

**RANCANG BANGUN *BACKEND* SISTEM INFORMASI GEOGRAFIS
(SIG) BERBASIS *WEBSITE* UNTUK TATA KELOLA KEBUN SAWIT
MESUJI LAMPUNG**

(Skripsi)

Oleh

TYAS NAFARA ANDINI

NPM 2115061028



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG**

2026

**RANCANG BANGUN *BACKEND* SISTEM INFORMASI GEOGRAFIS
(SIG) BERBASIS *WEBSITE* UNTUK TATA KELOLA KEBUN SAWIT
MESUJI LAMPUNG**

Oleh

TYAS NAFARA ANDINI

Skripsi

Sebagai Salah Satu Syarat untuk Mencapai Gelar

SARJANA TEKNIK

Pada

Jurusan Teknik Elektro

Fakultas Teknik Universitas Lampung



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG**

2026

ABSTRAK

RANCANG BANGUN *BACKEND* SISTEM INFORMASI GEOGRAFIS (SIG) BERBASIS *WEBSITE* UNTUK TATA KELOLA KEBUN SAWIT MESUJI LAMPUNG

Oleh

Tyas Nafara Andini

Pemanfaatan teknologi informasi dalam pengelolaan kebun kelapa sawit menjadi kebutuhan penting untuk meningkatkan efisiensi, akurasi, dan integrasi data, khususnya dalam pengelolaan data spasial dan non-spasial. Pada praktiknya, pengelolaan kebun sawit di Kabupaten Mesuji, Lampung masih menghadapi kendala akibat penggunaan sistem manual dan semi-digital yang menyulitkan proses pemantauan kondisi kebun serta pengambilan keputusan. Penelitian ini bertujuan untuk merancang dan membangun *backend* Sistem Informasi Geografis (SIG) berbasis *website* sebagai solusi dalam tata kelola kebun sawit. *Backend* sistem dikembangkan menggunakan *framework Laravel* dengan basis data *PostgreSQL* dan menerapkan arsitektur *RESTful API* untuk menyediakan layanan pengelolaan data kebun sawit yang terstruktur, meliputi data blok, zona, lahan, irigasi, jalan, pohon, tenaga kerja, peralatan, dan transportasi. Metode *Rapid Application Development (RAD)* digunakan dalam pengembangan sistem guna mempercepat proses pembangunan melalui pendekatan iteratif. Pengujian sistem dilakukan pada sisi backend menggunakan metode *black box testing* untuk memastikan seluruh fungsi sistem berjalan sesuai dengan kebutuhan yang telah ditetapkan. Hasil penelitian menunjukkan bahwa *backend WebGIS* yang dikembangkan mampu mengelola data kebun sawit secara terintegrasi dan seluruh fitur backend berfungsi dengan baik berdasarkan hasil pengujian, sehingga sistem ini dapat mendukung proses digitalisasi pengelolaan kebun sawit serta membantu pengambilan keputusan berbasis data secara lebih efektif dan sistematis.

Kata kunci: Sistem Informasi Geografis, *Backend WebGIS*, Kelapa Sawit, *Laravel*, *PostgreSQL*, *RESTful API*.

ABSTRACT

DESIGN AND DEVELOPMENT OF A WEB-BASED GEOGRAPHIC INFORMATION SYSTEM (GIS) BACKEND FOR OIL PALM PLANTATION MANAGEMENT IN MESUJI LAMPUNG

By

Tyas Nafara Andini

The utilization of information technology in oil palm plantation management has become an essential requirement to improve efficiency, accuracy, and data integration, particularly in managing spatial and non-spatial data. In practice, oil palm plantation management in Mesuji Regency, Lampung, still faces challenges due to the use of manual and semi-digital systems, which complicate the monitoring of plantation conditions and decision-making processes. This research aims to design and develop a web-based Geographic Information System (GIS) backend as a solution for oil palm plantation management. The system backend is developed using the Laravel framework with a PostgreSQL database and implements a RESTful API architecture to provide structured data management services, including data on blocks, zones, land, irrigation, roads, trees, labor, equipment, and transportation. The Rapid Application Development (RAD) method is applied to accelerate the system development process through an iterative approach. System testing is conducted on the backend using the black box testing method to ensure that all system functions operate according to the defined requirements. The results indicate that the developed WebGIS backend is capable of managing oil palm plantation data in an integrated manner, and all backend features function properly based on the testing results. Therefore, this system is expected to support the digitalization of oil palm plantation management and assist data-driven decision-making more effectively and systematically.

Keywords: Geographic Information System, WebGIS Backend, Oil Palm Plantation, Laravel, PostgreSQL, RESTful API.

Judul Skripsi

: RANCANG BANGUN *BACKEND* SISTEM
INFORMASI GEOGRAFIS (SIG) BERBASIS
WEBSITE UNTUK TATA KELOLA KEBUN
SAWIT MESUJI LAMPUNG

Nama Mahasiswa

: *Tyas Nafara Andini*

Nomor Pokok Mahasiswa

: 2115061028

Program Studi

: S1 Teknik Informatika

Jurusan

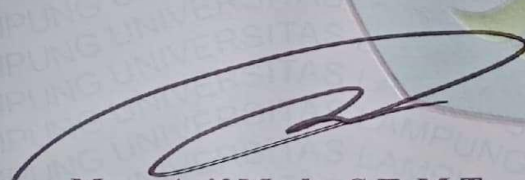
: Teknik Elektro Fakultas Teknik

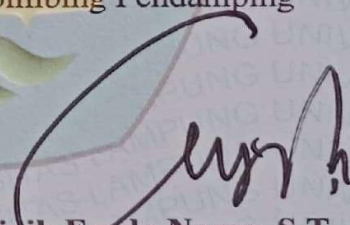
MENYETUJUI

1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping

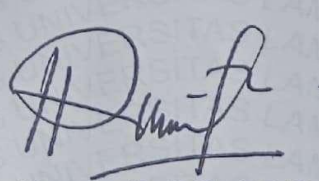

Mona Arif Muda, S.T., M.T.
NIP. 197111122000031002

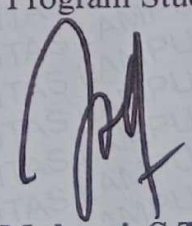

Ir. Gigih Forda Nama, S.T., M.T.I., IPM
NIP. 198307122008121003

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 : **Ir. Gigih Forda Nama, S.T., M.T.I., IPM**

Penguji : **Yessi Mulyani, S.T., M.T.**

2. Dekan Fakultas Teknik

Dr. H. Ahmad Herison, S.T., M.T.

NIP. 196910302000031001

Tanggal Lulus Ujian Skripsi : **28 Januari 2026**

SURAT PERNYATAAN

Saya yang bertanda tangan dibawah ini, menyatakan dengan sebenarnya bahwa skripsi saya yang berjudul “Rancang Bangun *Backend* Sistem Informasi Geografis (Sig) Berbasis *Website* Untuk Tata Kelola Kebun Sawit Mesuji Lampung” merupakan hasil karya saya sendiri. Semua hasil yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi saya ini merupakan salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, 13 Februari 2026

Pembuat Pernyataan



Tyas Nafara Andini

2115061028

RIWAYAT HIDUP



Tyas Nafara Andini lahir di Kota Metro, Provinsi Lampung, pada tanggal 15 September 2002. Penulis merupakan anak pertama dari dua bersaudara, putri dari pasangan Bapak Handi Kurniawan dan Ibu Apri Yanti.

Riwayat pendidikan dimulai dari Taman Kanak-Kanak (TK) Perwanida pada tahun 2009, kemudian dilanjutkan ke Sekolah Dasar (SD) Negeri 1 Metro dan diselesaikan pada tahun 2015. Pendidikan menengah pertama ditempuh di SMP Negeri 1 Metro dan lulus pada tahun 2018. Selanjutnya, pendidikan menengah atas diselesaikan di SMA Negeri 3 Metro pada tahun 2021. Pada tahun 2021, penulis diterima sebagai mahasiswa Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung melalui jalur SNMPTN (Seleksi Nasional Masuk Perguruan Tinggi Negeri). Selama masa perkuliahan, penulis aktif mengikuti berbagai kegiatan pengembangan diri dan organisasi. Penulis menjadi anggota Himpunan Mahasiswa Teknik Elektro (HIMATRO) pada periode 2021–2023, mengikuti program Merdeka Belajar Kampus Merdeka (MBKM) Studi Independen sebagai Web Fullstack Developer di PT Educa Sisfomedia Indonesia, serta menjalani Praktik Kerja Lapangan (PKL) di PT Angakasa Pura II KC Bandar Udara Radin Inten II pada periode Juni hingga Agustus 2024. Selain itu, penulis juga melaksanakan Kuliah Kerja Nyata (KKN) di Desa Banjar Ratu, Kecamatan Gunung Labuhan, Kabupaten Way Kanan, Provinsi Lampung.

MOTTO

“Allah tidak mengatakan hidup ini mudah. Tetapi Allah berjanji, bahwa
sesungguhnya bersama kesulitan ada kemudahan”

(Q.S Al-Insyirah: 5-6)

“Barang siapa menempuh jalan untuk mencari ilmu. Allah akan mudahkan
baginya jalan menuju surga”

(HR. Bukhari dan Muslim)

“Long Story Short, I Survived”

(Taylor Swift)

PERSEMBAHAN

Alhamdulillahirabbil‘alamin, segala puji bagi Allah SWT yang telah melimpahkan rahmat dan karunia-Nya sehingga skripsi ini dapat diselesaikan dengan baik. Shalawat serta salam semoga senantiasa tercurah kepada Nabi Muhammad SAW, teladan sepanjang masa.

Dengan tulus, kupersembahkan skripsi ini kepada

Kedua orang tuaku tercinta, Bapak Handi Kurniawan dan Ibu Apri Yanti, yang senantiasa melangitkan doa-doa terbaik sepanjang hidup mereka untukku. Terima kasih yang sebesar-besarnya atas segala kasih sayang, dukungan, dan pengorbanan yang tidak akan pernah bisa terbalaskan. Skripsi ini adalah bentuk kecil dari cinta dan rasa terima kasihku yang tak terhingga. Dan juga adikku tersayang, Muhammad Jeevan Qianno Andika, yang selalu menjadi penyemangat dan tidak pernah lupa memberikan ucapan selamat atas setiap pencapaianku. Semoga Allah SWT senantiasa melimpahkan berkah dan perlindungan-Nya kepada kita semua, Aamiin.

Rekan seperjuangan, seluruh dosen dan civitas akademik Jurusan Teknik Elektro, Universitas Lampung, terima kasih atas semangat, diskusi, ilmu, bimbingan, dan kebersamaan yang menjadikan perjalanan ini sangat bermakna.

Dan terakhir, untuk diriku sendiri, Tyas Nafara Andini, yang telah bertahan, berjuang, dan tak pernah berhenti percaya bahwa semuanya bisa diselesaikan.

SANWACANA

Alhamdulillahirabbil'alamin, puji syukur penulis panjatkan ke hadirat Allah SWT atas rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi berjudul “Rancang Bangun *Backend* Sistem Informasi Geografis (Sig) Berbasis *Website* Untuk Tata Kelola Kebun Sawit Mesuji Lampung”. Shalawat dan salam semoga tercurah kepada Nabi Muhammad SAW yang telah menjadi teladan dalam menuntut ilmu dan memperbaiki akhlak manusia. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Informatika, Fakultas Teknik, Universitas Lampung. Proses penyusunan skripsi ini merupakan perjalanan penuh tantangan dan pembelajaran. Keberhasilannya tak lepas dari bantuan, bimbingan, dan dukungan berbagai pihak. Dengan penuh rasa syukur, penulis mengucapkan terima kasih sebesar-besarnya kepada:

1. Ibu Herlinawati, S.T., M.T., selaku Ketua Jurusan Teknik Elektro Universitas Lampung.
2. Ibu Yessi Mulyani, S.T., M.T., selaku Ketua Program Studi Teknik Informatika, pembimbing akademik, sekaligus penguji, atas saran, masukan, dan ilmu yang telah diberikan demi perbaikan penelitian ini.
3. Bapak Mona Arif Muda, S.T.,M.T. selaku Pembimbing Utama, atas bimbingan, bantuan, dan arahan yang sangat berarti selama proses penelitian hingga terselesaikan dengan baik.
4. Bapak Ir. Gigih Forda Nama, S.T., M.T.I., IPM selaku Pembimbing Pendamping, atas masukan dan arahan yang diberikan dalam penyusunan penelitian ini.
5. Terutama dan paling utama kepada kedua orang tua, Papa dan Mama, serta adikku, atas doa, dukungan, semangat, dan bantuan materil yang tiada henti hingga skripsi ini dapat terselesaikan dengan baik.

6. Kepada diri sendiri, atas keteguhan dan usaha dalam menyelesaikan skripsi ini meskipun menghadapi berbagai rintangan.
7. Kepada rekan penelitian yang luar biasa: Reginia Putri Maharani, Azahra Nafisa, dan Vidya Adelia Anya, terima kasih telah berjalan bersama sejak awal penyusunan skripsi dengan segala lika-likunya.
8. Kepada sahabat-sahabat seperantauan yang sudah seperti keluarga: Hilda, Vania, Murti, Annisya, Pinka, Chandrika, Syifa, serta teman-teman seperjuangan lainnya yang tidak dapat disebutkan satu per satu.
9. Kepada teman-teman masa SMA: Humay, Choi, Jeki, dan Rifa yang selalu menjadi penghibur dan penyemangat di tengah berbagai tantangan.
10. Seluruh civitas akademik Jurusan Teknik Elektro Universitas Lampung, atas segala bantuan administratif dan dukungan selama proses penelitian.
11. Dan kepada semua pihak yang tidak dapat disebutkan satu per satu, terima kasih atas kontribusi, baik secara langsung maupun tidak langsung, dalam penyelesaian skripsi ini. Penelitian ini masih jauh dari kata sempurna, dikarenakan terbatasnya pengalaman dan pengetahuan yang dimiliki. Oleh karena itu, diharapkan segala bentuk saran serta masukan bahkan kritik yang membangun dari berbagai pihak. Semoga skripsi ini dapat bermanfaat bagi pada pembaca dan semua pihak khususnya dalam bidang pengembangan Backend Website.

Bandar Lampung, 25 Januari 2026

Penulis

Tyas Nafara Andini

DAFTAR ISI

DAFTAR ISI	xiii
DAFTAR GAMBAR	xvi
DAFTAR TABEL	xxiv
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian	3
1.4.1 Manfaat bagi Mahasiswa.....	3
1.4.2 Manfaat Bagi Lembaga dan Masyarakat.....	4
1.4.3 Manfaat Bagi Universitas Lampung	4
1.5 Batasan Masalah.....	4
1.6 Sistematika Penulisan	4
II. TINJAUAN PUSTAKA	6
2.1 Dasar Teori	6
2.1.1 Kelapa Sawit	6
2.1.2 Sistem Informasi Geografis.....	7
2.1.3 <i>Website</i>	8
2.1.4 <i>Backend</i>	8
2.1.5 <i>Framework Laravel</i>	9
2.1.6 <i>PostgreSQL</i>	9

2.1.7	<i>RESTful API</i>	10
2.1.8	<i>Unified Modelling Language (UML)</i>	11
2.1.9	<i>Postman</i>	11
2.1.10	<i>Black Box Testing</i>	12
2.1.11	<i>Metode RAD (Rapid Application Development)</i>	13
2.1.13	<i>Hosting</i>	15
2.2	Penelitian Terkait	16
III.	METODOLOGI PENELITIAN	44
3.1	Waktu dan Tempat Penelitian	44
3.2	<i>Capstone Project</i>	45
3.3	Alat dan Bahan Penelitian	46
3.3.1	Alat	46
3.3.2	Bahan Penelitian	47
3.4	Tahapan Penelitian	47
3.4.1	Pengumpulan Data	49
3.4.2	Pengembangan Sistem	50
3.4.3	Penulisan Laporan	58
IV.	HASIL DAN PEMBAHASAN	59
4.1	Rancangan Awal	59
4.1.1	<i>User Design</i>	59
4.1.2	<i>Construction</i>	76
4.2	Iterasi 1	181
4.2.1	<i>User Design</i>	181
4.2.2	<i>Constraction</i>	182
4.3	Performa <i>Backend</i> Sistem	195
4.4	<i>Cutover</i>	197

V. KESIMPULAN DAN SARAN	203
5.1 Kesimpulan.....	203
5.2 Saran.....	204
DAFTAR PUSTAKA.....	205
LAMPIRAN.....	210

DAFTAR GAMBAR

Gambar 2.1 Metode <i>Rapid Application Development</i> (RAD)	14
Gambar 2.2 Hasil Penelitian Terkait <i>Code</i> Menampilkan Data <i>User</i>	17
Gambar 2.3 Hasil Penelitian Terkait Pengujian Mendapatkan Data <i>User</i>	17
Gambar 2.4 Hasil Penelitian Terkait Admin <i>Dashboard</i>	18
Gambar 2.5 Hasil Penelitian Terkait Hasil Pengujian Menggunakan <i>Postman</i>	18
Gambar 2.6 Hasil Penelitian Terkait Implementasi Tampilan Halaman Grafik....	19
Gambar 2.7 Hasil Penelitian Terkait Implementasi Tampilan Halaman GIS.....	20
Gambar 2.8 Hasil Penelitian Terkait Tampilan Halaman <i>Dashboard</i>	21
Gambar 2.9 Hasil Penelitian Terkait Pengujian <i>Blackbox</i>	21
Gambar 2.10 Hasil Penelitian Terkait <i>Source Code File Api.php</i>	22
Gambar 2.11 Hasil Penelitian Terkait Pengujian CRUD Kelas	22
Gambar 2.12 Hasil Penelitian Terkait <i>Form</i> Peta Sebaran.....	23
Gambar 2.13 Hasil Penelitian Terkait <i>Form</i> Data Desa.....	24
Gambar 2.14 Hasil Penelitian Terkait Implementasi	25
Gambar 2.15 Hasil Penelitian Terkait Hasil Pengujian.....	25
Gambar 2.16 Hasil Penelitian Terkait Perancangan Arsitektur <i>Geospatial</i> API... 26	
Gambar 2.17 Hasil Penelitian Terkait <i>Layer Overlay</i> antara Peta Desa	27
Gambar 2.18 Hasil Penelitian Terkait Respon <i>JSON GET</i> Kapal.....	28
Gambar 2.19 Hasil Penelitian Terkait Menu View Data Kapal	28
Gambar 2.20 Hasil Penelitian Terkait Pengiriman Request Ke <i>Web Service</i>	29
Gambar 2.21 Hasil Penelitian Terkait Halaman <i>Dashboard</i> Paket Wisata	29
Gambar 2.22 Hasil Penelitian Terkait <i>Distribution Map Page</i>	31
Gambar 2.23 Hasil Penelitian Terkait Maps Page Design	32
Gambar 2.24 Hasil Penelitian Terkait <i>Home page</i>	33
Gambar 2.25 Hasil Penelitian Terkait Data <i>Maintenance Page</i>	33
Gambar 2. 26 Hasil Penelitian Terkait <i>Implement Of The Scale Web GIS</i>	34

Gambar 2.27 Hasil Penelitian Terkait Halaman Admin	36
Gambar 2.28 Hasil Penelitian Terkait Peta Sebaran Pasien <i>Covid-19</i> (Web)	36
Gambar 2.29 Hasil Penelitian Terkait <i>A Suitability Analysis Was Conducted</i>	37
Gambar 2. 30 <i>Decision Support System Integrated With The AgriMIS</i>	38
Gambar 2. 31 Hasil Penelitian Terkait <i>Mapping Page</i>	39
Gambar 2.32 Hasil Penelitian Terkait <i>Node Data page</i>	40
Gambar 2.33 Hasil Penelitian Terkait <i>System Testing Page</i>	40
Gambar 2.34 Hasil Penelitian Terkait <i>Web-Based CPD</i>	42
Gambar 2.35 Hasil Penelitian Terkait <i>My Location Page</i>	43
Gambar 2.36 Hasil Penelitian Terkait <i>Admin Dashboard Page</i>	43
Gambar 3.1 <i>Capstone Project</i>	45
Gambar 3.2 Tahapan Penelitian	48
Gambar 4.1 Arsitektur Perangkat Lunak.....	59
Gambar 4.2 Use Case Diagram Staff Kebun	61
Gambar 4.3 Use Case Diagram Manajer Kebun.....	62
Gambar 4.4 Activity Diagram Login	63
Gambar 4.5 Activity Diagram Staf Kebun.....	64
Gambar 4.6 Activity Diagram Melihat Peta.....	65
Gambar 4.7 Activity Diagram Tenaga Kerja dan Peralatan	66
Gambar 4.8 Activity Diagram Distribusi dan Produksi	67
Gambar 4.9 Activity Diagram Data Lahan	68
Gambar 4.10 Activity Diagram Data Irigasi	69
Gambar 4. 11 Activity Diagram Data Pohon	70
Gambar 4.12 Activity Diagram Data Jalan	71
Gambar 4.13 Activity Diagram Data Alat.....	72
Gambar 4.14 Activity Diagram Data Transportasi.....	73
Gambar 4.15 Activity Diagram Data Pekerja	74
Gambar 4.16 Activity Diagram Manajemen Pengguna	75
Gambar 4. 17 Konfigurasi .env	77
Gambar 4. 18 Struktur Direktori Backend	78
Gambar 4.19 Route Authetication.....	80
Gambar 4.20 AuthController.....	81

Gambar 4.21 API Login	82
Gambar 4.22 API Logout	83
Gambar 4.23 Route Menampilkan Data Upload Peta	83
Gambar 4.24 Controller Menampilkan Data Upload Peta	84
Gambar 4.25 API Menampilkan Data Upload Peta	85
Gambar 4.26 Route Menambahkan Data Upload Peta	86
Gambar 4.27 Controller Menambahkan Data Upload Peta	86
Gambar 4.28 API Menambahkan Data Upload Peta.....	87
Gambar 4.29 Route Mengedit Data Upload Peta.....	88
Gambar 4.30 Controller Mengedit Data Upload Peta.....	89
Gambar 4.31 API Mengedit Data Upload Peta	90
Gambar 4.32 Route Menghapus Data Upload Peta	91
Gambar 4.33 Controller Menghapus Data Upload Peta.....	91
Gambar 4.34 API Menghapus Data Upload Peta.....	92
Gambar 4.35 Route Menampilkan Data Blok.....	93
Gambar 4.36 Controller Menampilakan Data Blok	93
Gambar 4.37 API Menampilkan Data Blok	94
Gambar 4.38 Route Menambahkan Data Blok	95
Gambar 4.39 Controller Menambahkan Data Blok	95
Gambar 4.40 API Menambahkan Data Blok.....	97
Gambar 4.41 Route Mengedit Data Blok.....	97
Gambar 4.42 Controller Mengedit Data Blok.....	98
Gambar 4.43 API Mengedit Data Blok	99
Gambar 4.44 Route Menghapus Data Blok	99
Gambar 4.45 Controller Menghapus Data Blok	100
Gambar 4.46 API Menghapus Data Blok.....	100
Gambar 4.47 Route Menampilkan Data Zone	101
Gambar 4.48 Controller Menampilkan Data Zona	101
Gambar 4.49 API Menampilkan Data Zona.....	102
Gambar 4.50 Route Menambahkan Data Zona.....	103
Gambar 4.51 Controller Menambahkan Data Zona.....	104
Gambar 4.52 API Menambahkan Data Zona	105

