

**RANCANG BANGUN WEBSITE VISUALISASI DATA HUTAN BAKAU
UNTUK OPTIMALISASI PENDATAAN HUTAN BAKAU PETANGORAN
DENGAN METODE *RAD***

(Skripsi)

**Oleh:
RAHMAT RAMADHAN
2015061037**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024**

**RANCANG BANGUN WEBSITE VISUALISASI DATA HUTAN BAKAU
UNTUK OPTIMALISASI PENDATAAN HUTAN BAKAU PETANGORAN
DENGAN METODE *RAD***

**Oleh:
RAHMAT RAMADHAN**

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai
Gelar SARJANA TEKNIK**

**Pada
Program Studi Teknik Informatika Jurusan Teknik Elektro
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG**

2024

ABSTRAK

RANCANG BANGUN WEBSITE VISUALISASI DATA HUTAN BAKAU UNTUK OPTIMALISASI PENDATAAN HUTAN BAKAU PETANGORAN DENGAN METODE *RAD*

Oleh

RAHMAT RAMADHAN

Hutan Mangrove Petangoran di Provinsi Lampung merupakan kawasan ekowisata yang tidak hanya menawarkan keindahan alam, tetapi juga memiliki peran strategis dalam mitigasi bencana dan penyerapan karbon. Pohon bakau berfungsi sebagai pemecah ombak alami untuk mengurangi dampak tsunami dan gelombang laut, serta membantu menyerap karbon dioksida (CO₂) untuk mengurangi efek pemanasan global. Namun, monitoring pohon *mangrove* di kawasan ini masih dilakukan secara manual, yang menyebabkan kendala seperti rendahnya efisiensi waktu, ketidakakuratan data, dan terbatasnya jangkauan area. Penelitian ini bertujuan untuk mengatasi masalah tersebut dengan mengembangkan sebuah *website* berbasis sistem informasi yang mendukung visualisasi data *mangrove* secara akurat dan efisien. *Website* ini dirancang menggunakan metode *Rapid Application Development* (RAD), yang dilakukan sebanyak dua kali iterasi. Fokus penelitian adalah pada pengembangan *frontend website* yang menampilkan peta zona, dan titik-titik pohon *mangrove* berdasarkan data yang diperoleh melalui pengolahan citra oleh tim *backend*. Hasil penelitian menunjukkan bahwa data berhasil dikirim melalui API, diterima, dan divisualisasikan dengan baik dalam *website*. Proses pendataan pohon bakau yang sebelumnya membutuhkan waktu lima hingga sepuluh menit kini dapat dipercepat menjadi sepuluh hingga tiga puluh detik. Pengujian pada kebutuhan fungsional menggunakan *black-box testing* dan telah sesuai dengan skenario uji maupun tampilan yang diinginkan oleh pengguna. Sedangkan pengujian non-fungsional menggunakan *User Experience Questionnaire* (UEQ) yang melibatkan 30 responden dengan hasil pengujian kategori *Excellent* dalam hal daya tarik, ketepatan, stimulasi, dan kebaruan inovasi, selanjutnya parameter kejelasan, dan efisiensi mendapatkan kategori *Good*.

Kata kunci : Konservasi *Mangrove*, *Rapid Application Development*, *User Experience Questionnaire*, Visualisasi Data, *Website*.

ABSTRACT

DESIGN AND DEVELOPMENT OF A DATA VISUALIZATION WEBSITE FOR OPTIMIZING MANGROVE DATA COLLECTION

By

RAHMAT RAMDAHAN

The Mangrove Forest of Petangoran in Lampung Province is an ecotourism area that not only offers natural beauty but also plays a strategic role in disaster mitigation and carbon absorption. Mangrove trees act as natural wave breakers, reducing the impact of tsunamis and ocean waves, while also absorbing carbon dioxide (CO₂) to mitigate global warming effects. However, the manual data collection process for mangrove trees in this area poses challenges such as low time efficiency, data inaccuracies, and limited coverage. This study aims to address these issues by developing an information system-based website to support accurate and efficient mangrove data visualization. The website was designed using the Rapid Application Development (RAD) method, conducted in two iterations. The focus of this research was the frontend development of the website, which displays zone maps and mangrove tree points based on data processed by the backend team using image analysis. The results show that the data was successfully transmitted through the API, received, and visualized effectively on the website. The mangrove tree data collection process, which previously took five to ten minutes, has now been reduced to ten to thirty seconds. Functional testing using black-box testing confirmed compliance with test scenarios and user expectations. Meanwhile, non-functional testing using the User Experience Questionnaire (UEQ) involving 30 respondents revealed results in the Excellent category for attractiveness, accuracy, stimulation, and innovation, while clarity and efficiency achieved a Good rating.

Keywords: Data Visualization, Mangrove Conservation, Rapid Application Development, User Experience Questionnaire, Website.

Judul Skripsi

: **RANCANG BANGUN WEBSITE
VISUALISASI DATA HUTAN BAKAU
UNTUK OPTIMALISASI PENDATAAN
HUTAN BAKAU PETANGORAN
DENGAN METODE RAD**

Nama Mahasiswa

: **Rahmat Ramadhan**

Nomor Pokok Mahasiswa

: **2015061037**

Program Studi

: **Teknik Informatika**

Jurusan

: **Teknik Elektro**

Fakultas

: **Teknik**

MENYETUJUI

1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping

Mona Arif Muda, S.T.,M.T.
NIP. 197111122000031002

Mahendra Pratama, S.T., M.Eng.
NIP. 199112152019031013

2. Mengetahui

Ketua Jurusan
Teknik Elektro

Ketua Program
Studi Teknik Informatika

Herlinawati, S.T., M.T.
NIP. 197103141999032001

Yessi Mulyani, S.T.,M.T.
NIP. 197312262000122001

MENGESAHKAN

1. Tim Penguji

Ketua

: **Mona Arif Muda, S.T.,M.T.**



Sekretaris

: **Mahendra Pratama, S.T., M.Eng.**



Penguji

: **Ir. Ing. Hery Dian Septama, S.T., IPM**



2. Dekan Fakultas Teknik



Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc.

NIP. 197509282001121002



Tanggal Lulus Ujian Skripsi : 29 November 2024

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini , menyatakan bahwa skripsi saya dengan judul “Rancang Bangun *Webste* Visualisasi Data Hutan Bakau Untuk Optimalisasi Pendataan Hutan Bakau Petangoran Dengan Metode RAD” dibuat oleh saya sendiri. Semua hasil yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila pernyataan saya tidak benar dan terbukti bahwa skripsi ini merupakan salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, Desember 2024

Penulis,



Rahmat Ramadhan

2015061037

RIWAYAT HIDUP



Penulis dilahirkan di Jakarta, pada tanggal 1 Desember 2000. Penulis merupakan anak pertama dari pasangan Bapak Oktianto dan Ibu Nurqmari Salim. Penulis menyelesaikan pendidikannya di TK Kartika 1 pada tahun 2007, SD Negeri 03 Cipinang Melayu pada tahun 2013, SMP Negeri 109 Jakarta pada tahun 2016, dan SMAS Pusaka 1 Jakarta pada tahun 2019. Pada tahun 2020 penulis terdaftar sebagai mahasiswa Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik Universitas Lampung melalui jalur SBMPTN. Selama menjadi mahasiswa, penulis memfokuskan untuk mendalami bidang konsentrasi Rekayasa Perangkat Lunak, dan melakukan beberapa kegiatan, antara lain:

1. Menjadi anggota Himpunan Mahasiswa Teknik Elektro Universitas Lampung, Departemen Pengembangan Keteknikan Mahasiswa Teknik Elektro Universitas Lampung pada tahun 2020.
2. Menjadi anggota Himpunan Mahasiswa Teknik Elektro Universitas Lampung, Departemen Pengembangan Keteknikan Mahasiswa Teknik Elektro Universitas Lampung pada tahun 2021.
3. Mengikuti kegiatan Kampus Merdeka program Kredensial Mikro Mahasiswa Indonesia (KMMI) pada bidang pengembangan aplikasi *android* dengan *flutter* pada tahun 2022.
4. Mengikuti program Studi Independen Kampus Merdeka dari Kementerian Pendidikan dan Budaya dengan mengambil kelas *Android Developer* di Dicoding Indonesia pada tahun 2022.
5. Melaksanakan Kuliah Kerja Nyata MBKM di Desa Campang Jaya, Kecamatan Sukabumi, Kota Bandar Lampung, Provinsi Lampung pada bulan Desember sampai dengan Januari 2023.
6. Mengikuti program Studi Independen Kampus Merdeka dari Kementerian Pendidikan dan Budaya dengan mengambil kelas *Cloud Computing* di

Bangkit *Academy* Indonesia pada tahun 2023.

7. Mengikuti Magang sebagai *Information Technology Development* di divisi Sistem Informasi pada Kementerian Pemuda dan Olahraga Republik Indonesia (KEMENPORA) tahun 2023.
8. Mengikuti kegiatan Digital Talent Scholarship program *Fresh Graduate Academy* pada bidang *Android Developer with React native* dengan judul project *Move Apps* pada tahun 2024.

MOTTO

Gantungkan cita-cita mu setinggi langit! Bermimpilah setinggi langit. Jika engkau jatuh, engkau akan jatuh di antara bintang-bintang
(Ir. Soekarno)

Kalau engkau menginginkan sesuatu, jika hal tersebut tidak bisa dibeli maka buatlah dengan tanganmu sendiri
(Rahmat Ramadhan)

Jangan pernah menyerah pada suatu hal, karena dititik itulah kalian akan meningkatkan kualitas diri kita
(Tara Arts)

Usaha tanpa doa adalah kesombongan, doa tanpa usaha adalah kebohongan.
Jangan lupa akhiri dengan tawakkal
(Anonymous)

Jika kau mendapat sebuah masalah janganlah melihat dari sudut pandangmu saja..
kerana kau dapat melihat kelebihan dan kekurangan dari kedua nya
(Anonymous)



Sujud syukur kupersembahkan kepada Allah سُبْحَانَهُ وَتَعَالَى, Tuhan Yang Maha Esa dan Maha Besar. Berkat limpahan rahmat-Mu saya bisa menjadi pribadi yang bertaqwa, beriman, dan berilmu. Semoga dengan keberhasilan yang telah dicapai ini saya dapat menuju masa depan yang lebih baik dan dapat menggapai cita-cita serta selalu berada di jalan-Mu.

KUPERSEMBAHKAN SKRIPSI INI TERUNTUK:

“Ayahanda Oktianto dan Ibunda Nurqamari Salim atas dukungan dan kasih sayang yang diberikan mulai dari saya ada di dunia ini sampai saya sudah besar seperti sekarang ini. Terima kasih kepada Ayah dan Ibu atas doa yang tak henti-hentinya dipanjatkan serta pengorbanan yang tak terhitung nilainya. Semoga dengan ilmu dan cita-cita yang saya dapatkan kelak akan menjadi amal jariyah bagi Ayah dan Ibu”

“Terima Kasih untuk Om Teddy Hidayat yang selalu memberikan perhatian, pengertian, dukungan, dan semangat dalam setiap langkah dalam perkuliahan dan dalam penyusunan skripsi ini hingga saat ini.”

“Terima Kasih untuk Tamte Destriana yang selalu merawat saya dan memberikan perhatian, pengertian, dukungan, dan semangat dalam setiap langkah hingga saat ini.”

“Diriku sendiri. Maaf untuk malam-malam panjang dengan mata yang sulit tertidur, kepala yang sakit, dan lelah pikiran. Terima Kasih telah berjuang dan bekerjasama selama ini. Kamu hebat.”

“Terima Kasih kepada teman-teman Teknik Informatika 2020 yang bersama dan berkembang bersama selama perkuliahan di kampus tercinta Universitas Lampung. Semoga kelak kita semua yang kita impikan akan menjadi kenyataan.”

“Terima Kasih kepada seseorang yang selalu hadir dan selalu memberi bantuan, semangat, motivasi saat saya jatuh, dan memberikan saya tujuan hidup serta yang selalu mengingatkan saya untuk bersyukur saat saya bahagia. Semoga Allah selalu memberikan kita kebahagiaan dan bertanggung jawab dengan apapun jalan yang kita pilih kedepannya.”

SANWACANA

Segala puji syukur kehadiran Allah SWT, Tuhan semesta alam yang Maha Pengasih lagi Maha Penyayang, atas limpahan rahmat, taufik serta hidayah-Nya sehingga penulis dapat menyelesaikan penulisan skripsi dengan judul **“Rancang Bangun *Webste* Visualisasi Data Hutan Bakau Untuk Optimalisasi Pendataan Hutan Bakau Petangoran Dengan Metode RAD”**. Sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Informatika Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung.

Penulisan skripsi ini tidak lepas dari bantuan, bimbingan, motivasi, dan saran yang diberikan dari semua pihak. Oleh karena itu, pada kesempatan ini penulis ingin mengucapkan terima kasih kepada :

1. Allah ﷻ yang senantiasa memberikan kemudahan dan kelancaran kepada penulis serta Rasulullah Muhammad ﷺ yang telah menjadi suri tauladan selama penelitian berlangsung;
2. Kedua orang tua yang sangat penulis sayangi, Papa Oktianto dan Ibu Nurqamari Salim yang senantiasa mengajarkan kebaikan, memberikan perhatian, semangat, materi, kasih sayang, do'a terbaik, dukungan, serta kepercayaan atas setiap keputusan yang diambil oleh penulis hingga detik ini;
3. Kepada Om Teddy Hidayat yang selalu memberikan perhatian, pengertian, dukungan, dan semangat dalam setiap langkah dalam perkuliahan dan dalam penyusunan skripsi ini hingga saat ini.
4. Kepada almarhum tante Destriana yang selalu memberikan perhatian, membantu dari saya pertama saya sekolah hingga saya di titik ini, mendukung dalam pilihan yang saya buat, dan yang menjadi salah satu sosok yang mendidik saya untuk tetap tegar dalam menghadapi situasi yang sulit.

5. Bapak Dr. Eng. Helmy Fitriawan, S.T., M.Sc. selaku Dekan Fakultas Teknik Universitas Lampung;
6. Ibu Herlinawati, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung;
7. Ibu Yessi Mulyani, S.T., M.T. selaku Ketua Program Studi Teknik Informatika Universitas Lampung dan telah membantu proses kelancaran pengerjaan penelitian;
8. Mona Arif Muda, S.T., M.T. selaku pembimbing utama yang telah bersedia meluangkan waktu untuk memberikan bimbingan, saran, motivasi dan kesabaran dalam mengerjakan penelitian hingga selesai;
9. Bapak Mahendra Pratama, S.T., M.Eng. selaku Pembimbing Pendamping yang telah bersedia meluangkan waktu untuk memberikan bimbingan dan arahan selama penyusunan skripsi;
10. Ing. Hery Dian Septama, S.T., IPM. selaku Pembimbing Akademik yang telah memberikan bimbingan dan motivasi selama perkuliahan disetiap semester, dan Selaku Penguji yang telah memberikan banyak saran dan masukan sehingga penulis dapat menyelesaikan skripsi;
11. Seluruh Dosen Teknik Elektro dan Informatika atas bimbingan dan kesabarannya dalam mendidik penulis;
12. Mbak Rika selaku Admin Program Studi Teknik Informatika yang telah banyak membantu penulis dalam urusan administrasi selama perkuliahan dan penelitian;
13. Hesti Yuana Mahardika selaku pasangan hidupku tersayang yang selalu hadir dan menyemangati saya untuk selalu mengerjakan skripsi, lalu berkontribusi, memberikan masukan, dan membantu penulis selama proses pelaksanaan penelitian ini berlangsung;
14. Keluarga besar Teknik Elektro Angkatan 2020 yang telah menjadi teman seperjuangan sejak mahasiswa baru. Terima kasih atas bantuan, motivasi, kebersamaan, dan kekeluargaan yang telah diberikan;
15. Semua pihak yang telah membantu dan mendukung penulis dalam menyelesaikan penyusunan skripsi ini yang tidak dapat disebutkan satu persatu;

Penulis berharap agar laporan ini dapat menjadi referensi bagi pengembangan keilmuan di bidang Teknik Informatika. Oleh karena itu, semoga penelitian ini bermanfaat bagi yang membacanya

Bandar Lampung, 19 Desember
2024

Penulis,

Rahmat Ramadhan
NPM. 2015061037

DAFTAR ISI

DAFTAR ISI	I
DAFTAR GAMBAR	III
DAFTAR TABEL	V
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	3
1.5 Batasan Masalah	4
1.6 Sistematika Penulisan	4
II. TINJAUAN PUSTAKA	6
2.1 <i>Mangrove</i>	6
2.2 <i>Rapid Application Development</i>	7
2.3 <i>Sistem Informasi</i>	9
2.4 <i>Website</i>	9
2.5 <i>Unified Modeling Language (UML)</i>	10
2.6 <i>Bootstrap</i>	11
2.7 <i>Pengujian Black-box Testing</i>	12
2.8 <i>Database</i>	13
2.9 <i>Figma</i>	13
2.10 <i>DeepForest</i>	14

2.11	<i>REST API</i>	15
2.12	<i>PostgreSQL</i>	15
2.13	<i>Postman</i>	15
2.14	<i>Quantum Geographic Information System</i>	16
2.15	<i>User Experience Questionnaire</i>	16
2.16	Penelitian Terkait.....	18
III.	METODOLOGI PENELITIAN	21
3.1	Waktu dan Tempat Penelitian.....	21
3.2	Alat dan Bahan Penelitian	24
3.3	Tahapan Penelitian.....	25
3.3.1.	Metode Pengumpulan Data.....	26
3.3.2.	Metode Pengembangan Sistem	27
IV.	HASIL DAN PEMBAHASAN	57
4.1	Hasil dan Pembahasan.....	57
4.1.1	Iterasi Pertama.....	58
4.1.2	Iterasi Kedua	70
4.1.3	<i>User Experience Questionnaire (UEQ)</i>	87
4.1.4	Analisis.....	92
V.	KESIMPULAN DAN SARAN	96
5.1	Kesimpulan.....	96
5.2	Saran.....	97
	DAFTAR PUSTAKA	98
	LAMPIRAN	101

