each step, place a marker, and add the text to the marker's infowindow.  
    // Also attach the marker to an array so we can keep track of it and remove it  
    // when calculating new routes.  
    var myRoute = directionResult.routes[0].legs[0];  
  
    //console.log(directionResult.routes[0].legs[0]);  
  
    for (var i = 0; i < myRoute.steps.length; i++) {  
      var marker = markerArray[i] = markerArray[i] || new  
google.maps.Marker();  
      //marker.setMap(map);  
      //marker.setPosition(myRoute.steps[i].start_location);  
      //marker.setIcon('http://maps.google.com/mapfiles/ms/icons/blue-dot.png');  
      attachInstructionText(  
        stepDisplay, marker, myRoute.steps[i].instructions, map);  
    }  
  }  
}
```

```

function attachInstructionText(stepDisplay, marker, text, map) {
    google.maps.event.addListener(marker, 'click', function() {
        // Open an info window when the marker is clicked on, containing the
text
        // of the step.
        stepDisplay.setContent(text);
        stepDisplay.open(map, marker);
    });
}

function computeTotalDistance(result) {
    var total = 0;
    var myroute = result.routes[0];
    var terdekat = 0;

    terdekat = myroute.legs[0].steps[0].distance.value;

    //console.log(result);
    for (var i = 0; i < myroute.legs.length; i++) {
        total += myroute.legs[i].distance.value;
    }
    total = total / 1000;
    document.getElementById('total').innerHTML = total + ' km';
    document.getElementById('terdekat').innerHTML = (terdekat / 1000) + '
km';// + terdekat + ' m';
}
</script>

```

A.14 Kode program melihat data flora

Kode program melihat data cabang terletak pada class flora.

Penulisan kode program ini dapat dilihat pada tabel A.14.

Tabel A.14 Kode program melihat data flora

```

<div class="page-header">
    <h1>Flora</h1>
</div>
<div class="panel panel-default">
    <div class="panel-heading">
        <form class="form-inline">
            <input type="hidden" name="m" value="flora" />
            <div class="form-horizontal">
                <input class="form-control" type="text" placeholder="Pencarian. . ."
name="q" value="<?=$_GET['q']?>" />
                <button class="btn btn-success"> Refresh</button>
                <a class="btn bg-green" href="?m=flora_tambah"></span>

```



```

Tambah</a>
    </div>
</form>
</div>
<div class="container">
    <div class="wrapper1">
        <div class="row">
            <div class="col-sm-12">
                <table class="table table-responsive table-bordered">
                    <thead>
                        <tr class="nw">
                            <th>No</th>
                            <th>Gambar</th>
                            <th>Nama Tempat</th>
                            <th>Lat</th>
                            <th>Lng</th>
                            <th>Lokasi</th>
                            <th>Aksi</th>
                        </tr>
                    </thead>
                    <?php
                    $q = esc_field($_GET['q']);

                    $sql = "SELECT *
                        FROM tb_flora p
                        WHERE nama_tempat LIKE '%$q%'
                        ORDER BY id_tempat";
                    $rows = $db->get_results($sql);

                    foreach($rows as $row):?>
                        <tr>
                            <td><?==+$no?></td>
                            <td></td>
                            <td><?=$row->nama_tempat?></td>
                            <td><?=$row->lat?></td>
                            <td><?=$row->lng?></td>
                            <td><?=$row->lokasi?></td>
                            <td class="nw">
                                <a
                                    class="btn
                                    btn-xs
                                    btn-warning"
                                    href="?m=flora_ubah&ID=<?=$row->id_tempat?>"><span
                                    class="fa
                                    fa-
                                    edit"></span></a>
                                <a
                                    class="btn
                                    btn-xs
                                    btn-danger"
                                    href="aksi.php?act=flora_hapus&ID=<?=$row->id_tempat?>"
                                    onclick="return
                                    confirm('Hapus data?')"><span class="fa fa-trash"></span></a>
                            </td>

```

```

    </tr>
    <?php endforeach; ?>
  </table>
</div>
</div>
</div>
</div>

```

A.15 Kode program mengubah data flora

Kode program mengubah data flora terletak pada class flora_ubah.

Penulisan kode program ini dapat dilihat pada tabel A.15

Tabel A.15 Kode program mengubah data flora

```

<?php
    $row = $db->get_row("SELECT * FROM tb_flora WHERE
id_tempat='$_GET[ID]");
?>
<div class="page-header">
    <h1>Ubah Flora</h1>
</div>
<div class="container">
    <div class="wrapper1">
        <div class="col-sm-6">
            <?php if($_POST) include'aksi.php'?>
            <form method="post" action="?m=flora_ubah&ID=<?=$row->
id_tempat?" enctype="multipart/form-data">
                <div class="form-horizontal">
                    <label>Nama Flora <span class="text-danger">*</span></label>
                    <input class="form-control" type="text" name="nama_tempat"
value="<?=$row->nama_tempat?"/>
                </div>
                <div class="form-horizontal">
                    <label>Gambar <span class="text-danger">*</span></label>
                    <input class="form-control-file" type="file" name="gambar" />
                    <p class="help-block">Kosongkan jika tidak mengubah gambar</p>
                    
                </div>
                <div class="form-horizontal">
                    <label>Latitude <span class="text-danger">*</span></label>
                    <input class="form-control" type="text" id="lat" name="lat"
value="<?=$row->lat?"/>
                </div>
                <div class="form-horizontal">
                    <label>Longitude <span class="text-danger">*</span></label>
                    <input class="form-control" type="text" id="lng" name="lng"

```

```

value="<?=$row->lng?>"/>
</div>
<div class="form-horizontal">
  <label>Lokasi <span class="text-danger">*</span></label>
  <input class="form-control" type="text" name="lokasi"
value="<?=$row->lokasi?>"/>
</div>
<div class="form-horizontal">
  <label>Keterangan</label>
  <textarea class="form-control"rows="3"
name="keterangan"><?=$row->keterangan?></textarea>
</div>
<div class="form-horizontal">
  <button class="btn btn-primary"><span class="fa fa-save"></span>
Simpan</button>
  <a class="btn btn-danger" href="?m=flora"><span class="fa fa-
arrow-left"></span> Kembali</a>
</div>
</form>
</div>
</div>
<div class="col-md-6">
  <div id="map" style="height: 400px;"></div>
</div>
</div>
<script>
var defaultCenter = {
  lat : <?=$row->lat * 1?>,
  lng : <?=$row->lng * 1?>
};
function initMap() {

var map = new google.maps.Map(document.getElementById('map'), {
  zoom: <?=get_option('default_zoom')?>,
  center: defaultCenter
});

var marker = new google.maps.Marker({
  position: defaultCenter,
  map: map,
  title: 'Click to zoom',
  draggable:true
});

marker.addListener('drag', handleEvent);

```

```

marker.addListener('dragend', handleEvent);

var infowindow = new google.maps.InfoWindow({
  content: '<h4>Drag untuk pindah lokasi</h4>'
});

infowindow.open(map, marker);
}

function handleEvent(event) {
  document.getElementById('lat').value = event.latLng.lat();
  document.getElementById('lng').value = event.latLng.lng();
}