Gambar 4.53 Route Mengedit Data Zona	105
Gambar 4.54 Controller Mengedit Data Zona	106
Gambar 4.55 API Mengedit Data Zona.....	107
Gambar 4.56 Route Menghapus Data zona.....	108
Gambar 4.57 Controller Menghapus Data zona.....	108
Gambar 4.58 API Menghapus Data zona.....	109
Gambar 4.59 Route Menampilkan Data Lahan.....	109
Gambar 4.60 Controller Menampilkan Data Lahan.....	110
Gambar 4.61 API Menampilkan Data Lahan	111
Gambar 4.62 Route Menambahkan Data Lahan	111
Gambar 4.63 Controller Menambahkan Data Lahan	112
Gambar 4.64 API Menambahkan Data Lahan	113
Gambar 4.65 Route Mengedit Data Lahan	114
Gambar 4.66 Controller Mengedit Data Lahan.....	114
Gambar 4.67 API Mengedit Data Lahan.....	115
Gambar 4.68 Route Menghapus Data Lahan	116
Gambar 4.69 Controller Menghapus Data Lahan	116
Gambar 4.70 API Menghapus Data Lahan	117
Gambar 4.71 Route Menampilkan Data Irigasi	117
Gambar 4.72 Controller Menampilkan Data Irigasi	118
Gambar 4.73 API Menampilkan Data Irigasi.....	119
Gambar 4.74 Route Menambahkan Data Irigasi.....	119
Gambar 4.75 Controller Menambahkan Data Irigasi.....	120
Gambar 4.76 API Menambahkan Data Irigasi	121
Gambar 4.77 Route Mengedit Data Irigasi	122
Gambar 4.78 Controller Mengedit Data Irigasi	122
Gambar 4.79 API Mengedit Data Irigasi.....	123
Gambar 4.80 Route Menghapus Data Irigasi.....	124
Gambar 4.81 Controller Menghapus Data Irigasi	124
Gambar 4.82 API Menghapus Data Irigasi	125
Gambar 4. 83 Route Menampilkan Data Pohon	125
Gambar 4. 84 Controller Menampilkan Data Pohon	126

Gambar 4. 85 API Menampilkan Data Pohon.....	127
Gambar 4. 86 Route Menghitung Data Pohon	127
Gambar 4. 87 Controller Menghitung Data Pohon	128
Gambar 4. 88 API Menghitung Data Pohon	129
Gambar 4. 89 Route Menambahkan Data Pohon.....	129
Gambar 4. 90 Controller Menambahkan Data Pohon.....	130
Gambar 4. 91 API Menambahkan Data Pohon	131
Gambar 4. 92 Route Mengedit Data Pohon	131
Gambar 4. 93 Controller Mengedit Data Pohon	132
Gambar 4. 94 API Mengedit Data Pohon.....	133
Gambar 4. 95 Route Menghapus Data Pohon.....	134
Gambar 4. 96 Controller Menghapus Data Pohon	134
Gambar 4. 97 API Menghapus Data Pohon	135
Gambar 4.98 Route Menampilkan Data Jalan	135
Gambar 4.99 Controller Menampilkan Data Jalan	136
Gambar 4.100 API Menampilkan Data Jalan.....	137
Gambar 4.101 Route Menambahkan Data Jalan.....	138
Gambar 4.102 Controller Menambahkan Data Jalan.....	138
Gambar 4.103 API Menambahkan Data Jalan	139
Gambar 4.104 Route Mengedit Data Jalan	140
Gambar 4.105 Controller Mengedit Data Jalan	141
Gambar 4.106 API Mengedit Data Jalan.....	142
Gambar 4.107 Route Menghapus Data Jalan.....	142
Gambar 4.108 Controller Menghapus Data Jalan.....	143
Gambar 4.109 API Menghapus Data Jalan	143
Gambar 4.110 Route Menampilkan Data Alat	144
Gambar 4.111 Controller Menampilkan Data Alat	144
Gambar 4.112 API Menampilkan Data Alat	145
Gambar 4.113 Route Menambahkan Data Alat	146
Gambar 4.114 Controller Menambahkan Data Alat.....	146
Gambar 4.115 Menambahkan Data Alat	147
Gambar 4.116 Route Mengedit Data Alat.....	148

Gambar 4.117 Controller Mengedit Data Alat	148
Gambar 4.118 API Mengedit Data Alat	149
Gambar 4.119 Route Menghapus Data Alat.....	150
Gambar 4.120 Controller Menghapus Data Alat.....	150
Gambar 4.121 API Menghapus Data Alat.....	151
Gambar 4.122 Route Menampilkan Data Transportasi.....	151
Gambar 4.123 Controller Menampilkan Data Transportasi	152
Gambar 4.124 API Menampilkan Data Transportasi	152
Gambar 4.125 Route Menambahkan Data Transportasi	153
Gambar 4. 126 Controller Menambahkan Data Transportasi	153
Gambar 4.127 API Menambahkan Data Transportasi.....	154
Gambar 4.128 Route Mengedit Data Transportasi.....	155
Gambar 4.129 Controller Mengedit Data Transportasi.....	155
Gambar 4.130 API Mengedit Data Transportasi	156
Gambar 4.131 Route Menghapus Data Transportasi	156
Gambar 4.132 Controller Menghapus Data Transportasi	157
Gambar 4.133 API Menghapus Data Transportasi.....	158
Gambar 4.134 Route Menampilkan Data Pekerja.....	158
Gambar 4.135 Controller Menampilkan Data Pekerja.....	159
Gambar 4.136 API Menampilkan Data Pekerja	159
Gambar 4.137 Route Menambahkan Data Pekerja	160
Gambar 4.138 Controller Menambahkan Data Pekerja	160
Gambar 4.139 API Menambahkan Data Pekerja	161
Gambar 4.140 Route Mengedit Data Pekerja	162
Gambar 4.141 Controller Mengedit Data Pekerja.....	162
Gambar 4.142 API Mengedit Data Pekerja.....	163
Gambar 4.143 Route Menghapus Data Pekerja	164
Gambar 4.144 Controller Menghapus Data Pekerja	164
Gambar 4.145 API Menghapus Data Pekerja	165
Gambar 4.146 Route Menampilkan Manajemen Pengguna	166
Gambar 4.147 Controller Menampilkan Manajemen Pengguna.....	166
Gambar 4.148 API Menampilkan Manajemen Pengguna.....	167

Gambar 4.149 Route Menambahkan Manajemen Pengguna	167
Gambar 4.150 Controller Menambahkan Manajemen Pengguna	168
Gambar 4.151 API Menambahkan Manajemen Pengguna	169
Gambar 4.152 Route Mengedit Manajemen Pengguna	169
Gambar 4.153 Controller Mengedit Manajemen Pengguna	170
Gambar 4.154 API Mengedit Manajemen Pengguna.....	171
Gambar 4.155 Route Menghapus Manajemen Pengguna	171
Gambar 4.156 Controller Menghapus Manajemen Pengguna	172
Gambar 4.157 API Menghapus Manajemen Pengguna	173
Gambar 4.158 Activity Diagram Melihat Peta.....	181
Gambar 4.159 Route Menampilkan Data Blok Berdasarkan Tanggal	182
Gambar 4.160 Controller Menampilkan Data Blok Berdasarkan Tanggal	183
Gambar 4.161 API Menampilkan Data Blok Berdasarkan Tanggal	184
Gambar 4.162 Route Menampilkan Data Zona Berdasarkan Tanggal.....	185
Gambar 4. 163 Controller Menampilkan Data Zona Berdasarkan Tanggal.....	185
Gambar 4.164 API Menampilkan Data Zona Berdasarkan Tanggal	186
Gambar 4.165 Route Menampilkan Data Irigasi Berdasarkan Tanggal.....	187
Gambar 4.166 Controller Menampilkan Data Irigasi Berdasarkan Tanggal.....	188
Gambar 4. 167 API Menampilkan Data Irigasi Berdasarkan Tanggal	189
Gambar 4.168 Route Menampilkan Data Jalan Berdasarkan Tanggal.....	190
Gambar 4.169 Controller Menampilkan Data Jalan Berdasarkan Tanggal.....	190
Gambar 4.170 API Menampilkan Data Jalan Berdasarkan Tanggal	191
Gambar 4. 171 Route Menampilkan Data Pohon Berdasarkan Zona	192
Gambar 4. 172 Controller Menampilkan Data Pohon Berdasarkan Zona	192
Gambar 4. 173 API Menampilkan Data Pohon Berdasarkan Zona	193
Gambar 4.174 Instalasi Stack Teknologi.....	200
Gambar 4.175 Clone Repositori GitHub.....	200
Gambar 4.176 Instalasi Dependensi.....	200
Gambar 4.177 Konfigurasi Nginx.....	201
Gambar 4.178 Aktivasi Konfigurasi	202
Gambar 4.179 Optimisasi Laravel	202
Gambar 4.180 Manajemen Permission	202

Gambar 6. 1 Kegiatan Observasi Kebun Sawit Mesuji Tim Riset URO	213
Gambar 6. 2 Kegiatan Pemetaan Kebun Sawit Tim Riset URO	214

DAFTAR TABEL

Tabel 3.1 Waktu Pelaksanaan Penelitian	44
Tabel 3.2 Alat Penelitian	46
Tabel 3.3 <i>User Story</i> dan <i>Acceptance Criteria</i>	50
Tabel 3.4 Kebutuhan Fungsional.....	54
Tabel 3.5 Kebutuhan <i>Non</i> Fungsional	55
Tabel 4.1 <i>Use Case Diagram</i>	60
Tabel 4.2 <i>Black Box Testing</i>	174
Tabel 4.3 <i>Black Box Testing</i> Iterasi 2	194
Tabel 4.4 Hasil Pengujian Performa <i>Backend</i> Sistem.....	197
Tabel 4.5 Spesifikasi <i>VPS CloudAja</i>	198
Tabel 4.6 Konfigurasi <i>DNS</i>	199

I. PENDAHULUAN

1.1 Latar Belakang

Kelapa sawit merupakan salah satu komoditas unggulan Indonesia yang memiliki peran strategis dalam perekonomian nasional. Menurut data dari Pusat Data dan Sistem Informasi Pertanian Sekretariat Jenderal Kementerian Pertanian tahun 2022, ekspor kelapa sawit Indonesia mencapai *USD* 31,49 miliar dengan volume 36,52 juta ton. Tidak hanya berkontribusi terhadap devisa negara, sektor ini juga menyerap banyak tenaga kerja, terutama di daerah-daerah yang mengandalkan perkebunan sebagai penggerak ekonomi lokal, seperti Kabupaten Mesuji, Provinsi Lampung.

Namun, besarnya potensi tersebut belum sepenuhnya diiringi dengan pengelolaan yang efisien. Banyak pengelola kebun sawit masih menggunakan pencatatan manual atau semi-digital seperti buku catatan dan *file spreadsheet*. Metode ini tidak hanya menyita waktu, tetapi juga rawan kesalahan, duplikasi data, serta menyulitkan dalam pemantauan dan analisis data ketika skala kebun semakin luas. Ketidaksesuaian data laporan dengan kondisi di lapangan menjadi salah satu kendala utama dalam pengambilan keputusan yang cepat dan tepat.

Salah satu penyebab permasalahan ini adalah minimnya adopsi teknologi informasi yang mendukung sistem pengelolaan kebun secara *real-time* dan terintegrasi. Keterbatasan dalam integrasi data, terutama data spasial dan *non-spasial*, menyebabkan pengelola kesulitan memantau kondisi kebun secara menyeluruh. Selain itu, masih rendahnya pemanfaatan Sistem Informasi Geografis (SIG) juga turut membatasi efisiensi dalam pemetaan dan pengelolaan lahan sawit.

Jika permasalahan ini tidak segera diatasi, maka produktivitas, efisiensi, dan akurasi manajemen kebun sawit akan terus menurun. Kondisi ini dapat menghambat perkembangan sektor perkebunan dan menurunkan daya saing komoditas kelapa sawit di pasar global.

Adapun solusi yang dapat dilakukan adalah pengembangan Sistem Informasi Geografis (SIG) berbasis *website* yang memungkinkan integrasi antara data spasial—seperti data pohon, data zona, data lahan, dan jaringan irigasi—dengan data *non*-spasial seperti data pekerja, data peralatan kebun, dan informasi transportasi kebun. Dalam sistem ini, peran *backend website* sangat krusial sebagai pusat logika bisnis dan pemrosesan data, yang menyediakan layanan *API* bagi pengguna untuk mengakses dan mengelola informasi secara efisien melalui antarmuka *web*.

Dengan mengembangkan *backend website* yang andal, maka proses pengelolaan kebun sawit dapat dilakukan secara lebih terstruktur, *real-time*, dan dapat diakses oleh berbagai pihak yang terlibat. *Backend* juga menjadi fondasi penting untuk memastikan sistem dapat dikembangkan lebih lanjut dan terintegrasi dengan sistem atau antarmuka *frontend* di masa mendatang.

Oleh karena itu, penelitian ini difokuskan pada perancangan dan pengembangan *backend website* untuk Sistem Informasi Geografis (SIG) kebun sawit, dengan studi kasus di Kabupaten Mesuji, Lampung. Penelitian ini diharapkan dapat memberikan kontribusi terhadap upaya digitalisasi pengelolaan kebun sawit, khususnya dalam meningkatkan efisiensi, transparansi, dan ketepatan pengambilan keputusan berbasis data.

1.2 Rumusan Masalah

Berdasarkan permasalahan di atas, adapun rumusan masalah yang diangkat dalam penelitian ini adalah:

1. Bagaimana perancangan *backend webgis* sistem informasi geografis kebun sawit mesuji?
2. Bagaimana cara memastikan bahwa fitur-fitur yang dikembangkan dalam *backend WebGIS* telah berjalan sesuai dengan kebutuhan melalui proses pengujian yang tepat?

1.3 Tujuan Penelitian

Adapun tujuan penelitian dari penelitian ini adalah:

1. Merancang *backend WebGIS* untuk sistem informasi geografis kebun sawit di Mesuji yang mampu mengelola data spasial dan non-spasial secara terstruktur.
2. Melakukan pengujian terhadap fitur-fitur pada *backend WebGIS* menggunakan metode *black box testing* guna memastikan bahwa sistem telah berjalan sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan.

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah:

1.4.1 Manfaat bagi Mahasiswa

- a. Sebagai bentuk penerapan ilmu pengetahuan dan keterampilan praktis yang telah diperoleh selama mengikuti perkuliahan.
- b. Mengasah kreativitas dan kemampuan inovasi serta *skill* mahasiswa sehingga nantinya siap dalam menghadapi persaingan di dunia kerja.
- c. Memperluas wawasan dan pengalaman dalam merancang suatu karya khususnya dalam bidang teknologi.

1.4.2 Manfaat Bagi Lembaga dan Masyarakat

- a. Membantu pihak pengelola kebun sawit dalam mengintegrasikan dan mengelola data kebun sawit secara optimal dan terstruktur.
- b. Mendukung pengambilan keputusan berbasis data dalam pengelolaan kebun sawit untuk meningkatkan produktivitas dan kinerja operasional.
- c. Berkontribusi pada pengelolaan kebun sawit yang lebih berkelanjutan, sehingga mendukung kesejahteraan masyarakat di sekitar kebun sawit.

1.4.3 Manfaat Bagi Universitas Lampung

Sebagai sarana referensi di perpustakaan Universitas Lampung mengenai permasalahan yang terkait dengan penulisan Tugas Akhir ini.

1.5 Batasan Masalah

Agar tidak meluas dari maksud dan tujuan penelitian ini, maka permasalahannya dibatasi sebagai berikut:

1. Penelitian ini difokuskan pada pengembangan *backend WebGIS* yang berperan dalam penyediaan layanan *API RESTful* untuk keperluan integrasi.
2. Pengujian sistem dilakukan pada sisi *backend* menggunakan metode *blackbox testing* untuk memastikan fungsionalitas sesuai dengan kebutuhan dan spesifikasi yang telah dirancang.

1.6 Sistematika Penulisan

Untuk mempermudah penyusunan dan pemahaman isi skripsi yang berjudul "Rancang Bangun *Backend* Sistem Informasi Geografis Dalam Tata Kelola Kebun Sawit (Studi Kasus: Kebun Sawit Mesuji Lampung)", sistematika penulisan disusun sebagai berikut:

BAB I : PENDAHULUAN

Bab ini menguraikan latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan

BAB II : TINJAUAN PUSTAKA

Bab ini membahas teori-teori yang relevan mengenai kelapa sawit, sistem informasi geografis (GIS), *Website*, *backend*, *Laravel*, *PostgreSQL*, *RESTful API*, metode pengujian *blackbox testing*, *Unified Modelling Language (UML)*, dan metode *RAD*.

BAB II : METODOLOGI PENELITIAN

Bab ini menguraikan metode penelitian yang digunakan, mulai dari waktu dan tempat penelitian, alat dan bahan yang digunakan serta alur penelitiannya

BAB IV : HASIL DAN PEMBAHASAN

Bab ini memuat hasil penelitian dan analisisnya yang mencakup implementasi backend sistem informasi geografis kebun sawit, fitur – fitur dan *API* yang dikembangkan, Hasil pengujian menggunakan *blackbox testing*.

BAB V : PENUTUP

Bab ini menyajikan kesimpulan dari penelitian yang dilakukan serta saran yang diberikan berdasarkan hasil analisis.

II. TINJAUAN PUSTAKA

2.1 Dasar Teori

2.1.1 Kelapa Sawit

Kelapa sawit (*Elaeis guineensis* Jacq) merupakan tanaman tropis asal Afrika Barat yang kini menjadi komoditas utama di Indonesia. Tanaman ini memiliki batang tunggal yang tumbuh tegak, dapat mencapai ketinggian 20–30 meter, dan memiliki masa produktif antara 25 hingga 30 tahun. Daunnya tersusun secara menyirip dengan warna hijau tua, dan pada usia muda batangnya masih tertutup oleh pelepah. Buah kelapa sawit berwarna merah dengan tiga lapisan utama, yaitu *eksokarp* (kulit luar), *mesokarp* (daging buah), dan *endokarp* (lapisan keras pelindung biji), di mana bagian dalamnya mengandung minyak nabati yang sangat bernilai secara ekonomi. Terdapat tiga varietas utama kelapa sawit, yaitu *Dura*, *Pisifera*, dan *Tenera*. Varietas *Tenera*, hasil persilangan antara *Dura* dan *Pisifera*, paling umum dibudidayakan karena memiliki kandungan minyak tinggi dan cangkang tipis, sehingga lebih efisien dalam produksi minyak [1].

Industri kelapa sawit memainkan peran penting dalam perekonomian nasional. Berdasarkan data dari Direktorat Jenderal Perkebunan (Ditjenbun) tahun 2022, sektor ini menyumbangkan sekitar 3,5% terhadap Produk Domestik Bruto (PDB) dan mencakup 13,5% dari total ekspor nonmigas Indonesia. Industri ini juga menerima lebih dari 16 juta tenaga kerja secara langsung maupun tidak langsung, menjadikannya sebagai salah satu fondasi dalam pembangunan ekonomi [2].

PT Sampoerna Agro Tbk merupakan perusahaan publik Indonesia yang bergerak di bidang perkebunan dan pengolahan hasil kelapa sawit, serta komoditas lain seperti sagu dan karet. Didirikan pada Juni 1993 dengan nama PT Selapan Jaya, perusahaan

ini mulai menjalankan kegiatan komersial pada November 1998, dan berganti nama menjadi PT Sampoerna Agro Tbk pada tahun 2007. Lokasi perkebunan perusahaan tersebar di berbagai daerah di Indonesia, termasuk Sumatera Selatan, Kalimantan Barat dan Tengah, serta Riau. Selain memproduksi benih kelapa sawit unggul jenis DxP Sriwijaya, perusahaan ini juga bermitra dengan petani lokal melalui skema kemitraan plasma.

2.1.2 Sistem Informasi Geografis

Sistem Informasi Geografis (SIG) merupakan suatu sistem komputer yang dirancang untuk menangkap, menyimpan, memeriksa, mengintegrasikan, memanipulasi, menganalisis, dan menampilkan data yang berkaitan dengan posisi geografis di permukaan bumi. SIG tidak hanya berfungsi sebagai alat untuk membuat peta digital, tetapi lebih jauh lagi sebagai sistem analisis yang mampu mengolah data spasial guna mendukung proses pengambilan keputusan yang berbasis lokasi. Melalui integrasi antara data spasial (lokasi) dan atribut non-spasial (informasi deskriptif), SIG memungkinkan pengguna untuk mengidentifikasi pola, memahami hubungan antar fenomena geografis, serta menganalisis perubahan dalam ruang dan waktu. Selain itu, SIG memiliki peran strategis dalam berbagai bidang seperti perencanaan wilayah, pengelolaan sumber daya alam, pemetaan infrastruktur, mitigasi bencana, hingga pengawasan lingkungan [3].

Sistem Informasi Geografis (SIG) merupakan jenis sistem informasi yang menyajikan data secara visual melalui peta sebagai antarmukanya. SIG dibangun berdasarkan konsep lapisan (*layer*) serta hubungan antar data. Pertama kali dikembangkan pada tahun 1960, SIG awalnya bertujuan untuk menangani permasalahan yang berkaitan dengan geografi. Namun, setelah empat dekade, fungsinya meluas ke berbagai sektor, seperti pemantauan penyakit menular seperti demam berdarah, analisis kriminalitas seperti kerusuhan, hingga bidang pariwisata [4].

2.1.3 Website

Website atau *Web* didefinisikan sebagai sebuah sistem universal yang memungkinkan informasi dihubungkan dan diakses secara bebas melalui jaringan komputer. *Web* dirancang untuk menghubungkan berbagai jenis data dalam bentuk *hypertext*, sehingga pengguna dapat menjelajah dari satu informasi ke informasi lain hanya dengan satu klik. *Web* bukan hanya sebagai alat untuk penelitian ilmiah atau perdagangan elektronik, tetapi sebagai medium global untuk berbagi ide, kreativitas, dan pengetahuan manusia [5].