DAFTAR GAMBAR

Gambar 2. 1 Ilustrasi Rapid Aplication Development].....	7
Gambar 2. 2 <i>Bootstrap</i>	11
Gambar 2. 3 <i>Database (Firebase)</i>	13
Gambar 2. 4 <i>Figma</i>	13
Gambar 2. 5 Item Pengujian UEQ	18
Gambar 2. 6 Ilustrasi <i>Rapid Aplication Development</i>	27
Gambar 3. 1 <i>Capstone</i> Project Diagram	23
Gambar 3. 2 <i>Use Case</i> Diagram.....	36
Gambar 3. 3 <i>Activity</i> Diagram <i>Login</i>	38
Gambar 3. 4 <i>Activity</i> Diagram Melihat Data.....	39
Gambar 3. 5 <i>Activity</i> Diagram Homepage Admin	40
Gambar 3. 6 Rancangan Tampilan <i>Dashboard</i>	41
Gambar 3. 7 Rancangan Tampilan Pencarian Data.....	42
Gambar 3. 8 Rancangan Visualisasi Data Hutan Bakau	43
Gambar 3. 9 Rancangan Tampilan Detail Visualisasi Data	44
Gambar 3. 10 Rancangan Halaman <i>Login</i>	45
Gambar 3. 11 Instalasi <i>NodeJs</i>	46
Gambar 3. 12 <i>Node Version</i>	47
Gambar 3. 13 Instalasi <i>Webpack</i>	47
Gambar 3. 14 Konfigurasi <i>Webpack</i>	48
Gambar 3. 15 Menjalankan <i>Webpack</i>	48
Gambar 3. 16 Skrip <i>Webpack</i>	49
Gambar 3. 17 Struktur <i>Folder</i>	50
Gambar 3. 18 <i>Endpoint API</i>	52
Gambar 3. 19 <i>Endpoint API</i>	53

Gambar 4. 1 Halaman Beranda	58
Gambar 4. 2 Halaman Data Bakau.....	59
Gambar 4. 3 Sequence Diagram Halaman Data Bakau	60
Gambar 4. 4 Halaman Data <i>Map</i> Admin.....	62
Gambar 4. 5 Fungsi Memuat Data	63
Gambar 4. 6 Halaman <i>Upload Map</i> Admin Iterasi dua	65
Gambar 4. 7 Diagram <i>Overlay Map</i>	65
Gambar 4. 8 <i>Sequence</i> Diagram <i>Upload Map</i>	67
Gambar 4. 9 Halaman Dashboard Admin	70
Gambar 4. 10 Halaman Data Bakau Iterasi Dua	71
Gambar 4. 11 Halaman Data Bakau Iterasi Dua	72
Gambar 4. 12 Diagram <i>Map</i> Iterasi dua.....	73
Gambar 4. 13 Halaman <i>Map Mangrove</i> Admin Iterasi kedua	75
Gambar 4. 14 Diagram <i>Map</i> Admin Iterasi dua.....	76
Gambar 4. 15 Halaman Layanan.....	78
Gambar 4. 16 Halaman Artikel	79
Gambar 4. 17 Halaman <i>Login</i>	80
Gambar 4. 18 Halaman Akun.....	80
Gambar 4. 19 Halaman Artikel Admin	81
Gambar 4. 20 Halaman Layanan Admin.....	81
Gambar 4. 21 Data Hasil Kuesioner <i>User Experience Questionnaire (UEQ)</i>	88
Gambar 4. 22 Data Perubahan Nilai <i>User Experience Questionnaire (UEQ)</i>	89
Gambar 4. 23 Nilai Rata-Rata <i>User Experience Questionnaire (UEQ)</i>	90
Gambar 4. 24 Grafik Benchmark data hasil <i>User Experience Questionnaire (UEQ)</i>	91
Gambar 4. 25 Koefisien <i>Reliabilitas Alpha Cronbach</i>	91
Gambar 4. 26 Rata-rata dari Hasil Pengukuran <i>User Experience Questionnaire (UEQ)</i> [11]	92
Gambar 4. 27 Sampel Zona Bakau Petangoran	94
Gambar 4. 28 Hasil Perhitungan Setelah Penelitian	95

DAFTAR TABEL

Tabel 2. 1 Benchmark Interval Skala UEQ	17
Tabel 3. 1 Jadwal Penelitian	21
Tabel 3. 2 Alat dan Bahan	24
Tabel 3. 3 Parameter Kebutuhan	28
Tabel 3. 4 <i>User Stories</i>	30
Tabel 3. 5 Tabel Klasifikasi <i>MoSCoW</i>	32
Tabel 3. 6 Kebutuhan Fungsional	34
Tabel 3. 7 Kebutuhan Non-Fungsional	35
Tabel 3. 8 <i>Endpoint API</i>	54
Tabel 4. 1 <i>Testing Black Box</i> Iterasi Pertama	68
Tabel 4. 2 Pengujian Menampilkan Daftar Data Bakau	68
Tabel 4. 3 Pengujian Mengelola <i>Map Mangrove</i>	69
Tabel 4. 4 Pengujian <i>Upload Map Overlay</i>	69
Tabel 4. 5 Pengujian <i>Black Box Testing</i> Kedua	82
Tabel 4. 6 Melihat <i>dashboard</i> admin	83
Tabel 4. 7 Pengujian Menampilkan Daftar Data Bakau	83
Tabel 4. 8 Pengujian Mengelola <i>Map Mangrove</i>	84
Tabel 4. 9 Pengujian Login Akun Admin	84
Tabel 4. 10 Pengujian Melihat Halaman Layanan	85
Tabel 4. 11 Pengujian Melihat Halaman Artikel	85
Tabel 4. 12 Pengujian Melihat Akun Admin	86
Tabel 4. 13 Pengujian Mengelola Artikel	86
Tabel 4. 14 Pengujian Mengelola Layanan	87
Tabel 4. 15 Pengujian <i>Logout</i> Admin	87

I. PENDAHULUAN

1.1 Latar Belakang

Hutan *mangrove* memainkan peran penting dalam upaya mitigasi penanggulangan bencana tsunami. Selain itu pohon bakau memiliki kemampuan pemecah ombak alami, karena bakau mampu mengurangi ketinggian dan energi hempasan ombak yang berdampak meminimalkan kerusakan yang ditimbulkan. Selain itu hutan *Mangrove* Petengoran yang terletak di Provinsi Lampung merupakan kawasan ekowisata yang menawarkan keindahan alam serta keanekaragaman hayati yang luar biasa. Namun, keberadaan hutan *mangrove* ini memiliki nilai penting yang tidak hanya terbatas pada estetika dan keanekaragaman hayati. Keberadaan hutan *Mangrove* memiliki peran dalam menyediakan sumber daya alam seperti kayu, hingga hewan seperti ikan dan terumbu karang yang dapat mendukung kehidupan masyarakat sekitar [1].

Di sisi lain, tidak hanya berperan dalam mitigasi bencana ombak laut ataupun tsunami. Bakau juga berperan penting dalam penyerapan karbon dioksida (CO_2) dari atmosfer, sehingga berkontribusi dalam mengurangi efek gas rumah kaca yang memicu pemanasan global. Salah satu cara yang efektif untuk mengurangi dampak kerusakan ketika terjadi bencana alam dan pemanasan global adalah memaksimalkan potensi ini, diperlukan upaya konservasi dan pemantauan yang terintegrasi.

Sayangnya, proses pendataan yang masih dilakukan secara manual, seperti pengambilan gambar dan perhitungan pohon *mangrove* secara langsung, menghadapi sejumlah kendala. Kendala tersebut meliputi rendahnya efisiensi waktu, kebutuhan tenaga kerja yang besar, risiko ketidakakuratan data, terbatasnya jangkauan area, serta sulitnya pemantauan secara berkala. Oleh karena itu, penting

untuk merancang sistem informasi berbasis *website* yang mampu mendukung pendataan *mangrove* secara akurat dan terintegrasi. Saat ini, URO telah menjalin kerja sama dengan pengelola Hutan *Mangrove* Petengoran dalam melakukan pendataan.

Kondisi ini menunjukkan perlunya inovasi dalam metode pendataan hutan *mangrove* untuk meningkatkan efisiensi dan akurasi. Pendekatan berbasis teknologi sangat dibutuhkan seperti penggunaan *drone*, pemetaan digital, dan sistem otomatisasi lainnya sehingga dapat menjadi solusi untuk mengatasi tantangan ini. Dengan demikian, proses pendataan *mangrove* di Petangoran dapat mendukung pengelolaan yang lebih baik dan berkelanjutan. Sehingga tujuan akhir dapat membantu pihak konservasi bakau atau pihak luar untuk melihat data bakau petangoran.

Berdasarkan permasalahan tersebut, penelitian ini akan membahas perancangan visualisasi data *mangrove* menggunakan metode *Rapid Application Development* (RAD). Metode ini menekankan pengembangan aplikasi secara cepat melalui proses iteratif dan umpan balik berkelanjutan, sehingga memungkinkan pengembangan *website* dalam waktu singkat namun tetap memenuhi kebutuhan pengguna.

Dengan demikian, skripsi ini akan berfokus pada perancangan *website* sistem informasi Hutan *Mangrove* Petengoran menggunakan metode *Rapid Application Development*. Tujuannya adalah untuk mendukung upaya konservasi dan pemantauan hutan *mangrove* secara berkelanjutan, serta menyediakan data yang akurat dan mudah diakses bagi berbagai pemangku kepentingan.

1.2 Rumusan Masalah

Adapun rumusan masalah dari penelitian ini adalah sebagai berikut :

1. Bagaimana perancangan sistem informasi pemantauan pohon bakau yang terkomputerisasi di Hutan *Mangrove* Petengoran Dinas Pariwisata Kabupaten Pesawaran ?
2. Bagaimana implementasi sistem informasi pemantauan pohon bakau dapat meningkatkan efisiensi Pelaku konservasi dalam pendataan tanaman bakau di hutan *mangrove* Petangoran Dinas Pariwisata Kabupaten Pesawaran.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Menyatukan pendataan dan pelaporan data hutan bakau dalam satu *platform* terintegrasi.
2. Membantu memudahkan pekerjaan pegawai dan URO Unila dalam menjalankan proses pendataan data pohon bakau di wilayah *mangrove* petangoran.
3. Membangun suatu sistem manajemen informasi menyajikan pendataan data hutan bakau berbasis *website* pada Hutan *Mangrove* Petengoran Dinas Pariwisata Kabupaten Pesawaran.
4. Mempermudah pengguna dan pihak observasi dalam melakukan melihat data tanaman bakau di hutan *Mangrove* Petangoran.

1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut :

1. Penelitian ini diharapkan dapat memberikan kontribusi signifikan untuk meningkatkan kinerja pegawai, sehingga proses pencatatan data bakau dapat dilakukan dengan lebih efektif dan efisien tanpa memerlukan pencatatan manual.
2. Ketersediaan literatur terkait perancangan sistem informasi pendataan pohon bakau dengan pendekatan berbasis *website*.

3. Memudahkan Masyarakat dalam melihat informasi mengenai besaran Hutan *Mangrove* Petengoran secara lebih detail.

1.5 Batasan Masalah

Penelitian ini memiliki batasan masalah yaitu :

1. Pada pengembangan *website* hanya mengembangkan perangkat lunak visualisasi data pohon bakau di wilayah Hutan *Mangrove* Petangoran berbasis *website*.
2. Sistem belum mengimplementasikan Teknik *Enkripsi*.
3. Sistem ini membahas mengenai visualisasi data bakau di wilayah Hutan *Mangrove* Petengoran Dinas Pariwisata Kabupaten Pesawaran dalam bentuk *map* dan tabel.
4. Sistem saat ini belum mendukung penambahan fitur untuk pengelolaan ID kanopi.
5. Sistem belum mengintegrasikan proses pengolahan data mentah untuk digunakan oleh *modul machine learning*.

1.6 Sistematika Penulisan

Dalam pembahasan laporan penelitian ini, sistematika penulisan yang digunakan adalah sebagai berikut :

BAB I PENDAHULUAN

Pada Bab Pendahuluan ini berisi tentang latar belakang penelitian pengembangan aplikasi, rumusan masalah penelitian, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan penelitian Perancangan dan Implementasi Sistem Informasi Pendataan tanaman bakau Hutan *Mangrove* Petengoran Dinas Pariwisata Kabupaten Pesawaran.

BAB II TINJAUAN PUSTAKA

Pada Bab ini berisi tentang penelitian sebelumnya dan pengetahuan tentang *mangrove*, *rapid application development* (RAD), sistem informasi, *website*,

unified model language (UML), bootstrap, pengujian black box testing, database, figma, deepforest, restAPI, postgresSQL, postman, quantum geographic information system (QGIS), user experience Questionnaire dan penelitian terkait Perancangan dan Implementasi Sistem Informasi Pendataan tanaman bakau Hutan *Mangrove* Petengoran Dinas Pariwisata Kabupaten Pesawaran.

BAB III METODE PENULISAN

Pada Bab ini berisi tentang mengenai metode dan perancangan dari sistem yang digunakan dalam penelitian Perancangan dan Implementasi Sistem Informasi Pendataan tanaman bakau Hutan *Mangrove* Petengoran Dinas Pariwisata Kabupaten Pesawaran.

BAB IV PEMBAHASAN

Pada bab pembahasan berisi tentang hasil penelitian berupa aplikasi untuk visualisasi hutan mangrove. Pada bab ini menjelaskan tentang hasil dan pengujian dari Perancangan dan Implementasi Sistem Informasi Pendataan tanaman bakau Hutan *Mangrove* Petengoran Dinas Pariwisata Kabupaten Pesawaran.

BAB V KESIMPULAN DAN SARAN

Pada Bab ini berisi tentang kesimpulan dan saran pengembang hasil penelitian lanjutan untuk hasil yang lebih baik lagi dari penelitian Perancangan dan Implementasi Sistem Informasi Pendataan tanaman bakau Hutan *Mangrove* Petengoran Dinas Pariwisata Kabupaten Pesawaran.

DAFTAR PUSTAKA

LAMPIRAN

II. TINJAUAN PUSTAKA

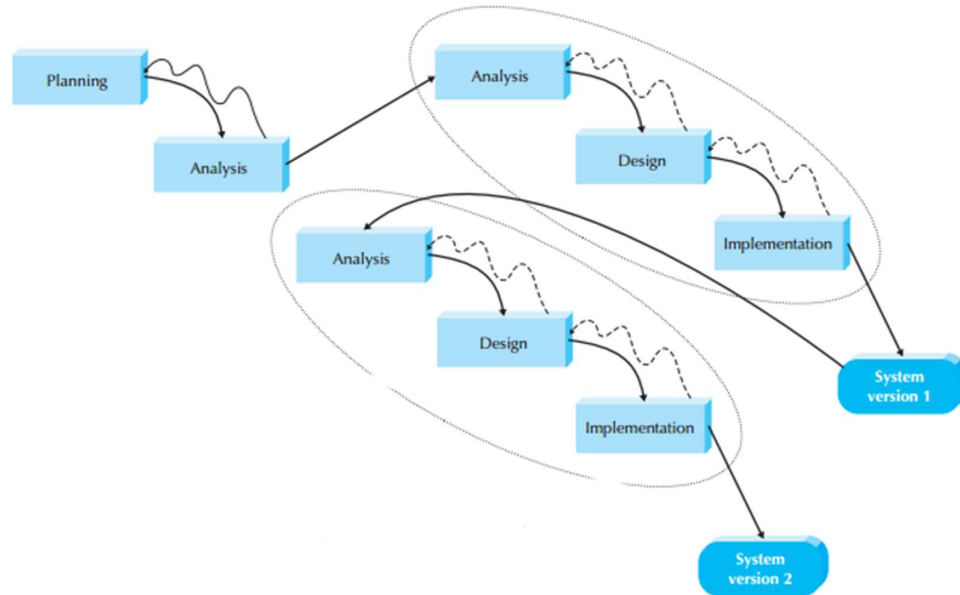
2.1 *Mangrove*

Mangrove, sebagai jenis tanaman yang menghuni habitat air asin atau laut, membawa beragam manfaat dan nilai ekonomis yang tak terbantahkan. Sebagai penjaga alam pesisir, *mangrove* memiliki adaptasi khusus yang memungkinkannya bertahan di lingkungan yang keras, seperti air payau, tanah berlumpur, dan perubahan pasang surut. Selain memberikan perlindungan fisik, bagian-bagian tanaman *mangrove* memiliki potensi untuk dijadikan bahan baku dalam industri kosmetik, farmasi, atau sebagai tambahan tekstil. Keberadaan hutan *mangrove* memainkan peran vital dalam menyediakan sumber daya alam yang mendukung kehidupan masyarakat sekitarnya, mulai dari kayu hingga jenis-jenis ikan, krustasea, dan terumbu karang [1]. Reproduksi *mangrove* yang unik, dengan biji yang cenderung lebih besar dan menggunakan metode vivipar, menambahkan keistimewaan pada tanaman ini. Dengan kemampuannya mengapung sebelum akhirnya tenggelam, biji *mangrove* melibatkan periode tertentu yang bervariasi tergantung pada jenisnya [1]. Keseluruhan, keragaman manfaat dan karakteristik *mangrove* menciptakan sebuah ekosistem yang sangat bernilai, mendorong perlindungan, dan pengelolaan berkelanjutan untuk menjaga kehidupan di ekosistem pesisir.

Selain manfaat tersebut, kanopi pohon *mangrove* memiliki peran penting dalam menjaga keberlangsungan ekosistem *mangrove*. Kanopi pohon tidak hanya berfungsi sebagai habitat bagi fauna, tetapi mempengaruhi tingkat pencahayaan yang mencapai permukaan tanah, yang pada akhirnya memengaruhi keberhasilan regenerasi spesies *mangrove*. Penilaian terhadap kanopi pohon *mangrove* dapat digunakan sebagai indikator kesehatan ekosistem *mangrove*, karena kepadatan dan cakupan kanopi menunjukkan tingginya keanekaragaman hayati serta fungsi ekologis *mangrove* dalam lingkungan pesisir [1]. Pengamatan pada bagian kanopi

menjadi salah satu metode untuk menghitung jumlah pohon dan tingkat kesehatan vegetasi pada suatu area *mangrove* tertentu, di mana kanopi yang lebih luas biasanya berhubungan dengan pohon yang lebih dewasa dan sehat.