$(function(){
  initMap();
})
</script>

```

A.16 Kode program melihat data fauna

Kode program melihat data fauna terletak pada class fauna.

Penulisan kode program ini dapat dilihat pada tabel A.16

Tabel A.16 Kode program melihat data fauna

```

<div class="page-header">
  <h1>Fauna</h1>
</div>
<div class="panel panel-default">
  <div class="panel-heading">
    <form class="form-inline">
      <input type="hidden" name="m" value="fauna" />
      <div class="form-horizontal">
        <input class="form-control" type="text" placeholder="Pencarian. . ."
name="q" value="<?=$_GET['q']?>" />
        <button class="btn btn-success"></span> Refresh</button>
        <a class="btn bg-green" href="?m=fauna_tambah"></span>
Tambah</a>
      </div>
    </form>
  </div>
<div class="container">
  <div class="wrapper1">
    <div class="row">
      <div class="col-sm-12">
        <table class="table table-responsive table-bordered">
<thead>

```

```

        <tr class="nw">
            <th>No</th>
            <th>Gambar</th>
            <th>Nama Tempat</th>
            <th>Lat</th>
            <th>Lng</th>
            <th>Lokasi</th>
            <th>Aksi</th>
        </tr>
    </thead>
    <?php
    $q = esc_field($_GET['q']);

    $sql = "SELECT *
    FROM tb_fauna p
    WHERE nama_tempat LIKE '%$q%'
    ORDER BY id_tempat";
    $rows = $db->get_results($sql);

    foreach($rows as $row):?>
    <tr>
        <td><?==+$no?></td>
        <td></td>
        <td><?=$row->nama_tempat?></td>
        <td><?=$row->lat?></td>
        <td><?=$row->lng?></td>
        <td><?=$row->lokasi?></td>
        <td class="nw">
            <a
                class="btn
                btn-xs
                btn-warning"
                href="?m=fauna_ubah&ID=<?=$row->id_tempat?"><span
                class="fa fa-
                edit"></span></a>
            <a
                class="btn
                btn-xs
                btn-danger"
                href="aksi.php?act=fauna_hapus&ID=<?=$row->id_tempat?"
                onclick="return
                confirm('Hapus data?')"><span class="fa fa-trash"></span></a>
            </td>
        </tr>
    <?php endforeach; ?>
    </table>
</div>
</div>
</div>
</div>

```

A.17 Kode program mengubah data fauna

Kode program mengubah data fauna terletak pada class fauna_ubah. Penulisan kode program ini dapat dilihat pada tabel A.18.

Tabel A.17 Kode program mengubah data fauna

```

<?php
    $row = $db->get_row("SELECT * FROM tb_fauna WHERE
id_tempat='$_GET[ID]");
?>
<div class="page-header">
    <h1>Ubah Fauna</h1>
</div>
<div class="container">
    <div class="wrapper1">
        <div class="col-sm-6">
            <?php if($_POST) include'aksi.php'?>
            <form method="post" action="?m=fauna_ubah&ID=<?=$row-
>id_tempat?" enctype="multipart/form-data">
                <div class="form-horizontal">
                    <label>Nama Tempat <span class="text-danger">*</span></label>
                    <input class="form-control" type="text" name="nama_tempat"
value="<?=$row->nama_tempat?"/>
                </div>
                <div class="form-horizontal">
                    <label>Gambar <span class="text-danger">*</span></label>
                    <input class="form-control-file" type="file" name="gambar" />
                    <p class="help-block">Kosongkan jika tidak mengubah gambar</p>
                    
                </div>
                <div class="form-horizontal">
                    <label>Latitude <span class="text-danger">*</span></label>
                    <input class="form-control" type="text" id="lat" name="lat"
value="<?=$row->lat?"/>
                </div>
                <div class="form-horizontal">
                    <label>Longitude <span class="text-danger">*</span></label>
                    <input class="form-control" type="text" id="lng" name="lng"
value="<?=$row->lng?"/>
                </div>
                <div class="form-horizontal">
                    <label>Lokasi <span class="text-danger">*</span></label>
                    <input class="form-control" type="text" name="lokasi"
value="<?=$row->lokasi?"/>
                </div>
            </div>
        </div>
    </div>

```

```
        <label>Keterangan</label>
        <textarea class="form-control" name="keterangan"><?=$row-
>keterangan?></textarea>
    </div>
    <div class="form-horizontal">
        <button class="btn btn-primary"><span class="fa fa-save"></span>
Simpan</button>
        <a class="btn btn-danger" href="?m=fauna"><span class="fa fa-
arrow-left"></span> Kembali</a>
    </div>
</form>
</div>
</div>
<div class="col-md-6">
    <div id="map" style="height: 400px;"></div>
</div>
<script>
var defaultCenter = {
    lat : <?=$row->lat * 1?>,
    lng : <?=$row->lng * 1?>
};
function initMap() {

    var map = new google.maps.Map(document.getElementById('map'), {
        zoom: <?=get_option('default_zoom')?>,
        center: defaultCenter
    });

    var marker = new google.maps.Marker({
        position: defaultCenter,
        map: map,
        title: 'Click to zoom',
        draggable:true
    });

    marker.addListener('drag', handleEvent);
    marker.addListener('dragend', handleEvent);

    var infowindow = new google.maps.InfoWindow({
        content: '<h4>Drag untuk pindah lokasi</h4>'
    });