Halaman *website* merupakan dokumen yang disusun menggunakan format *HTML* (*Hyper Text Markup Language*) dan umumnya dapat diakses melalui protokol *HTTP*. Protokol ini berfungsi untuk mengirim informasi dari server situs *web* ke pengguna melalui peramban (*web browser*). Halaman *web* dapat bersifat statis maupun dinamis, dan biasanya menjadi bagian dari struktur yang saling terhubung melalui jaringan tautan (*hyperlink*) [6].

2.1.4 Backend

Backend merupakan bagian dari pengembangan *web* yang bertanggung jawab untuk mengelola logika aplikasi, pemrosesan data, server, serta komunikasi dengan *database*. *Backend* berbeda dengan *frontend*, karena *backend* bekerja di sisi server dan tidak berinteraksi langsung dengan pengguna. Tugas utama *backend* adalah menerima permintaan dari *frontend*, memprosesnya (misalnya mengambil, menyimpan, atau memanipulasi data dalam *database*), lalu mengirimkan hasilnya kembali ke *frontend* [7].

Pengembangan *backend* dapat dilakukan menggunakan berbagai bahasa pemrograman, seperti *Java*, *Python*, *PHP*, dan *JavaScript*. Manajemen basis data dan server merupakan bagian penting dalam *backend* sebuah aplikasi. Basis data digunakan untuk menyimpan data dan terdiri dari dua jenis, yaitu basis data relasional dan non-relasional. Basis data non-relasional menyimpan data dalam format yang fleksibel, sedangkan basis data relasional menyimpan data dalam

bentuk tabel. Server *web* menyimpan kode dan memproses aplikasi. Server menangani komunikasi antara pengguna dan aplikasi melalui *HTTP*, yang merupakan singkatan dari *HyperText Transfer Protocol* [8].

2.1.5 *Framework Laravel*

Laravel adalah *framework PHP* modern yang dapat memudahkan pengembangan aplikasi *web* dengan struktur yang bersih dan mudah dipelihara. *Framework* ini menerapkan prinsip desain *SOLID*, *dependency injection*, dan *IoC container* untuk mengelola dependensi secara efisien. *Laravel* juga menggunakan *service providers* untuk mengatur layanan sistem secara optimal. Dengan pemisahan tanggung jawab antar lapisan aplikasi, *Laravel* mendukung pengembangan sistem yang fleksibel, scalable, dan tahan terhadap perubahan [9].

Laravel adalah *framework PHP* berbasis *MVC* yang mempermudah pengembangan aplikasi *web*. *Framework* ini menyajikan fitur seperti *Artisan*, *Blade*, dan *Eloquent ORM* untuk mempercepat proses *coding* dan pengelolaan data. *Laravel* dipilih karena mampu menghasilkan sistem yang efisien, mudah dipelihara, dan responsif, sehingga cocok digunakan oleh pemula maupun pengembang berpengalaman [10].

2.1.6 *PostgreSQL*

PostgreSQL adalah sistem manajemen basis data relasional berbasis objek yang bersifat *open source*, dikembangkan dari proyek *POSTGRES* di *UC Berkeley*. Sistem ini dirancang fleksibel, kuat, dan dapat diperluas, serta mendukung fitur lanjutan seperti tipe data kompleks, *inheritance*, *rules*, dan *trigger*. *PostgreSQL* mematuhi standar *SQL* dan mendukung ekstensi, menjadikannya ideal untuk aplikasi modern. Fitur unggulannya termasuk *multiversion concurrency control (MVCC)* untuk mendukung transaksi simultan tanpa konflik, serta sistem transaksi yang andal dengan *rollback* dan *recovery*. *PostgreSQL* cocok digunakan untuk berbagai skala aplikasi, dari kecil hingga enterprise [11].

PostgreSQL mampu bersaing dengan berbagai vendor *database* relasional ternama seperti *Oracle*, *MySQL*, dan *SQL Server*. Sistem ini telah digunakan secara luas di berbagai sektor, mulai dari lembaga pemerintahan, instansi publik, hingga perusahaan swasta. Sebagai *DBMS* lintas platform, *PostgreSQL* dapat dijalankan di berbagai sistem operasi [12].

2.1.7 *RESTful API*

RESTful API (*Representational State Transfer Application Programming Interface*) merupakan sebuah antarmuka layanan *web* yang dirancang berdasarkan prinsip-prinsip arsitektur *REST*, sebagaimana diperkenalkan pertama kali oleh Roy Fielding dalam disertasinya pada tahun 2000. *RESTful API* memungkinkan komunikasi antara sistem melalui protokol *HTTP* dengan pendekatan berbasis *resource*, di mana setiap *resource* diidentifikasi secara unik menggunakan *Uniform Resource Locator* (*URL*). Operasi terhadap *resource* dilakukan menggunakan metode standar *HTTP*, yaitu *GET* untuk mengambil data, *POST* untuk membuat data baru, *PUT* untuk memperbarui data yang ada, dan *DELETE* untuk menghapus data. *RESTful API* bersifat *stateless*, artinya setiap *request* dari klien harus membawa seluruh informasi yang diperlukan untuk memproses permintaan tersebut tanpa bergantung pada sesi yang disimpan di server. Representasi data yang dipertukarkan antara klien dan server biasanya menggunakan format ringan seperti *JSON* (*JavaScript Object Notation*) atau *XML* (*Extensible Markup Language*), yang memungkinkan pertukaran data secara efisien. Selain itu, *RESTful API* mengandalkan penggunaan kode status *HTTP* untuk memberikan informasi mengenai hasil dari setiap permintaan, seperti 200 OK untuk permintaan yang berhasil, 201 *Created* untuk *resource* yang berhasil dibuat, 400 *Bad Request* untuk permintaan yang salah, serta 404 *Not Found* ketika *resource* yang diminta tidak ditemukan. Dengan karakteristik sederhana, konsisten, dan skalabel, *RESTful API* menjadi salah satu standar utama dalam pengembangan sistem terdistribusi modern, karena dapat memudahkan integrasi antar platform dan memungkinkan pengembangan aplikasi yang fleksibel, modular, serta mudah dikembangkan lebih lanjut [13].

2.1.8 *Unified Modelling Language (UML)*

Unified Modeling Language (UML) adalah sebuah bahasa pemodelan standar yang digunakan untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan artefak-artefak dalam sistem perangkat lunak. *UML* berfungsi sebagai alat bantu komunikasi yang menggambarkan desain sistem dari berbagai sudut pandang, dengan tujuan untuk meningkatkan pemahaman, konsistensi, dan keteraturan dalam pengembangan sistem berbasis objek. Bahasa ini tidak hanya berguna dalam tahap perancangan, tetapi juga mendukung seluruh siklus hidup pengembangan perangkat lunak, mulai dari analisis kebutuhan hingga implementasi dan pemeliharaan [14].

UML memperkenalkan seperangkat elemen dasar, seperti *actor*, *class*, *interface*, *component*, dan *node*, yang masing-masing mewakili komponen penting dalam sistem. Elemen-elemen ini digambarkan melalui diagram-diagram standar untuk memperjelas hubungan antar bagian sistem. Secara keseluruhan, *UML* menyediakan berbagai jenis diagram yang dikelompokkan menjadi tiga kategori utama: diagram struktur (*structural diagrams*), diagram perilaku (*behavioral diagrams*), dan diagram interaksi (*interaction diagrams*). Diagram struktur, seperti diagram kelas dan diagram objek, menggambarkan aspek statis dari sistem. Diagram perilaku, seperti diagram aktivitas dan diagram *use case*, menunjukkan aspek dinamis dan alur kerja dari sistem. Sedangkan diagram interaksi, seperti diagram urutan dan diagram komunikasi, fokus pada bagaimana objek dalam sistem saling berinteraksi melalui pesan [15].

2.1.9 *Postman*

Postman merupakan alat yang sangat berguna dalam pengujian dan pengembangan *API*, karena memungkinkan pengguna untuk mengeksplorasi dan memahami cara kerja *API* secara menyeluruh. Dengan *Postman*, proses pengujian *API* menjadi lebih sistematis dan efisien, baik untuk keperluan eksploratif maupun untuk otomatisasi. Dalam konteks pengujian perangkat lunak, *Postman* tidak hanya membantu mengirim permintaan *HTTP*, tetapi juga menyediakan berbagai fitur

lanjutan seperti *mock server*, pengujian keamanan, dan pengujian performa. Pengguna dapat membuat skenario pengujian lengkap dengan parameter, *header*, dan *body*, serta memanfaatkan skrip *JavaScript* ringan untuk memvalidasi respons *API* secara otomatis [16].

Postman merupakan sebuah platform yang digunakan untuk kolaborasi dalam pengembangan *API*. Aplikasi ini dikembangkan oleh Abhinav Asthana, seorang *programmer* dan *desainer* asal Bangalore, India. *Postman* dirancang untuk mempermudah proses pengujian, pengembangan, serta pendokumentasian *API*. Dengan antarmuka yang sederhana, *Postman* memungkinkan pengujian *API* dilakukan secara efisien dan cepat. Mekanisme kerjanya didasarkan pada pengelompokan permintaan (*request*) berdasarkan metode *request*, *URL*, serta parameter-parameter yang digunakan dalam permintaan tersebut [17].

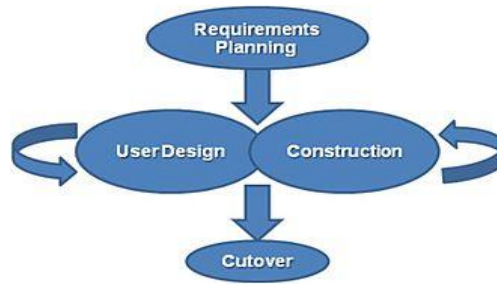
2.1.10 Black Box Testing

Black box testing adalah salah satu metode pengujian perangkat lunak yang berfokus pada evaluasi fungsi perangkat lunak berdasarkan spesifikasi yang diberikan, tanpa mempertimbangkan struktur internal atau logika implementasi program. Dalam pendekatan ini, penguji memperlakukan program seperti sebuah "kotak hitam", di mana yang diamati hanya input yang diberikan dan *output* yang dihasilkan, tanpa pengetahuan tentang bagaimana proses di dalam sistem berlangsung. Tujuan utama *black box testing* adalah untuk menemukan kesalahan atau ketidaksesuaian antara *output* aktual dan *output* yang diharapkan berdasarkan spesifikasi fungsi sistem. Metode ini melibatkan perancangan data uji berdasarkan semua kemungkinan kondisi input, termasuk input valid, input tidak valid, kondisi batas (*boundary conditions*), serta berbagai skenario penggunaan nyata. Meskipun dalam teori ideal diperlukan pengujian semua kemungkinan input untuk menemukan seluruh kesalahan, dalam praktiknya hal tersebut tidak memungkinkan karena jumlah kombinasi input yang tidak terbatas. Oleh karena itu, strategi yang diterapkan dalam *black box testing* adalah memilih subset data uji yang representatif dan berpotensi tinggi untuk menemukan kesalahan dalam jumlah

pengujian yang wajar. Pendekatan ini juga menegaskan pentingnya merancang kasus uji yang tidak hanya mengonfirmasi bahwa sistem bekerja sesuai harapan, tetapi juga mengungkap kemungkinan perilaku sistem yang tidak sesuai. Dengan demikian, *black box testing* menjadi alat yang efektif dalam meningkatkan keandalan perangkat lunak dan memastikan kesesuaian fungsional sistem terhadap kebutuhan pengguna tanpa bergantung pada pemahaman detail teknis implementasi [18].

2.1.11 Metode RAD (*Rapid Application Development*)

Rapid Application Development (RAD) adalah sebuah pendekatan siklus hidup pengembangan aplikasi yang dirancang untuk mempercepat pembuatan sistem sekaligus meningkatkan kualitas hasilnya dibandingkan metode tradisional. Pendekatan ini memanfaatkan kekuatan alat bantu pengembangan modern, seperti *CASE tools* dan *code generators*, untuk mempercepat berbagai tahap analisis, desain, hingga implementasi. RAD berfokus pada keterlibatan intensif pengguna dalam seluruh proses pengembangan, penggunaan prototipe untuk mengurangi risiko kesalahan spesifikasi, serta pengembangan sistem secara iteratif sehingga memungkinkan penyesuaian berkelanjutan. Dengan penggunaan tim kecil yang terlatih dan manajemen proyek yang ketat, *RAD* mampu menghasilkan sistem dengan biaya lebih rendah, waktu pengembangan lebih singkat, dan kualitas yang lebih tinggi. Selain itu, *RAD* mendorong pembangunan infrastruktur informasi yang terintegrasi, di mana sistem baru dikembangkan dengan mempertimbangkan keterhubungan dan kesinambungan data di seluruh organisasi. Pendekatan ini sangat efektif dalam dunia bisnis yang dinamis, di mana kecepatan merespons perubahan pasar menjadi faktor kunci keberhasilan.



Gambar 2.1 Metode *Rapid Application Development (RAD)*

Siklus hidup *Rapid Application Development (RAD)* terdiri dari empat fase utama. Pertama, *Requirements Planning Phase* di mana kebutuhan bisnis diidentifikasi melalui workshop intensif bersama pengguna. Kedua, *User Design Phase*, melibatkan pengguna dan pengembang dalam mendesain model proses, data, dan antarmuka sistem menggunakan *prototyping* dan *CASE tools*. Ketiga, *Construction Phase*, yaitu tahap pembangunan sistem secara cepat dengan bantuan otomatisasi, sambil tetap menerima umpan balik pengguna. Terakhir, *Cutover Phase*, meliputi pengujian akhir, pelatihan pengguna, konversi data, dan peluncuran sistem ke produksi. Keempat fase ini menekankan keterlibatan aktif pengguna dan pengembangan iteratif untuk menghasilkan sistem berkualitas tinggi dalam waktu singkat [19].

2.1.12 Nginx

Nginx merupakan salah satu perangkat lunak *web server open-source* yang banyak digunakan untuk mengelola lalu lintas jaringan pada aplikasi berbasis web. *Nginx* memiliki arsitektur berbasis *event-driven* dan *asynchronous* yang memungkinkan perangkat lunak ini menangani banyak koneksi secara simultan dengan penggunaan memori yang efisien. Pendekatan tersebut menjadikannya unggul dalam hal kecepatan, skalabilitas, dan kestabilan pada sistem dengan permintaan tinggi. Selain berfungsi sebagai web server konvensional, *Nginx* juga dapat difungsikan sebagai *reverse proxy*, *cache HTTP*, dan *load balancer*, sehingga mampu mempercepat respon permintaan pengguna dan mengurangi beban kerja *server backend*. Dalam implementasinya, *Nginx* menyediakan kemampuan untuk mengelola lalu-lintas *inbound* dan *outbound*, mendukung enkripsi *SSL/TLS* untuk

menjamin keamanan komunikasi antara klien dan *server*, serta fitur pengaturan ulang *URL (rewrite engine)* dan kontrol akses (*access control*) berdasarkan alamat *IP* atau kredensial tertentu guna meningkatkan keamanan sistem.[20]

Selanjutnya, dalam konteks konfigurasi tingkat lanjut dan integrasi arsitektur aplikasi, *Nginx* menawarkan fleksibilitas tinggi untuk penyesuaian modul pihak ketiga, pengaturan *logging*, serta pengoptimalan performa melalui teknik seperti kompresi *respons*, *caching* lanjutan, dan pengaturan *worker process* sesuai kemampuan *hardware*. Sistem ini dapat bekerja dengan berbagai *framework* aplikasi (misalnya *PHP-FastCGI*) dan juga berfungsi sebagai *mail proxy* atau aplikasi pengantara (*proxy*) untuk skala besar. Melalui modul dan skrip yang tepat, pengguna dapat melakukan debug konfigurasi, memecahkan masalah umum seperti limitasi koneksi dan *file descriptor*, serta melakukan tuning trafik agar sistem responsif dan stabil dalam kondisi beban kerja besar.[21]

2.1.13 Hosting

Web hosting adalah layanan yang memungkinkan individu atau organisasi untuk menyimpan situs *web* mereka di server yang dapat diakses melalui internet. Layanan ini memastikan situs web tetap online dan dapat diakses oleh pengunjung di seluruh dunia. Ada beberapa jenis layanan *hosting*, termasuk *shared hosting*, di mana beberapa pengguna berbagi sumber daya server yang sama; *VPS hosting*, yang menawarkan lebih banyak kontrol dan sumber daya server pribadi; *dedicated hosting*, di mana satu server sepenuhnya didedikasikan untuk satu pengguna; dan *cloud hosting*, yang memungkinkan skalabilitas dan fleksibilitas lebih besar dengan memanfaatkan beberapa server. Dalam memilih layanan *hosting*, ada beberapa faktor penting yang perlu dipertimbangkan, seperti keandalan, keamanan, kecepatan, dan dukungan pelanggan. Nama domain juga berperan penting karena bekerja bersama dengan *hosting* untuk memastikan situs web dapat diakses melalui nama domain yang mudah diingat. Pemilihan penyedia *hosting* yang tepat dan pengelolaan situs web yang efisien, mulai dari instalasi aplikasi dan sistem

manajemen konten (*CMS*) hingga pengaturan *email* dan *database*, sangat penting untuk keberhasilan sebuah situs web [22].

2.2 Penelitian Terkait

Dalam penyusunan skripsi ini, sebagian terinspirasi dari penelitian – penelitian sebelumnya. Beberapa penelitian yang relevan mencakup pengembangan Sistem Informasi Geografis (SIG) berbasis *web*, pengelolaan data spasial menggunakan *PostgreSQL* dan *PostGIS*, serta implementasi *backend* menggunakan *framework Laravel*. Hal tersebut dilakukan untuk mendukung pelaksanaan penelitian ini dan menghindari plagiarisme. Berikut ini akan dijabarkan penelitian-penelitian terkait sebelumnya.

2.2.1 Pengembangan Backend Sistem Kearsipan Dokumen menggunakan *Framework Laravel* di CV Nakula Sadewa

Penelitian ini dilakukan oleh Arief Hidayatullah dan Dianni Yusuf pada tahun 2024. Penelitian ini bertujuan untuk mengatasi permasalahan pengelolaan arsip konvensional di CV Nakula Sadewa, yang selama ini masih menggunakan penyimpanan sederhana seperti *Google Drive*. Penelitian ini menggunakan metode pengembangan *Rapid Application Development (RAD)* untuk mempercepat pembuatan sistem. *Backend* sistem dikembangkan menggunakan *framework Laravel*. *Database* yang digunakan berbasis *MySQL*. Proses pengujian dilakukan melalui unit testing menggunakan *tools Postman*, dengan fokus pada pengujian fungsi-fungsi *API* secara individual.

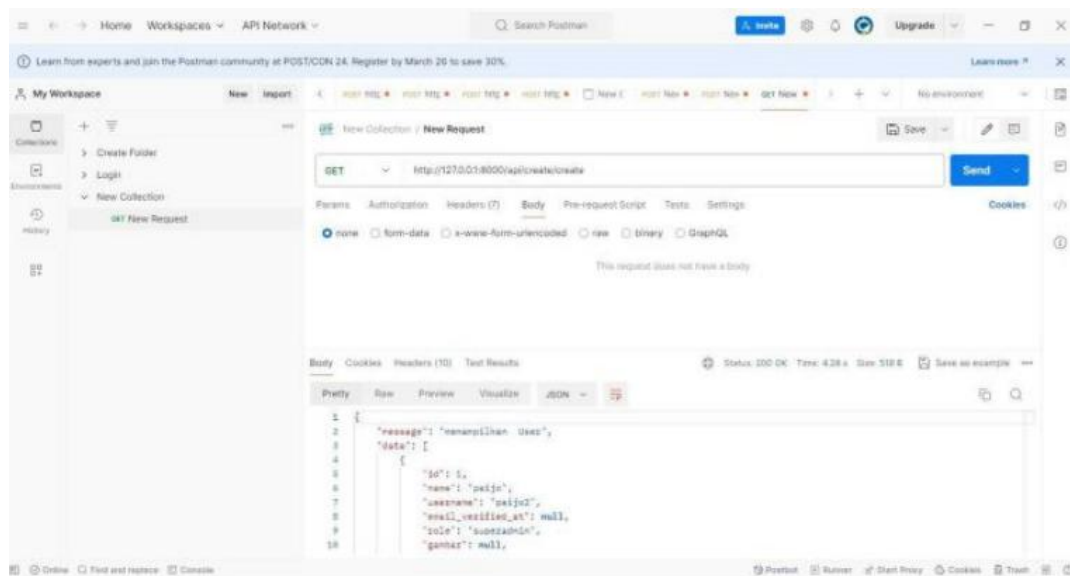
Hasil dari penelitian ini menunjukkan bahwa seluruh fungsi backend, seperti pengelolaan data pengguna, *folder*, *file*, *reset password*, dan *upload file*, dapat berjalan dengan baik dan menghasilkan respons 200 OK saat diuji menggunakan *Postman*. Implementasi *backend* yang dilakukan mencakup proses pembuatan, pembacaan, pembaruan, dan penghapusan data (*CRUD*), serta pengelolaan hak akses pengguna. Sistem yang dikembangkan dinilai berhasil meningkatkan efisiensi pengelolaan dokumen di perusahaan. [23]

```

1 public function index(Request $request){
2     $keyword = $request->input('name');
3
4     $users = $keyword
5     ? User::where('name', 'LIKE', '%'. $keyword. '%')->get()
6     : User::all();
7     return response()->json([
8         'message' => 'menampilkan User',
9         'data' => $users
10    ]);
11 }

```

Gambar 2.2 Hasil Penelitian Terkait *Code* Menampilkan Data *User* dan Pencarian Data *User*



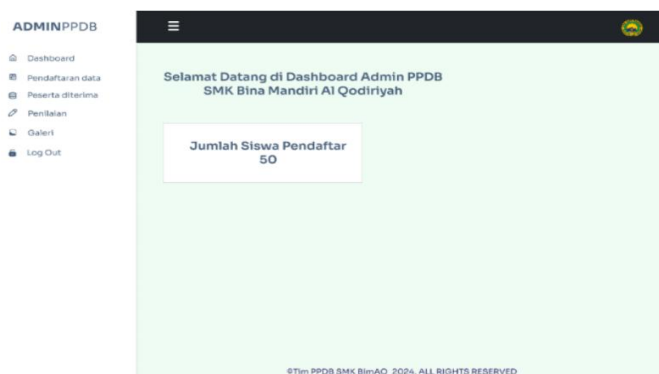
Gambar 2.3 Hasil Penelitian Terkait Pengujian Mendapatkan Data *User*

2.2.2 Perancangan *Backend* Pada Sistem Informasi Penerimaan Peserta Didik Baru (PPDB) SMK Bina Mandiri Al Qodiriyah Srono

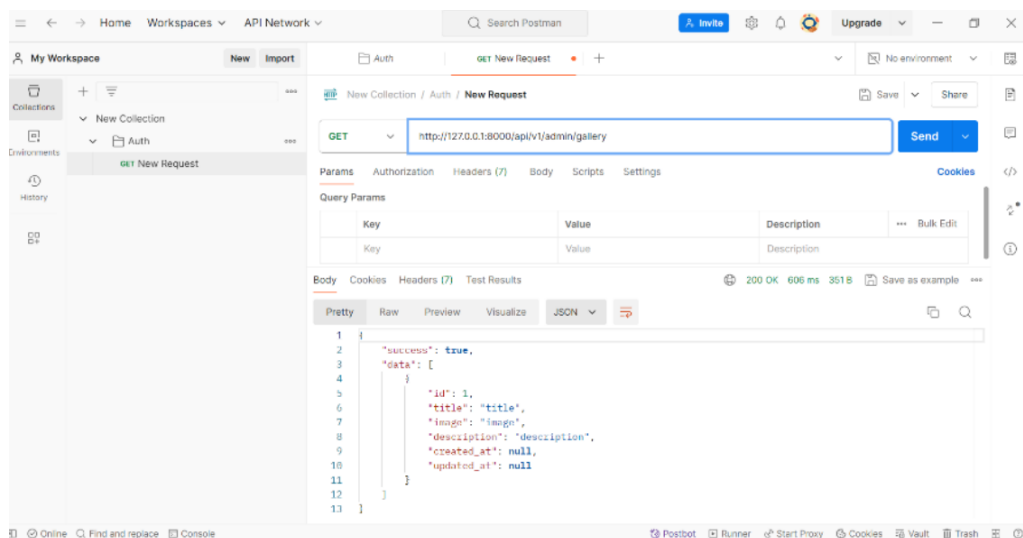
Penelitian ini dilakukan oleh Wandha Dese Fitriani, Alfin Hidayat, dan I Wayan Suardinata pada tahun 2024. Penelitian ini bertujuan untuk menggantikan sistem pendaftaran peserta didik baru yang sebelumnya dilakukan secara manual menggunakan buku besar, dengan sistem informasi berbasis web yang lebih efektif dan efisien. Metode pengembangan yang digunakan dalam penelitian ini adalah *Rapid Application Development (RAD)*, karena dinilai mampu mempercepat proses pembuatan sistem dengan pendekatan berbasis iterasi dan *prototyping*. *Backend* sistem ini dibangun menggunakan *framework Laravel 9* dengan bahasa pemrograman *PHP*, sedangkan *MySQL* digunakan sebagai basis data. Perancangan

sistem dilakukan dengan pendekatan *UML*, termasuk pembuatan *Use Case Diagram*, *Activity Diagram*, serta *Entity Relationship Diagram (ERD)* untuk menggambarkan relasi antar data dalam *database*.