2.2 *Rapid Application Development*



Gambar 2. 1 Ilustrasi *Rapid Application Development* [2]

Dalam proses pengembangan *website*, metode yang digunakan adalah *Rapid Application Development* (RAD). RAD adalah sebuah metode pengembangan perangkat lunak yang berfokus pada pengembangan aplikasi secara cepat, dengan melalui pengulangan dan feedback yang berulang-ulang [3]. Pendekatan ini memungkinkan pengembang untuk dengan cepat merancang, membangun, dan menguji solusi perangkat lunak baru.

RAD memanfaatkan teknik pemodelan dan *prototyping* yang cepat untuk mempercepat proses pengembangan. Dalam setiap iterasi, tim pengembangan merancang, membangun, menguji, dan merevisi produk perangkat lunak sampai mencapai hasil yang diinginkan. Proses ini memungkinkan *developer* untuk

mengidentifikasi dan memperbaiki masalah sejak dini dalam siklus pengembangan, sehingga mengurangi risiko kegagalan proyek.

Meskipun *RAD* memungkinkan pengembangan yang lebih cepat, tetapi tetap berlandaskan pada prinsip-prinsip ilmiah dan metodologi yang solid. Oleh karena itu, penggunaan *RAD* tidak mengesampingkan pentingnya penelitian dan dokumentasi. Sebaliknya, *RAD* mendorong penggunaan jurnal dan dokumentasi lainnya untuk melacak kemajuan proyek, memahami tantangan yang dihadapi, dan mencatat solusi yang telah diimplementasikan.

Dengan demikian, penggunaan metode *Rapid Application Development* dalam pengembangan *website* memungkinkan *developer* untuk merespons dengan cepat terhadap perubahan kebutuhan, sambil tetap mempertahankan kualitas dan integritas produk akhir. Selain itu, pendekatan ini memungkinkan *developer* untuk belajar dan beradaptasi sepanjang proses pengembangan, sehingga meningkatkan efisiensi dan efektivitas mereka.

Metode pengembangan perangkat lunak *Rapid Application Development* (RAD) melibatkan empat fase esensial, yang dijabarkan sebagai berikut:

1. Perencanaan (*Planning*): Tahap awal ini melibatkan penyelidikan dan studi kelayakan. Keputusan untuk melanjutkan pengembangan sistem bergantung pada penilaian teknis, ekonomis, dan operasional yang disetujui.
2. Analisis (*Analysis*): Setelah perencanaan, fase analisis dilakukan dalam rangka pengembangan rekayasa perangkat lunak. Proses ini dimulai dengan pemahaman mendalam mengenai kebutuhan pengguna terhadap sistem yang akan dikembangkan. Metode seperti diskusi, wawancara, observasi, studi literatur, dan sejenisnya digunakan untuk mengumpulkan informasi dan data yang dibutuhkan dalam pengembangan sistem, hingga menetapkan implementasi selanjutnya.
3. Desain (*Design*): Pada tahap desain, pengembang terlibat dalam diskusi untuk menentukan kebutuhan pengguna dan melakukan perbaikan pada

desain apabila diperlukan. Pembuatan desain sistem berfungsi sebagai panduan dalam pengembangan, dan partisipasi serta masukan pengguna dianggap sangat krusial untuk mencapai tujuan pengembangan sistem.

4. Implementasi (*Implementation*): Tahap implementasi melibatkan transformasi desain sistem menjadi kode program menggunakan bahasa pemrograman. Saran dan masukan pengguna memegang peranan utama dalam mencapai tujuan pengembangan, dan apabila hasilnya belum memenuhi kebutuhan pengguna, pengembang akan kembali ke tahap perancangan pengguna. Jika kode program sudah sesuai dengan kebutuhan, tahap pengujian dilakukan menggunakan metode black box testing untuk memastikan kesalahan sistem minimal dan kesesuaiannya dengan kebutuhan pengguna.

2.3 Sistem Informasi

Sistem informasi merupakan suatu sistem yang mengintegrasikan interaksi manusia, data, informasi, dan teknologi untuk beroperasi dalam tugas pengumpulan data, penyimpanan data, serta penghasilan informasi yang berbentuk keputusan dan tindakan. Penerapan sistem informasi dapat dilakukan pada berbagai sektor, termasuk dalam lingkup bisnis, pemerintahan, pendidikan, kesehatan, dan sektor lainnya. Pemanfaatan sistem informasi memiliki potensi untuk membantu organisasi meningkatkan efektivitas dan efisiensi dengan mengoptimalkan penggunaan sumber daya serta membuat keputusan yang lebih optimal berdasarkan ketersediaan informasi.

2.4 Website

Website adalah suatu kumpulan halaman yang saling terhubung di internet dan dapat diakses melalui peramban *web*. Setiap halaman memiliki potensi untuk menyajikan teks, gambar, video, atau elemen multimedia lainnya, serta dilengkapi dengan tautan atau *hyperlink* yang mengarahkan pengguna ke halaman-halaman lain di dalam situs *web* tersebut atau bahkan ke situs *web* lainnya. Tiap situs *web* memiliki alamat khusus yang dikenal sebagai URL (*Uniform Resource Locator*)

untuk memungkinkan akses oleh pengguna melalui peramban *web* seperti Chrome, Firefox, Safari, atau Edge.

Desain dan fungsionalitas situs *web* bervariasi tergantung pada tujuan dan kebutuhan masing-masing. Beberapa situs *web* sederhana mungkin hanya berisi beberapa halaman statis dengan teks dan gambar, sementara situs *web* lain dapat mencakup fitur interaktif, pangkalan data, atau aplikasi yang lebih kompleks.

Jenis-jenis situs *web* secara umum dapat dikelompokkan sebagai berikut:

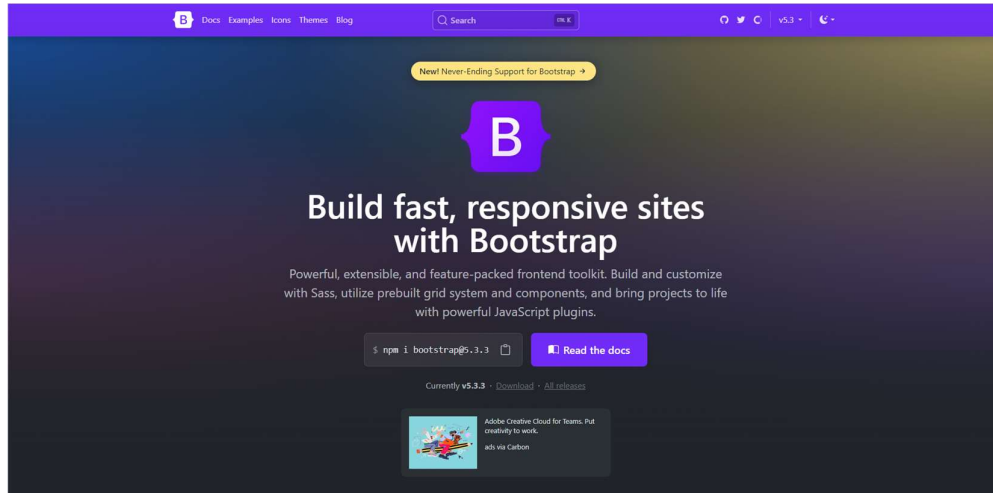
1. Situs *web* statis : Jenis situs *web* ini memiliki konten yang tetap dan tidak berubah secara dinamis berdasarkan interaksi pengguna atau data yang disimpan. Artinya, konten pada situs *web* statis akan selalu sama setiap kali situs tersebut diakses. Contoh umum meliputi halaman informasi sederhana, portofolio pribadi, dan situs *web* perusahaan yang tidak memerlukan pembaruan konten secara teratur.
2. Situs *web* dinamis : Situs *web* dinamis mengubah kontennya berdasarkan interaksi pengguna atau data yang disimpan dalam sebuah basis data. Ini berarti konten pada situs *web* dinamis secara otomatis diperbarui atau berubah secara real-time sesuai dengan permintaan atau tindakan pengguna.
3. Situs *web* interaktif : Jenis situs *web* ini memungkinkan pengguna untuk berinteraksi secara aktif dengan kontennya, seperti mengisi formulir, mengirim komentar, memilih opsi, dan bermain game. Contoh situs *web* interaktif meliputi platform media sosial seperti *Facebook* atau *Twitter*, toko online dengan keranjang belanja interaktif, situs *web* permainan, dan aplikasi *web* seperti *Google Docs* yang memungkinkan kolaborasi *real-time* antar pengguna.

2.5 *Unified Modeling Language* (UML)

Unified Modeling Language (UML) adalah standarisasi bahasa pemodelan untuk pengembangan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek [4]. *Unified Modeling Language* (UML) memegang peran penting dalam proses pengembangan perangkat lunak. Sebagai

bahasa pemodelan visual, *UML* mendukung desain dan pemodelan sistem perangkat lunak yang kompleks.

2.6 *Bootstrap*



Gambar 2. 2 *Bootstrap*

Bootstrap adalah kerangka kerja *CSS* yang sangat terkenal dan difavoritkan dalam pengembangan situs *web*. Dikembangkan oleh *Twitter*, *Bootstrap* telah menjadi pilihan utama bagi banyak pengembang *web* karena kemampuannya untuk menyederhanakan dan mempercepat proses pembuatan situs *web* yang responsif dan berorientasi *mobile* [5].

Salah satu keunggulan utama dari *Bootstrap* adalah ketersediaan berbagai komponen dan elemen yang memungkinkan pengembang untuk dengan mudah membangun tata letak halaman, membuat navigasi yang intuitif, menampilkan konten dengan gaya yang menarik, dan banyak lagi. Dengan demikian, *Bootstrap* memberikan kerangka kerja yang kokoh dan efisien bagi mereka yang ingin menghasilkan situs *web* dengan kualitas tinggi dalam waktu singkat.

Selain itu, *Bootstrap* mengintegrasikan konsep desain responsif dengan sangat baik. Artinya, situs *web* yang dikembangkan menggunakan *Bootstrap* dapat secara otomatis menyesuaikan tampilannya untuk berbagai ukuran layar dan perangkat,

meningkatkan pengalaman pengguna tanpa harus membuat versi situs yang berbeda untuk setiap perangkat atau resolusi layar.

Kemampuan *Bootstrap* untuk berkomunikasi secara harmonis dengan semua browser utama menjadi keunggulan besar. Dengan dukungan untuk fitur HTML5 dan CSS3 terbaru, *Bootstrap* memastikan bahwa situs *web* yang dikembangkan menggunakan kerangka kerja ini tetap relevan dan dapat diakses dengan baik di lingkungan online yang terus berkembang [5].

2.7 Pengujian *Black-box Testing*

Pengujian sistem merupakan langkah akhir dalam proses pengembangan perangkat lunak yang bertujuan untuk menilai kinerja dan fungsinya serta mendeteksi kesalahan yang dapat segera diperbaiki. Pengujian ini mencakup beberapa metode, di antaranya functional technique dan structural technique.

Salah satu metode yang digunakan dalam functional technique adalah Black Box Testing, yaitu metode pengujian perangkat lunak yang tidak memperhatikan struktur internal atau logika pemrograman sistem. Pada metode ini, pengujian berfokus pada input dan output yang dihasilkan oleh sistem sesuai dengan spesifikasi yang telah ditetapkan oleh pengembang, tanpa memerlukan pengetahuan penguji mengenai bahasa pemrograman yang digunakan [6] .

Dengan kata lain, Black Box Testing didasarkan pada persyaratan atau spesifikasi desain dari entitas perangkat lunak yang diuji [6]. Contoh hasil yang diharapkan, yang sering disebut sebagai test oracle, meliputi spesifikasi persyaratan/desain, nilai yang dihitung secara manual, dan hasil simulasi. Secara umum, pengujian fungsional menitikberatkan pada perilaku eksternal dari entitas perangkat lunak yang diuji.

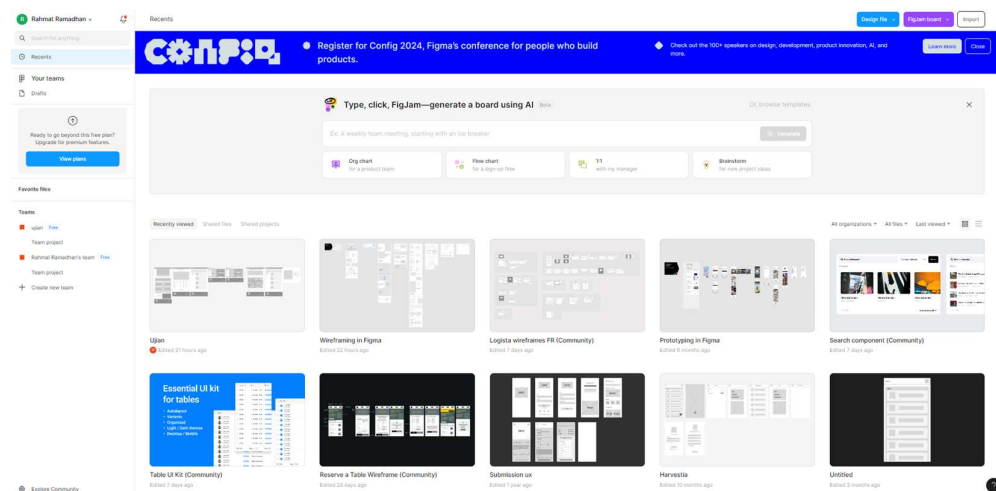
2.8 Database



Gambar 2. 3 Database (Firebase)

Database adalah sekumpulan data yang saling terhubung sedemikian rupa sehingga dapat disimpan, dimanipulasi, dan dipanggil oleh pengguna. Dalam sistem informasi perhitungan pohon bakau, digunakan *firestore database NoSQL* sebagai media penyimpanan data secara *real-time*. *Database NoSQL* sendiri memiliki empat karakteristik utama, yaitu kunci-nilai, berbasis dokumen, berbasis kolom, dan berbasis grafik. *Firebase* sendiri memiliki kemampuan untuk mengoptimalkan akses data dalam hitungan mikrodetik daripada menggunakan *web-service*.

2.9 Figma



Gambar 2. 4 Figma

Figma adalah *platform* desain kolaboratif yang sangat efektif untuk pembuatan *wireframe*, langkah awal dalam desain UI/UX. Memungkinkan tim bekerja bersama

dalam satu dokumen secara *real-time*, *Figma* memudahkan pembuatan *wireframe* yang konsisten dan interaktif. Dengan alat vektor *editing* dan fitur *prototyping*, *desainer* dapat merinci struktur dan alur pengguna, sementara komponen dan *Libraries* mempercepat proses. *Figma* memudahkan pengorganisasian konten dan *export wireframe* untuk berbagi dengan *stakeholder*, mendukung kolaborasi yang efisien dalam tahap awal pengembangan.

2.10 *DeepForest*

DeepForest adalah paket *Python* yang dirancang untuk mendeteksi pohon individu dalam citra *RGB* resolusi tinggi melalui metode *deep learning*. Metode ini memungkinkan *delineasi* mahkota pohon, yang merupakan langkah penting dalam analisis hutan. *DeepForest* dilatih menggunakan lebih dari 30 juta mahkota pohon yang dihasilkan secara algoritma dari 22 hutan, serta 10.000 mahkota yang dilabeli secara manual dari enam hutan [7]. Proses pelatihan dimulai dengan menghasilkan data pelatihan lokal dengan melabeli pohon dalam citra menggunakan kotak pembatas. *DeepForest* juga mendukung *transfer learning*, yang memungkinkan pengguna untuk melakukan *fine-tuning* model yang telah ada dengan data pelatihan yang lebih sedikit, hanya memerlukan sekitar 1.000 pohon yang dilabeli secara manual untuk mendapatkan hasil yang signifikan.

Setelah model dilatih, pengguna dapat mengevaluasi kinerjanya menggunakan data uji terpisah, dengan metrik *mean average precision* (mAP) digunakan untuk mengukur kinerja prediksi. *DeepForest* dapat membuat prediksi untuk area seluas 1 km² dalam waktu sekitar 7 menit pada satu *CPU*, dan memiliki dukungan untuk menjalankan di sumber daya *GPU* untuk meningkatkan kecepatan prediksi [7]. Selain itu, *DeepForest* memudahkan pengguna untuk mengeksport prediksi untuk analisis lebih lanjut, menjadikannya alat yang kuat untuk analisis penginderaan jauh yang lebih luas, termasuk estimasi biomassa dan klasifikasi spesies.

2.11 *REST API*

REST API (Representational State Transfer Application Programming Interface) adalah gaya arsitektur perangkat lunak yang memfasilitasi pertukaran data antar aplikasi melalui Internet dan bertindak sebagai perantara, menerima permintaan dan mengirimkan tanggapan menggunakan protokol *HTTP* yang umum digunakan. *REST API* didasarkan pada prinsip *Representational State Transfer (REST)* dan menggunakan metode *HTTP* seperti *GET*, *POST*, *PUT*, dan *DELETE* untuk berinteraksi dengan data atau sumber daya di server [8].

2.12 *PostgreSQL*

PostgreSQL adalah salah satu database *open source* paling populer, fitur utamanya adalah fleksibilitas dan kemampuan beradaptasi di seluruh industri. Keuntungan utama *PostgreSQL* adalah ekstensibilitasnya, yang memungkinkan pengembang menambahkan fungsionalitas *database* baru tanpa mengorbankan proyek aslinya. Banyak perusahaan memanfaatkan fungsionalitas dan ekosistem *PostgreSQL* yang kaya untuk membangun aplikasi yang kompleks dan sukses, yang pada gilirannya menciptakan permintaan yang signifikan terhadap *PostgreSQL* melebihi kapasitas server tunggal. Ekstensi Citus merupakan pengembangan penting yang mengubah *PostgreSQL* menjadi sistem manajemen basis data terdistribusi (DDBMS), memungkinkan skalabilitas yang lebih besar sambil menjaga kompatibilitas dengan fitur-fitur *PostgreSQL* terbaru dan alat pendukung dengan biaya pengembangan minimal [9].