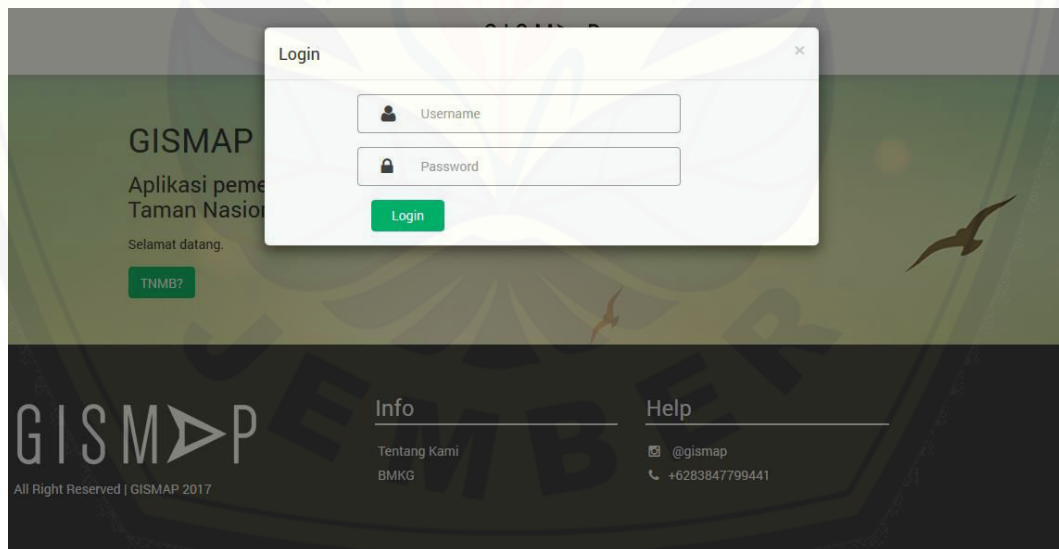
    infowindow.open(map, marker);
}
```

```
function handleEvent(event) {  
    document.getElementById('lat').value = event.latLng.lat();  
    document.getElementById('lng').value = event.latLng.lng();  
}  
  