Penelitian ini menghasilkan sistem informasi PPDB berbasis web yang mendukung proses pendaftaran calon peserta didik secara *online*, verifikasi data, input nilai seleksi, dan pengumuman hasil seleksi. Aplikasi memungkinkan calon siswa untuk mendaftar, mengunggah berkas, serta melihat hasil seleksi daring. Di sisi admin, sistem memudahkan manajemen data siswa, termasuk pengelolaan data pendaftar, nilai ujian, dan validasi berkas. Fitur *backend* telah diuji dan dinyatakan berjalan baik, siap digunakan untuk mendukung PPDB di SMK Bina Mandiri Al Qodiriyah.[24]



Gambar 2.4 Hasil Penelitian Terkait Admin *Dashboard*

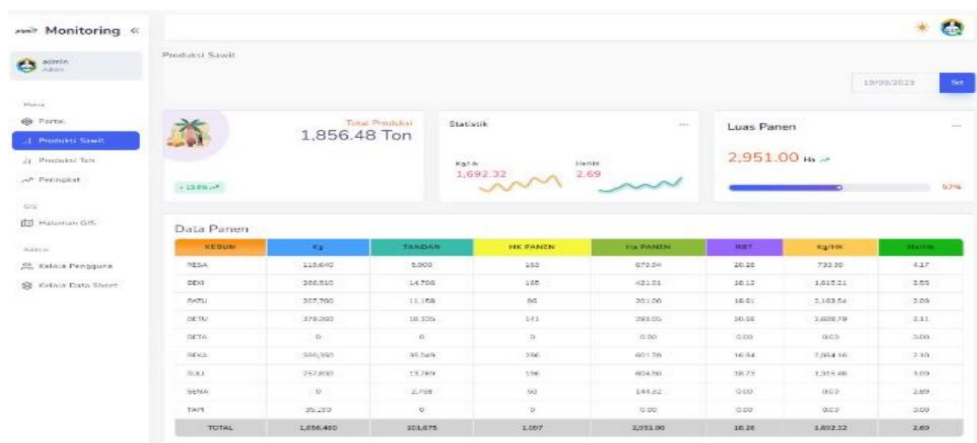


Gambar 2.5 Hasil Penelitian Terkait Hasil Pengujian Menggunakan *Postman*

2.2.3 Sistem Monitoring Produksi Menggunakan *Laravel* Dan *CorkBootstrap*

Penelitian ini dilakukan oleh Abim Ari Bimantara dan Rakhmat Dedi Gunawan pada tahun 2023. Penelitian ini bertujuan untuk membangun sistem monitoring produksi berbasis web di PT Perkebunan Nusantara VII (PTPN 7) untuk mengatasi permasalahan pencatatan produksi yang sebelumnya dilakukan secara manual melalui file PDF. Dalam pengembangan sistem, peneliti menggunakan metode *Extreme Programming (XP)*, yang mencakup tahap perencanaan, desain, pengkodean, dan pengujian. Alat dan teknologi yang digunakan meliputi *framework Laravel* untuk *backend*, template *Cork-Bootstrap 5* untuk tampilan antarmuka, *phpMyAdmin* untuk manajemen *database*, dan *Visual Studio Code* sebagai *text editor*. Sistem ini dilengkapi dengan fitur Web GIS untuk menampilkan data spasial lokasi kebun sawit dan teh. Untuk pengujian kualitas sistem, metode yang digunakan adalah ISO 25010 yang menilai aspek *functionality*, *reliability*, *security*, *performance efficiency*, dan *operability*.

Hasil penelitian menunjukkan bahwa sistem monitoring produksi yang dikembangkan mampu mempercepat dan mempermudah pengolahan data produksi di PTPN 7. Sistem ini memungkinkan pengguna untuk mengakses data produksi secara dinamis dalam bentuk grafik dan tabel, serta memetakan informasi geografis kebun. Pengujian menggunakan standar ISO 25010 menghasilkan skor rata-rata 88,05%, yang menunjukkan bahwa sistem memenuhi kriteria kualitas dengan kategori "sangat layak". Dengan demikian, sistem ini dinilai mampu meningkatkan efisiensi pemantauan produksi dan memberikan pengalaman pengguna yang lebih baik dibandingkan metode manual sebelumnya. [25]



Gambar 2.6 Hasil Penelitian Implementasi Tampilan Halaman Grafik Sawit

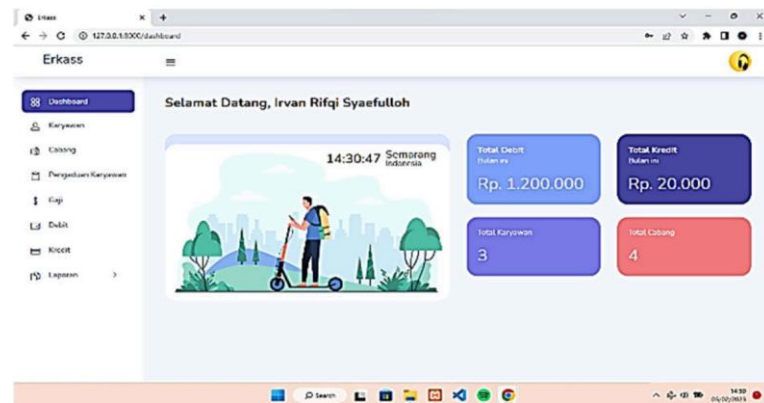


Gambar 2.7 Hasil Penelitian Terkait Implementasi Tampilan Halaman GIS

2.2.4 Pembuatan *Backend* Sistem Informasi Kepegawaian pada PT Erka Service Solusindo untuk Pengguna Admin Berbasis Website Menggunakan Kerangka Kerja *Laravel*

Penelitian ini dilakukan oleh Irvan Rifqi Syaefulloh, Adnan Fauzi, dan Ilmam Fauzi Hashbil Alim dari Universitas Diponegoro dan diterbitkan dalam Jurnal Teknik Komputer pada tahun 2023. Penelitian ini menggunakan metode pengembangan *Waterfall* dalam membangun sistem. Alat dan teknologi yang digunakan meliputi *framework Laravel* untuk pengembangan *backend*, *database MySQL* untuk manajemen data, serta arsitektur pemrograman *MVC (Model-View-Controller)* sebagai pola pengembangan aplikasi. Untuk pengujian, metode yang digunakan adalah *black-box testing*, di mana pengujian dilakukan dengan memberikan berbagai input untuk mengevaluasi apakah keluaran yang dihasilkan sesuai dengan fungsionalitas yang diharapkan.

Hasil penelitian ini adalah terciptanya sistem informasi kepegawaian berbasis web yang mencakup pengelolaan data pegawai, data keuangan, data kehadiran, dan data gaji. Sistem ini berhasil diimplementasikan dengan semua fitur berfungsi sesuai harapan berdasarkan hasil pengujian. Sistem ini memudahkan admin PT Erka Service Solusindo dalam pengelolaan data kepegawaian, menggantikan sistem manual berbasis *Microsoft Excel* yang sebelumnya digunakan.[26]



Gambar 2.8 Hasil Penelitian Terkait Tampilan Halaman *Dashboard*

No	Skenario pengujian	Kesimpulan
1	Pengguna membuka menu Karyawan pada <i>sidebar</i>	Sesuai
2	Pengguna menekan tombol tambah dan kemudian mengisi data karyawan yang baru, kemudian pengguna menekan tombol simpan	Sesuai
3	Pengguna menekan tombol detail pada halaman karyawan	Sesuai
4	Pengguna menekan tombol edit pada halaman karyawan	Sesuai
5	Pengguna menekan tombol hapus pada halaman karyawan	Sesuai

Gambar 2.9 Hasil Penelitian Terkait Pengujian *Blackbox*

2.2.5 Penerapan *REST API Laravel* sebagai Fondasi *Back-end* Aplikasi *G-MOOC 4D*

Penelitian berjudul ini dilakukan oleh Iman Nurjaman, Fandy Setyo Utomo, dan Nandang Hermanto dari Universitas Amikom Purwokerto, dan diterbitkan dalam *Journal of Informatics and Interactive Technology (JIITE)* pada bulan April 2024. Penelitian ini menggunakan metode pengembangan perangkat lunak *Waterfall*, yang terdiri dari tahap analisis kebutuhan, desain, implementasi, dan pengujian. Dalam pengembangan *backend* aplikasi *G-MOOC 4D*, digunakan *framework Laravel* dengan arsitektur *REST API* untuk pertukaran data yang aman dan efisien dalam format *JSON*. Basis data yang digunakan adalah *MySQL*, dan pengujian dilakukan dengan metode *Black Box* menggunakan aplikasi *Postman*.

Hasil dari penelitian ini adalah berhasil dibangunnya *REST API* yang terdiri dari sejumlah *endpoint* yang memungkinkan tim *frontend* untuk berinteraksi dengan *backend* secara efisien dan aman. Semua *endpoint* yang dikembangkan berhasil

diuji dengan hasil yang sesuai harapan, sehingga mendukung integrasi sistem *G-MOOC 4D* dalam menyediakan akses pembelajaran inklusif untuk penyandang disabilitas.[27]

```
Route::controller(KelasController::class)->group(function () {
    Route::get('kelas', 'index');
    Route::get('search/kelas', 'search');
    Route::get('kelas/name/{name}', 'showByName');

    Route::middleware('auth.admin')->group(function () {
        Route::post('kelas/create', 'store');
        Route::get('kelas/{id_kelas}', 'show');
        Route::post('kelas/update/{id_kelas}', 'update');
        Route::delete('kelas/delete/{id_kelas}', 'destroy');
    });
});
```

Gambar 2.10 Hasil Penelitian Terkait *Source Code File Api.php*

No.	Endpoint	Skenario pengujian	Hasil yang diharapkan	Hasil pengamatan	keterangan
1.	/api/kelas/create	Mengirimkan permintaan POST dengan bearer token, name, image, id_level, dan description.	Mendapatkan respons 201 Created dengan pesan "Kelas berhasil dibuat".	Sesuai dengan yang diharapkan	Diterima
2.	/api/kelas/{id_kelas}	Mengirimkan permintaan GET dengan bearer token dan id_kelas.	Mendapatkan respons 200 OK dengan data kelas yang sesuai dengan ID yang diminta.	Sesuai dengan yang diharapkan	Diterima
3.	/api/kelas/update/{id_kelas}	Mengirimkan permintaan POST dengan bearer token, id_kelas, name, image, id_level, dan description.	Mendapatkan respons 201 Created dengan pesan "Kelas berhasil diupdate".	Sesuai dengan yang diharapkan	Diterima
4.	/api/kelas/delete/{id_kelas}	Mengirimkan permintaan DELETE dengan bearer token dan id_kelas.	Mendapatkan respons 200 OK dengan pesan "Kelas berhasil dihapus".	Sesuai dengan yang diharapkan	Diterima

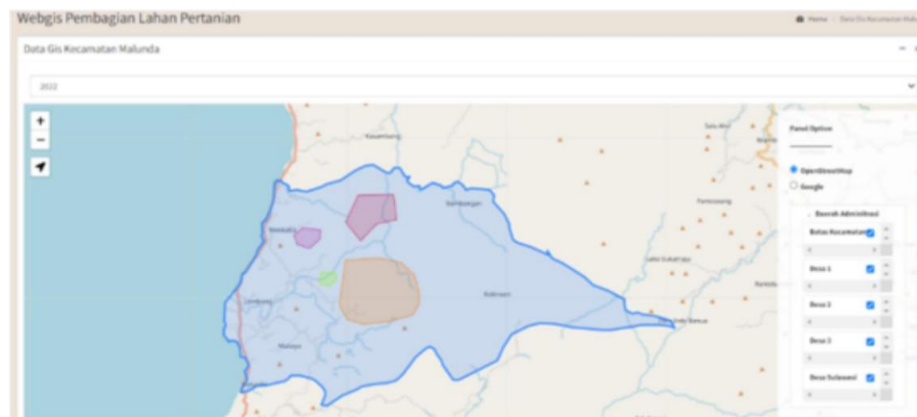
Gambar 2.11 Hasil Penelitian Terkait Pengujian *CRUD* Kelas

2.2.6 Sistem Informasi Lahan Pertanian Kabupaten Polewali Mandar Berbasis GIS

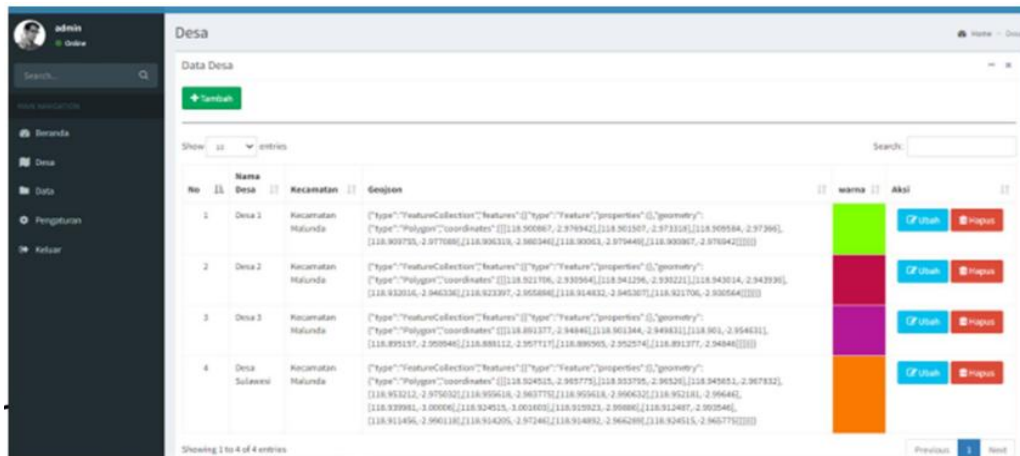
Penelitian berjudul "Sistem Informasi Lahan Pertanian Kabupaten Polewali Mandar Berbasis GIS" disusun oleh Fatmawati, Akhmad Qashlim, dan Ashabul Kahpi dari Universitas Al Asyariah Mandar, dan dipublikasikan dalam Jurnal Peqguruang: Seri Konferensi pada bulan Mei 2023. Tujuan utama penelitian ini adalah untuk merancang dan membangun sistem informasi geografis guna memetakan sebaran hasil pertanian di Kecamatan Polewali, sehingga informasi terkait lokasi lahan dan jenis tanaman pangan dapat disampaikan secara lebih cepat, akurat, dan mudah

diakses. Data dalam penelitian ini dikumpulkan melalui wawancara, observasi lapangan, dan studi pustaka. Sistem dibangun menggunakan bahasa pemrograman *PHP* dan *HTML* untuk antarmuka pengguna, serta *MySQL* sebagai basis data. Fitur-fitur dalam sistem meliputi *dashboard* admin, form peta sebaran, form input data desa dan kecamatan, yang mendukung pengelolaan dan visualisasi data pertanian secara spasial.

Hasil dari penelitian ini menunjukkan bahwa sistem informasi yang dikembangkan berhasil menampilkan peta persebaran hasil pertanian secara interaktif di wilayah Kabupaten Polewali Mandar. Sistem ini mempermudah pengguna, baik dari pihak internal pemerintah maupun masyarakat umum, dalam mengetahui jenis tanaman, lokasi produksi, dan data tabulasi pertanian berdasarkan desa dan kecamatan. Dengan adanya fitur-fitur seperti pemetaan lokasi pertanian dan *input* data produksi, sistem ini mampu memberikan informasi yang lebih jelas dan luas mengenai potensi pertanian setempat. Selain itu, sistem ini dinilai mampu menjadi alat bantu dalam pengambilan keputusan, sosialisasi hasil pertanian, dan mendukung transparansi informasi di sektor pertanian daerah.[28]



Gambar 2.12 Hasil Penelitian Terkait Form Peta Sebaran



No	Nama Desa	Kecamatan	Gejgon	warna	Aksi
1	Desa 1	Kecamatan Matunda	[{"type":"FeatureCollection","features":[{"type":"Feature","properties":{"geometry":{"type":"Polygon","coordinates":[[[18.900867,2.970942],[18.902507,2.973318],[18.909584,2.973961],[18.909750,2.977086],[18.906310,2.980346],[18.909051,2.979446],[18.900867,2.970942]]]]}}}]}		Ubah Hapus
2	Desa 2	Kecamatan Matunda	[{"type":"FeatureCollection","features":[{"type":"Feature","properties":{"geometry":{"type":"Polygon","coordinates":[[[18.921706,2.930964],[18.942296,2.930221],[18.943014,2.943936],[18.932016,2.946336],[18.923397,2.950886],[18.914832,2.945307],[18.921706,2.930964]]]]}}}]}		Ubah Hapus
3	Desa 3	Kecamatan Matunda	[{"type":"FeatureCollection","features":[{"type":"Feature","properties":{"geometry":{"type":"Polygon","coordinates":[[[18.891377,2.948461],[18.901344,2.949631],[18.901,2.954631],[18.899137,2.959446],[18.888112,2.957171],[18.880305,2.952574],[18.891377,2.948461]]]]}}}]}		Ubah Hapus
4	Desa Sulaeni	Kecamatan Matunda	[{"type":"FeatureCollection","features":[{"type":"Feature","properties":{"geometry":{"type":"Polygon","coordinates":[[[18.924505,2.985775],[18.933795,2.985261],[18.945851,2.987832],[18.953212,2.975032],[18.959418,2.983779],[18.959418,2.980632],[18.952181,2.986446],[18.939981,2.980046],[18.924505,2.983803],[18.925823,2.980886],[18.912487,2.982046],[18.911496,2.986118],[18.914205,2.972482],[18.914892,2.964289],[18.924505,2.985775]]]]}}}]}		Ubah Hapus

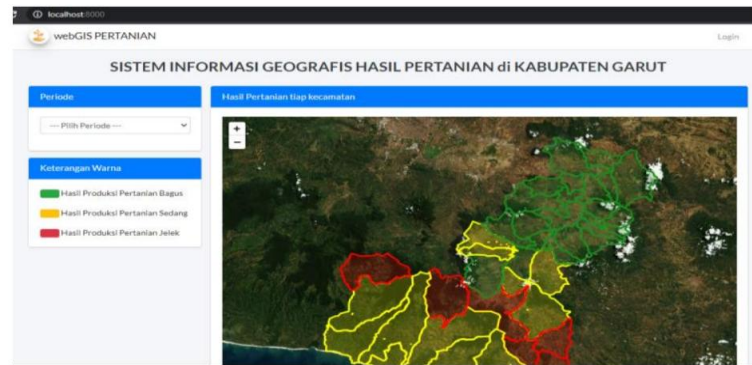
Gambar 2.13 Hasil Penelitian Terkait Form Data Desa

2.2.7 Rancang Bangun Sistem Informasi Geografis Hasil Pertanian di Kabupaten Garut

Penelitian berjudul “Rancang Bangun Sistem Informasi Geografis Hasil Pertanian di Kabupaten Garut” dilakukan oleh Raden Erwin Gunadhi Rahayu dan Anwar Fauzi pada tahun 2020. Penelitian ini bertujuan untuk mengembangkan sistem informasi geografis yang mampu memetakan data hasil pertanian, khususnya komoditas padi, di wilayah Kabupaten Garut. Sistem ini dibangun untuk mendukung Dinas Pertanian dalam memantau daerah-daerah penghasil padi secara lebih efektif. Metodologi yang digunakan adalah *Rational Unified Process (RUP)*, dengan tahapan *Inception*, *Elaboration*, *Construction*, dan *Transition*. Sistem dirancang menggunakan pendekatan *Unified Modeling Language (UML)*, seperti *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*. Dalam tahap pengujian, peneliti menggunakan metode *black-box testing* untuk memastikan sistem berjalan sesuai dengan fungsinya. Walaupun detail teknis seperti *framework* dan *database* tidak dijelaskan secara spesifik, sistem berbasis web ini memiliki antarmuka pengguna untuk aktor admin dan penyuluh pertanian.