2.13 *Postman*

Postman adalah perangkat lunak untuk pengujian, pengembangan, dan dokumentasi *API*. Hal memungkinkan tim untuk berkolaborasi secara *real-time* dalam satu lingkungan, sehingga memudahkan pembuatan permintaan *API* yang konsisten dan interaktif. Alat pengujian otomatis dan kemampuan simulasi memungkinkan pengembang untuk merinci struktur dan proses data, sementara koleksi dan variabel lingkungan mempercepat proses. Aplikasi ini juga memudahkan pengorganisasian permintaan dan ekspor koleksi untuk dibagikan

kepada pemangku kepentingan, sehingga memungkinkan kolaborasi yang efisien pada tahap awal pengembangan *API*.

2.14 *Quantum Geographic Information System*

Quantum Geographic Information System (QGIS) merupakan perangkat lunak *GIS open-source*, *QGIS* sendiri menyediakan antarmuka untuk banyak program perangkat lunak sistem informasi 37 geografis (SIG) yang relevan, seperti *SAGA*, *GDAL*, *GRASS*, atau *LasTools* [10]. *SIG* merupakan suatu sistem yang mampu menyimpan, mengumpulkan, mentransformasi, memanipulasi, memadukan, pengolahan dan analisis data secara simultan serta mampu menyajikan data tersebut dalam sebuah informasi mengenai fenomena geografis suatu wilayah [10]. *QGIS* adalah alat yang penting dalam *remotesensing*. Dengan menggunakan *QGIS*, pengguna dapat melakukan berbagai jenis analisis pada data citra satelit, mulai dari visualisasi sederhana hingga analisis spasial yang kompleks. Aplikasi ini dapat menyediakan data, melihat, mengubah, dan menganalisis data yang bersifat *geospasial*. Teknologi *remotesensing* yang dilengkapi *SIG* mampu menghasilkan data lebih akurat, karena mampu mengumpulkan, menyimpan, mengolah dan menganalisis, serta menyajikan data dan informasi dari suatu objek atau fenomena [10].

2.15 *User Experience Questionnaire*

User Experience Questionnaire (UEQ) adalah metode pengujian aplikasi yang berisikan serangkaian pertanyaan yang digunakan untuk mengumpulkan saran dan masukan dari pengguna tentang interaksi penggunaan aplikasi. Pengujian ini bermaksud untuk menilai kualitas pengalaman pengguna dan mengidentifikasi fitur yang perlu ditingkatkan. Berikut faktor penilaian *User Experience Questionnaire* (UEQ) adalah sebagai berikut [11]:

1. Daya Tarik : Apakah user menyukai atau tidak aplikasi?
2. Kejelasan : Apakah mudah untuk memahami aplikasi dan apakah mudah untuk dipelajari?
3. Efisiensi : Bisakah pengguna dapat menyelesaikan tugas dengan sistem yang

tidak membuang waktu, tenaga, dan biaya?

4. Ketepatan : Apakah user terarah terhadap sistem?
5. Simulasi : Apakah user menarik dan memotivasi untuk menggunakan sistem?
6. Kebaruan : Apakah aplikasi inovatif dan kreatif?

Hasil pengujian *User Experience Questionnaire* (UEQ) berupa nilai rata-rata yang akan dikategorikan kedalam bentuk diagram Benchmark. Sehingga untuk memahami hasil kategori penilaian dengan skala yang tercantum dalam Tabel 2.1.

Tabel 2. 1 *Benchmark Interval Skala UEQ* [11]

	<i>Att.</i>	<i>Eff.</i>	<i>Per.</i>	<i>Dep.</i>	<i>Sti.</i>	<i>Nov.</i>
<i>Excellent</i>	≥ 1.75	≥ 1.78	≥ 1.9	≥ 1.65	≥ 1.55	≥ 1.4
<i>Good</i>	≥ 1.52	≥ 1.47	≥ 1.56	≥ 1.48	≥ 1.31	≥ 1.05
	< 1.75	< 1.78	< 1.9	< 1.65	< 1.55	< 1.4
<i>Above average</i>	≥ 1.17	≥ 0.98	≥ 1.08	≥ 1.14	≥ 0.99	≥ 0.71
	< 1.52	< 1.47	< 1.56	< 1.48	< 1.31	< 1.05
<i>Below average</i>	≥ 0.7	≥ 0.54	≥ 0.64	≥ 0.78	≥ 0.5	≥ 0.3
	< 1.17	< 0.98	< 1.08	< 1.14	< 0.99	< 0.71
<i>Bad</i>	< 0.7	< 0.54	< 0.64	< 0.78	< 0.5	< 0.3

Berdasarkan 6 faktor penilaian UEQ terdapat 26 item dengan format *seven stage semantic differential* yang berisi pernyataan negatif dan positif pada kuesioner yang diacak bertujuan untuk melihat konsistensi jawaban responden, Adapun detail dan item dapat dilihat sebagai berikut:

	1	2	3	4	5	6	7		
menyusahkan	☐	☐	☐	☐	☐	☐	☐	menyenangkan	1
tak dapat dipahami	☐	☐	☐	☐	☐	☐	☐	dapat dipahami	2
kreatif	☐	☐	☐	☐	☐	☐	☐	monoton	3
mudah dipelajari	☐	☐	☐	☐	☐	☐	☐	sulit dipelajari	4
bermanfaat	☐	☐	☐	☐	☐	☐	☐	kurang bermanfaat	5
membosankan	☐	☐	☐	☐	☐	☐	☐	mengasyikkan	6
tidak menarik	☐	☐	☐	☐	☐	☐	☐	menarik	7
tak dapat diprediksi	☐	☐	☐	☐	☐	☐	☐	dapat diprediksi	8
cepat	☐	☐	☐	☐	☐	☐	☐	lambat	9
berdaya cipta	☐	☐	☐	☐	☐	☐	☐	konvensional	10
menghalangi	☐	☐	☐	☐	☐	☐	☐	mendukung	11
baik	☐	☐	☐	☐	☐	☐	☐	buruk	12
rumit	☐	☐	☐	☐	☐	☐	☐	sederhana	13
tidak disukai	☐	☐	☐	☐	☐	☐	☐	menggembirakan	14
lazim	☐	☐	☐	☐	☐	☐	☐	terdepan	15
tidak nyaman	☐	☐	☐	☐	☐	☐	☐	nyaman	16
aman	☐	☐	☐	☐	☐	☐	☐	tidak aman	17
memotivasi	☐	☐	☐	☐	☐	☐	☐	tidak memotivasi	18
memenuhi ekspektasi	☐	☐	☐	☐	☐	☐	☐	tidak memenuhi ekspektasi	19
tidak efisien	☐	☐	☐	☐	☐	☐	☐	efisien	20
jelas	☐	☐	☐	☐	☐	☐	☐	membingungkan	21
tidak praktis	☐	☐	☐	☐	☐	☐	☐	praktis	22
terorganisasi	☐	☐	☐	☐	☐	☐	☐	berantakan	23
atraktif	☐	☐	☐	☐	☐	☐	☐	tidak atraktif	24
ramah pengguna	☐	☐	☐	☐	☐	☐	☐	tidak ramah pengguna	25
konservatif	☐	☐	☐	☐	☐	☐	☐	inovatif	26

Gambar 2. 5 Item Pengujian UEQ

2.16 Penelitian Terkait

Untuk menjamin keabsahan penelitian ini, penulis akan menyajikan sejumlah penelitian sebelumnya yang dilakukan oleh peneliti lain. Berikut rangkuman singkat penelitian terdahulu terkait penelitian yang akan dijadikan referensi oleh penulis :

1. Rancang Bangun Aplikasi Data Pengolah Kelapa Sawit Pada PT.PERKEBUNAN NUSANTARA VII SUNGAI NIRU Berbasis *Website*
 Penelitian "Pengembangan Aplikasi Data Pengolahan Kelapa Sawit Berbasis *Web* dengan *Metode Rapid Application Development* (RAD) menggunakan PHP dan *MySQL*: Studi Kasus PT. Perkebunan Nusantara VII Sungai Niru" ditulis oleh Muhamad Ar Rafiq, Khana Wijaya, dan Rishi Suparianto pada tahun 2023. Penelitian ini menjelaskan tentang pembuatan aplikasi data pengolahan kelapa sawit berbasis *web* untuk PT. Perkebunan Nusantara VII Sungai Niru. Dalam

pengembangannya, aplikasi ini menggunakan *tool Visual Studio Code*. Hasil pengujian aplikasi dilakukan menggunakan metode *black box* [12].

2. Analisis Perhitungan Individu Pohon *Mangrove* Pada Citra *Unmanned Aerial Vehicle* (UAV) Menggunakan Metode *Local Maxima*

Penelitian Analisis Perhitungan Individu Pohon *Mangrove* Pada Citra *Unmanned Aerial Vehicle* (UAV) Menggunakan dengan Metode *Local Maxima* Studi kasus dilakukan di Pulau Rupat Utara, Kabupaten Bengkalis, Provinsi Riau. Penelitian ini bertujuan untuk mempermudah perhitungan jumlah pohon *mangrove* dengan efisiensi tinggi. Hasilnya diuji dengan evaluasi model menggunakan data sample dan visual perhitungan pohon *mangrove* dengan minimum ketinggian 5 meter dan 4 meter [13].

3. Sistem Informasi *Website* Pariwisata Hutan *Mangrove* di Kelurahan Belawan Sicanang Kecamatan Medan Belawan Sebagai Media Promosi

Penelitian ini membahas pengembangan sistem informasi *web* pariwisata hutan *mangrove* di Kelurahan Belawan Sicanang sebagai media promosi yang efektif, berbasis SDLC. Studi kasus dilakukan di Kelurahan Belawan Sicanang, Kecamatan Medan Belawan, oleh Randi Rian Putra pada tahun 2019. Penelitian ini menjelaskan tentang implementasi sistem meliputi halaman utama, profil, informasi publik, galeri, kontak, *login* admin, dan halaman admin. Hasilnya diuji dengan pengujian sistem menggunakan data sesungguhnya dalam jangka waktu tertentu oleh analis sistem bersama dengan user [14].

4. Penelitian ini membahas pembangunan *dashboard* lokasi rawan tanah longsor di Indonesia berbasis data bencana tanah longsor dari tahun 2011-2014, menggunakan metode studi literatur dan eksperimen. Studi kasus dilakukan di Indonesia oleh peneliti pada tahun 2018, menjelaskan pola kejadian bencana tanah longsor di Indonesia serta pembangunan *dashboard* menggunakan Tableau. Hasil penelitian diuji dengan analisis data empiris untuk mengidentifikasi daerah rawan tanah longsor dan merancang kesiapsiagaan mitigasi bencana [15].

5. Penelitian ini membahas pengembangan sistem informasi pemetaan objek wisata pesisir di Kota Manado, Kota Bitung, dan Kabupaten Minahasa Utara berbasis

Web GIS. Penelitian ini menggunakan *metode Rapid Application Development* (RAD) dan merupakan studi kasus yang dilakukan oleh Aditya Lapu Kalua, Samuel Pondaag, dan Fadhila Mohune pada tahun 2023. Penelitian ini menjelaskan tentang informasi detail objek wisata, termasuk lokasi, harga masuk, jam operasional, fasilitas, deskripsi, dan gambar objek wisata. Hasil dari penelitian ini diuji dengan skor 100% untuk menunjukkan bahwa aplikasi yang dikembangkan tidak mengalami error dalam fungsi-fungsinya [16].

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Waktu penelitian dilakukan mulai bulan Maret 2024 sampai dengan oktober 2024 yang berlokasi di Hutan *Mangrove* Petengoran Dinas Pariwisata Kabupaten Pesawaran yang bertempat di Jl. Raya Kedondong, Way Layap Kec. Gedong Tataan Kabupaten Pesawaran, Provinsi Lampung – Indonesia. Untuk lebih jelasnya berikut adalah tabel kegiatan selama penelitian berlangsung.

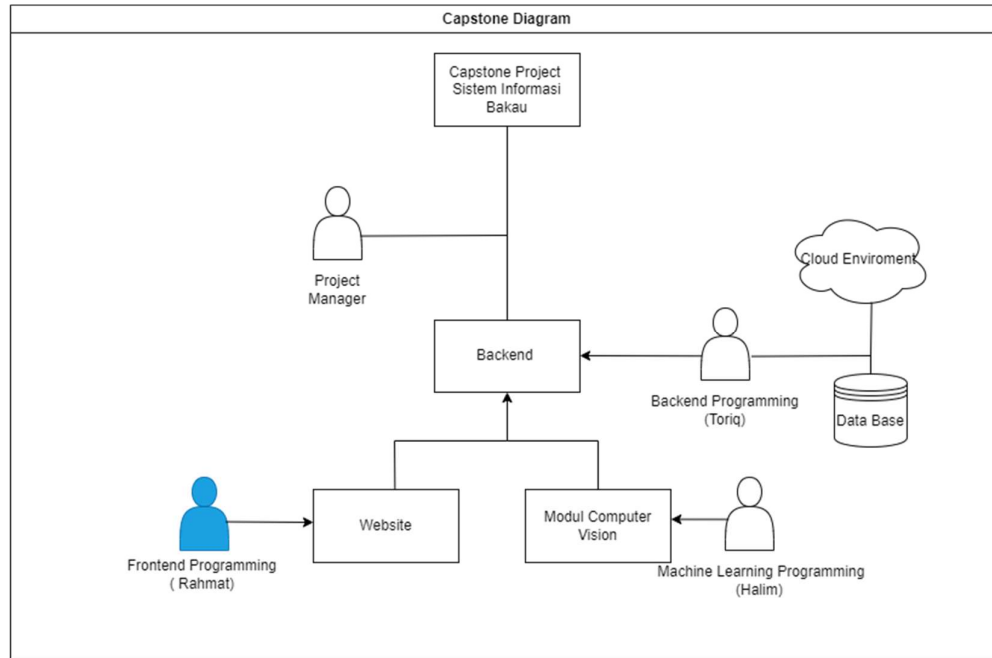
Tabel 3. 1 Jadwal Penelitian

No .	Kegiatan	Waktu Penelitian							
		Maret	April	Mei	Juni	Juli	Agustus	September	Oktober
1	Penelitian Awal								
	a. Studi Literatur								
	b. Wawancara								
	c. Analisis Kebutuhan Sistem								
2	<i>Stage-I System Development Iteration</i>								
	a. Analisis Kebutuhan Sistem								
	b. Desain Sistem								
	c. Pengembangan Sistem								
	d. Pengujian Sistem								

Tabel 3. 1 Jadwal Penelitian (lanjutan)

No .	Kegiatan	Waktu Penelitian							
		Maret	April	Mei	Juni	Juli	Agustus	September	Oktober
3	<i>Stage-2 System Development Iteration</i>								
	a. Analisis Kebutuhan Sistem								
	b. Desain Sistem								
	c. Pengembangan Sistem								
	d. Pengujian Sistem								
4	<i>Penulisan Akhir</i>								
	a. Pembuatan Laporan								

Diagram Capstone Project

Gambar 3. 1 *Capstone Project Diagram*

Capstone Project ini melibatkan empat aktor utama: *Project Manager*, *Backend Programming*, *Machine Learning Programming*, dan *Frontend Programming*. *Project Manager* bertanggung jawab dalam merencanakan, mengkoordinasikan, dan memastikan proyek berjalan sesuai jadwal dan mencapai tujuan yang ditetapkan.

Backend Programming memiliki tugas penting dalam mempersiapkan infrastruktur teknis proyek. Mereka membuat lingkungan *cloud* yang akan menjadi tempat untuk melakukan *deploy model computer vision*, database, dan *website*. Ini melibatkan persiapan server, pengaturan database, dan implementasi *API*.

Selanjutnya, *Machine Learning Programming* bertugas membuat model untuk fungsi *computer vision* yang ada di *frontend website*. Model ini nantinya akan di-deploy di lingkungan *cloud* yang telah disiapkan oleh *Backend Programming*. Mereka memastikan bahwa *model* bekerja dengan baik dan dapat diintegrasikan dengan baik ke dalam sistem.

Frontend, bertugas membuat bagian *website* yang akan dihubungkan ke backend. Ini melibatkan pembuatan user interface, implementasi fungsi-fungsi pada *website*, dan integrasi dengan backend. *API* memastikan bahwa *website* berfungsi dengan baik dan dapat berinteraksi dengan pengguna dengan lancar.

Setiap aktor memiliki peran yang penting dan saling terkait satu sama lain dalam proyek ini. Semua ini bekerja bersama untuk mencapai tujuan akhir proyek, yaitu pembuatan *website* yang berfungsi dengan baik dan memenuhi kebutuhan pengguna.

3.2 Alat dan Bahan Penelitian

Perangkat keras (hardware) dan perangkat lunak (software) yang digunakan dalam penelitian ini adalah sebagai berikut:

Tabel 3. 2 Alat dan Bahan

No	Perangkat	Spesifikasi	Deskripsi
1.	Laptop	Intel Core i5, RAM 16GB, RTX 3050 dengan sistem operasi Windows 11	Perangkat keras yang digunakan dalam proses pembuatan Aplikasi
2.	<i>Visual Studio Code</i>	<i>Version 1.74.3</i>	Perangkat lunak yang digunakan dalam proses pembuatan Aplikasi
3.	<i>Figma</i>	-	Perangkat lunak untuk merancang <i>wireframe</i> antarmuka pengguna

Tabel 3. 2 Alat dan Bahan (Lanjutan)

No	Perangkat	Spesifikasi	Deskripsi
4.	<i>Firebase</i>	<i>Platform Database</i>	<i>Database</i> tempat penyimpanan data yang akan digunakan
5.	<i>Node Package Manager (NPM)</i>	Versi 6.14.17	<i>Package Manager</i> untuk menginstall <i>library</i>
6.	<i>Bootstrap</i>	Versi 5.3	<i>Framework CSS</i> untuk pembuatan bagian <i>frontend website</i>
7.	<i>Javascript</i>	Javascript ES6	Bahasa pemrograman untuk pengembangan <i>website</i>
8.	<i>NodeJS</i>	Versi 18.16.1	<i>Runtime environment</i> JavaScript sebagai kebutuhan <i>back-end developer</i>
9.	<i>Browser Google Chrome</i>	Versi 120.0.6	Perangkat lunak yang membantu dalam proses

3.3 Tahapan Penelitian

Tahapan penelitian yang dilakukan terbagi menjadi tiga bagian yaitu pengumpulan data, pengembangan *website*, dan penulisan laporan.