$(function){  
    initMap();  
}  
</script>
```

B. Tampilan Sistem

B.1 Halaman *login*

Halaman *login* merupakan tampilan awal sistem yang digunakan sebagai sarana *user* untuk mengakses sistem sesuai hak akses yang dimiliki masing-masing user. Untuk melakukan *login* sistem user harus mengisi *username* dan *password* pada kolom yang telah disediakan dan mengklik tombol “Login”. Halaman Login dapat dilihat pada Gambar B.1.



Gambar B.1 Halaman *login*

B.2 Fitur melihat data pemetaan fauna










Fitur ini dapat diakses oleh admin pusat, fitur ini menampilkan data fauna berupa no, gambar, nama fauna, lat, lng, dan lokasi seperti pada Gambar B.2. Terdapat juga tombol “Tambah” jika admin ingin menambah data fauna dan

ikon edit jika admin ingin mengubah data fauna. Fitur ini dapat diakses admin dengan memilih menu “inventaris” kemudian menekan submenu “Fauna”.

GISM▷P Admin Inventaris Peta

Fauna

Pencarian... Refresh Tambah

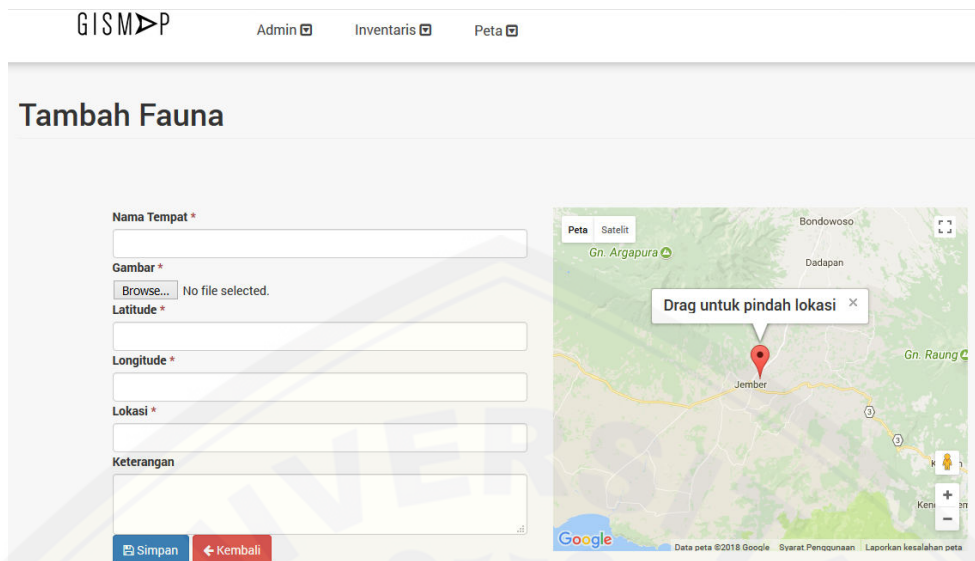
No	Gambar	Nama Tempat	Lat	Lng	Lokasi	Aksi
1		Kijang	-8.39450797241986	113.87083059375004	Andongrejo, Kab. Banyuwangi	 
2		Banteng	-8.466505912757164	113.80216604296879	Andongrejo, Kab. Jember	 
3		Monyet Ekor Panjang	-8.450205667499695	113.84885793750004	Andongrejo, Kab. Jember	 

Gambar B.2 Tampilan Data Fauna

Terdapat kolom pencarian untuk membantu pengguna ketika mencari data, caranya dengan mengetikkan kata kunci kemudian menekan enter maka akan otomatis muncul hasil pencariannya. Tombol “Refresh” digunakan untuk memuat ulang daftar data fauna.

B.3 Fitur menambah data pemetaan fauna

Fitur ini dapat diakses dengan mengklik tombol “Tambah” yang ada di tampilan daftar fauna. Fitur ini memungkinkan admin dapat menambah data fauna. Adapun isi dari tampilannya berupa kolom nama, gambar, *latitude*, *longitude*, lokasi, dan keterangan seperti pada Gambar B.3. Admin hanya perlu mengisi setiap kolom kemudian menekan tombol simpan maka data fauna akan ditambahkan ke *database*. Pada kolom *latitude* dan *longitude* diisi koordinat lokasi atau dengan menggeser ikon lokasi pada kolom peta.



The screenshot shows the 'Tambah Fauna' (Add Fauna) form in the GISMAP application. The form has the following fields:

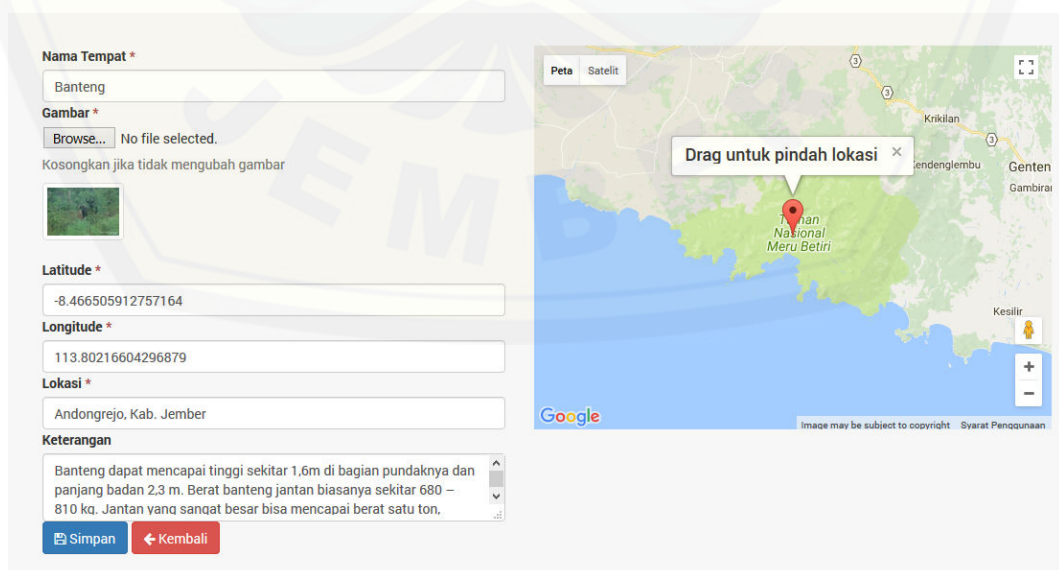
- Nama Tempat ***: A text input field.
- Gambar ***: A file upload field with a 'Browse...' button and the text 'No file selected.'
- Latitude ***: A text input field.
- Longitude ***: A text input field.
- Lokasi ***: A text input field.
- Keterangan**: A text area for additional information.

At the bottom of the form are two buttons: 'Simpan' (Save) and 'Kembali' (Back). To the right of the form is a Google Maps interface showing a location in Jember, with a red pin and a tooltip that says 'Drag untuk pindah lokasi' (Drag to move location).

Gambar B.3 Tampilan tambah data fauna

B.4 Fitur mengubah data pemetaan fauna

Fitur ini dapat diakses dengan mengklik ikon edit yang ada di tampilan daftar fauna. Fitur ini memungkinkan admin dapat mengubah data fauna. Adapun isi dari tampilannya berupa kolom nama, gambar, *latitude*, *longitude*, lokasi, dan keterangan seperti pada Gambar B.4. Admin hanya perlu mengisi setiap kolom kemudian menekan tombol simpan maka data fauna akan diperbarui di *database*. Pada kolom *latitude* dan *longitude* diisi koordinat lokasi atau dengan menggeser ikon lokasi pada kolom peta.



The screenshot shows the 'Tambah Fauna' form with the following data entered:

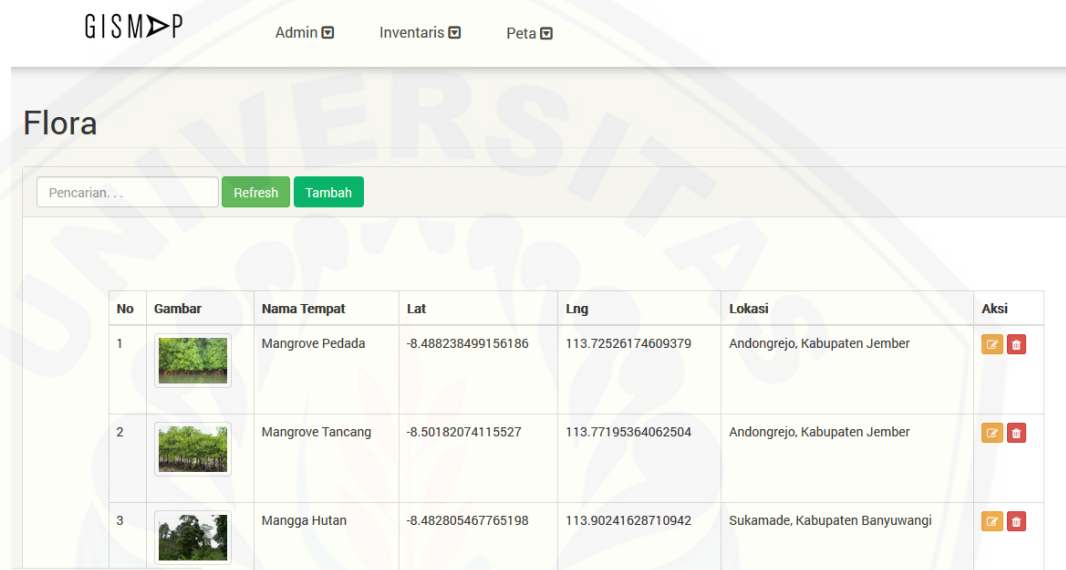
- Nama Tempat ***: Banteng
- Gambar ***: A small image of a banteng is displayed.
- Latitude ***: -8.466505912757164
- Longitude ***: 113.80216604296879
- Lokasi ***: Andongrejo, Kab. Jember
- Keterangan**: Banteng dapat mencapai tinggi sekitar 1,6m di bagian pundaknya dan panjang badan 2,3 m. Berat banteng jantan biasanya sekitar 680 – 810 kg. Jantan yang sangat besar bisa mencapai berat satu ton.










At the bottom of the form are two buttons: 'Simpan' (Save) and 'Kembali' (Back). To the right of the form is a Google Maps interface showing a location in Meru Betiri National Park, with a red pin and a tooltip that says 'Drag untuk pindah lokasi' (Drag to move location).

Gambar B.4 Tampilan ubah data fauna

B.5 Fitur melihat data pemetaan flora

Fitur ini dapat diakses oleh admin pusat, fitur ini menampilkan data flora berupa no, gambar, nama fauna, lat, lng, dan lokasi seperti pada Gambar B.5. Terdapat juga tombol “Tambah” jika admin ingin menambah data flora dan ikon edit jika admin ingin mengubah data flora. Fitur ini dapat diakses admin dengan memilih menu “inventaris” kemudian menekan submenu “Flora”.



No	Gambar	Nama Tempat	Lat	Lng	Lokasi	Aksi
1		Mangrove Pedada	-8.488238499156186	113.72526174609379	Andongrejo, Kabupaten Jember	 