Hasil dari penelitian ini menunjukkan bahwa sistem yang dibangun berhasil mempermudah proses pendataan dan pemetaan hasil produksi padi berdasarkan wilayah. Dengan adanya fitur pemetaan geografis digital, sistem memungkinkan visualisasi langsung terhadap data seperti luas tanam, luas panen, dan tingkat produktivitas padi per kecamatan. Pengujian fungsionalitas sistem terhadap fitur

login, manajemen pengguna, penyuluh, dan hasil pertanian menunjukkan bahwa semua fitur berjalan dengan baik. Penelitian ini memberikan kontribusi penting dalam penggunaan sistem informasi geografis sebagai media bantu untuk pengambilan keputusan berbasis data spasial di sektor pertanian daerah.[29]



Gambar 2.14 Hasil Penelitian Terkait Implementasi

No	Aktor	Harapan	Keterangan
1	<i>Login</i>	Masuk ke aplikasi	Berhasil
2	<i>Logout</i>	Keluar dari aplikasi	Berhasil
3	Kelola penyuluh	Dapat mengelola penyuluh diantaranya menambah, menghapus, mengubah melihat	Berhasil
4	Kelola user	Dapat mengelola user diantaranya menambah, menghapus, mengubah melihat	Berhasil
5	Kelola hasil pertanian	Dapat mengelola hasil pertanian diantaranya menambah, menghapus, mengubah melihat	Berhasil

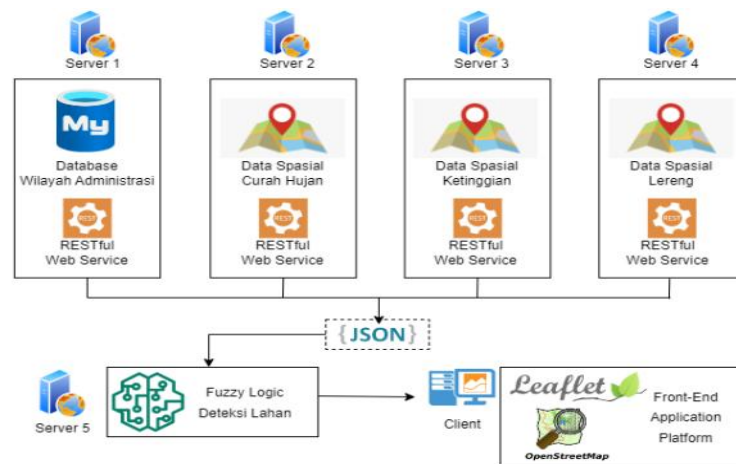
Gambar 2.15 Hasil Penelitian Terkait Hasil Pengujian

2.2.8 Arsitektur Geospasial *API* dengan *Laravel* untuk Sistem Deteksi Kesesuaian Lahan Pertanian

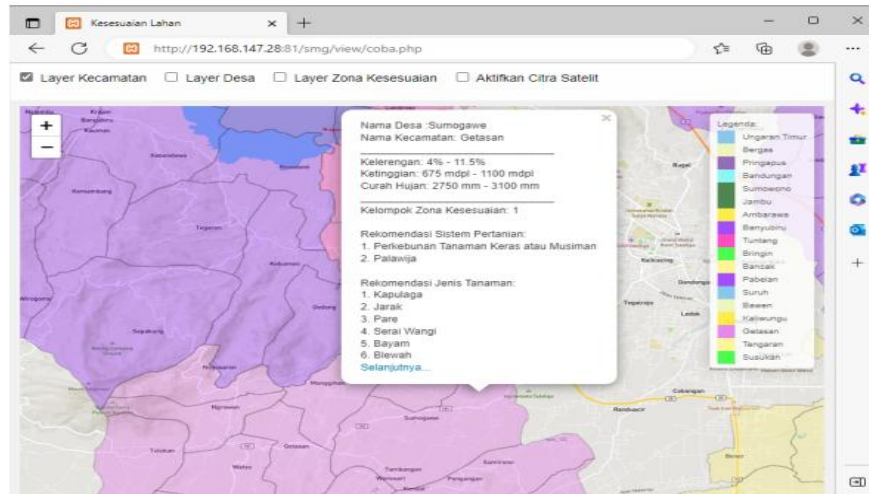
Penelitian ini dilakukan oleh Yeremia Alfa Susetyo dari Universitas Kristen Satya Wacana dan dipublikasikan pada tahun 2023 dalam Jurnal Sistemasi. Tujuan dari penelitian ini adalah membangun arsitektur Geospasial *API* berbasis *Laravel* untuk mendeteksi kesesuaian lahan pertanian di Kabupaten Semarang. Penelitian ini menggunakan pendekatan *microservices RESTful API* dan menerapkan *framework Laravel* sebagai *backend processing platform*. Data yang digunakan terdiri dari data spasial (peta curah hujan, ketinggian, kelerengan) dan data *non*-spasial (zonasi kesesuaian lahan), yang dikemas dalam format *GeoJSON*. Penelitian ini melibatkan 5 *server node*: empat server menyediakan *API* untuk data spasial, dan satu server (Server 5) bertindak sebagai *client* yang menggabungkan dan memproses data

dengan algoritma *Fuzzy Logic*. Metodologi penelitian meliputi tahapan pengumpulan data, pembangunan platform *backend*, penerapan *fuzzy logic* untuk pengelompokan wilayah, hingga pembuatan frontend berbasis peta interaktif menggunakan *Leaflet* dan *OpenStreetMap*.

Hasil penelitian menunjukkan bahwa sistem *Geospatial API* yang dibangun mampu menangani data spasial yang kompleks dan heterogen secara efisien. Sebanyak 6 API utama berhasil dikembangkan, masing-masing mewakili data wilayah, curah hujan, ketinggian, dan kemiringan, serta dua layanan tambahan untuk informasi kesesuaian lahan. Data yang dihasilkan berbentuk *GeoJSON*, yang kemudian divisualisasikan dalam bentuk peta digital interaktif di sisi *frontend*. Sistem juga mampu memberikan rekomendasi tanaman berdasarkan zona kesesuaian lahan dengan memanfaatkan *Fuzzy Inference System (FIS)*. Pengujian *API* dilakukan menggunakan *Postman*, dan hasilnya menunjukkan *API* dapat merespon dengan cepat dan sesuai format yang diharapkan. Penelitian ini membuktikan bahwa arsitektur backend berbasis *Laravel REST API* dapat diintegrasikan secara fleksibel dengan berbagai *platform frontend* dan layak diterapkan dalam sistem informasi pertanian berskala besar.[30]



Gambar 2.16 Hasil Penelitian Terkait Perancangan Arsitektur Geospatial *API* dengan *Laravel* untuk Sistem Deteksi



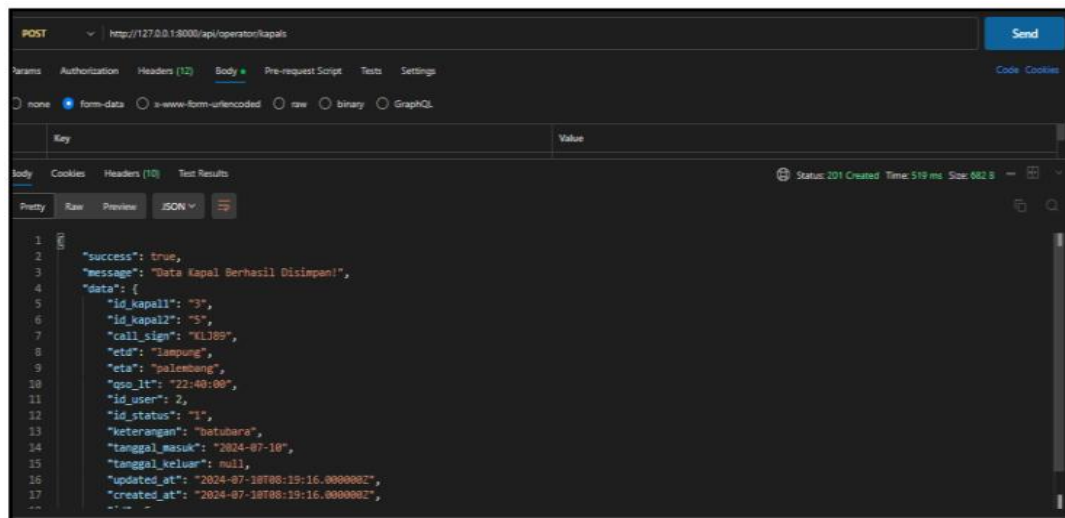
Gambar 2.17 Hasil Penelitian Terkait *Layer Overlay* antara Peta Desa dan Informasi Zona Kesesuaian Lahan

2.2.9 Implementasi *Rest Api* Pada Sistem Pendataan Kapal Menggunakan *Laravel 9* Dan *Nuxtjs*

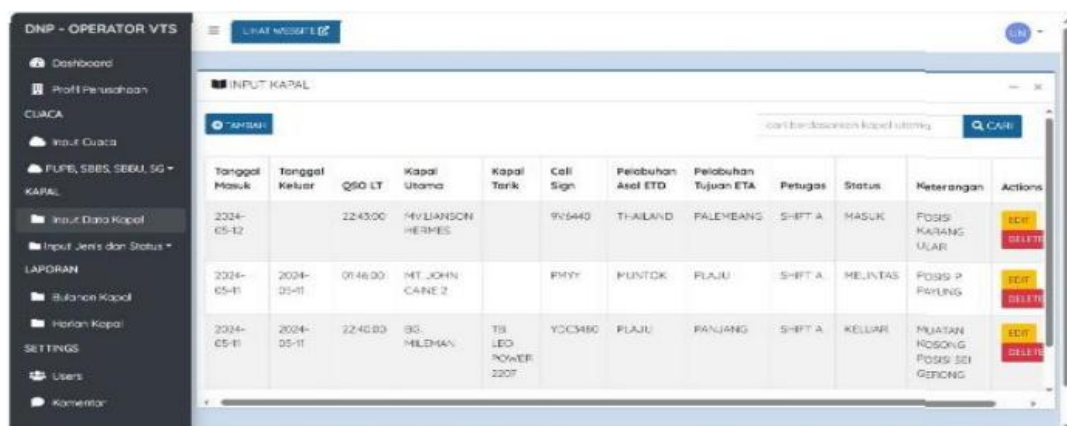
Penelitian ini dilakukan oleh Laurensius Sirwanti Saputra dan Hendrik Fery Herdiyatomoko dari Fakultas Sains dan Teknologi, Universitas Katolik Musi Charitas, Palembang, dan diterbitkan pada tahun 2024 dalam *Journal Computer and Technology*. Penelitian ini bertujuan mengembangkan sistem pendataan keluar-masuk kapal berbasis *website* untuk Distrik Navigasi Tipe A Kelas I Palembang dengan menggunakan teknologi *REST API* sebagai *Web Service*. Sistem ini dibangun menggunakan *framework Laravel 9* untuk *backend*, *NuxtJs* untuk *frontend*, *MySQL* sebagai *database server*, serta menggunakan *Postman* sebagai alat pengujian *API*. Metode pengembangan perangkat lunak yang digunakan adalah metode *Waterfall*, dengan tahapan komunikasi, perencanaan, pemodelan, konstruksi, dan *deployment*.

Hasil penelitian menunjukkan bahwa sistem pendataan kapal yang dibangun berhasil menggantikan metode manual sebelumnya dan mendukung operasional pelaporan harian dan bulanan kapal dengan lebih efisien. Implementasi *REST API* memungkinkan pertukaran data berbasis *JSON* antara sistem internal dan eksternal secara cepat dan akurat. Hasil pengujian menggunakan *black-box testing*, *white-box testing*, dan pengujian *response time* dengan *Postman* menunjukkan bahwa seluruh fungsi berjalan baik dan memenuhi kebutuhan yang ditetapkan. Secara keseluruhan,

sistem ini dinilai efektif, andal, dan mampu mendukung integrasi dengan sistem lain melalui *REST API*. [31]



Gambar 2.18 Hasil Penelitian Terkait Respon *JSON GET* Kapal



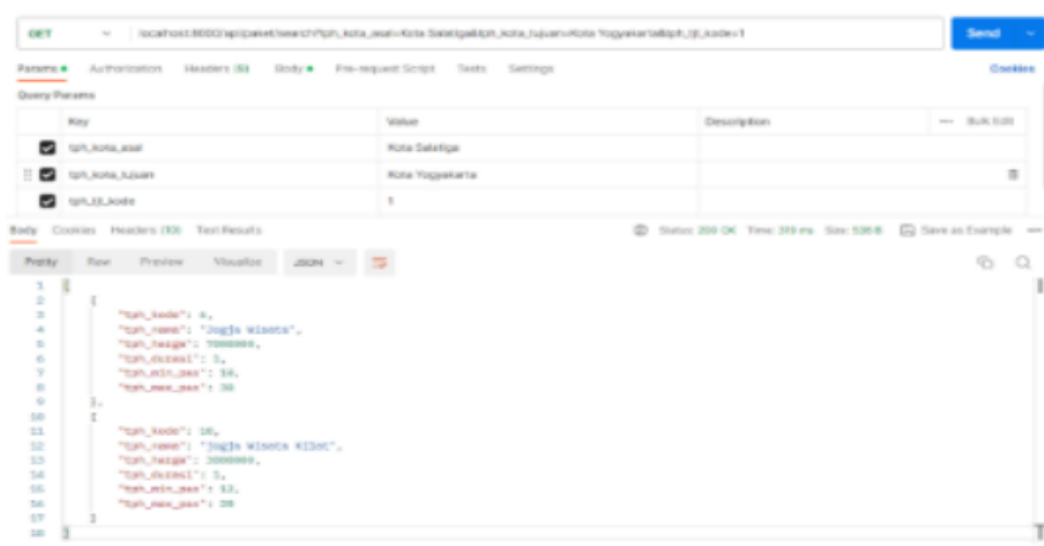
Gambar 2.19 Hasil Penelitian Terkait Menu *View Data* Kapal

2.2.10 Perancangan *Web Service REST API* Menggunakan *PHP* dan *Framework Laravel* di Tenta *Tour Salatiga*

Penelitian ini dilakukan oleh Rudi Purnomo, yang membahas pengembangan layanan *web service* untuk mendukung sistem informasi di perusahaan Tenta *Tour Salatiga*. Penelitian ini diterbitkan pada tahun 2023. Metode yang digunakan dalam penelitian adalah metode pengembangan perangkat lunak berbasis rekayasa kebutuhan dan implementasi *REST API*. *Framework* yang digunakan adalah *Laravel*, dengan bahasa pemrograman *PHP*, dan *database* yang digunakan adalah *MySQL*. Untuk pengujian layanan, digunakan *Postman* sebagai alat untuk menguji *endpoint REST API* serta memastikan fungsi *CRUD* (*Create, Read, Update, Delete*)

berjalan sesuai spesifikasi. Sistem dibangun dengan pendekatan arsitektur berbasis *Model-View-Controller (MVC)*.

Penelitian menunjukkan bahwa penerapan sistem terpadu berbasis *web service* dengan arsitektur *REST* berhasil menyediakan layanan pengolahan data secara responsif dan lintas platform berkat interoperabilitas yang tinggi. Sistem ini memungkinkan akses melalui berbagai perangkat dan mendukung pengembangan yang fleksibel dan efisien. Hasil *performance testing* menunjukkan rata-rata waktu request sebesar 1216 ms, dengan *throughput* mencapai 1734 *request* per menit dan transfer data sebesar 11,7 KB per *response*. Meskipun sistem sudah responsif dan efisien, masih terdapat ruang untuk pengembangan, seperti penambahan fitur promosi, pemesanan, dan sistem pembayaran melalui *payment gateway*, serta integrasi lebih luas ke aplikasi web maupun mobile melalui *API*. [32]



Gambar 2.20 Hasil Penelitian Terkait Pengiriman Request Ke *Web Service*

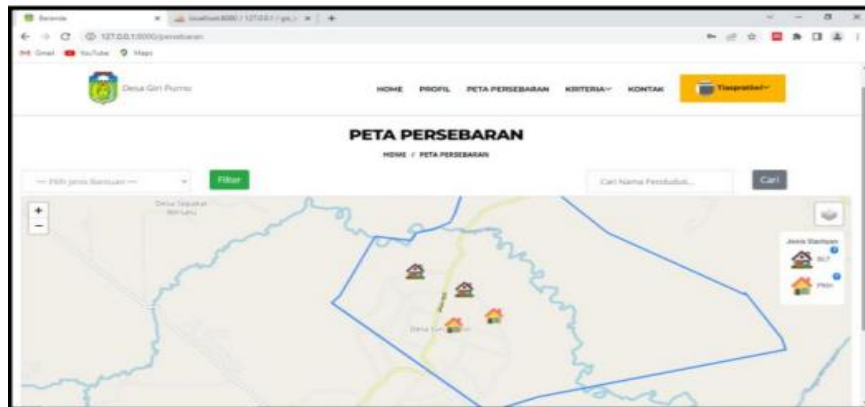


Gambar 2.21 Hasil Penelitian Terkait Halaman *Dashboard* Paket Wisata

2.2.11 Implementation of Web-Based Geographical Information System Mapping Recipients of Social Assistance in Giri Purno Village

Penelitian ini ditulis oleh Dwi Winarti, Lido Sabda Lesmana, dan Tias Pratiwi dari Universitas Dharmas Indonesia dan diterbitkan pada Agustus 2023. Penelitian ini bertujuan untuk mengembangkan Sistem Informasi Geografis berbasis web guna memetakan rumah-rumah penerima bantuan sosial di Desa Giri Purno, Kecamatan Rimbo Ilir, Kabupaten Tebo. Metode penelitian yang digunakan adalah metode *Waterfall*, yang mencakup tahapan analisis kebutuhan, perancangan sistem, implementasi, integrasi, serta pemeliharaan sistem. Dalam pengembangan sistem, peneliti menggunakan beberapa alat dan teknologi utama, antara lain *framework Laravel* sebagai *backend*, *Leaflet.js* untuk visualisasi peta, serta *OpenStreetMap* sebagai peta dasar. Untuk manajemen basis data, sistem ini memanfaatkan *MySQL*, sedangkan pengujian sistem dilakukan melalui implementasi fitur-fitur sistem berdasarkan rancangan *UML* seperti *use case*, *activity*, dan *sequence diagram*. *Tools* tambahan yang digunakan dalam proses pengembangan termasuk *XAMPP* untuk server lokal dan *Bootstrap* untuk antarmuka pengguna.

Hasil penelitian menunjukkan bahwa sistem yang dibangun mampu menampilkan data penerima bantuan sosial beserta informasi pendukungnya secara spasial, seperti lokasi rumah, RT, dan kategori bantuan yang diterima. Selain itu, sistem juga menyediakan fitur rute menuju rumah penerima bantuan, sehingga dapat memudahkan pihak desa maupun instansi terkait dalam pendistribusian bantuan dan pemantauan secara geografis. Dengan tampilan peta yang interaktif dan data yang terorganisir, sistem ini dinilai berhasil meningkatkan efisiensi dalam pengelolaan data bantuan sosial di tingkat desa.[33]



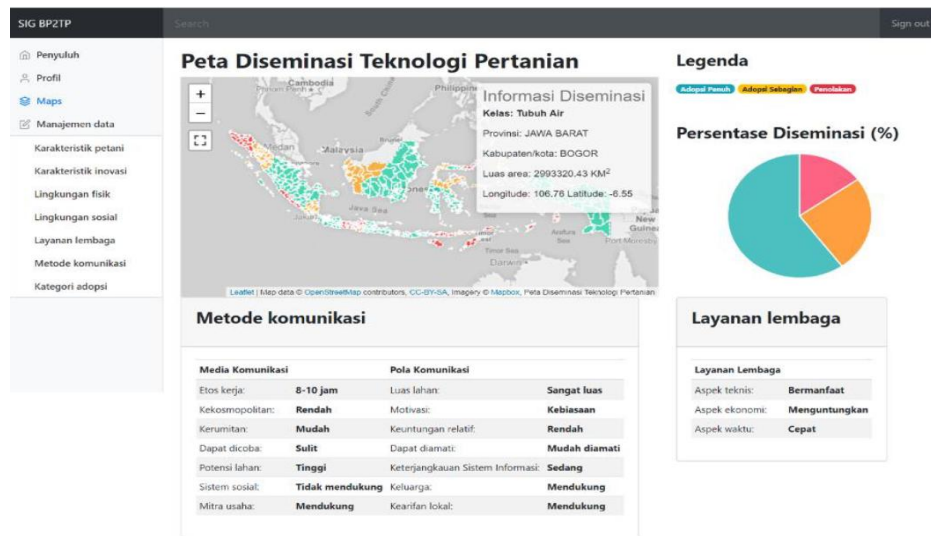
Gambar 2.22 Hasil Penelitian Terkait Distribution Map Page

2.2.12 Geographic Information System for Mapping Agricultural Technology Dissemination

Penelitian ini ditulis oleh Tegar Karangga A dan Heni Sulistiani dari Universitas Teknokrat Indonesia, dan dipublikasikan pada tahun 2024. Penelitian ini bertujuan untuk mengembangkan sistem informasi geografis berbasis *web* yang memetakan penyebaran teknologi pertanian di berbagai wilayah Indonesia. Metode yang digunakan dalam pengembangan sistem ini adalah *Predictive Software Development Life Cycle (SDLC)*. Proses pengembangan sistem mencakup tahapan perencanaan, analisis, desain (menggunakan diagram *use case*, *class*, *sequence*, dan *flowchart*), implementasi, pengujian (*black-box testing*), dan pemeliharaan. Dalam implementasinya, sistem menggunakan *framework Laravel* sebagai *backend*, *PostgreSQL* dengan ekstensi *PostGIS* untuk *database* spasial, serta *Leaflet.js* sebagai pustaka visualisasi peta. Visualisasi data dilakukan dalam format *GeoJSON*, yang diubah dari *query* spasial melalui fungsi *ST_AsGeoJSON*, dan ditampilkan menggunakan antarmuka berbasis *web* dengan *Bootstrap*.