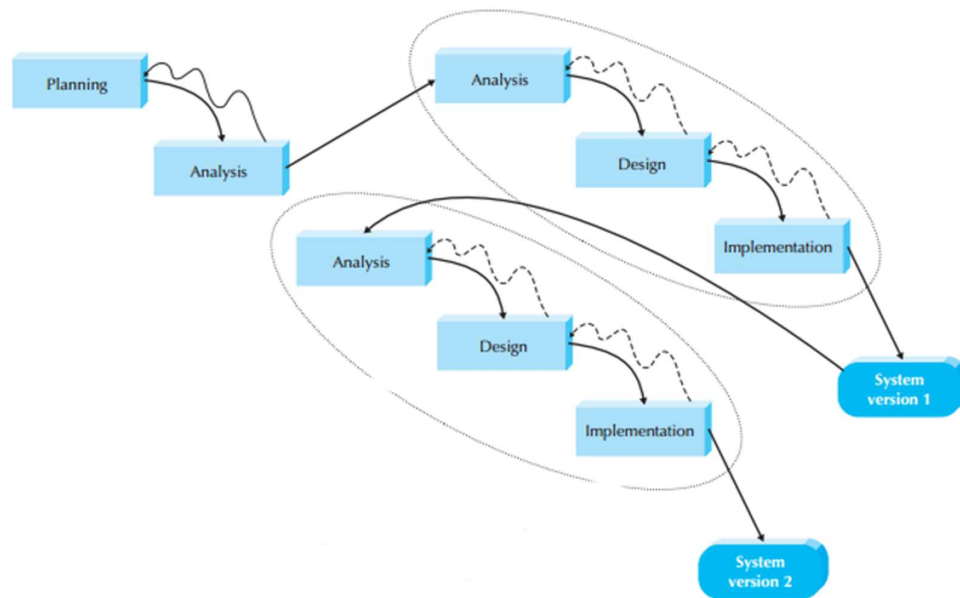
3.3.1. Metode Pengumpulan Data

Pengumpulan data merupakan kegiatan penelitian yang melibatkan pencarian informasi secara langsung pada objek penelitian untuk menjawab permasalahan yang diteliti. Dalam konteks ini, penulis menggunakan beberapa metode pengumpulan data yaitu observasi, wawancara, dan studi literatur.

- a) Metode Observasi digunakan untuk mengamati proses perhitungan pohon bakau di Hutan *Mangrove* Petengoran Dinas Pariwisata Kabupaten Pesawaran. Melalui observasi, data dan informasi terkait proses pendataan bakau yang masih dilakukan secara manual dengan menghitung luas besaran daerahnya tanpa menghitung banyak pohon dalam sebuah zona.
- b) Metode wawancara dilakukan dengan penanggung jawab Hutan Mangrove Petengoran untuk memahami secara mendalam proses pencatatan, langkah-langkah yang dilakukan di lokasi pendataan tanaman bakau, serta kebutuhan yang diperlukan untuk mendukung proses tersebut. Wawancara ini juga bertujuan menggali informasi mengenai kebutuhan pihak pengelola, yang menginginkan bantuan pendataan secara otomatis dan terintegrasi melalui sebuah platform. Sistem ini diharapkan dapat menggantikan metode manual, seperti menghitung pohon satu per satu langsung di rawa mangrove, sehingga dapat meningkatkan efisiensi waktu, meminimalkan kesalahan pencatatan, dan memperluas jangkauan area pendataan. Informasi yang diperoleh dari wawancara ini menjadi dasar penting dalam perancangan sistem informasi pendataan Hutan Mangrove Petengoran.
- c) Metode Studi Literatur digunakan untuk mengumpulkan informasi teoritis yang mendukung penelitian. Studi literatur menjadi landasan teori untuk pembuatan sistem informasi pendataan hutan bakau berbasis *website*. Tujuan dari tahap ini adalah memudahkan Masyarakat dalam melihat informasi dan data dari objek wisata hutan bakau petangoran serta data tersebut dapat dijadikan acuan sebagai pengukur saat ingin dilakukan pendataan produksi oksigen saat ingin dilakukan penjualan produksi oksigen untuk gas rumah kaca.

3.3.2. Metode Pengembangan Sistem

Metode pengembangan *website* di dalam penelitian ini adalah menggunakan metode. *Rapid Application Development* sebuah metode pengembangan perangkat lunak yang menekankan pada pendekatan iteratif dan inkremental. Pendekatan ini memungkinkan pengembang untuk dengan cepat merancang, membangun, dan menguji solusi perangkat lunak. Tahapan-tahapan dari metode *rapid application development* yaitu *planning*, *analysis*, *design*, dan *implementation*.



Gambar 2. 6 *Ilustrasi Rapid Application Development* [2]

3.3.2.1. Planning

Pada awal tahapan pengembangan sistem, diperlukan tahap *planning* yang mencakup analisis awal dan studi kelayakan. Tahap ini bertujuan untuk memastikan bahwa sistem yang akan dikembangkan layak dari segi teknis, ekonomis, dan operasional, serta mendapatkan persetujuan untuk dilanjutkan ke tahap pengembangan. Dalam proses ini, pengembangan aplikasi direncanakan untuk dilakukan dalam dua iterasi. Hal ini mengacu pada tahap parameter klasifikasi MoSCoW, serta kebutuhan teknis terdapat fitur yang memerlukan pengembangan tambahan, termasuk integrasi dan penambahan *API* baru yang harus disediakan oleh

tim *back-end*. Dengan demikian, iterasi ini dirancang untuk memastikan setiap kebutuhan teknis dapat terpenuhi secara optimal sebelum sistem diselesaikan sepenuhnya. awalan tahapan pengembangan sistem diperlukannya tahan *Planning* dimana tahapan ini akan dilakukannya analisis awal dan studi kelayakan. Parameter agar sistem dapat dikatakan layak jika secara teknis, ekonomis, dan operasional sudah memenuhi dan disetujui untuk dilakukan pengembangan sistemnya.

Tabel 3. 3 Parameter Kebutuhan

Parameter	Kebutuhan
Teknis	1. Memilih platform pengembangan <i>web</i> yang sesuai untuk merancang sistem informasi perhitungan pohon bakau, yang harus memungkinkan integrasi data. 2. Antarmuka Pengguna: a. Sistem dapat menampilkan data baik secara raw data ataupun data yang sudah diolah menjadi <i>map</i> b. Admin dapat, menambah gambar tahunan <i>Mangrove</i> Petangoran. c. Admin dapat, menghapus <i>file</i> koordinat zona tahunan <i>Mangrove</i> Petangoran. d. Admin dapat, menghapus gambar zona tahunan <i>mangrove</i> petangoran. e. Admin dapat, menambah, mengubah, dan menghapus artikel, data, dan data diri admin. f. Sistem dapat menampilkan artikel/berita mengenai hutan <i>mangrove</i> petangoran

Tabel 3. 3 Parameter Kebutuhan (Lanjutan)

Parameter	Kebutuhan
Ekonomis	1. Biaya Operasional: Perencanaan untuk biaya operasional setelah peluncuran, termasuk biaya hosting, dan pemeliharaan,. 2. Pendapatan Potensial: Evaluasi potensi pendapatan dari sistem informasi, misalnya melalui peningkatan efisiensi kerja pegawai.
Operasional	1. Manajemen Konten: Sistem yang memungkinkan administrator untuk mengelola dan memperbarui konten website serta data perhitungan pohon bakau secara berkala. 2. Pemantauan dan Dukungan: Layanan dukungan yang tersedia untuk memastikan ketersediaan sistem dan resolusi cepat terhadap masalah pengguna.

3.3.2.2. Analysis

Tahap selanjutnya setelah tahapan *planning* dalam proses pengembangan perangkat lunak ini dilanjutkan dengan tahapan *analysis* kebutuhan sistem. Pada tahap analisis, terdapat upaya kolaboratif dengan pihak pengelola *mangrove* Petangoran untuk membahas proses pengumpulan data, mengatasi permasalahan yang ada, dan mengidentifikasi perlunya suatu sistem informasi untuk mengumpulkan data tanaman *mangrove* di kawasan hutan *mangrove* Petangoran, yang akan selanjutnya dikembangkan. Tahap selanjutnya membuat *user story* dan menentukan kebutuhan fungsional dan non-fungsional sebagai berikut :

1. User Stories dan Acceptance Criteria

User story adalah cara untuk mendeskripsikan kebutuhan sistem dalam bahasa alami agar dapat dipahami dengan mudah oleh pengguna akhir yang mungkin tidak berpengalaman di bidang IT. *Acceptance criteria* merupakan kumpulan kondisi

yang menetapkan kebutuhan pengguna yang harus terpenuhi dalam pembuatan produk. *User stories* dan *acceptance criteria* dari Sistem adalah sebagai berikut :

Tabel 3. 4 *User Stories*

Kode	User Story	Acceptance Criteria
Pengguna		
US-01	Sebagai pengguna umum, saya ingin melihat data tentang Hutan <i>Mangrove</i> Petengoran baik dalam bentuk data tabel maupun grafik agar saya dapat memahami lebih baik tentang kondisi dan perkembangan hutan <i>mangrove</i> tersebut.	• Data tentang Hutan <i>Mangrove</i> Petengoran dapat diakses dari halaman terkait di <i>website</i>. • Data <i>mangrove</i> berupa tabel dan grafik harus tersedia dengan jelas dan mudah dimengerti. • Grafik harus interaktif dan dapat disesuaikan berdasarkan preferensi pengguna.
US-02	Sebagai pengguna umum, saya ingin melihat artikel atau berita terbaru mengenai Hutan <i>Mangrove</i> Petengoran agar saya dapat memperoleh informasi terkini tentang kondisi dan kegiatan di hutan <i>mangrove</i> tersebut.	• Artikel atau berita terbaru mengenai Hutan <i>Mangrove</i> Petengoran dapat diakses dengan mudah dari halaman utama <i>website</i>. • Artikel atau berita terbaru terupdate secara berkala dan menampilkan informasi yang relevan. • Pengguna dapat dengan jelas melihat tanggal publikasi artikel atau berita tersebut.

Tabel 3. 4 *User Stories* (Lanjutan 1)

Kode	User Story	Acceptance Criteria
Admin		
US-03	Sebagai admin, saya ingin memiliki akses penuh untuk mengelola konten seperti artikel, data, dan informasi diri admin di dalam sistem agar saya dapat memperbarui informasi dengan mudah dan cepat.	• Admin harus dapat masuk ke dalam sistem dengan menggunakan akun dan kata sandi yang valid. • Setelah masuk, admin harus memiliki akses penuh untuk mengelola konten seperti zona <i>mangrove</i>, data <i>mangrove</i>, artikel, data, dan informasi diri admin. • Perubahan yang dilakukan oleh admin harus tersimpan dan terlihat langsung di halaman website.
US-04	Sebagai <i>admin</i> , saya ingin dapat melihat, mengubah dan, menghapus data hasil <i>mangrove</i> ,	• Terdapat halaman edit data hutan bakau, dengan titik koordinat dan zona. • Terdapat fungsi hapus data hutan bakau, dengan varietas dan id kanopi pohon bakau.
US-05	Sebagai <i>admin</i> , saya ingin berpindah dari halaman admin ke halaman beranda utama sehingga saya dapat data hutan bakau dan artikel.	• Terdapat fitur logout dan login yang berfungsi untuk masuk dan keluar dari halaman admin.

Tabel 3. 4 *User Stories* (Lanjutan 2)

Kode	User Story	• Acceptance Criteria
US-06	Sebagai admin, saya ingin dapat mengunggah data peta setiap tahunnya dan menampilkan peta tersebut.	Terdapat halaman Upload admin yang berfungsi mengunggah peta dalam format jpg dan tif ke server, dan dapat melihat peta tersebut.
US-07	Sebagai <i>admin</i> , saya ingin melihat visualisasi diagram jumlah layanan, data <i>mangrove</i> dan artikel.	Terdapat halaman <i>dashboard</i> admin yang berfungsi untuk melihat informasi data <i>mangrove</i> , jumlah layanan, dan artikel.

Berdasarkan *user story* pada tabel 3.4 selanjutnya akan dilakukan pengelompokkan berdasarkan urgensi dari prioritas masing-masing fitur menggunakan metode *MoSCoW* menjadi empat kategori *Must Have*, *Should Have*, *Could Have*, dan *Wont Have* untuk lebih jelasnya sebagai berikut :

Tabel 3. 5 Tabel Klasifikasi MoSCoW

<i>Must Have</i>	
US-01	Data <i>mangrove</i> dalam bentuk tabel dan grafik dapat dilihat oleh pengguna umum.
US-04	Admin dapat melihat, mengubah, dan menghapus data hasil <i>mangrove</i> .
US-06	Admin dapat mengunggah dan menampilkan peta tahunan.

Tabel 3. 5 Tabel Klasifikasi MoSCoW (Lanjutan)

<i>Should Have</i>	
US-03	Admin memiliki akses penuh untuk mengelola konten (zona <i>mangrove</i> , artikel, dan data admin).
US-07	Admin dapat melihat visualisasi diagram jumlah layanan, data <i>mangrove</i> , dan artikel di <i>dashboard</i>
<i>Could Have</i>	
US-02	Pengguna umum dapat mengakses artikel atau berita terbaru mengenai Hutan Mangrove Petengoran.
US-05:	Admin dapat berpindah dari halaman admin ke halaman utama menggunakan fitur <i>logout</i> .

Berdasarkan hasil klasifikasi *MoSCoW*, fitur-fitur dikelompokkan ke dalam dua iterasi, pada iterasi pertama dilakukan pengembangan fitur yang dikategorikan sebagai *Must Have* dimana mencakup US-01, US-04, dan US-06 yang berfokus pada visualisasi data Bakau. Selanjutnya pada iterasi kedua mencakup fitur *Should Have* dan *Could Have*, yaitu US-03, US-07, US-02, dan US-05.

2. Analisa Kebutuhan

Analisa kebutuhan yang digunakan pada pengembangan Sistem Pendataan yaitu kebutuhan fungsional dan non fungsional. Kebutuhan fungsional dan non fungsional berdasarkan hasil analisis adalah sebagai berikut:

a. Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang berisi proses yang akan dilakukan oleh sistem. Berikut kebutuhan fungsional dari Sistem Pendataan Pohon Bakau:

Tabel 3. 6 Kebutuhan Fungsional

Role	Kebutuhan Fungsional
Admin	
US-03	Melihat Data Hutan Bakau
US-03	Melihat Artikel
US-03	Melihat Visualisasi Data Hutan bakau
US-03	Mengelola Artikel
US-03	Mengelola Data Admin
US-04	Mengelola Data Mentah Hutan Bakau
US-05	Melakukan <i>Login</i>
US-05	Melakukan <i>Logout</i>
US-06	Menginput Data Hutan Bakau
US-06	Mengelola Data Hutan Bakau
Pengguna	
US-01	Melihat Visualisasi Data Hutan Bakau
US-02	Melihat Artikel
US-02	Melihat Layanan

b. Kebutuhan Non-Fungsional

Kebutuhan non fungsional merupakan batasan kebutuhan yang dimiliki oleh sistem. Berikut kebutuhan non fungsional dari Sistem Pendataan Hutan Bakau:

Tabel 3. 7 Kebutuhan Non-Fungsional

Kebutuhan Non Fungsional	Deskripsi
<i>Availability</i>	Sistem dijalankan menggunakan koneksi internet dan berjalan sesuai dengan jam aktivitas di Hutan <i>Mangrove</i> Petangoran.
<i>Portability</i>	Sistem ini mampu berjalan diberbagai sistem operasi yang mampu menggunakan <i>web browser</i> .
<i>Security</i>	Pada sistem ini memiliki <i>username</i> dan <i>password</i> untuk admin yang dapat melakukan pengelolaan data.
<i>Build</i>	Sistem dibangun menggunakan bahasa pemrograman <i>HTML</i> , <i>CSS</i> , dan <i>JavaScript</i> .