2		Mangrove Tancang	-8.50182074115527	113.77195364062504	Andongrejo, Kabupaten Jember	 
3		Mangga Hutan	-8.482805467765198	113.90241628710942	Sukamade, Kabupaten Banyuwangi	 

Gambar B.5 Tampilan data flora

Terdapat kolom pencarian untuk membantu pengguna ketika mencari data, caranya dengan mengetikkan kata kunci kemudian menekan enter maka akan otomatis muncul hasil pencariannya. Tombol “Refresh” digunakan untuk memuat ulang daftar data flora.

B.6 Fitur menambah data pemetaan flora

Fitur ini dapat diakses dengan mengklik tombol “Tambah” yang ada di tampilan daftar flora. Fitur ini memungkinkan admin dapat menambah data fauna. Adapun isi dari tampilannya berupa kolom nama, gambar, *latitude*, *longitude*, lokasi, dan keterangan seperti pada Gambar B.6. Admin hanya perlu mengisi setiap kolom kemudian menekan tombol simpan maka data flora akan ditambahkan ke *database*. Pada kolom *latitude* dan *longitude* diisi koordinat lokasi atau dengan menggeser ikon lokasi pada kolom peta.

Tambah Flora

Nama Flora *

Gambar *

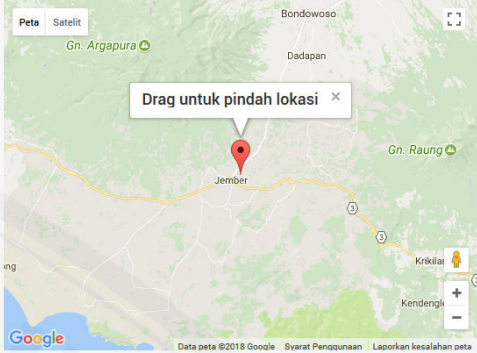
Browse... No file selected.

Latitude *

Longitude *

Lokasi *

Keterangan



Gambar B.6 Tampilan tambah data flora

B.7 Fitur mengubah data pemetaan flora


Fitur ini dapat diakses dengan mengklik ikon edit yang ada di tampilan daftar flora. Fitur ini memungkinkan admin dapat mengubah data flora. Adapun isi dari tampilannya berupa kolom nama, gambar, *latitude*, *longitude*, lokasi, dan keterangan seperti pada Gambar B.7. Admin hanya perlu mengisi setiap kolom kemudian menekan tombol simpan maka data flora akan diperbarui di *database*. Pada kolom *latitude* dan *longitude* diisi koordinat lokasi atau dengan menggeser ikon lokasi pada kolom peta.

Nama Flora *

Gambar *

Browse... No file selected.

Kosongkan jika tidak mengubah gambar




Latitude *

Longitude *

Lokasi *

Keterangan

Buah pedada termasuk ke dalam kelas Angiospermae, tumbuhan biji terbuka yang memiliki propagule atau bakal buah yang sangat unik. Memiliki kemampuan mengapung dan memiliki akar napas untuk membantu dalam proses respirasi serta reproduksinya. Polinasi yang



Gambar B.7 Tampilan ubah data flora

C. Pengerjaan Sistem

C.1 Target pengerjaan

Tabel C.1 Rencana Target Pengerjaan

Minggu	Tugas	Rincian
1	Desain	Pembuatan usecase
		Pembuatan CRC card
		Pembuatan class diagram
		Testing desain
2	Implementasi kode modul 1	ST01
		ST02
		ST03
		ST04
		ST05
		ST06
		Testing
3	Implementasi kode modul 2	ST07
		ST08
		ST09
		ST10
		Testing
4	Testing	Test performa web
		Test fitur web

C.2 Laporan pengerjaan

Tabel C.2 Hasil rencana pengerjaan

Minggu	Tugas	Rincian	Hasil	Checkout
1	Desain	Pembuatan usecase	Gambar	Tepat
		Pembuatan CRC card	Gambar	Tepat
		Pembuatan class diagram	Gambar	Tepat
		Testing desain	Gambar	Tepat
2	Implementasi kode modul 1	ST01	Tabel gambar ,	Tepat
		ST02	Tabel ,	Tepat

Minggu	Tugas	Rincian	Hasil	Checkout
			gambar	
		ST03	Tabel gambar ,	Tepat
		ST04	Tabel gambar ,	Tepat
		ST05	Tabel gambar ,	Tepat
		ST06	Tabel gambar ,	Tepat
		Testing	Tabel	Tepat
3	Implementasi kode modul 2	ST07	Tabel gambar ,	Tepat
		ST08	Tabel gambar ,	Tepat
		ST09	Tabel gambar ,	Tepat
		ST10	Tabel gambar ,	Tepat
		Testing	Tabel	Tepat
4	Testing	Test performa web	Gambar	Tepat
		Test fitur web	Tabel	Tepat

C.3. Feedback

Tabel C.3 Umpan balik

Sumber	Tahapan	Rincian
Klien	Sistem	Login hanya untuk 1 admin
		Tombol logout tidak bisa
Tim	Desain	Class diagram kurang jelas
	Sistem	Rute untuk flora dan fauna tidak ada jalurnya
		Tambahkan radius

D. Testing

D.1 Uji Implementasi Modul 1

Tabel D.1 *Testing* modul 1

Fitur	Fungsi	Hasil	Check out
<i>Login</i>	Menampilkan halaman <i>login</i> , mengisi <i>username</i>	Sistem berhasil menampilkan halaman <i>login</i> , mengisi	√

Fitur	Fungsi	Hasil	Check out
	dan <i>password</i> , mengklik tombol <i>login</i> , masuk ke sistem	<i>username</i> dan <i>password</i> , mengklik tombol <i>login</i> , masuk ke sistem	
Tambah Pemetaan Cabang	Menampilkan <i>form</i> tambah data cabang, mengisi <i>form</i> , inputan peta statis, mengklik tombol tambah	Sistem berhasil menampilkan <i>form</i> tambah data cabang, mengisi <i>form</i> , inputan peta statis, mengklik tombol tambah	√
Lihat Pemetaan Cabang	Menampilkan data peta (nama, gmbr, long, lng, populasi, lokasi), menampilkan tombol tambah data, menampilkan aksi ubah (<i>icon</i>)	Sistem berhasil menampilkan data peta (nama, gmbr, long, lng, populasi, lokasi), menampilkan tombol tambah data, menampilkan aksi ubah (<i>icon</i>)	√
Ubah Pemetaan Cabang	Menampilkan <i>form</i> ubah data cabang, mengisi <i>form</i> , inputan peta statis, mengklik tombol ubah	Sistem berhasil menampilkan <i>form</i> ubah data cabang, mengisi <i>form</i> , inputan peta statis, mengklik tombol ubah	√