Hasil dari penelitian ini adalah sistem *GIS* berbasis *web* yang mampu menampilkan peta interaktif penyebaran teknologi pertanian. Peta ini menampilkan klasifikasi adopsi teknologi (adopsi penuh, sebagian, atau penolakan) pada wilayah-wilayah tertentu. Selain itu, sistem juga menyediakan halaman manajemen data untuk karakteristik petani, termasuk lokasi geografis dan tingkat adopsi teknologi. Sistem ini diharapkan dapat membantu peneliti dan penyuluh pertanian dalam merancang

strategi penyebaran teknologi yang lebih efektif dan sesuai dengan karakteristik lokal.[34]



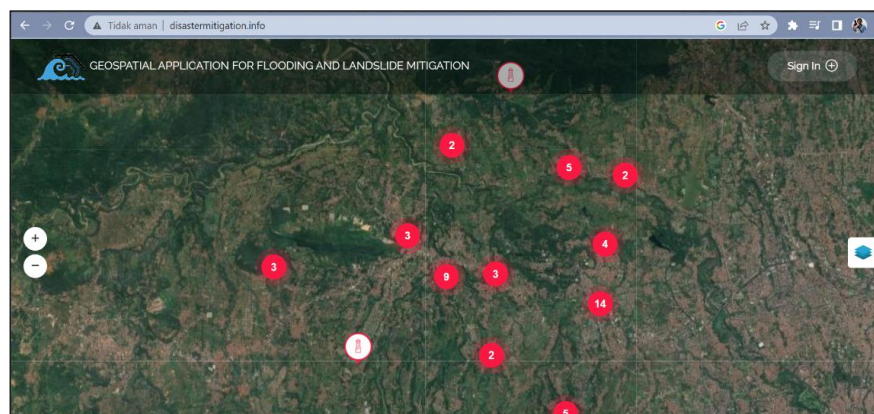
Gambar 2.23 Hasil Penelitian Terkait *Maps Page Design*

2.2.13 The Designing of Geoinformation Backend-Frontend to Improve Landslide Mitigation Application

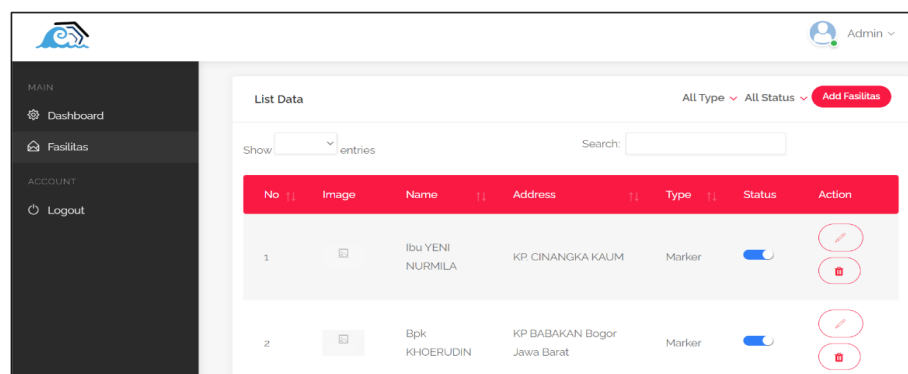
Penelitian ini dilakukan oleh Iksal Yanuarsyah, Syarbaini Ahmad, dan Nurkaliza Khalid dipublikasikan pada tahun 2024. Tujuan penelitian ini adalah merancang *backend* dan *frontend* sistem geoinformasi untuk mendukung aplikasi mitigasi bencana longsor berbasis *web* dan *mobile*. Pengembangan dilakukan dengan metode *Waterfall Model*, mencakup tahap perencanaan, desain, implementasi, dan pengujian. *Backend* dibangun menggunakan *PHP* dan *Node.js* dalam arsitektur *client-server*, dengan komunikasi data berbasis *JSON* melalui *Web Service API*. *Frontend* dikembangkan menggunakan *HTML*, *CSS*, dan *JavaScript*, serta *Leaflet JS* sebagai pustaka pemetaan interaktif. Sistem juga mengimplementasikan perhitungan jarak evakuasi menggunakan rumus *Spherical Law of Cosines*. Diagram UML seperti *use case*, *activity*, *sequence*, dan *class* digunakan dalam mendesain dan memodelkan sistem.

Hasil penelitian menunjukkan bahwa sistem yang dikembangkan mampu menampilkan data spasial seperti peta rawan longsor, titik evakuasi, dan rute penyelamatan secara interaktif. Sistem menyediakan antarmuka yang mudah digunakan baik untuk administrator maupun pengguna umum, serta mendukung

pembaruan data secara *real-time* melalui *web* dan *mobile*. Visualisasi data dilakukan dalam bentuk lapisan peta dan data tabular, memungkinkan pengguna melihat informasi lokasi, kondisi potensi bencana, serta jalur evakuasi yang paling efisien. Sistem ini juga memiliki fitur untuk mengelola dan mengedit layer peta, serta mendukung pemantauan kondisi secara langsung melalui antarmuka *web*. Dengan kombinasi *backend* yang responsif dan *frontend* yang intuitif, sistem ini berhasil meningkatkan efektivitas penyampaian informasi spasial dalam konteks mitigasi bencana.[35]



Gambar 2.24 Hasil Penelitian Terkait *Home page*



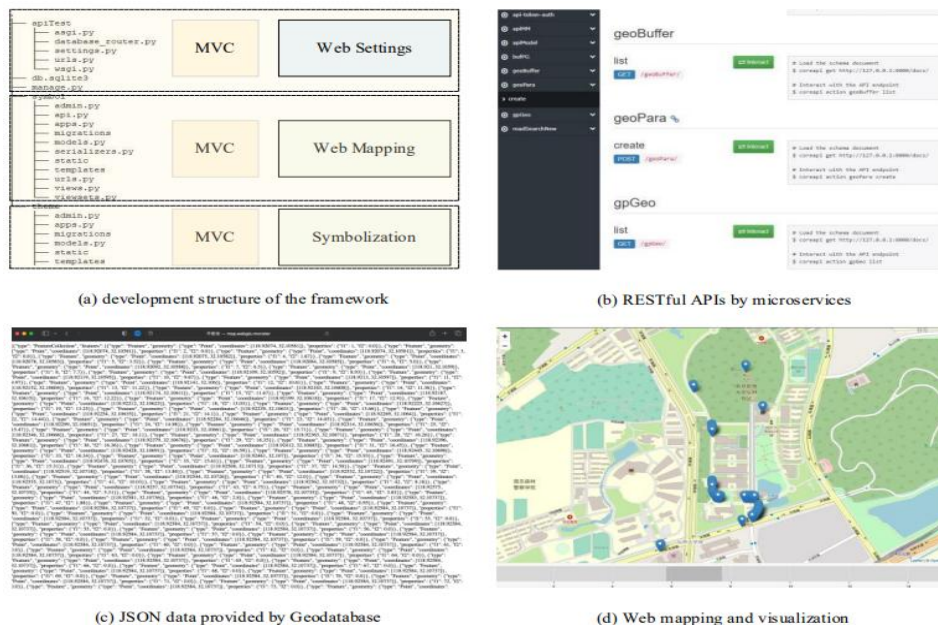
Gambar 2.25 Hasil Penelitian Terkait *Data Maintenance Page*

2.2.14 The Design of Scalable Web GIS Microservice Framework for Undergraduate Education

Penelitian ini dilakukan oleh Haiyang Lyu, Lizhi Miao, Donglai Jiao, dan Qiqi Zhong dari Nanjing University of Posts and Telecommunications, China, pada tahun 2023. Tujuan dari penelitian ini adalah merancang kerangka kerja *Web GIS* berbasis *microservice* yang berskala untuk meningkatkan kemampuan mahasiswa

dalam praktik *GIS*. Metode yang digunakan adalah pengembangan sistem menggunakan pendekatan *RESTful API* dengan arsitektur *Model-View-Controller (MVC)*. *Backend* dikembangkan menggunakan *framework Python Django*, dengan integrasi *PostgreSQL* dan ekstensi *PostGIS* sebagai basis data spasial. Visualisasi data dilakukan menggunakan *JavaScript* dan pustaka seperti *Leaflet*, *OpenLayers*, dan *D3.js*. Sistem juga dikemas menggunakan *Docker* dan dideploy melalui *Nginx* dengan *Web Server Gateway Interface (WSGI)*.

Hasil dari penelitian menunjukkan bahwa kerangka kerja yang dirancang berhasil mengintegrasikan berbagai mata kuliah GIS seperti *Web GIS*, *GIS Principles*, *GIS Algorithms*, *GIS Software Development*, dan *GeoDatabase* ke dalam satu sistem pembelajaran yang terpadu. Mahasiswa mampu mengembangkan *Web GIS* secara menyeluruh, mulai dari perancangan *database* spasial, pengembangan *RESTful API*, hingga visualisasi data spasial di *browser*. Kegiatan praktik dilakukan dalam berbagai bentuk seperti eksperimen terbuka, tugas akhir, proyek inovasi, dan kompetisi profesional. Melalui pendekatan ini, mahasiswa menunjukkan peningkatan signifikan dalam keterampilan teknis *GIS*, pemahaman lintas mata kuliah, serta kemampuan *problem solving* berbasis sistem informasi geografis.[36]

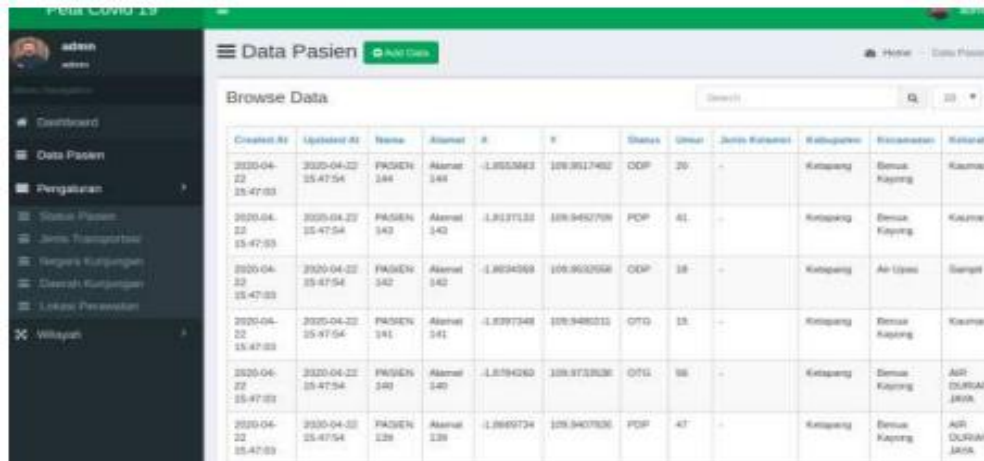


Gambar 2. 26 Hasil Penelitian Terkait Implement Of The Scale Web GIS
Microservice Framework

2.2.15 Melihat Peta Penyebaran Pasien COVID-19 dengan Kombinasi *QGIS* dan *Framework Laravel*

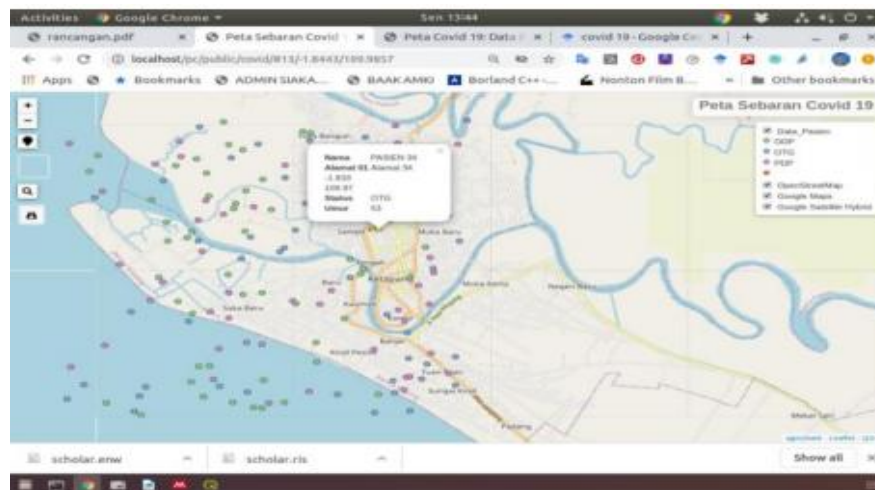
Penelitian ini dilakukan oleh Muhammad Ullil Fahri dari Akademi Manajemen Komputer dan Informatika (AMKI) Ketapang, dan dipublikasikan dalam Jurnal Teknologi Terpadu pada tahun 2020. Penelitian ini bertujuan untuk membantu pemerintah daerah Kabupaten Ketapang dalam melacak penyebaran pasien COVID-19 secara visual melalui pemetaan digital. Metode yang digunakan adalah digitalisasi pemetaan dengan menggunakan *QGIS* untuk pengolahan data spasial dan *framework Laravel* untuk pengembangan aplikasi web berbasis Sistem Informasi Geografis (SIG). Penelitian ini memanfaatkan data koordinat pasien yang dikategorikan ke dalam status *ODP*, *OTG*, *PDP*, dan Positif COVID-19, kemudian diolah dalam *QGIS* dan ditransformasikan ke dalam format *web* menggunakan plugin *QGIS2Web*. Aplikasi ini diimplementasikan pada sistem operasi *Ubuntu* 18.04, menggunakan *Apache2*, *PHP* 7, dan *MySQL* sebagai basis data.

Hasil dari penelitian ini menunjukkan bahwa kombinasi *QGIS* dan *Laravel* dapat menghasilkan aplikasi pemetaan pasien COVID-19 yang akurat dan mudah diakses melalui website oleh pihak yang berkepentingan, seperti rumah sakit dan pemerintah daerah. Sistem memungkinkan pengguna untuk menambahkan data pasien berdasarkan koordinat x dan y, yang secara otomatis akan divisualisasikan di peta dengan warna sesuai status pasien. Aplikasi ini juga mendukung fitur seperti *highlight* saat *hover*, *filter* berdasarkan status, dan geolokasi pengguna. Dengan rancangan database yang telah dinormalisasi, sistem mampu menangani jumlah data pasien yang besar dan disiapkan untuk menjadi *open source* agar dapat digunakan lebih luas. Penelitian ini membuktikan bahwa teknologi SIG berbasis *web* sangat efektif untuk visualisasi penyebaran penyakit secara *real-time* dan mendukung pengambilan keputusan yang cepat dalam situasi darurat.[37]



Created At	Updated At	Name	Alamat	R	S	Status	Umur	Jenis Kelamin	Kabupaten	Kecamatan	Kelurahan
2020-04-22 15:47:53	2020-04-22 15:47:54	PASIEH	Alamat	-1.8553863	109.9617462	ODP	20	-	Katapang	Benua Kaping	Kasman
2020-04-22 15:47:53	2020-04-22 15:47:54	PASIEH	Alamat	-1.8337133	109.9492706	POD	41	-	Katapang	Benua Kaping	Kasman
2020-04-22 15:47:53	2020-04-22 15:47:54	PASIEH	Alamat	-1.8034068	109.9602058	ODP	18	-	Katapang	Air Upan	Sangit
2020-04-22 15:47:53	2020-04-22 15:47:54	PASIEH	Alamat	-1.8397348	109.9480211	OTG	15	-	Katapang	Benua Kaping	Kasman
2020-04-22 15:47:53	2020-04-22 15:47:54	PASIEH	Alamat	-1.8794260	109.9733036	OTG	86	-	Katapang	Benua Kaping	AIR DURAN JARA
2020-04-22 15:47:53	2020-04-22 15:47:54	PASIEH	Alamat	-1.8669724	109.9407836	POD	47	-	Katapang	Benua Kaping	AIR DURAN JARA

Gambar 2.27 Hasil Penelitian Terkait Halaman Admin



Gambar 2.28 Hasil Penelitian Terkait Peta Sebaran Pasien Covid-19 (Web)

2.2.16 An Approach for Common Web GIS Platform for Agriculture Sector Modernization in Sri Lanka

Penelitian ini dilakukan oleh S.T.S. De Silva, R.P. De Silva, dan N.D.K. Dayawansa dari University of Peradeniya, dan dipublikasikan pada tahun 2025. Penelitian ini bertujuan untuk mengembangkan *platform WebGIS* nasional bernama *AgriMIS* yang dapat digunakan oleh seluruh pemangku kepentingan di sektor pertanian Sri Lanka untuk mengelola data spasial dan *non-spasial* secara terintegrasi. Platform ini dibangun dengan pendekatan *cloud-based microservice architecture* menggunakan *Golang*, *Laravel*, *React Native*, *Python*, serta *database MongoDB* dan *MySQL*. Platform *AgriMIS* dihosting secara *hybrid* menggunakan *Google Cloud Platform (GCP)* dan *Microsoft Azure* untuk menguji interoperabilitas dan skalabilitas. Selain itu, *AgriMIS* juga menerapkan standar internasional seperti ISO

19115, *OGC*, serta keamanan siber dengan *OWASP 10* dan *CIS 20*. Akses data dilakukan melalui *REST API*, dan platform ini mendukung format *GeoJSON* untuk manajemen data spasial.

Hasil dari penelitian menunjukkan bahwa *AgriMIS* mampu memfasilitasi pengumpulan dan visualisasi data petani, lahan, pupuk, tanaman, dan panen secara *real-time*, baik melalui platform *WebGIS* maupun aplikasi *Android native* yang juga dapat digunakan secara *offline*. Platform ini mendukung berbagai analisis spasial seperti *proximity analysis*, *suitability analysis*, dan *forecasting fertilizer requirements*, serta memungkinkan integrasi dengan sistem pendukung keputusan (*Decision Support System*) melalui *API*. Dalam uji coba, platform ini digunakan oleh proyek pertanian cerdas yang didanai *World Bank* dan berhasil menampilkan sebaran petani, jenis tanaman, serta kebutuhan pupuk hingga tingkat wilayah administratif terkecil. Penelitian ini menunjukkan bahwa platform geospasial terpadu seperti *AgriMIS* berpotensi besar untuk meningkatkan efisiensi pengambilan keputusan di sektor pertanian dan menjadi tulang punggung digitalisasi data agrikultur nasional.[38]



Gambar 2.29 Hasil Penelitian Terkait *A Suitability Analysis Was Conducted On Fertilizer Requirements Against The Farmer*



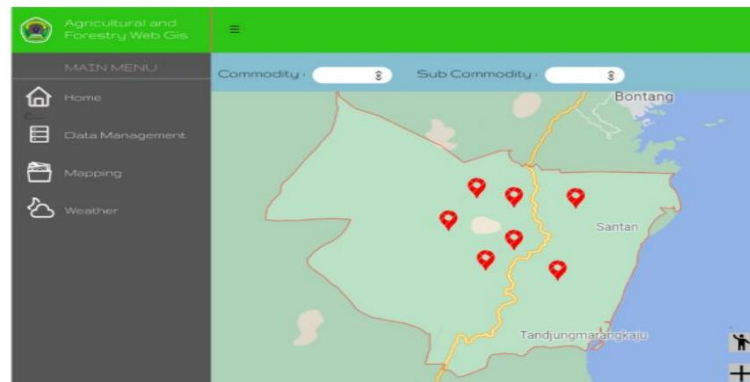
Gambar 2. 30 *Decision Support System Integrated With The AgriMIS Platform To Generate Statistical*

2.2.17 Development Web-GIS of Commodity Information System for Agriculture, Establishment and Forestry in Marangkayu District

Penelitian ini dilakukan oleh Adelia Juli Kardika, Aulia Khoirunnita, Salman, Saharuddin, dan Indah Muliana, dan dipublikasikan pada tahun 2022 dalam *International Journal of Education and Management Engineering (IJEME)*. Penelitian ini bertujuan untuk mengembangkan sistem informasi komoditas berbasis Web-GIS di Kecamatan Marangkayu, Kabupaten Kutai Kartanegara, Kalimantan Timur, untuk menyajikan informasi spasial terkait lokasi, harga, produksi, dan distribusi komoditas pertanian, perkebunan, dan kehutanan. Metode yang digunakan adalah *Rapid Application Development (RAD)*, yang mencakup empat tahap: *requirement planning*, *user design*, *construction*, dan *cut-over*. Sistem dibangun menggunakan *PHP*, *JavaScript*, *HTML*, dan *framework Bootstrap*, dengan basis data *PostgreSQL* serta ekstensi *PostGIS* untuk menangani data spasial. Pengujian sistem dilakukan dengan metode *black-box testing* untuk memastikan semua fungsionalitas sistem berjalan sesuai dengan kebutuhan pengguna.

Hasil penelitian menunjukkan bahwa aplikasi Web-GIS yang dibangun berhasil memenuhi kebutuhan pengguna dalam menyajikan informasi komoditas secara interaktif dan terintegrasi. Sistem ini memiliki 25 fungsionalitas, termasuk pencarian lokasi berdasarkan nama pemilik, desa, jenis komoditas, serta pelaporan harga dan produksi. Pengguna dapat melihat rute ke lokasi komoditas, mengelola data spasial dan atribut komoditas, serta melakukan analisis visual terhadap

perkembangan harga dan produksi komoditas dari waktu ke waktu. Admin yang terverifikasi dapat menambah, mengubah, dan menghapus data. Aplikasi ini dinilai efektif dalam memperkenalkan potensi wilayah Marangkayu serta memberikan kemudahan akses informasi kepada masyarakat, pemerintah, dan pelaku pasar dalam pengambilan keputusan berbasis data spasial.[39]

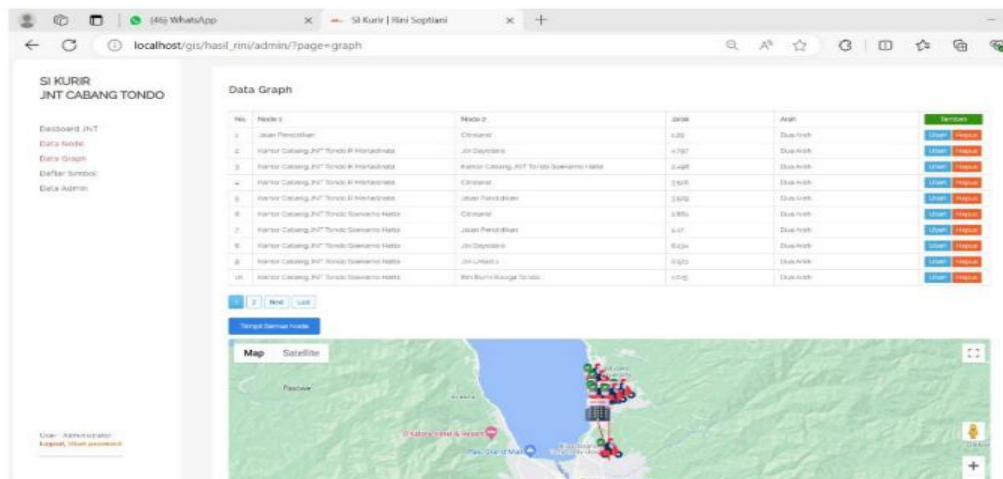


Gambar 2. 31 Hasil Penelitian Terkait *Mapping Page*

2.2.18 Geographical Information System Shortes Path Delivery Of Goods Using The Bellman-Ford And Dijkstra Algorithm (Case Study J&T Palu City)

Penelitian berjudul “*Geographical Information System Shortest Path Delivery of Goods Using the Bellman-Ford and Dijkstra Algorithm (Case Study J&T Palu City)*” dilakukan oleh Rini Septiani, Yuri Yudhaswana Joeфри, Rizka Ardiansyah, Septiano Anggun Pratama, dan Rahmah Laila dari Universitas Tadulako, dan dipublikasikan dalam JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika) pada bulan Juni 2025. Penelitian ini bertujuan untuk mengoptimalkan rute pengiriman barang menggunakan sistem informasi geografis (GIS) dengan algoritma *Bellman-Ford* dan Dijkstra dalam konteks pengiriman oleh *J&T Express* di Kota Palu. Metode yang digunakan adalah pendekatan kuantitatif eksperimental, dengan perancangan sistem berbasis *web* yang mengintegrasikan data jarak antar titik dari *Google Maps*. Algoritma *Bellman-Ford* digunakan untuk menangani graf berbobot negatif dan mendeteksi siklus negatif, sedangkan Dijkstra digunakan untuk graf dengan bobot positif dan memiliki efisiensi komputasi lebih tinggi. Pengujian sistem dilakukan menggunakan metode *Black Box Testing*.