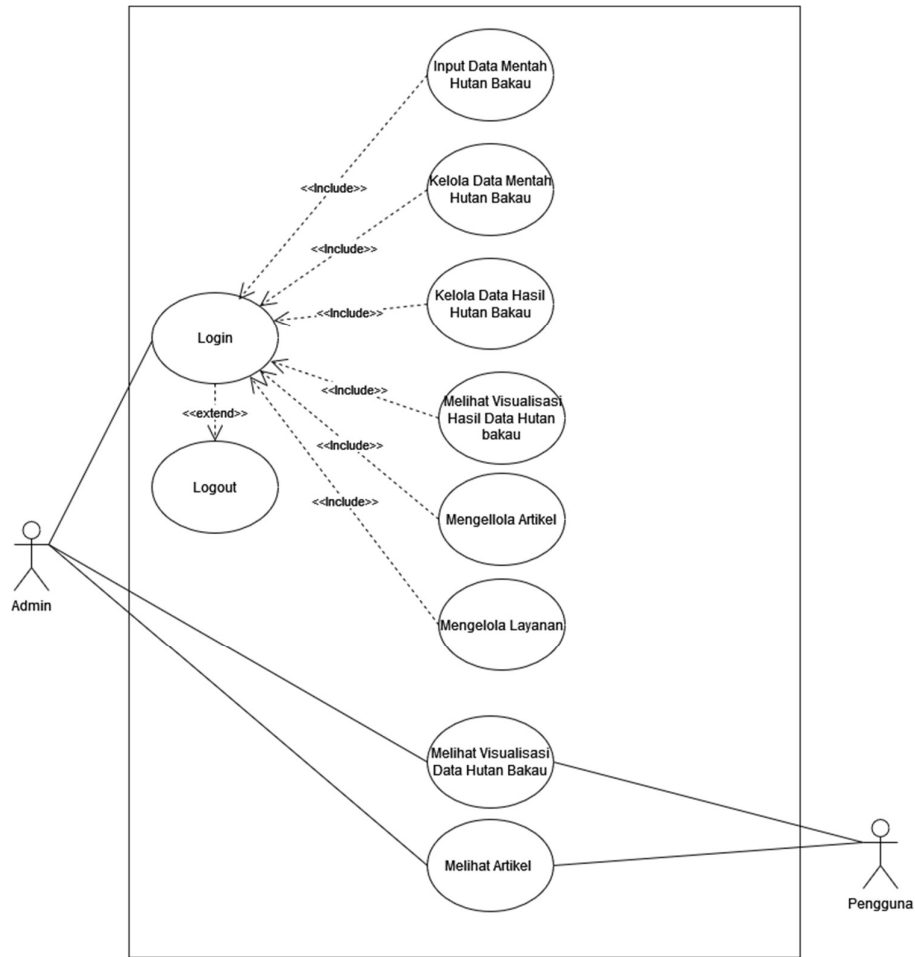
3.3.2.3. Desain (Perancangan)

Setelah menganalisis masalah dan bentuk *Analysis* sistem yang akan dikembangkan, desain antarmuka pengguna menjadi tahap dalam pengembangan. Tujuan dari langkah ini adalah untuk menyajikan tampilan sistem yang akan digunakan. Salah satu keuntungan dari *UML* adalah membantu pengembang dengan menyediakan informasi grafis kepada pengguna atau bahasa pemodelan visual. *UML* memodelkan sistem berbasis objek, memberikan deskripsi model, dan berfungsi sebagai blueprint untuk informasi yang lebih mendalam saat merancang program. Informasi berikut ini termasuk dalam *Unified Modelling Language* (*UML*) seperti *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, *Interface System*, dan instalasi *environment* dengan rincian sebagai berikut:

1. Use Case Diagram

Proses pembuatan *use case diagram* mengilustrasikan representasi grafis yang bertujuan untuk menunjukkan operasi dan aktivitas sebuah sistem. Dengan menunjukkan hubungan interaksi antara aktor dan sistem, *use case diagram* menggambarkan aspek fungsional dari sebuah aplikasi. Langkah pertama melibatkan identifikasi aktor-aktor yang perlu dibuat dalam sistem. Selanjutnya,

peran-peran yang dapat dimainkan oleh setiap aktor diidentifikasi. Berikut adalah *use case diagram* untuk sistem informasi hutan bakau:



Gambar 3. 2 *Use Case Diagram*

Use Case Diagram sistem informasi visualisasi data hutan bakau di wilayah Petangoran menggambarkan interaksi antara dua aktor, yaitu Admin dan Pengguna, dengan sistem. Aktor pertama adalah Admin yang memiliki berbagai fungsi setelah melakukan *login*. Admin dapat melakukan operasi *Create*, *Read*, *Update*, dan *Delete* (CRUD) pada data mentah, data hasil/jadi, layanan dan artikel. Data yang dikelola berupa data koordinat, dan visualisasi data hutan bakau. Selain itu, Admin dapat memasukkan data mentah baru dalam format file Gambar dan *GeoJson* untuk

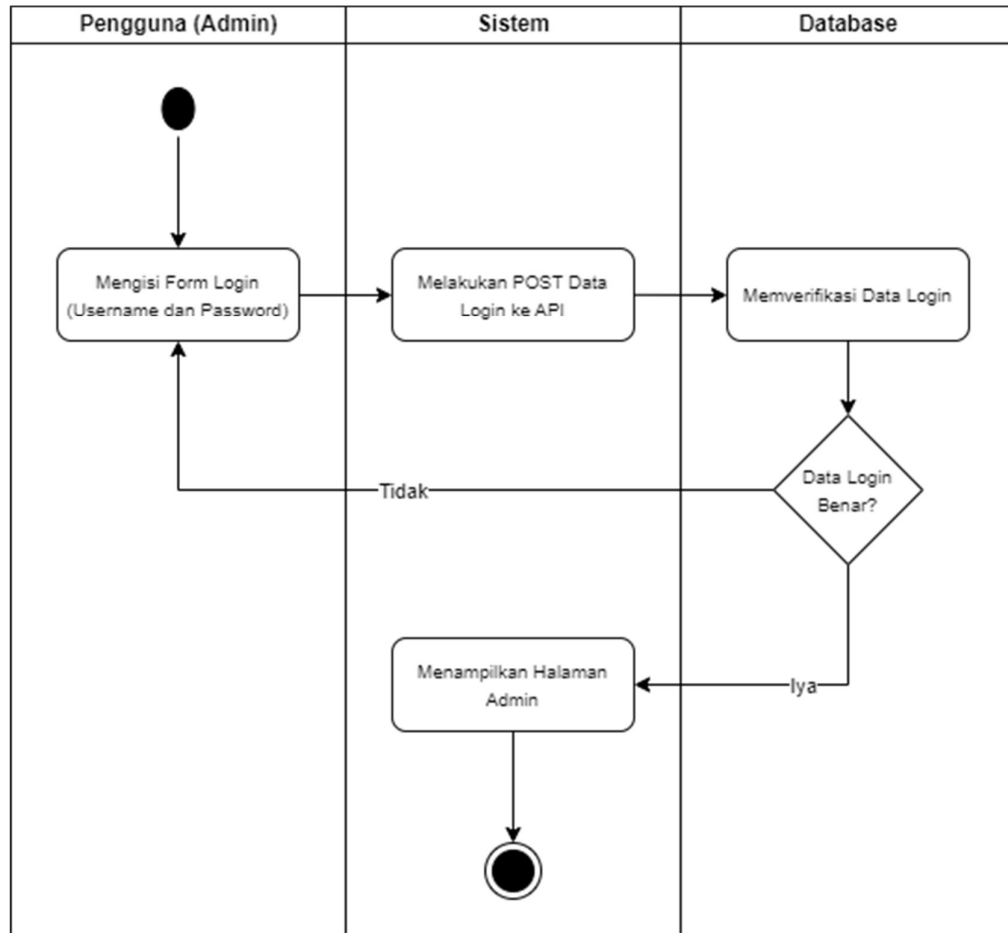
file ini berisikan data data koordinat zona hutan bakau. Aktor kedua adalah Pengguna yang dapat melihat visualisasi data hutan bakau dan membaca artikel tanpa perlu melakukan *login*. Visualisasi data disajikan dalam tabel dan peta zona *mangrove* petangoran yang telah dimasukan oleh admin. Dengan demikian, sistem ini memfasilitasi pengelolaan dan akses informasi tentang hutan bakau di wilayah Petangoran.

2. *Activity Diagram*

Activity Diagram merupakan representasi alur kerja atau alur aktivitas yang telah direncanakan dalam sebuah sistem. *Activity Diagram* dibuat untuk menunjukkan interaksi antara aktor dan sistem sesuai dengan penggunaannya, mulai dari awal hingga akhir sistem tersebut. Diagram ini menampilkan urutan di mana setiap aktor menyelesaikan tugas mereka. *Activity Diagram* terdiri dari bagian-bagian yang memiliki bentuk yang berbeda dan dihubungkan oleh panah. Panah-panah ini menunjukkan urutan dimana berbagai aktivitas berlangsung dari awal hingga akhir. Diagram aktivitas Sistem Informasi Hutan Bakau terlihat seperti berikut:

a) *Activity Diagram Login*

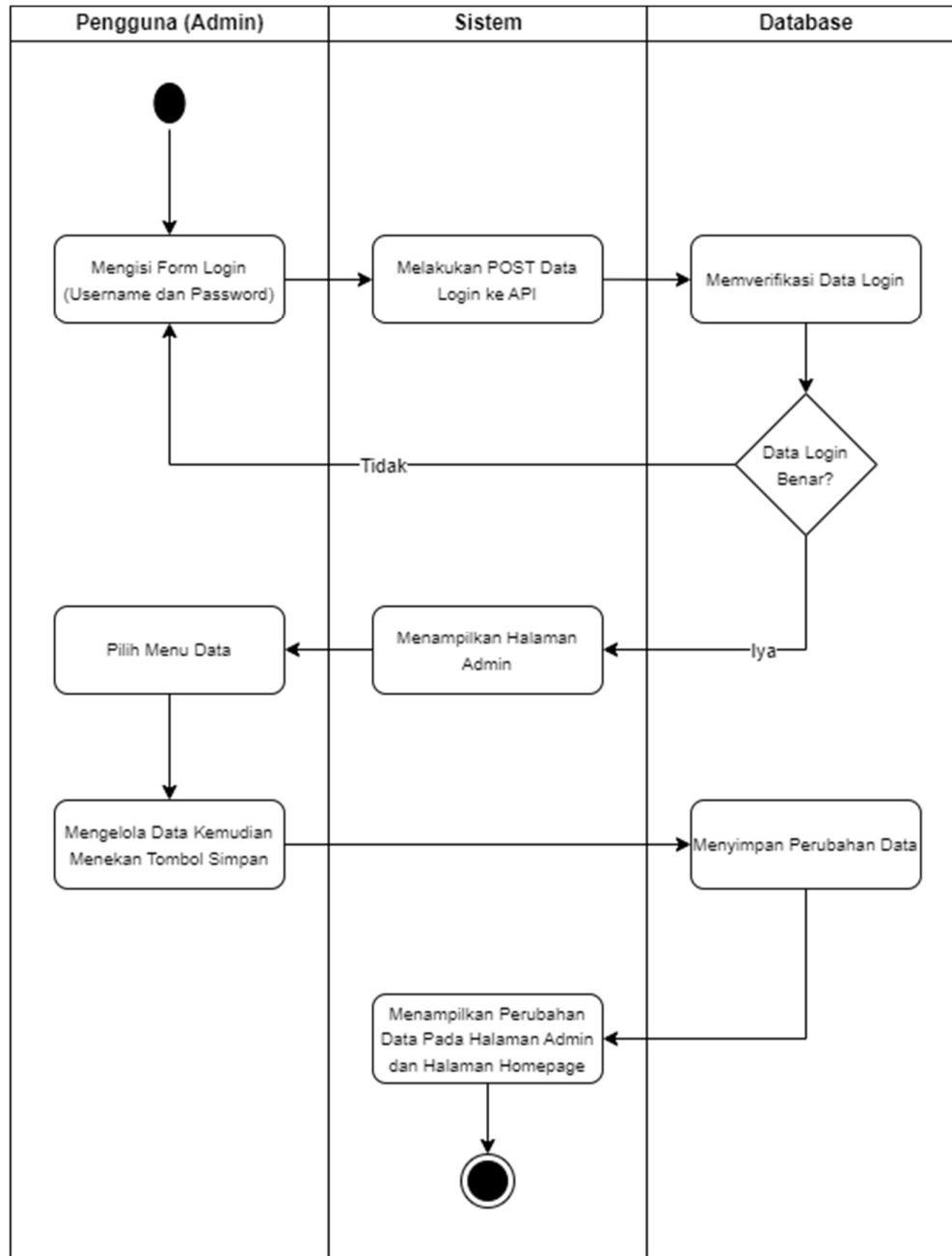
Activity diagram *login* merupakan gambaran aktivitas admin untuk melakukan *login* ke sistem sebagai autentifikasi dalam mengelola data. Sebelum berada pada halaman admin diminta untuk memasukkan data akun yang telah terdaftar dengan menggunakan *username* dan *password*. Jika admin salah mengisi *username* dan *password* maka admin akan dialihkan ke halaman *homepage*.



Gambar 3. 3 *Activity Diagram Login*

b) *Activity Diagram Mengelola Data*

Activity diagram mengelola data merupakan gambaran aktivitas admin untuk melakukan perubahan data seperti *Create, Read, Update, dan Delete* (CRUD) data *mangrove*, artikel, layanan, dan profil admin.

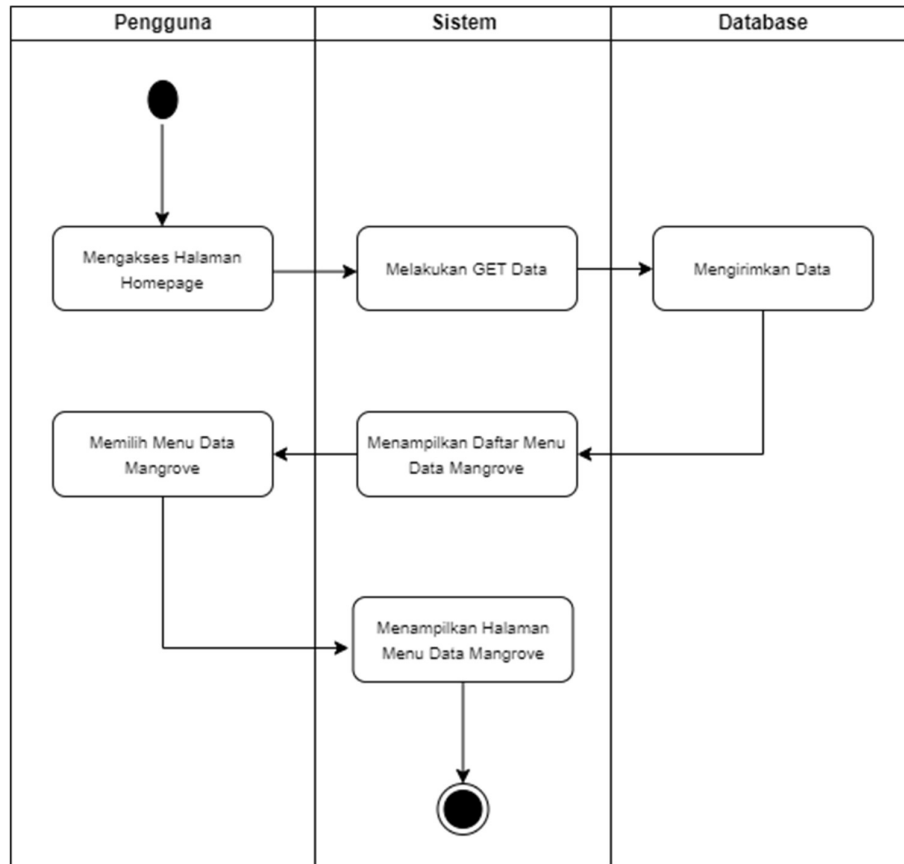


Gambar 3. 4 *Activity Diagram* Melihat Data

c) *Activity Diagram* Melihat Data

Activity diagram melihat data merupakan gambaran aktivitas pengguna untuk melihat halaman *homepage* yang berisi data *mangrove*, artikel, layanan, dan

informasi lainnya. Pada halaman *homepage* pengguna tidak melakukan *login* karena fungsi halaman *homepage* hanya menampilkan informasi yang telah disediakan oleh admin.



Gambar 3. 5 Activity Diagram Homepage Admin

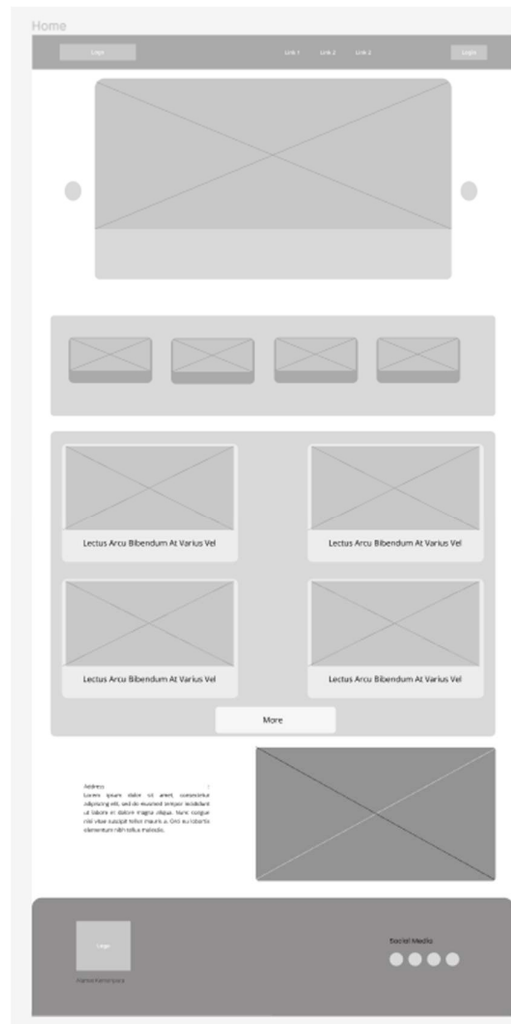
3. Interface System

Interface system adalah representasi grafis dari sistem yang akan dibuat sesuai dengan kebutuhan pengguna. *Figma* digunakan sebagai berikut dalam proses pembuatan antarmuka sistem:

1) Tampilan Halaman Beranda

Pada halaman beranda menampilkan landing page dimana itu merupakan artikel, lalu informasi mengenai alamat dan peta hutan *mangrove* petangoran, dan