Tambah Pemetaan Pos Pantau	Menampilkan <i>form</i> tambah data pos pantau, mengisi <i>form</i> , inputan peta statis, mengklik tombol tambah	Sistem berhasil menampilkan <i>form</i> tambah data pos pantau, mengisi <i>form</i> , inputan peta statis, mengklik tombol tambah	√
Lihat Pemetaan Pos Pantau	Menampilkan data peta (nama, gmbr, long, lng, populasi, lokasi), menampilkan tombol tambah data, menampilkan aksi ubah (<i>icon</i>)	Sistem berhasil menampilkan data peta (nama, gmbr, long, lng, populasi, lokasi), menampilkan tombol tambah data, menampilkan aksi ubah (<i>icon</i>)	√
Ubah Pemetaan Pos Pantau	Menampilkan <i>form</i> ubah data pos pantau, mengisi <i>form</i> , inputan peta statis, mengklik tombol ubah	Sistem berhasil menampilkan <i>form</i> ubah data pos pantau, mengisi <i>form</i> , inputan peta statis, mengklik tombol ubah	√
Tambah Pemetaan Fauna	Menampilkan <i>form</i> tambah data fauna, mengisi <i>form</i> , inputan peta statis, mengklik tombol tambah	Sistem berhasil menampilkan <i>form</i> tambah data fauna, mengisi <i>form</i> , inputan peta statis, mengklik tombol tambah	√
Lihat Pemetaan	Menampilkan data peta (nama, gmbr, long, lng, populasi, lokasi),	Sistem berhasil menampilkan data peta (nama, gmbr, long, lng, populasi, lokasi),	√

Fitur	Fungsi	Hasil	Check out
Fauna	menampilkan tombol tambah data, menampilkan aksi ubah (<i>icon</i>)	menampilkan tombol tambah data, menampilkan aksi ubah (<i>icon</i>)	
Ubah Pemetaan Fauna	Menampilkan form ubah data fauna, mengisi <i>form</i> , inputan peta statis, mengklik tombol ubah	Sistem berhasil menampilkan form ubah data fauna, mengisi <i>form</i> , inputan peta statis, mengklik tombol ubah	√
Tambah Pemetaan Flora	Menampilkan form tambah data flora, mengisi <i>form</i> , inputan peta statis, mengklik tombol tambah	Sistem berhasil menampilkan form tambah data flora, mengisi <i>form</i> , inputan peta statis, mengklik tombol tambah	√
Lihat Pemetaan Flora	Menampilkan data peta (nama, gmbr, long, lng, populasi, lokasi), menampilkan tombol tambah data, menampilkan aksi ubah (<i>icon</i>)	Sistem berhasil menampilkan data peta (nama, gmbr, long, lng, populasi, lokasi), menampilkan tombol tambah data, menampilkan aksi ubah (<i>icon</i>)	√
Ubah Pemetaan Flora	Menampilkan <i>form</i> ubah data flora, mengisi <i>form</i> , inputan peta statis, mengklik tombol ubah	Sistem berhasil menampilkan <i>form</i> ubah data flora, mengisi <i>form</i> , inputan peta statis, mengklik tombol ubah	√

D.2 Uji Implementasi Modul 2

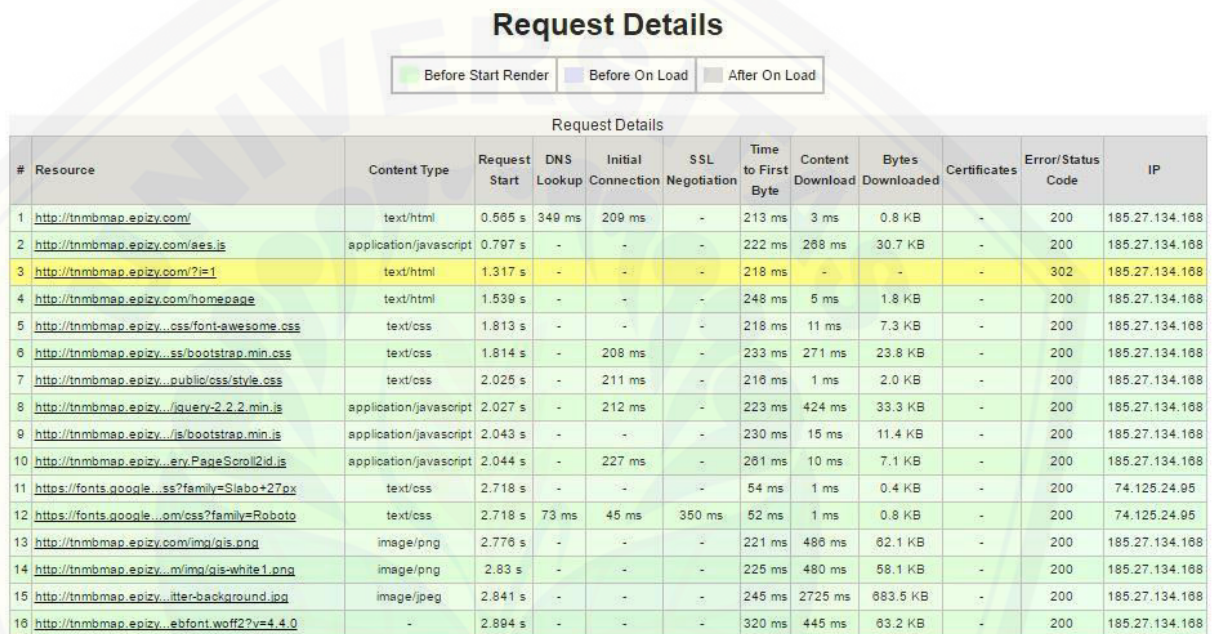
Tabel D.2 *Testing* modul 2

Fitur	Fungsi	Hasil	Check out
Lihat Daftar Peta Cabang	Menampilkan peta dengan <i>mark label</i> bertanda pin lokasi, menampilkan tombol detail ketika <i>marker</i> diklik	Sistem berhasil menampilkan peta dengan <i>mark label</i> bertanda pin lokasi, menampilkan tombol detail ketika <i>marker</i> diklik	√
Lihat Detail Peta Cabang	Menampilkan peta statis lokasi dari peta yang diklik, menampilkan detail dari lokasi yang dipilih	Sistem berhasil menampilkan peta statis lokasi dari peta yang diklik, menampilkan detail dari lokasi yang dipilih	√
Lihat Rute	Menampilkan rute yang	Sistem berhasil menampilkan	√

Fitur	Fungsi	Hasil	Check out
Cabang	bisa dilalui untuk mencapai lokasi tujuan dengan titik awal adalah lokasi sistem diakses	rute yang bisa dilalui untuk mencapai lokasi tujuan dengan titik awal adalah lokasi sistem diakses	
Lihat Detail Rute Cabang	Menampilkan rute yang bisa dilalui untuk mencapai lokasi tujuan dengan titik awal adalah lokasi sistem diakses dengan detail per jalan yang akan dilalui	Sistem berhasil menampilkan rute yang bisa dilalui untuk mencapai lokasi tujuan dengan titik awal adalah lokasi sistem diakses dengan detail per jalan yang akan dilalui	√
Lihat Daftar Peta Pos Pantau	Menampilkan peta dengan <i>mark label</i> bertanda pin lokasi, menampilkan tombol detail ketika <i>marker</i> diklik	Sistem berhasil menampilkan peta dengan <i>mark label</i> bertanda pin lokasi, menampilkan tombol detail ketika <i>marker</i> diklik	√
Lihat Detail Peta Pos Pantau	Menampilkan peta statis lokasi dari peta yang diklik, menampilkan detail dari lokasi yang dipilih	Sistem berhasil menampilkan peta statis lokasi dari peta yang diklik, menampilkan detail dari lokasi yang dipilih	√
Lihat Rute Pos Pantau	Menampilkan rute yang bisa dilalui untuk mencapai lokasi tujuan dengan titik awal adalah lokasi sistem diakses	Sistem berhasil menampilkan rute yang bisa dilalui untuk mencapai lokasi tujuan dengan titik awal adalah lokasi sistem diakses	√
Lihat Detail Rute Pos Pantau	Menampilkan rute yang bisa dilalui untuk mencapai lokasi tujuan dengan titik awal adalah lokasi sistem diakses dengan detail per jalan yang akan dilalui	Sistem berhasil menampilkan rute yang bisa dilalui untuk mencapai lokasi tujuan dengan titik awal adalah lokasi sistem diakses dengan detail per jalan yang akan dilalui	√