Hasil penelitian menunjukkan bahwa penerapan algoritma *Bellman-Ford* dan Dijkstra dalam GIS pengiriman barang dapat meningkatkan efisiensi operasional seperti penghematan waktu tempuh, biaya bahan bakar, dan biaya pemeliharaan kendaraan. Dalam uji perbandingan, algoritma *Dijkstra* menunjukkan keunggulan dalam waktu komputasi untuk graf berbobot positif, sementara *Bellman-Ford* lebih unggul pada kondisi graf tertentu yang kompleks atau memiliki bobot negatif. Sistem yang dikembangkan juga dilengkapi dengan fitur visualisasi rute, halaman input data simpul (*node*) dan graf, serta *dashboard* pengelolaan pengiriman. Penelitian ini memberikan kontribusi penting dalam pengembangan *backend* sistem GIS logistik, yang relevan untuk aplikasi *WebGIS* dalam bidang lain seperti perkebunan atau pertanian.[40]



Gambar 2.32 Hasil Penelitian Terkait *Node Data* page

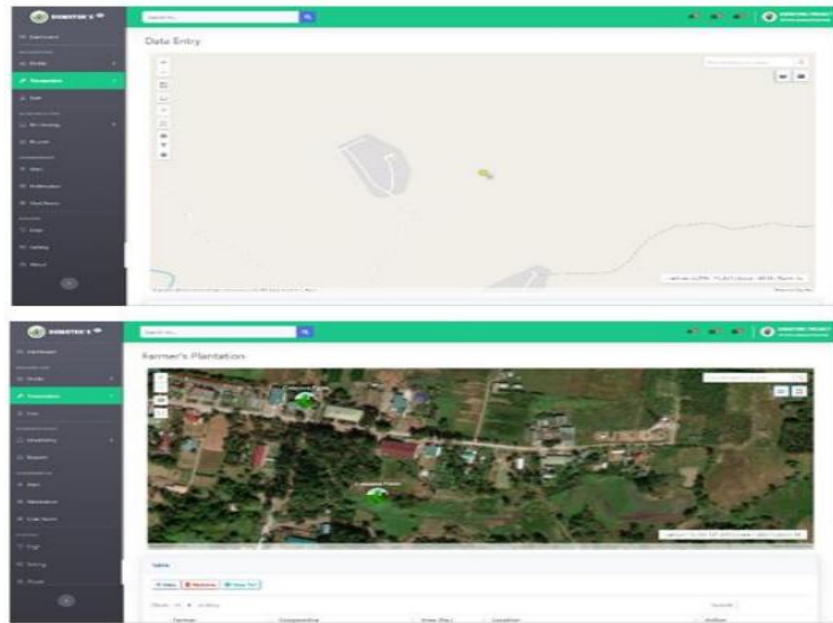
Test Case	Test Scenario	Testing Results
Opening the website	Open the website address	
login	Enter the username and password according to the database then press the login button	Valid
View node data	Pressing the node data button on the left sidebar	Valid
Add node data	Add node data and press the save button	Valid
Change node data	Modify the contents of the selected node data and press the save button.	Valid
Delete node data	Press the delete button on one of the data nodes, then press the OK button	Valid
View graph data	Pressing the graph data button on the left sidebar	Valid
Add graph data	Add the node data and press the save button	Valid
Changing graph data	Modify the contents of the selected graph data, then press the save button	Valid
Delete graph data	Press the delete button on one of the graph data, then press the OK button	Valid

Gambar 2.33 Hasil Penelitian Terkait *System Testing Page*

2.2.19 Drone-based Geographical Information System (GIS) Mapping of Cassava Phytoplasma Disease (CPD) for Precision Agriculture

Penelitian ini ditulis oleh Irma T. Plata, Edward B. Panganiban, Darios B. Alado, Allan C. Taracatac, Bryan B. Bartolome, dan Freddie Rick E. Labuanan dari Isabela State University, Filipina, serta dipublikasikan pada Februari 2022 dalam *International Journal of Emerging Technology and Advanced Engineering*. Tujuan dari penelitian ini adalah untuk mengembangkan sistem pemetaan penyakit CPD (*Cassava Phytoplasma Disease*) pada tanaman singkong menggunakan teknologi *drone* dan sistem informasi geografis (GIS) berbasis web. Pengembangan sistem dilakukan menggunakan metode *Rapid Application Development (RAD)*, dengan pendekatan pemrograman berbasis *PHP* (versi 7.1) dan *CodeIgniter 3.1* sebagai *framework*. Untuk pengolahan data spasial dan visualisasi peta, digunakan *ArcGIS JavaScript API*, sedangkan basis data dikelola melalui *MySQL* dengan bantuan alat seperti *SQLyog*. Antarmuka sistem dikembangkan menggunakan *HTML5*, *CSS3*, *JavaScript*, *JQuery*, *Ajax*, dan *Bootstrap 4*, dengan editor *Sublime Text 3*, serta diuji menggunakan browser untuk implementasi web.

Hasil dari penelitian ini menunjukkan bahwa sistem yang dikembangkan berhasil mendeteksi keberadaan penyakit *CPD* pada tanaman singkong secara visual melalui peta *heatmap* berbasis web. Sistem mampu memproses data dari *drone* berupa koordinat dan gambar, yang kemudian diolah menjadi informasi spasial tentang tingkat keparahan penyakit. Penggunaan *color intensity*, *pixel intensity*, dan *ratio plotting* dalam visualisasi *heatmap* memungkinkan identifikasi area yang terinfeksi dengan warna gradasi (kuning untuk ringan, *oranye* untuk sedang, dan merah untuk berat). Selain itu, sistem ini juga memungkinkan pengguna untuk melihat data tanaman berdasarkan lokasi, nama petani, varietas singkong, dan jumlah batang yang terinfeksi. Secara keseluruhan, penelitian ini membuktikan bahwa pemanfaatan *drone* dan *Web-GIS* dengan *backend* yang terintegrasi dapat membantu petani dan lembaga pertanian dalam melakukan deteksi dini dan pengendalian penyakit tanaman secara efisien dan akurat.[41]



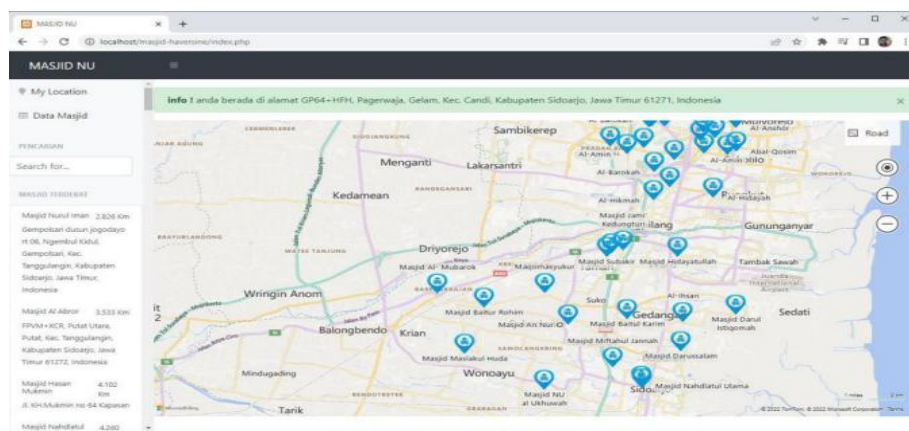
Gambar 2.34 Hasil Penelitian *Web-Based CPD Monitoring and Mapping System*

2.2.20 *Webgis Application For Searching Nu Mosque In East Java Using The Haversine Method*

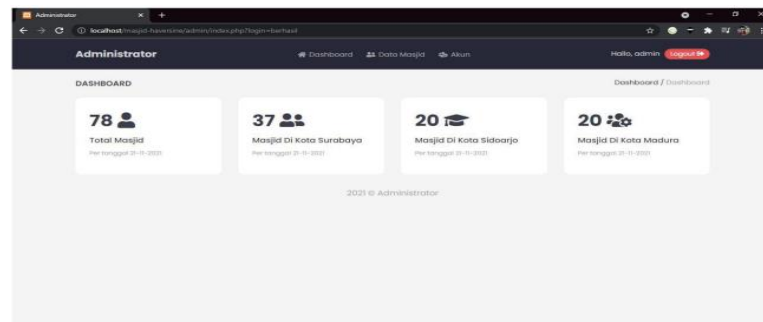
Penelitian ini dilakukan oleh Rizqi Ibnu Tarmizi, Mas Nurul Hamidah, dan Arif Arizal, pada bulan Desember 2022. Tujuan penelitian ini adalah mengembangkan aplikasi WebGIS yang dapat membantu pengguna, khususnya warga Nahdlatul Ulama (NU), dalam menemukan lokasi masjid NU terdekat di wilayah Jawa Timur. Metode yang digunakan dalam sistem ini adalah metode *Haversine*, yaitu rumus untuk menghitung jarak terpendek antara dua titik di permukaan bumi berdasarkan koordinat lintang dan bujur. Sistem ini dikembangkan dengan menggunakan teknologi WebGIS, *database MySQL*, serta algoritma *Haversine* untuk menghitung jarak secara presisi berdasarkan data spasial. Struktur sistem dijelaskan melalui *flowchart*, *context diagram*, dan *data flow diagram (DFD) level 0*, dengan dua entitas utama: *admin* dan *user*. Admin bertugas menambahkan dan mengelola data masjid, sementara user menggunakan fitur pencarian berdasarkan lokasi terkini.

Hasil dari penelitian menunjukkan bahwa sistem *WebGIS* yang dibangun berhasil menampilkan lokasi masjid NU di peta digital dan menghitung jarak dari lokasi pengguna ke masjid terdekat dengan akurat. Fitur utama sistem meliputi tampilan

lokasi pengguna, tampilan detail masjid, serta *dashboard* admin untuk mengelola data. Pengujian dilakukan dengan membandingkan hasil perhitungan jarak menggunakan sistem dengan hasil dari *Google Maps*. Dari lima kasus uji yang dilakukan, tiga di antaranya menghasilkan nama masjid dan jarak yang sama, sedangkan dua lainnya berbeda karena masjid tersebut tidak termasuk dalam kategori masjid NU. Hasil ini menunjukkan bahwa sistem mampu menjalankan fungsinya dengan baik dalam mendukung pencarian spasial secara akurat dan efisien, serta dapat digunakan sebagai alat bantu dalam menyebarkan informasi tempat ibadah berbasis komunitas.[42]



Gambar 2.35 Hasil Penelitian Terkait *My Location Page*



Gambar 2.36 Hasil Penelitian Terkait Admin *Dashboard Page*

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

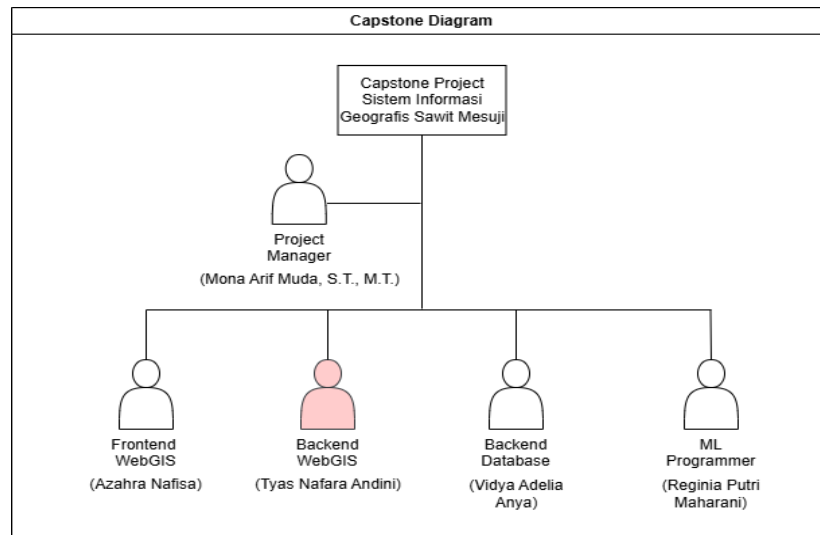
Waktu penelitian berlangsung selama kurang lebih 6 bulan semenjak bulan Mei hingga bulan Oktober 2025. Penelitian ini dilakukan di Gedung H jurusan teknik Elektro Universitas Lampung.

Tabel 3.1 Waktu Pelaksanaan Penelitian

No.	Aktivitas	Mei	Juni	Juli	Agustus	September	Oktober
1.	Pengumpulan Data						
2.	<i>Requirements Planning</i>						
3.	<i>User Design</i>						
4.	<i>Construction</i>						
5.	<i>Cutover</i>						
6.	Pengerjaan Laporan						

3.2 Capstone Project

Diagram di atas menjelaskan arsitektur dan alur kerja dalam pengembangan Sistem Informasi Geografis (SIG) Kebun Sawit berbasis *website*.



Gambar 3.1 Capstone Project

Capstone Project ini melibatkan lima aktor utama yaitu *Project Manager*, *Frontend WebGis*, *Backend WebGis*, *Backend Database*, dan *ML Programmer*. *Project Manager*, bertanggung jawab atas keseluruhan perencanaan, koordinasi, dan pengawasan jalannya pengembangan sistem. Di bawahnya terdapat tim pengembang yang terdiri dari beberapa spesialis dengan peran berbeda. *Frontend WebGIS Developer*, yang bertugas merancang dan mengembangkan tampilan antarmuka sistem agar dapat digunakan dengan baik oleh pengguna. *Backend WebGIS Developer*, bertanggung jawab dalam pengolahan logika bisnis di sisi *server* serta integrasi *API* yang diperlukan dalam sistem. *Backend Database Developer*, yang berfokus pada perancangan dan pengelolaan basis data untuk menyimpan informasi penting yang digunakan dalam sistem *WebGIS*. *Machine Learning (ML) Programmer*, yang berperan dalam mengembangkan model pembelajaran mesin yang digunakan untuk analisis data geografis terkait perkebunan sawit di Mesuji. 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.

3.3 Alat dan Bahan Penelitian

3.3.1 Alat

Pada penelitian ini, terdapat beberapa alat penelitian yang digunakan untuk mendukung dan menunjang kelancaran setiap proses penelitian yang dilakukan. Adapun alat yang digunakan dalam penelitian ini adalah sebagai berikut:

Tabel 3.2 Alat Penelitian

No.	Perangkat	Deskripsi	Spesifikasi
1.	<i>Laptop</i>	Perangkat keras yang digunakan dalam pengembangan sistem.	<i>ASUS Vivobook M1403QA</i> , prosesor <i>AMD Ryzen 7 5800H</i> (3.20 GHz), <i>RAM</i> 16 GB, GPU <i>AMD Radeon Graphics</i> (496 MB), penyimpanan <i>SSD</i> 477 GB, sistem operasi <i>Windows 11 Home Single Language Version 25H2 (Build 26200.7019)</i>
2.	<i>Visual Studio Code</i>	Perangkat lunak yang digunakan untuk mengolah kode program.	Versi 1.105.1 (<i>user setup</i>)
3.	<i>Laravel</i>	<i>Framework</i> yang digunakan dalam perancangan sistem.	<i>Framework Laravel</i> 11.46.1
4.	<i>UMLet</i>	<i>Website</i> yang digunakan untuk mendesain <i>requirement</i> .	-
5.	<i>Mouse</i>	Perangkat keras yang digunakan untuk menggerakkan kursor.	-
7.	<i>PHP</i>	Bahasa pemrograman yang digunakan pengembangan <i>web</i> .	Versi 8.2.12

8.	<i>Nginx</i>	<i>Web server</i> yang digunakan untuk menjalankan dan melayani aplikasi <i>Laravel</i> di lingkungan <i>hosting</i> CloudAja. <i>Nginx</i> berfungsi sebagai <i>reverse proxy</i> dan <i>load balancer</i> , yang memastikan koneksi API dapat berjalan cepat dan stabil.	<i>nginx version: nginx/1.18.0 (Ubuntu)</i>
9.	<i>CloudAja</i>	Platform layanan cloud yang digunakan untuk melakukan hosting <i>backend Laravel</i> . CloudAja menyediakan infrastruktur server berbasis cloud dengan dukungan <i>Nginx</i> dan PHP-FPM sehingga API dapat diakses secara online oleh frontend WebGIS kebun sawit.	-

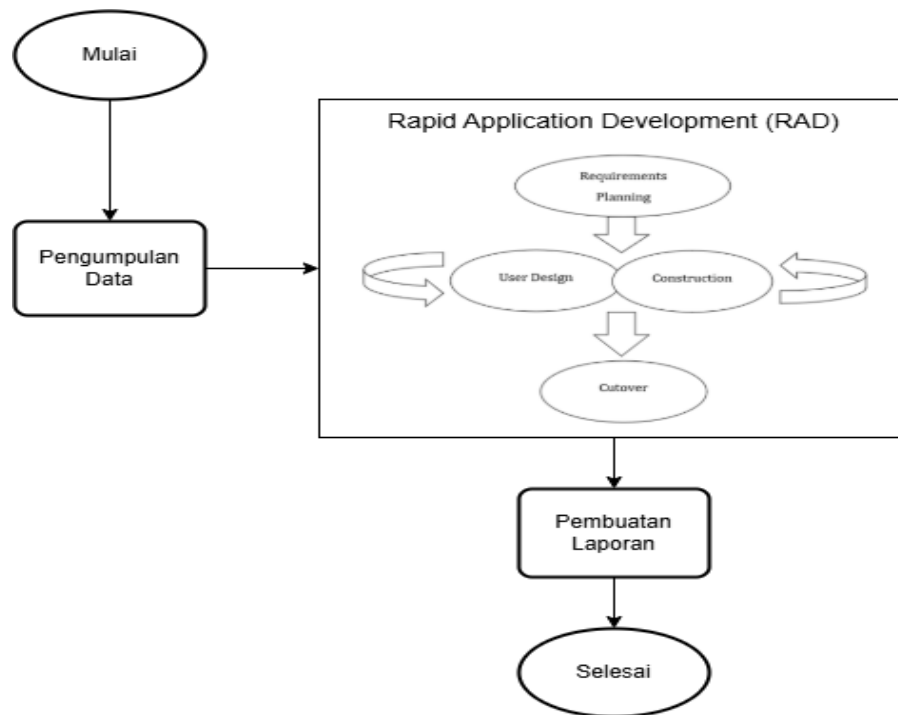
3.3.2 Bahan Penelitian

Dalam penelitian ini, bahan penelitian yang digunakan untuk mendukung proses pengembangan Sistem Informasi Geografis (SIG) kebun sawit di Mesuji berbasis web berupa data peta ortofoto kebun sawit di Kabupaten Mesuji, Provinsi Lampung. Data ortofoto tersebut diperoleh dari Tim Riset URO dan digunakan sebagai dasar dalam pemetaan serta pengolahan data spasial kebun sawit.

3.4 Tahapan Penelitian

Tahapan dalam penelitian ini mencakup proses pengumpulan data, pengembangan sistem, dan pembuatan laporan. Tujuannya adalah menghasilkan sistem *backend* yang mampu mengelola data spasial dan *non-spasial* secara terstruktur dan mendukung pengelolaan informasi geografis secara efisien. Berikut ini adalah

flowchart yang menunjukkan setiap tahapan dalam proses pengembangan *backend WebGIS* untuk kebun sawit:



Gambar 3.2 Tahapan Penelitian

Penelitian ini diawali dengan tahap pengumpulan data yang bertujuan untuk mengidentifikasi permasalahan serta kebutuhan pengguna terhadap sistem yang akan dibangun. Data dikumpulkan melalui observasi, wawancara, maupun studi literatur untuk mendapatkan gambaran yang akurat mengenai kondisi dan proses yang ada. Setelah data terkumpul, penelitian dilanjutkan ke dalam tahapan *Rapid Application Development (RAD)* yang terdiri dari beberapa fase utama. Fase pertama adalah perencanaan kebutuhan, yaitu proses mendefinisikan kebutuhan sistem berdasarkan data yang telah dikumpulkan agar sistem yang dikembangkan sesuai dengan tujuan dan harapan pengguna. Selanjutnya dilakukan desain sistem, di mana rancangan awal sistem disusun dalam bentuk sketsa antarmuka, struktur data, serta alur proses sistem.

Tahap berikutnya adalah pengembangan, yaitu proses pembuatan sistem berdasarkan desain yang telah dirancang. Dalam tahap ini dilakukan pembuatan kode program, pengujian fungsi, serta penyempurnaan sistem secara iteratif hingga

sistem berjalan sesuai harapan. Setelah sistem selesai dikembangkan, tahap selanjutnya adalah implementasi, yaitu menerapkan sistem ke lingkungan kerja sebenarnya. Pada tahap ini juga dilakukan pengujian sistem menggunakan metode *Black Box Testing* untuk memastikan bahwa sistem berfungsi dengan baik dari sisi pengguna. Setelah sistem berhasil diimplementasikan, tahap terakhir dalam penelitian ini adalah pembuatan laporan. Pada tahap ini, seluruh proses, temuan, serta hasil dari pengembangan sistem didokumentasikan dalam bentuk laporan penelitian sebagai bentuk pertanggungjawaban ilmiah dan sebagai referensi untuk pengembangan sistem di masa mendatang. Dengan demikian, penelitian dinyatakan selesai setelah seluruh tahapan tersebut dilalui.