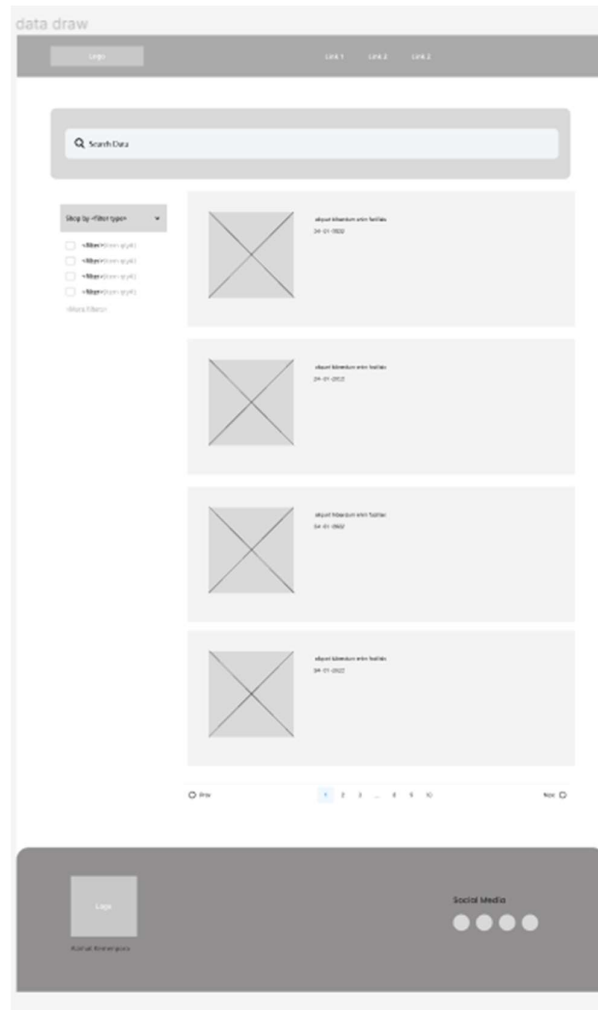
menampilkan layanan lain yang dimiliki pengurus hutan *mangrove* petangoran.
rancangan



Gambar 3. 6 Rancangan Tampilan *Dashboard*

2) Tampilan Halaman Draw Visualisasi Data

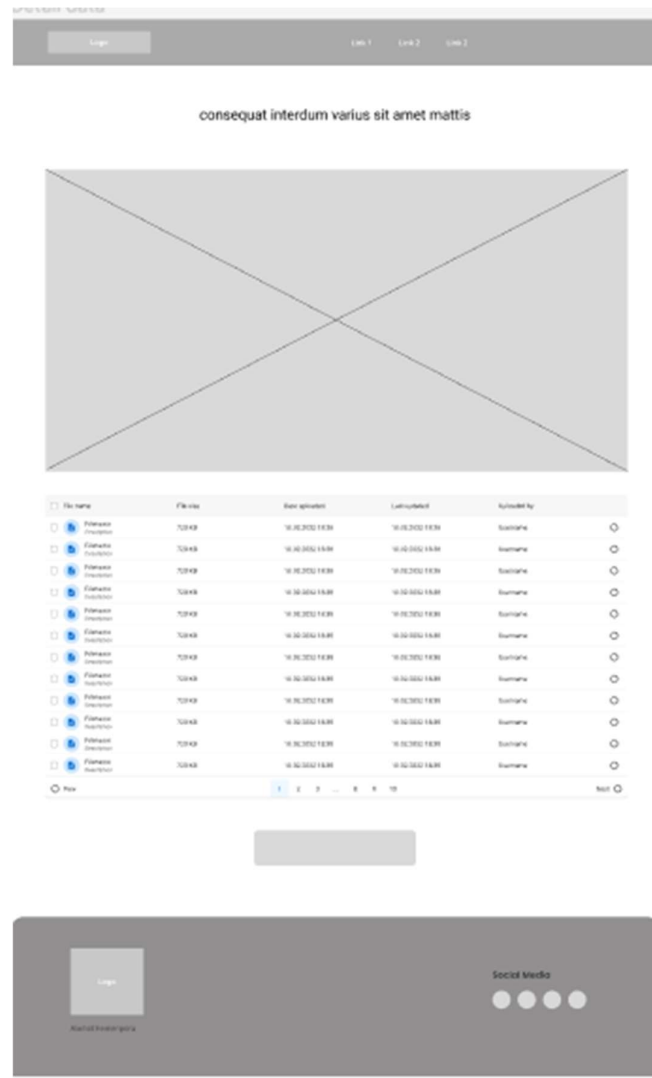
Pada Halaman draw visualisasi data berisikan list data-data yang dimiliki oleh hutan *mangrove* petangoran, secara menyeluruh, terdapat filter dan pencarian yang berfungsi untuk mencari secara spesifik untuk data tersebut.



Gambar 3. 7 Rancangan Tampilan Pencarian Data

3) Tampilan Halaman Visualisasi Data Hutan Bakau Petangoran

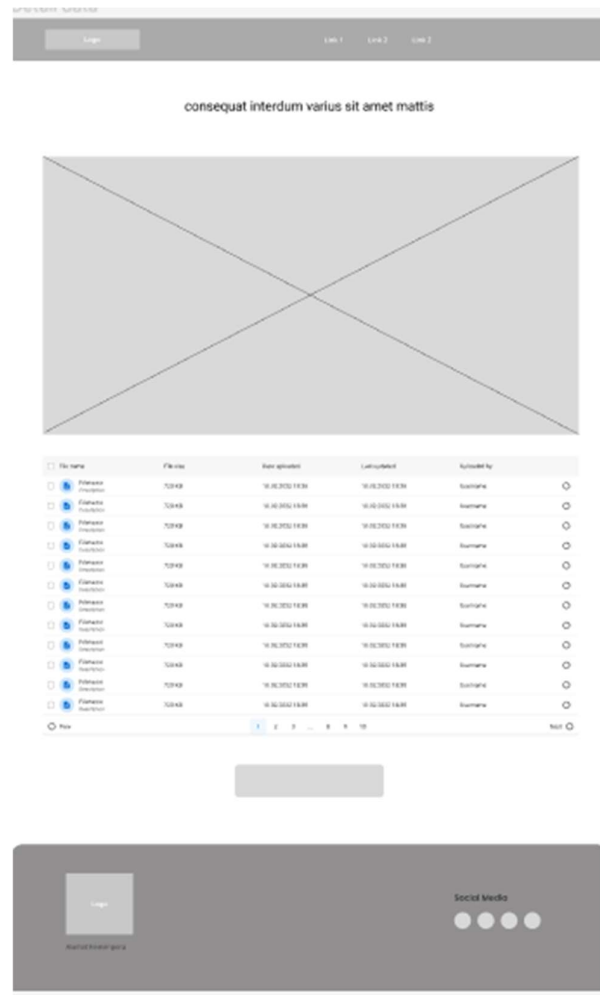
Pada halaman visualisasi data hutan bakau berisikan visualisasi data berupa *map* dan tabel yang berisi data-data mengenai hutan bakau yang berada di hutan *mangrove* petangoran.



Gambar 3. 8 Rancangan Visualisasi Data Hutan Bakau

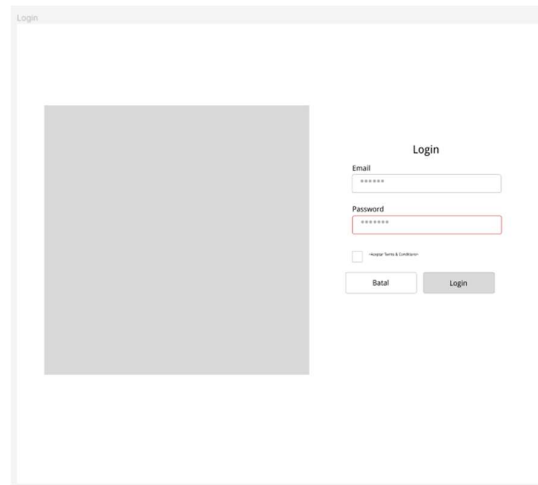
4) Tampilan Halaman Detail Visualisasi data

Pada halaman detail visualisasi data berisikan visualisasi data sesuai dengan format yang telah diatur terlebih dahulu atau pada bagian atas merupakan visualisasi data berbentuk *map* dan untuk bagian bawahnya merupakan visualisasi data berupa tabel.



Gambar 3. 9 Rancangan Tampilan Detail Visualisasi Data

5) Rancangan Tampilan Halaman *Login Admin*



Gambar 3. 10 Rancangan Halaman *Login*

Pada halaman login admin dapat mengisi username dan password bertujuan untuk dapat mengelola Data dan artikel Rancangan tampilan dapat dilihat pada gambar diatas.

3.3.2.4. *Implementation*

Pada tahapan ini adalah tahap pengembangan desain menjadi sebuah sistem dari rancangan desain *wireframe* yang telah dibuat tahapan desain. Pada tahapan ini sistem dibuat dengan menggunakan *bootstrap* sebagai *framework CSS* dalam pembuatan bagian *frontend website*, *NodeJS* sebagai *runtime environment JavaScript* dalam pembuatan fungsi dan tampilan *website*, dan *ExpressJS* sebagai *framework NodeJS* dalam pembuatan *backend*. Selanjutnya *black-box* sebagai proses pengujian merupakan tahap akhir yang dilakukan untuk mengetahui kesesuaian output, input, dan fungsionalitas dari sistem. Selain itu tahapan pengujian sangat penting dalam menjamin kualitas sistem. Sedangkan pengujian *user experience* menggunakan *User Experience Questionnaire*.

Persiapan pembuatan aplikasi :

4. Instalasi enviroment

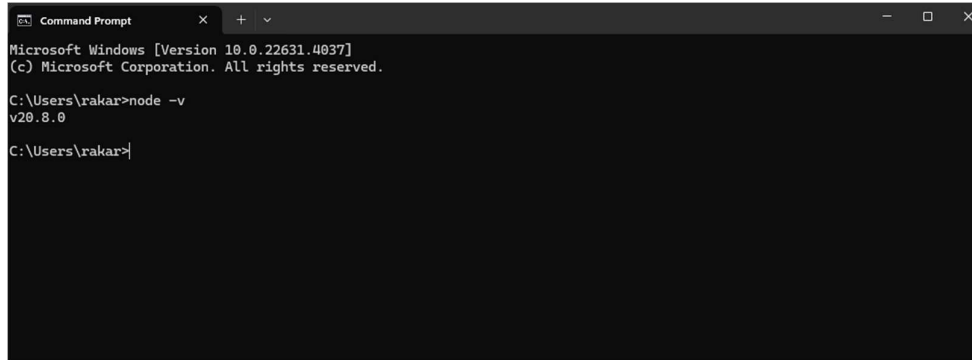
1) Instalasi dan Konfigurasi NodeJS



Gambar 3. 11 Instalasi *NodeJs*

NodeJS adalah lingkungan runtime yang menyediakan pustaka *JavaScript* untuk memperlancar pengembangan *website*. *NodeJS* memfasilitasi menjalankan server lokal menggunakan perintah yang tersedia. *NodeJS* dilengkapi dengan *Node Package Manager* (NPM), yang memungkinkan pengembang untuk mengelola ketergantungan dan paket *JavaScript* eksternal secara efektif.

Langkah awal melibatkan mengunduh *NodeJS* dari situs *web* resmi di <https://nodejs.org/> dan memilih versi *LTS* yang direkomendasikan yang cocok untuk stabilitas dengan sistem operasi (*Windows*, *macOS*, atau *Linux*). Selama proses konfigurasi *NodeJS*, sangat penting untuk memastikan bahwa *NodeJS* dan *Node Package Manager* (NPM) terpasang dengan benar dan berfungsi dengan optimal. Hal ini dapat dipastikan dengan membuka terminal atau command prompt dan memasukkan perintah “node -v” atau dapat dilihat seperti pada gambar 3.12 :



```

Microsoft Windows [Version 10.0.22631.4837]
(c) Microsoft Corporation. All rights reserved.

C:\Users\rakar>node -v
v20.8.0

C:\Users\rakar>

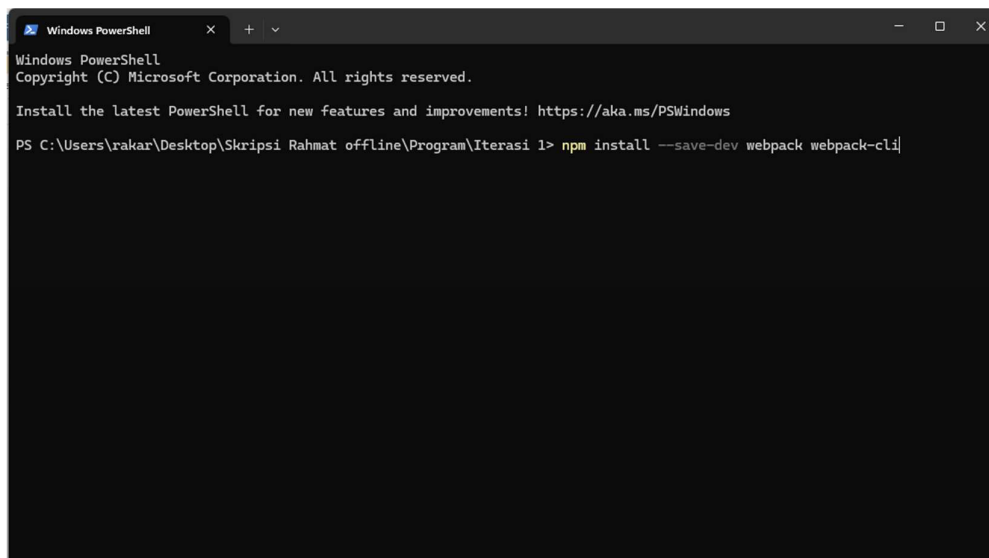
```

Gambar 3. 12 *Node Version*

2) Instalasi dan konfigurasi Webpack

Webpack adalah *bundler* yang digunakan untuk mengelola dan mengkompilasi modul *JavaScript* modern. Instalasi *Webpack* memiliki persyaratan yaitu *Node.js* dan *npm* sudah terpasang di komputer.

- a. Instalasi *Webpack* dan *Webpack CLI*: Setelah inisialisasi proyek, instal *webpack* dan *webpack-cli* sebagai dependencies pengembangan:



```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\rakar\Desktop\Skrripsi Rahmat offline\Program\Iterasi 1> npm install --save-dev webpack webpack-cli

```

Gambar 3. 13 Instalasi *Webpack*

- b. Membuat File konfigurasi *Webpack* (*webpack.config.js*) diperlukan untuk menentukan bagaimana proyek akan dikompilasi. Buat file di *root* proyek dengan isi seperti di bawah ini:

```

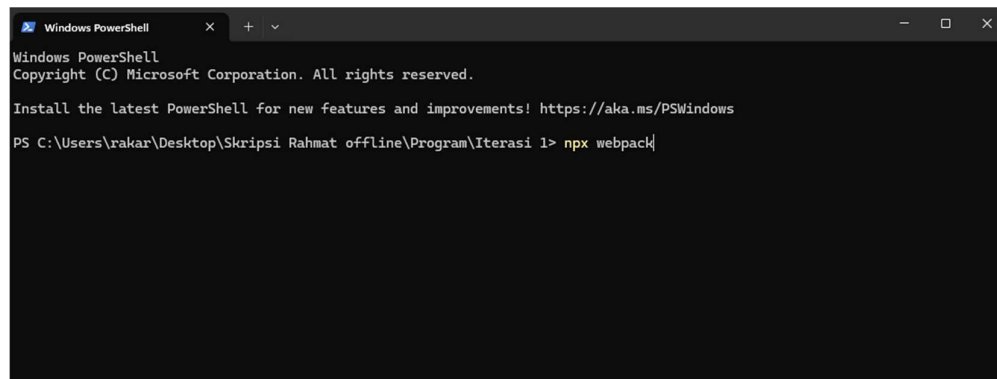
const path = require('path');
const { merge } = require('webpack-merge');
const common = require('./webpack.common');

module.exports = merge(common, {
  mode: 'development',
  devtool: 'inline-source-map',
  devServer: {
    static: path.resolve(__dirname, 'dist'),
    open: true,
    port: 8161,
    client: {
      overlay: {
        errors: true,
        warnings: true,
      },
    },
  },
  compress: true,
});

```

Gambar 3. 14 Konfigurasi Webpack

- c. Menjalankan *Webpack*: Untuk menjalankan *Webpack*, gunakan perintah:

Gambar 3. 15 Menjalankan *Webpack*

Webpack akan memproses semua file *JavaScript* yang dimulai dari file *index.js* dan menghasilkan file *bundle.js* di folder *dist*.

- d. Mengatur Skrip di *package.json*: Untuk mempermudah penggunaan *Webpack*, tambahkan skrip di file *package.json* agar *webpack* dapat dijalankan dengan

lebih singkat. untuk menjalankan *website* tersebut dapat dengan perintah npm run-dev(disesuaikan dengan script yang digunakan)

```
{
  "name": "MangrovePetangoran",
  "version": "1.0.0",
  "description": "Uro MangrovePetangoran",
  "main": "index.js",
  "scripts": {
    "start-dev": "webpack serve --config webpack.dev.js",
    "build": "webpack --config webpack.prod.js"
  },
}
```

Gambar 3. 16 Skrip *Webpack*

3) Membuat dan Menyusun Struktur Folder

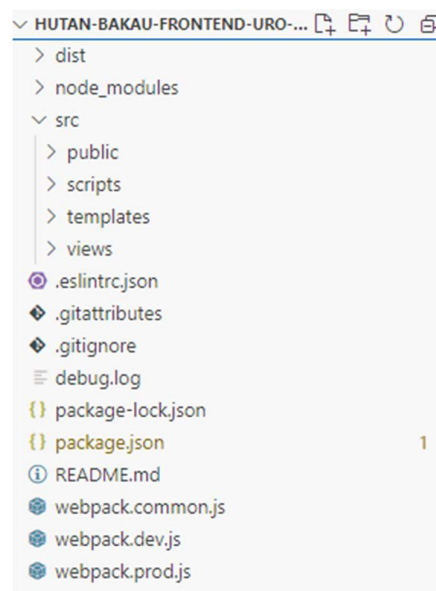
Pada tahap awal pengembangan sistem informasi visualisasi data *mangrove*, proses dimulai dengan pembuatan folder yang kemudian diberi nama sesuai dengan kebutuhan proyek. Folder tersebut kemudian dibuka menggunakan *Visual Studio Code*, yang akan digunakan sebagai editor kode. Selanjutnya, terminal dibuka dan perintah `npm init` dijalankan. Pengguna harus menjawab semua pertanyaan yang diajukan oleh *Node Package Manager* (NPM) untuk mengkonfigurasi proyek, atau dapat menggunakan flag `--y` pada terminal untuk menjawab secara otomatis dengan nilai default. Setelah menjawab seluruh pertanyaan, pastikan bahwa berkas `package.json` telah dibuat di dalam folder proyek.