Lihat Daftar Peta Fauna	Menampilkan peta dengan <i>mark label</i> bertanda pin lokasi dan radius perkiraan lokasi, menampilkan tombol detail ketika <i>marker</i> diklik	Sistem berhasil menampilkan peta dengan <i>mark label</i> bertanda pin lokasi dan radius perkiraan lokasi, menampilkan tombol detail ketika <i>marker</i> diklik	√
Lihat Daftar Peta Flora	Menampilkan peta dengan <i>mark label</i> bertanda pin lokasi dan	Sistem berhasil menampilkan peta dengan <i>mark label</i> bertanda pin lokasi dan radius	√

Fitur	Fungsi	Hasil	Check out
	radius perkiraan lokasi, menampilkan tombol detail ketika marker diklik	perkiraan lokasi, menampilkan tombol detail ketika marker diklik	

D.3 Uji Performa Sistem



Gambar D.1 Testing performa sistem

Note.

Resource: Alamat web yang diakses

Content Type: Tipe dari halaman yang diakses

Request Start: Mulai ditanggapinya permintaan pengaksesan web

Content Download: Waktu untuk mengunduh isi dari halaman yang ditampilkan pada web

D.4 Uji Black Box

Tabel. D.3 Black Box Testing

Fitur	Fungsi	Hasil	Check out
<i>Login</i>	Menampilkan halaman <i>login</i> , mengisi <i>username</i>	Sistem berhasil menampilkan halaman <i>login</i> , mengisi	√

Fitur	Fungsi	Hasil	Check out
	dan <i>password</i> , mengklik tombol <i>login</i> , masuk ke sistem	<i>username</i> dan <i>password</i> , mengklik tombol <i>login</i> , masuk ke sistem	
Tambah Pemetaan Cabang	Menampilkan <i>form</i> tambah data cabang, mengisi <i>form</i> , inputan peta statis, mengklik tombol tambah	Sistem berhasil menampilkan <i>form</i> tambah data cabang, mengisi <i>form</i> , inputan peta statis, mengklik tombol tambah	√
Lihat Pemetaan Cabang	Menampilkan data peta (nama, gmbr, long, lng, populasi, lokasi), menampilkan tombol tambah data, menampilkan aksi ubah (<i>icon</i>)	Sistem berhasil menampilkan data peta (nama, gmbr, long, lng, populasi, lokasi), menampilkan tombol tambah data, menampilkan aksi ubah (<i>icon</i>)	√
Ubah Pemetaan Cabang	Menampilkan <i>form</i> ubah data cabang, mengisi <i>form</i> , inputan peta statis, mengklik tombol ubah	Sistem berhasil menampilkan <i>form</i> ubah data cabang, mengisi <i>form</i> , inputan peta statis, mengklik tombol ubah	√
Tambah Pemetaan Pos Pantau	Menampilkan <i>form</i> tambah data pos pantau, mengisi <i>form</i> , inputan peta statis, mengklik tombol tambah	Sistem berhasil menampilkan <i>form</i> tambah data pos pantau, mengisi <i>form</i> , inputan peta statis, mengklik tombol tambah	√
Lihat Pemetaan Pos Pantau	Menampilkan data peta (nama, gmbr, long, lng, populasi, lokasi), menampilkan tombol tambah data, menampilkan aksi ubah (<i>icon</i>)	Sistem berhasil menampilkan data peta (nama, gmbr, long, lng, populasi, lokasi), menampilkan tombol tambah data, menampilkan aksi ubah (<i>icon</i>)	√
Ubah Pemetaan Pos Pantau	Menampilkan <i>form</i> ubah data pos pantau, mengisi <i>form</i> , inputan peta statis, mengklik tombol ubah	Sistem berhasil menampilkan <i>form</i> ubah data pos pantau, mengisi <i>form</i> , inputan peta statis, mengklik tombol ubah	√
Tambah Pemetaan Fauna	Menampilkan <i>form</i> tambah data fauna, mengisi <i>form</i> , inputan peta statis, mengklik tombol tambah	Sistem berhasil menampilkan <i>form</i> tambah data fauna, mengisi <i>form</i> , inputan peta statis, mengklik tombol tambah	√
Lihat Pemetaan	Menampilkan data peta (nama, gmbr, long, lng, populasi, lokasi),	Sistem berhasil menampilkan data peta (nama, gmbr, long, lng, populasi, lokasi),	√

Fitur	Fungsi	Hasil	Check out
Fauna	menampilkan tombol tambah data, menampilkan aksi ubah (<i>icon</i>)	menampilkan tombol tambah data, menampilkan aksi ubah (<i>icon</i>)	
Ubah Pemetaan Fauna	Menampilkan form ubah data fauna, mengisi <i>form</i> , inputan peta statis, mengklik tombol ubah	Sistem berhasil menampilkan form ubah data fauna, mengisi <i>form</i> , inputan peta statis, mengklik tombol ubah	√
Tambah Pemetaan Flora	Menampilkan form tambah data flora, mengisi <i>form</i> , inputan peta statis, mengklik tombol tambah	Sistem berhasil menampilkan form tambah data flora, mengisi <i>form</i> , inputan peta statis, mengklik tombol tambah	√
Lihat Pemetaan Flora	Menampilkan data peta (nama, gmbr, long, lng, populasi, lokasi), menampilkan tombol tambah data, menampilkan aksi ubah (<i>icon</i>)	Sistem berhasil menampilkan data peta (nama, gmbr, long, lng, populasi, lokasi), menampilkan tombol tambah data, menampilkan aksi ubah (<i>icon</i>)	√
Ubah Pemetaan Flora	Menampilkan <i>form</i> ubah data flora, mengisi <i>form</i> , inputan peta statis, mengklik tombol ubah	Sistem berhasil menampilkan <i>form</i> ubah data flora, mengisi <i>form</i> , inputan peta statis, mengklik tombol ubah	√