Langkah berikutnya adalah menginstal paket atau ketergantungan, seperti *Webpack* sebagai module bundler dan lingkungan pengembangan. Pemasangan paket dilakukan dengan mengetik perintah di terminal: `npm install nama-package --save-dev` dan `npm install nama-package`.

Untuk memastikan konsistensi penulisan kode JavaScript dalam proyek, perintah `npx eslint --init` digunakan untuk menginisialisasi ESLint. Jawablah seluruh pertanyaan yang diberikan oleh ESLint. Jika konfigurasi ESLint berhasil, akan

terbentuk berkas baru bernama `.eslintrc.json` di dalam folder proyek, jika memilih format JSON.

Setelah tahap ini selesai, langkah selanjutnya adalah menyusun struktur folder proyek. Folder utama diberi nama `src`, karena akan digunakan untuk menyimpan berbagai aset dan berkas *JavaScript* yang akan dibuat. Struktur folder dirancang dalam kelompok-kelompok terpisah untuk memudahkan pengelolaan aset-aset yang ada dan memastikan penulisan kode *JavaScript* yang lebih terstruktur, sebagaimana yang ditunjukkan pada gambar 3.17.



Gambar 3. 17 Struktur Folder

Penjelasan struktur folder sebagai berikut:

- a) `node_modules` : Direktori JavaScript yang berfungsi sebagai tempat penyimpanan maupun pengelolaan dan berisi modul-modul yang terdiri dari kode JavaScript, *dependencies*, dan berkas lain yang diperlukan untuk menjalankan aplikasi atau proyek NodeJS.
- b) `src → public` : Berfungsi untuk menyimpan aset yang dapat diakses secara *public* seperti *favicon*, *icons*, gambar, video, atau *font*.
- c) `src → styles` : Berfungsi untuk menyimpan berkas *styling* seperti CSS.
- d) `src → scripts` : Berfungsi untuk menyimpan berkas JavaScript dan

termasuk *entry point*.

- e) `src` → *templates* : Berfungsi untuk menyimpan aset *HTML*.
- f) `.eslintrc.json` : Berkas konfigurasi *ESLint* yang berfungsi membantu mengidentifikasi dan memperbaiki kode JavaScript
- g) `package-lock.json` : Berkas yang dihasilkan oleh *Node Package Manager* (NPM) saat menjalankan perintah “npm install” yang berfungsi untuk menyimpan informasi spesifik tentang versi paket dan *dependencies* yang diinstal dalam suatu proyek.
- h) `package.json` : Berkas konfigurasi dalam proyek NodeJS yang mendefinisikan metadata proyek, menyimpan *dependencies*, dan mengatur berbagai pengaturan proyek.

4) Konsumsi *API*

Pada penelitian ini, terdapat tahapan penting yang melibatkan konsumsi *API* yang telah dikembangkan oleh tim *backend*. Setiap *API* dirancang untuk mendukung pengambilan data, pengelolaan, dan interaksi dengan basis data *mangrove*. Tahapan konsumsi *API* melibatkan pemanggilan *endpoint-endpoint* menggunakan metode *HTTP* seperti *GET*, *POST*, *PUT*, dan *DELETE* [8]. Melalui tahapan ini, data yang dikumpulkan dari server backend kemudian ditampilkan atau diolah lebih lanjut di frontend, memungkinkan pengguna untuk mengelola informasi *mangrove* dengan lebih efektif. Berikut *source code* mengenai *endpoint API* yang dapat dilihat pada gambar 3.17.

```

import axios from 'axios';
const API_URL = 'http://35.219.109.18:5000'; // Replace with your actual API URL

const api = axios.create({
  baseURL: API_URL,
  headers: {
    'Content-Type': 'application/json',
  },
});

export const getMangroveData = async (wilayah, tahun, page = 1, perPage = 1000) => {
  const response = await axios.get(`${API_URL}/get_mangrove_data`, {
    params: { wilayah, tahun: tahun, page, per_page: perPage }
  });
  return response.data;
};

export const getMapData = async (wilayah, tahun, tahun_peta) => {
  const response = await axios.get(`${API_URL}/map_data`, {
    params: { wilayah, tahun: tahun, tahun_peta }
  });
  return response.data;
};

export const getJumlah = async (wilayah, tahun) => {
  const response = await axios.get(`${API_URL}/get_jumlah`, {
    params: { wilayah, tahun: tahun }
  });
  return response.data;
};

```

Gambar 3. 18 *Endpoint API* [A]

```

export const updateMangroveData = async (kanopiId, varietas) => {
  const response = await axios.put(`${API_URL}/update_mangrove_data`, {
    kanopi_id: kanopiId,
    varietas
  });
  return response.data;
};

export const deleteMangroveData = async (kanopiId, wilayah, tahun) => {
  const response = await axios.delete(`${API_URL}/delete_mangrove_data`, {
    params: { kanopi_id: kanopiId, wilayah, tahun: tahun }
  });
  return response.data;
};

export const deleteDetection = async (wilayah, tahun) => {
  const response = await axios.delete(`${API_URL}/delete_detection`, {
    params: { wilayah, tahun: tahun }
  });
  return response.data;
};

export const getFullImageUrl = (relativePath) => {
  return `${API_URL}${relativePath}`;
};

export const downloadCanopyShapefile = async (wilayah, tahun) => {
  const response = await axios.get(`${API_URL}/download-canopy-shapefile`, {
    params: { wilayah, tahun: tahun },
    responseType: 'blob',
  });

  const blob = new Blob([response.data], { type: 'application/zip' });

  const link = document.createElement('a');
  link.href = window.URL.createObjectURL(blob);
  link.download = `${wilayah}_${tahun}_canopy_bounding_boxes.zip`;
  link.click();
};

export const uploadMap = (formData) => {
  return api.post('/upload-map', formData, {
    headers: { 'Content-Type': 'multipart/form-data' }
  });
};

export default api;

```

Gambar 3. 19 *Endpoint API* [B]

berikut uraian dalam bentuk tabel yang menjelaskan fungsi-fungsi *API* yang terdapat dalam aplikasi, lengkap dengan metode *HTTP* dan parameter yang digunakan yang dapat dilihat pada tabel 3.7.

Tabel 3. 8 *Endpoint API*

Fungsi API	Metode HTTP	URL	Deskripsi	Parameter
getMangroveData	GET	/get_mangrove_data	Mengambil data <i>mangrove</i> berdasarkan wilayah, tahun, dan pagination/halaman .	wilayah, tahun, page, per_page
getJumlah	GET	/get_jumlah	Mengambil jumlah <i>mangrove</i> berdasarkan wilayah dan tahun tertentu.	wilayah, tahun
updateMangrove Data	PUT	/update_mangrove_data	Memperbarui data kanopi <i>mangrove</i> berdasarkan ID kanopi dan varietas tanaman.	kanopi_id, varietas
deleteMangrove Data	DELETE	/delete_mangrove_data	Menghapus data <i>mangrove</i> berdasarkan ID kanopi, wilayah, dan tahun.	kanopi_id, wilayah, tahun
deleteDetection	DELETE	/delete_detection	Menghapus data deteksi berdasarkan wilayah dan tahun tertentu.	wilayah, tahun
uploadMap	POST	/upload-map	Mengupload file peta menggunakan multipart/form-data.	formData (file peta)
getMapData	GET	/map_data	Mengambil koordinat kanopi bakau	wilayah, tahun: tahun, tahun_peta

Penjelasan tabel konsumsi *API* :

1. **getMangroveData** menggunakan metode **GET** untuk mengambil data *mangrove* berdasarkan wilayah, tahun, dan pagination. Parameter yang digunakan adalah wilayah (wilayah yang ingin diambil datanya), tahun (tahun data yang diinginkan), page (halaman data), dan per_page (jumlah data per halaman). *Endpoint* ini membantu mengambil data *mangrove* dalam jumlah besar dengan memanfaatkan fitur pagination agar lebih ringan dalam memuat data *mangrove* dalam sebuah zona.
2. **getJumlah** adalah fungsi **GET** yang mengambil jumlah data *mangrove* yang ada di wilayah tertentu untuk tahun yang ditentukan. Parameter yang digunakan adalah wilayah dan tahun. Fungsi ini memungkinkan pengguna mendapatkan informasi mengenai total jumlah pohon *mangrove* di wilayah dan tahun yang spesifik.
3. **updateMangroveData** menggunakan metode **PUT** untuk memperbarui data *mangrove* berdasarkan ID kanopi (*kanopi_id*) dan varietas (jenis *mangrove*). Fungsi ini mengirimkan permintaan untuk memperbarui data yang ada di server dengan informasi terbaru mengenai varietas tanaman *mangrove* di wilayah tersebut.
4. **deleteMangroveData** menggunakan metode **DELETE** untuk menghapus data *mangrove* berdasarkan ID kanopi, wilayah, dan tahun. Parameter yang digunakan adalah *kanopi_id* (ID kanopi yang akan dihapus), wilayah, dan tahun. Fungsi ini memberikan kemampuan untuk mengelola dan menghapus data *mangrove*.
5. **deleteDetection** menggunakan metode **DELETE** untuk menghapus data deteksi *mangrove* berdasarkan wilayah dan tahun. Fungsi ini digunakan saat pengguna ingin menghapus data deteksi terkait wilayah dan tahun tertentu, mungkin karena adanya kesalahan data atau kebutuhan pembaruan.
6. **uploadMap** menggunakan metode **POST** untuk mengunggah file peta dengan format multipart/form-data. Parameter yang dikirim dalam bentuk formData

digunakan untuk mengelola file yang diunggah oleh pengguna ke server.

7. **getMapData** menggunakan metode **GET** untuk mengambil data koordinat kanopi *mangrove* berdasarkan wilayah, tahun, dan tahun_peta. Parameter yang digunakan adalah wilayah, tahun, tahun_peta.

Setiap fungsi *endpoint* dirancang untuk berinteraksi dengan *API* secara spesifik dan mengelola data *mangrove* serta peta dalam penelitian ini

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, dapat ditarik Kesimpulan adalah sebagai berikut :

1. Sistem visualisasi data *mangrove* petangoran yang dikembangkan menggunakan *javascript* telah berhasil di kembangkan dan fitur visualisasi dapat berjalan dengan baik.
2. Berdasarkan pengujian *blackbox* testing mendapatkan hasil yang sesuai dengan tiap skenario yang diuji, serta telah sesuai tampilan dan fungsi yang diinginkan oleh pengguna
3. Berdasarkan pengujian *User Experience Questionnaire* (UEQ) yang telah dilakukan dengan total responden sebanyak 30 orang memperoleh hasil pengujian kategori *Excellent* dalam hal daya tarik, ketepatan, stimulasi, dan kebaruan inovasi, selanjutnya parameter kejelasan, dan efisiensi mendapatkan kategori *Good*.
4. Berdasarkan penelitian ini pada *website* telah berhasil menampilkan data kanopi dari *API* dan dapat menampilkan data kanopi di keempat zona *mangrove* petangoran dengan data *mangrove* yang diperoleh dengan metode pembacaan *Deepforest*.
5. Berdasarkan capaian penelitian menunjukkan adanya optimalisasi dalam proses pendataan hutan bakau di wilayah *mangrove* petangoran. peningkatan ini dimana sebelum ada penelitian memakan waktu lima hingga sepuluh menit pada perhitungan pohon bakau pada gambar zona 4.27, dan setelah dilakukan penelitian menjadi sepuluh hingga tiga puluh detik.
6. Berdasarkan penelitian berhasil mengembangkan sebuah website yang menampilkan data pohon bakau di zona hutan mangrove petangoran yang dapat diakses dengan mudah oleh masyarakat umum, pengelola kawasan, dan

peneliti. Selain meningkatkan efisiensi pendataan, platform ini juga berkontribusi dalam mendukung upaya konservasi hutan mangrove melalui penyediaan data yang terintegrasi.

5.2 Saran

1. Optimasi antarmuka: disarankan untuk melakukan penelitian *UI/UX* yang lebih mendalam guna meningkatkan keselarasan desain antarmuka, fungsionalitas, dan estetika sistem. Mengingat bahwa desain antarmuka pengguna dalam penelitian ini sebagian besar didasarkan pada masukan pengguna dan proses pengembangan dengan menggunakan metode *prototipe*, hal ini menjadi sangat penting.
2. Sistem informasi visualisasi data *mangrove* petangoran disarankan memiliki fitur input deteksi tanaman bakau, dengan tujuan mempermudah proses pendataan pohon *mangrove*.
3. Sistem informasi visualisasi data *mangrove* petangoran disarankan untuk meningkatkan desain *responsif* situs *website* agar antarmuka pengguna sistem dapat menyesuaikan dengan berbagai ukuran layar.
4. Sistem informasi visualisasi data *mangrove* petangoran disarankan terdapat pengembangan lebih lanjut yang menambahkan data yang ditampilkan seperti variasi tanaman, kesehatan tanaman, dan umur tanaman.

DAFTAR PUSTAKA

- [1] A. Jupri, “Inventarisasi Spesies Mangrove Di Teluk Kertasari, Sumbawa Barat Inventarization of Mangrove Species in Kertasari Bay, Western Sumbawa,” 2003. doi: <https://doi.org/10.24198/bjib.v5i1.317>.
- [2] A. Dennis, B. H. Wixom, and R. M. Roth, *Systems Analysis and Design, 6th Edition*. Wiley, 2014. [Online]. Available: <https://books.google.co.id/books?id=gu9EBQAAQBAJ>
- [3] Alfiansyah, W. Nurcahyo, and N. Faizah, “Aplikasi Sistem Informasi Personalial CV. Madya Mandiri Teknik Berbasis Web Dengan Metode Rapid Application Development (RAD),” *Jurnal Indonesia : Manajemen Informatika dan Komunikasi*, vol. 3, no. 2, pp. 74–85, Dec. 2022, doi: 10.35870/jimik.v3i2.89.
- [4] A. S. Rosa, *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika Bandung, 2015.
- [5] D. C. Foreman and D. Foreman, *Bootstrap 5 Foundations*. in The Random Knowledge Enthusiast. Independently published, 2021. [Online]. Available: <https://books.google.co.id/books?id=KN1tzgEACAAJ>
- [6] S. Nidhra, “Black Box and White Box Testing Techniques - A Literature Review,” *International Journal of Embedded Systems and Applications*, vol. 2, pp. 29–50, Jun. 2012, doi: 10.5121/ijesa.2012.2204.
- [7] Ben G. Weinstein, | Ethan P. White, Henry Senyondo, Gregoire Vincent, Mélaïne Aubry-Kientz, and Sergio Marconi, “DeepForest A Python package for RGB deep learning tree crown delineation,” *British Ecological Society*, vol. 11, no. 12, pp. 1743–1751, Aug. 2020, doi: <https://doi.org/10.1111/2041-210X.13472>.
- [8] J. Chan, R. Chung, and J. Huang, *Python API Development Fundamentals: Develop a full-stack web application with Python and Flask*. Packt Publishing Ltd, 2019.

- [9] U. Cubukcu, O. Erdogan, S. Pathak, S. Sannakkayala, and M. Slot, "Citrus: Distributed PostgreSQL for Data-Intensive Applications," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Association for Computing Machinery, Jun. 2021, pp. 2490–2502. doi: 10.1145/3448016.3457551.
- [10] Suarjaya Gede Dewa, Lanya Indayati, and Adi Ratna Putu Gusti i, "Aplikasi Remote Sensing dan Sig Untuk Pemetaan dan Informasi Sumberdaya Lahan Subak Sawah di Kecamatan Kuta dan Kuta Utara, Kabupaten Badung," vol. 6, no. 3, pp. 239–248, Jul. 2017, [Online]. Available: <https://ojs.unud.ac.id/index.php/JAT>
- [11] Mochammad Aldi Kushendriawan, Harry Budi Santoso, Panca O. Hadi Putra, and Martin Schrepp, "Evaluating User Experience of a Mobile Health Application 'Halodoc' using User Experience Questionnaire and Usability Testing," *Jurnal Sistem Informasi*, vol. 17, no. 1, pp. 58–71, Apr. 2021, doi: 10.21609/jsi.v17i1.1063.
- [12] M. A. Rafiq *et al.*, "Rancang Bangun Aplikasi Data Pengolahan Kelapa Sawit Pada PT. Perkebunan Nusantara VII Sungai Niru Berbasis Web," vol. 18, pp. 74–79, 2023.
- [13] T. Mukti, S. A. Hudjimartsu, and E. Hermawan, "Analisis Perhitungan Individu Pohon Mangrove Pada Citra Unmanned Aerial Vehicle(UAV) Menggunakan Metode Local Maxima," *INFOTECH journal*, vol. 9, no. 2, pp. 491–506, Aug. 2023, doi: 10.31949/infotech.v9i2.6480.
- [14] R. R. Putra, "Sistem Informasi Web Pariwisata Hutan Mangrove di Kelurahan Belawan Sicanang Kecamatan Medan Belawan Sebagai Media Promosi," 2019. doi: <https://doi.org/10.55583/arsy.v4i1.491>.
- [15] Ridho Darman, "Pembangunan Dashboard Lokasi Rawan Tanah Longsor di Indonesia Menggunakan Tableau," *Jurnal Teknik Informatika dan Sistem Informasi, Universitas Andalas*, vol. Volume 4 Nomor 2, no. 2, pp. 254–267, Aug. 2018, Accessed: Nov. 18, 2024. [Online]. Available: <https://journal.maranatha.edu/index.php/jutisi/article/view/1493>

- [16] A. L. Kalua, Pondaag S., and F. Mohune, “Sistem Informasi Pemetaan Objek Wisata Pesisir Kota Manado, Kota Bitung, dan Kab. Minahasa Utara Berbasis Web,” 2023.
- [17] “Tools User Experience Questionnaire .” Accessed: Oct. 28, 2024. [Online]. Available: <https://www.ueq-online.org/>
- [18] “User Experience Questionnaire Handbook,” 2024. Accessed: Oct. 27, 2024. [Online]. Available: <https://www.ueq-online.org/>