Lihat Daftar Peta Cabang	Menampilkan peta dengan <i>mark label</i> bertanda pin lokasi, menampilkan tombol detail ketika <i>marker</i> diklik	Sistem berhasil menampilkan peta dengan <i>mark label</i> bertanda pin lokasi, menampilkan tombol detail ketika <i>marker</i> diklik	√
Lihat Detail Peta Cabang	Menampilkan peta statis lokasi dari peta yang diklik, menampilkan detail dari lokasi yang dipilih	Sistem berhasil menampilkan peta statis lokasi dari peta yang diklik, menampilkan detail dari lokasi yang dipilih	√
Lihat Rute Cabang	Menampilkan rute yang bisa dilalui untuk mencapai lokasi tujuan dengan titik awal adalah lokasi sistem diakses	Sistem berhasil menampilkan rute yang bisa dilalui untuk mencapai lokasi tujuan dengan titik awal adalah lokasi sistem diakses	√
Lihat Detail Rute Cabang	Menampilkan rute yang bisa dilalui untuk mencapai lokasi tujuan	Sistem berhasil menampilkan rute yang bisa dilalui untuk mencapai lokasi tujuan	√

Fitur	Fungsi	Hasil	Check out
	dengan titik awal adalah lokasi sistem diakses dengan detail per jalan yang akan dilalui	dengan titik awal adalah lokasi sistem diakses dengan detail per jalan yang akan dilalui	
Lihat Daftar Peta Pos Pantau	Menampilkan peta dengan <i>mark label</i> bertanda pin lokasi, menampilkan tombol detail ketika <i>marker</i> diklik	Sistem berhasil menampilkan peta dengan <i>mark label</i> bertanda pin lokasi, menampilkan tombol detail ketika <i>marker</i> diklik	√
Lihat Detail Peta Pos Pantau	Menampilkan peta statis lokasi dari peta yang diklik, menampilkan detail dari lokasi yang dipilih	Sistem berhasil menampilkan peta statis lokasi dari peta yang diklik, menampilkan detail dari lokasi yang dipilih	√
Lihat Rute Pos Pantau	Menampilkan rute yang bisa dilalui untuk mencapai lokasi tujuan dengan titik awal adalah lokasi sistem diakses	Sistem berhasil menampilkan rute yang bisa dilalui untuk mencapai lokasi tujuan dengan titik awal adalah lokasi sistem diakses	√
Lihat Detail Rute Pos Pantau	Menampilkan rute yang bisa dilalui untuk mencapai lokasi tujuan dengan titik awal adalah lokasi sistem diakses dengan detail per jalan yang akan dilalui	Sistem berhasil menampilkan rute yang bisa dilalui untuk mencapai lokasi tujuan dengan titik awal adalah lokasi sistem diakses dengan detail per jalan yang akan dilalui	√
Lihat Daftar Peta Fauna	Menampilkan peta dengan <i>mark label</i> bertanda pin lokasi dan radius perkiraan lokasi, menampilkan tombol detail ketika <i>marker</i> diklik	Sistem berhasil menampilkan peta dengan <i>mark label</i> bertanda pin lokasi dan radius perkiraan lokasi, menampilkan tombol detail ketika <i>marker</i> diklik	√
Lihat Daftar Peta Flora	Menampilkan peta dengan <i>mark label</i> bertanda pin lokasi dan radius perkiraan lokasi, menampilkan tombol detail ketika <i>marker</i> diklik	Sistem berhasil menampilkan peta dengan <i>mark label</i> bertanda pin lokasi dan radius perkiraan lokasi, menampilkan tombol detail ketika <i>marker</i> diklik	√

E. Biaya Pengeluaran

Tabel E.1 Biaya pengeluaran

Pengeluaran	Rincian			Jumlah
	waktu	Upah per jam	Pengerjaan per hari (jam)	
Pengerjaan sistem	30 hari	10.000	8 jam	2.400.000
	Jumlah fitur	Upah per fitur		
Fitur	23	15.000		345.000
	Total			

F. Wawancara

Hasil wawancara

Narasumber

Adie Setyanto, S.Si.

Bagian pelayanan masyarakat

Pedoman wawancara

1. Siapakah yang sering datang untuk meminta data ke TNMB?
2. Data apa yang sering diminta?
3. Data apa yang biasanya paling dibutuhkan?

Hasil wawancara

No Pertanyaan

1. Siapa yang sering datang untuk meminta ijin penelitian dan pengambilan data ke TNMB?
Banyak, rata-rata dari kalangan mahasiswa kehutanan dari universitas luar kota
2. Kalau dari orang Jember sendiri?
Jarang, tapi ya ada mahasiswa pecinta alam yang melakukan penelitian disana
3. Kebanyakan dari peneliti biasanya meminta data apa?

Data inventaris hewan dan tumbuhan, biasanya meminta titik koordinat biar bisa maksimal dalam melakukan penelitian. Kemudian data rute yang bisa dilalui.

4. Dalam bentuk apa data yang diberikan?

Kalau ada yang minta ya nyari dulu di inventaris kita. Data inventaris kita dalam bentuk excel kita kasih ke pemohon

5. Data apa yang paling sering diminta, dan penting dibutuhkan peneliti?

Data inventaris flora dan fauna, itu yang selalu ada dalam permohonan penelitian

6. Semisal ada sistem yang bisa digunakan untuk memudahkan pemberian data kepada peneliti, bentuk sistem yang bagaimana agar bisa memenuhi kebutuhan permohonan penelitian tentang flora dan fauna?

Simpel saja, yang penting bisa menampilkan data dari inventaris flora dan fauna dalam 1 layar. Terus misal ada petanya untuk melihat posisi dari flora atau fauna itu juga bisa memudahkan peneliti saat melakukan kegiatan. Mungkin bisa juga ditambahi lokasi pos dan balai TNMB. Soalnya kebanyakan masih ada yang tidak tahu