3.4.1 Pengumpulan Data

Pengumpulan data merupakan salah satu tahapan penelitian yang bertujuan untuk memperoleh informasi langsung dari objek penelitian guna menjawab permasalahan yang dikaji. Dalam penelitian ini, penulis menerapkan beberapa metode pengumpulan data, yaitu observasi, wawancara, dan studi literatur.

1. Observasi dilakukan oleh *project manager* pada juni 2023 di area Perkebunan pohon kelapa sawit PT. Bina Sawit Makmur (Sampoerna Agro Group). untuk mengamati proses perhitungan jumlah pohon sawit di kebun sawit Mesuji. Melalui metode ini, diperoleh data dan informasi mengenai proses pendataan pohon sawit yang masih dilakukan secara manual.
2. Wawancara dilakukan secara langsung oleh project manager dengan narasumber yang berperan penting dalam sistem yang akan dikembangkan. Narasumber dalam penelitian ini adalah Manager PT Bina Sawit Makmur (Sampoerna Agro Group), yaitu Bapak Ruli Wandri. Tujuan wawancara ini adalah untuk menggali kebutuhan sistem, mengidentifikasi kendala yang dihadapi dalam pengelolaan data kebun sawit, serta memperoleh data-data eksklusif yang berkaitan dengan pengelolaan kebun sawit.
3. Studi Literatur digunakan untuk mengumpulkan berbagai referensi teoritis yang mendukung penelitian ini. Studi literatur berperan sebagai landasan teori dalam proses pengembangan sistem informasi geografis kebun sawit berbasis *website*.

3.4.2 Pengembangan Sistem

Pengembangan sistem pada penelitian ini menggunakan metode *Rapid Application Development (RAD)* yang terdiri dari empat tahapan utama, yaitu *Requirements Planning*, *User Design*, *Construction*, dan *Cutover*. Pengembangan *backend* dilaksanakan dengan waktu efektif selama ± 3 bulan, yang dialokasikan secara terstruktur pada setiap tahapan RAD guna mendukung proses perancangan, implementasi, pengujian, serta *deployment* sistem *backend*.

1. *Requirements Planning*

a. *User Stories dan Acceptance Criteria*

User story adalah cara sederhana untuk menggambarkan apa yang dibutuhkan sistem dari sudut pandang pengguna awam. *Acceptance criteria* adalah daftar persyaratan yang harus dipenuhi agar produk sesuai dengan kebutuhan pengguna.

Tabel 3.3 *User Story dan Acceptance Criteria*

Kode	<i>User Story</i>	<i>Acceptance Criteria</i>
US-01	Sebagai seorang manajer kebun, saya ingin dapat login ke dalam sistem agar dapat mengakses seluruh fitur pengelolaan data kebun.	• <i>Endpoint</i> dapat menerima data <i>email</i> dan <i>password</i> untuk proses <i>login</i>. • <i>Endpoint</i> dapat menampilkan halaman sesuai <i>role</i> pengguna (manajer kebun dan <i>staff</i>) setelah berhasil <i>login</i>. • <i>Endpoint</i> akan menolak akses jika pengguna tidak memiliki hak akses atau token tidak valid.
US-02	Sebagai seorang manajer kebun, saya ingin pengguna dapat melakukan logout agar sesi pengguna dapat diakhiri dengan aman.	• <i>Endpoint</i> dapat menghapus atau menonaktifkan token autentikasi yang sedang aktif. • <i>Endpoint</i> dapat memastikan token yang telah dihapus tidak dapat digunakan kembali.

		• Sistem menampilkan pesan konfirmasi <i>logout</i> berhasil.
US-03	Sebagai seorang manajer kebun, saya ingin dapat mengunggah dan menampilkan peta kebun dalam format <i>.tif</i> , agar peta dapat divisualisasikan di <i>dashboard</i> .	• <i>Endpoint</i> dapat menerima file peta dalam format <i>.tif</i> atau <i>.tiff</i> untuk diunggah. • <i>Endpoint</i> dapat menampilkan daftar peta yang telah diunggah beserta keterangan file-nya. • <i>Endpoint</i> dapat menampilkan layer peta di <i>dashboard</i> sesuai file yang tersimpan.
US-04	Sebagai seorang manajer kebun, saya ingin mendapatkan data hasil analisis jumlah pohon sawit beserta koordinatnya, agar dapat ditampilkan di <i>dashboard</i> dan divisualisasikan sebagai titik-titik di peta kebun sawit.	• <i>Endpoint</i> dapat mengambil data hasil analisis yang berisi id, koordinat, dan jumlah pohon dari model <i>Machine Learning</i> di <i>database</i>. • <i>Endpoint</i> dapat menampilkan data tersebut di <i>dashboard</i> dalam bentuk tabel. • <i>Endpoint</i> dapat menampilkan titik- titik pohon di layer peta sesuai koordinat yang diterima.
US-05	Sebagai seorang manajer kebun, saya ingin mendapatkan dan mengelola data blok kebun sawit, agar pengguna dapat memantau informasi blok di <i>dashboard</i> .	• <i>Endpoint</i> dapat menampilkan daftar blok beserta informasi detailnya. • <i>Endpoint</i> dapat mengirimkan data lokasi blok dalam format spasial untuk ditampilkan di peta. • <i>Endpoint</i> dapat menampilkan versi terakhir data blok berdasarkan tanggal tertentu.

		• <i>Endpoint</i> dapat menampilkan daftar pekerja yang terkait dengan blok tertentu.
US-06	Sebagai seorang manajer kebun, saya ingin mendapatkan dan mengelola data zona agar pengguna dapat melihat pembagian wilayah kebun di <i>dashboard</i> .	• <i>Endpoint</i> dapat menampilkan daftar zona beserta informasi detailnya (nama zona, luas, dan batas wilayah). • <i>Endpoint</i> dapat menampilkan data zona dalam format spasial untuk ditampilkan di peta. • <i>Endpoint</i> dapat menyimpan, memperbarui, dan menghapus data zona.
US-07	Sebagai seorang manajer pengguna, saya ingin mendapatkan data penggunaan dan riwayat lahan, agar pengguna dapat melihat perubahan penggunaan lahan beserta luasnya di <i>dashboard</i> .	• <i>Endpoint</i> dapat menampilkan daftar lahan beserta luas dan riwayat perubahannya. • <i>Endpoint</i> dapat menyimpan dan memperbarui data penggunaan lahan.
US-08	Sebagai seorang manajer kebun, saya ingin mendapatkan data kondisi saluran irigasi, agar pengguna dapat memantau kondisi irigasi dan memperbaruinya di <i>dashboard</i> .	• <i>Endpoint</i> dapat menampilkan daftar saluran irigasi beserta kondisi dan lokasinya. • <i>Endpoint</i> dapat mengirimkan data kondisi irigasi seperti “normal”, “perlu perhatian”, atau “rusak”. • <i>Endpoint</i> dapat menyimpan dan memperbarui data kondisi saluran irigasi.

US-09	Sebagai seorang manajer kebun, saya ingin mendapatkan dan mengelola data jalan agar pengguna dapat memantau kondisi serta lokasi jalan di <i>dashboard</i> .	• <i>Endpoint</i> dapat menampilkan daftar jalan beserta informasi detailnya (lebar, kondisi, dan lokasi). • <i>Endpoint</i> dapat menampilkan data jalan dalam format spasial untuk ditampilkan di peta. • <i>Endpoint</i> dapat menyimpan, memperbarui, dan menghapus data jalan.
US-10	Sebagai seorang manajer kebun, saya ingin mendapatkan dan mengelola data alat agar pengguna dapat memantau serta memperbarui informasi alat yang digunakan di kebun melalui <i>dashboard</i> .	• <i>Endpoint</i> dapat menampilkan daftar alat beserta informasi detailnya (nama alat, dan jenis penggunaan). • <i>Endpoint</i> dapat menambahkan, memperbarui, dan menghapus data alat.
US-11	Sebagai seorang manajer kebun, saya ingin mendapatkan dan mengelola data transportasi agar pengguna dapat memantau kendaraan operasional kebun di <i>dashboard</i> .	• <i>Endpoint</i> dapat menampilkan daftar transportasi beserta informasi detail (jenis kendaraan, dan kapasitasnya). • <i>Endpoint</i> dapat menyimpan, memperbarui, dan menghapus data transportasi.
US-12	Sebagai seorang manajer kebun, saya ingin mendapatkan dan mengelola data pekerja, agar pengguna dapat menampilkan, menambah, mengubah, dan	• <i>Endpoint</i> dapat menampilkan daftar seluruh pekerja beserta informasinya. • <i>Endpoint</i> dapat menyimpan, memperbarui, dan menghapus data pekerja.

	menghapus data pekerja di <i>dashboard</i> .	• <i>Endpoint</i> dapat memastikan data pekerja yang dihapus tidak menimbulkan relasi yang tidak valid.
US-13	Sebagai seorang manajer kebun, saya ingin mendapatkan dan mengelola data pengguna agar manajer kebun dapat menambahkan, memperbarui, atau menghapus akun pengguna sesuai hak akses.	• <i>Endpoint</i> dapat menampilkan daftar seluruh pengguna beserta perannya (<i>staff</i> dan manajer kebun). • <i>Endpoint</i> dapat menambahkan pengguna baru dengan validasi <i>email</i> dan peran tertentu. • <i>Endpoint</i> dapat memperbarui dan menghapus data pengguna. • <i>Endpoint</i> dapat memastikan autentikasi dan otorisasi berdasarkan <i>role-based access control</i> (RBAC).

b. Analisa Kebutuhan

Pada pengembangan Sistem Informasi Geografis ini, Analisa kebutuhan yang digunakan yaitu kebutuhan fungsional dan *non* fungsional. Berikut ini adalah kebutuhan fungsional dan *non* fungsional berdasarkan hasil analisis:

a) Kebutuhan Fungsional

Kebutuhan fungsional berisi proses yang akan dilakukan oleh sistem. Kebutuhan fungsional dari Sistem Informasi Geografis Kebun Sawit adalah sebagai berikut:

Tabel 3.4 Kebutuhan Fungsional

No.	Kebutuhan Fungsional
1.	Manajer dan <i>Staff</i> kebun dapat melakukan proses <i>login</i> pengguna dengan validasi <i>email</i> dan <i>password</i> .

2.	Manajer dan <i>Staff</i> kebun dapat melakukan <i>logout</i> untuk mengakhiri sesi pengguna secara aman.
3.	Manajer kebun dapat melihat ringkasan data berupa jumlah pohon, luas kebun, dan jumlah pekerja di dashboard.
4.	Manajer kebun dapat melihat dan mengelola <i>upload</i> peta.
5.	Manajer kebun dapat melihat dan mengelola data pohon sawit.
6.	Manajer kebun dapat melihat dan mengelola data lahan perkebunan
8.	Manajer kebun dapat melihat dan mengelola zona.
9.	Manajer kebun dapat melihat jaringan saluran irigasi di peta
10.	Manajer kebun dapat melihat data transportasi.
11.	Manajer kebun dapat melihat dan mengelola data pekerja
12.	Manajer kebun dapat melihat dan mengelola data alat pertanian
13.	Manajer kebun dapat melihat dan mengelola data Jalan
14.	Manajer kebun dapat melihat dan mengelola data pengguna.
15.	<i>Staff</i> kebun dapat melihat halaman utama yang menampilkan informasi ringkas tentang kebun sawit.
16.	<i>Staff</i> kebun dapat melihat deskripsi tentang sistem dan pengelola kebun sawit.
17.	<i>Staff</i> kebun dapat melihat informasi kontak penting terkait pengelola kebun sawit.
18.	<i>Staff</i> kebun dapat melihat peta interaktif yang menampilkan lokasi kebun sawit beserta informasi terkait.

b) Kebutuhan *Non Fungsional*

Tabel 3.5 Kebutuhan *Non Fungsional*

No.	Kebutuhan <i>Non Fungsional</i>	Deskripsi
1.	Performa	Sistem harus dapat memproses permintaan dalam waktu kurang dari 3 detik.
2.	Keamanan	

		Sistem harus dilengkapi dengan autentikasi menggunakan <i>email</i> dan <i>password</i> untuk mengakses fitur manajer kebun dan <i>staff</i> .
3.	Ketersediaan	Sistem dijalankan menggunakan koneksi internet dan berjalan 24 jam sehari.

2. *User Design*

Pada tahap berikutnya, dilakukan proses *user design* yang akan dikembangkan. Selama proses ini, pengguna dapat memberikan masukan terkait prototipe yang telah dikembangkan, sehingga perbaikan dapat dilakukan secara bertahap. Dengan metode ini, pengembangan sistem menjadi lebih efektif dan cepat. Tahapan ini mencakup pembuatan arsitektur perangkat lunak, *use case diagram*, *activity diagram*, dan *class diagram* untuk mendukung perancangan sistem.

a) **Arsitektur Perangkat Lunak**

Arsitektur perangkat lunak merupakan representasi struktural yang menggambarkan hubungan dan interaksi antar komponen dalam sistem, mencakup aspek struktural dan perilaku aplikasi. Tujuannya adalah untuk membantu pengambilan keputusan desain, komunikasi antar *stakeholder*, analisis kinerja, panduan pengembangan, serta identifikasi dan mitigasi risiko teknis. Proses ini diawali dengan identifikasi komponen dan penentuan peran masing-masing.

b) ***Use Case Diagram***

Pembuatan *use case diagram* dalam penelitian ini bertujuan untuk menggambarkan hubungan antara aktor dan sistem yang akan dikembangkan serta menjelaskan proses-proses yang terjadi dalam sistem. Selain itu, *use case diagram* juga digunakan untuk mengidentifikasi kebutuhan fungsional dari sistem secara menyeluruh, sehingga dapat diketahui layanan-layanan apa saja yang harus disediakan oleh sistem untuk memenuhi kebutuhan pengguna.

c) *Activity Diagram*

Activity diagram digunakan untuk menggambarkan alur aktivitas atau proses yang terjadi dalam sistem secara terperinci, mulai dari awal hingga akhir. Diagram ini menunjukkan urutan langkah, keputusan, serta interaksi antara aktor dan sistem. Selain membantu memahami logika dan alur kerja sistem, *activity diagram* juga berfungsi sebagai panduan dalam perancangan dan implementasi agar sistem yang dibangun sesuai dengan kebutuhan pengguna.

3. *Construction*

Tahap *Construction* merupakan fase inti dalam metode *Rapid Application Development (RAD)* yang berfokus pada proses implementasi sistem berdasarkan hasil perencanaan dan perancangan pada tahap sebelumnya. Pada fase ini, dilakukan pengembangan sisi *backend* dari sistem informasi berbasis *web* menggunakan *framework Laravel*. Backend dikembangkan untuk mengelola logika bisnis sistem, serta menyediakan layanan *Application Programming Interface (API)* yang bersifat *RESTful* sebagai penghubung antara antarmuka pengguna (*frontend*), basis data *PostgreSQL*, dan model *Machine Learning* yang digunakan untuk mendukung analisis dalam sistem.

Dalam tahap ini, *backend* bertugas memproses permintaan dari sisi *frontend* yang dibangun dengan *Bootstrap*, *Leaflet*, dan *JavaScript*, kemudian mengakses data dari basis data maupun hasil prediksi dari model *Machine Learning* sesuai kebutuhan. Pengembangan backend juga mencakup proses validasi data, autentikasi dan otorisasi pengguna, serta pengelolaan data melalui operasi *CRUD (Create, Read, Update, Delete)* yang diakses melalui *endpoint API*. Setelah seluruh fungsionalitas dikembangkan, dilakukan proses pengujian terhadap sistem *backend* menggunakan *black box testing* guna memastikan bahwa seluruh layanan berjalan sesuai dengan kebutuhan yang telah ditentukan. Hasil dari tahap ini adalah sistem *backend* yang telah siap diintegrasikan secara menyeluruh.

4. Cutover

Tahap *Cutover* merupakan fase akhir dalam metode *Rapid Application Development (RAD)* yang menandai proses transisi dari pengembangan menuju implementasi penuh sistem. Pada tahap ini, seluruh komponen sistem yang telah dikembangkan, termasuk *backend*, *frontend*, integrasi *database*, serta integrasi dengan model *Machine Learning*, digabungkan dan dipersiapkan untuk digunakan secara nyata. Aktivitas utama pada fase ini meliputi finalisasi pengujian sistem secara menyeluruh menggunakan *blackbox testing*, termasuk pengujian integrasi antarkomponen dan pengujian fungsionalitas sistem berdasarkan skenario penggunaan yang telah ditentukan. Selain itu, dilakukan pula proses *deployment* sistem ke *server* produksi agar dapat diakses oleh pengguna dengan menggunakan *hosting render*. Hasil dari tahap *Cutover* adalah sistem informasi geografis kebun sawit berbasis *web* yang telah siap digunakan secara operasional dan mendukung proses bisnis sesuai dengan tujuan yang ditetapkan dalam proyek.

3.4.3 Penulisan Laporan

Tahap penulisan laporan adalah tahap akhir dari penelitian ini. Pada tahap ini, dilakukan penulisan laporan yang memuat data dan informasi yang diperoleh selama proses penelitian berlangsung.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pengembangan yang telah dilakukan mengenai “*Sistem Informasi Geografis (SIG) dalam Tata Kelola Kebun Sawit (Backend)*”, dapat disimpulkan beberapa hal sebagai berikut:

1. Perancangan *backend WebGIS* untuk sistem informasi geografis kebun sawit di Mesuji telah berhasil dilakukan menggunakan *framework Laravel* dengan basis data *PostgreSQL*. Sistem ini mampu mengelola data spasial dan *non*-spasial secara terstruktur, mencakup entitas seperti blok, zona, lahan, irigasi, jalan, alat, transportasi, pekerja, serta data pohon sawit hasil proses *Machine Learning*. *Backend* juga telah terintegrasi dengan *frontend* sehingga data dapat divisualisasikan dalam peta digital secara interaktif.
2. Seluruh fitur yang dikembangkan telah diuji menggunakan metode *Black Box Testing* dan menunjukkan hasil sesuai dengan kebutuhan serta spesifikasi sistem. Setiap *endpoint API* berfungsi dengan baik dalam proses *CRUD*, autentikasi pengguna, dan pengelolaan data spasial. Hasil pengujian menunjukkan sistem berjalan stabil, dan siap digunakan untuk mendukung pengelolaan kebun sawit secara digital.

5.2 Saran

Adapun saran yang dapat diberikan untuk pengembangan dan penelitian selanjutnya adalah sebagai berikut:

1. Integrasi sistem *backend* dengan model *machine learning* secara langsung dapat dioptimalkan agar proses deteksi pohon, perhitungan jumlah, dan klasifikasi lahan berjalan otomatis tanpa intervensi manual.
2. Sistem *backend WebGIS* ini dapat dikembangkan lebih lanjut dengan menambahkan fitur analisis lanjutan seperti analisis kesehatan tanaman dan estimasi produktivitas kebun. Pengembangan dapat dilakukan melalui integrasi modul *Machine Learning* di sisi *backend* untuk memproses data spasial secara otomatis.

DAFTAR PUSTAKA

- [1] P. D. I. N. MP, *Pertumbuhan Planlet Kelapa Sawit Memiliki Mutu Akar Di Prenursery*. Cv. Azka Pustaka, 2022.
- [2] Direktorat Jenderal Perkebunan, *Statistik Perkebunan Indonesia: Komoditas Kelapa Sawit 2020–2022*. Kementerian Pertanian Republik Indonesia, 2022
- [3] R. F. Tomlinson, *Thinking about GIS: Geographic Information System Planning for Managers*. ESRI, Inc., 2007.
- [4] M. A. Hamdani and S. Utomo, "SISTEM INFORMASI GEOGRAFIS (SIG) PARIWISATA KOTA BANDUNG MENGGUNAKAN GOOGLE MAPS API DAN PHP," *J. Teknol. Inf. Dan Komun.*, vol. 11, no. 1, July 2021.
- [5] T. Berners-Lee and M. Fischetti, *Weaving the web: the original design and ultimative destiny of the World Wide Web by its inventor*, 1. paperback ed., [Repr.]. New York, NY: HarperBusiness, 2011.
- [6] Z. Kuroma and N. D. Saputro, "Perancangan Aplikasi Monitoring Jurnal Kegiatan Badan Pusat Statistik Kota Pekalongan Berbasis Web," in *Proceeding Science and Engineering National Seminar*, 2021.
- [7] R. Nicoara, *How to be a Web Developer: A Complete Beginner's Guide on What to Know and Where to Start*. Berkeley, CA: Apress, 2023.
- [8] S. Uunonen, *Backend as a Service in Web Development*, 2023.
- [9] T. Otwell, *Laravel: From Apprentice To Artisan*. 2013.
- [10] Z. Subecz, "Web-development with Laravel framework," *Gradus*, vol. 8, no. 1, pp. 211–218, 2021.

- [11] B. Momjian, *PostgreSQL: introduction and concepts*. Boston, Mass.: Addison-Wesley, 2001.
- [12] T. S Waruwu, "Implementasi Postgresql Sebagai Sistem Manajemen Basis Data Pada Pendaftaran Mahasiswa Baru Berbasis Web," *J. Mahajana Inf.*, vol. 4, no. 1, pp. 57–61, June 2019.
- [13] L. Richardson and S. Ruby, *RESTful Web Services*. O'Reilly Media, Inc., 2008.
- [14] G. Booch, J. Rumbaugh, and I. Jacobson, *The unified modeling language user guide*. in The Addison-Wesley object technology series. Reading (Mass.): Addison-Wesley, 1999.
- [15] D. W. T. Putra and R. Andriani, "Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD," *J. TeknoIf*, vol. 7, no. 1, p. 32, Apr. 2019.
- [16] D. Westerveld, *API Testing and Development with Postman: A practical guide to creating, testing, and managing APIs for automated software testing*. Packt Publishing Ltd, 2021.
- [17] A. S. Sari and R. Hidayat, "Designing website vaccine booking system using golang programming language and framework React JS," *Journal of Information System, Informatics and Computing*, vol. 6, no. 1, pp. 22–39, 2022.
- [18] G. J. Myers, C. Sandler, and T. Badgett, *The Art of Software Testing*. John Wiley & Sons, 2011.
- [19] J James Martin, *Rapid Application Development*. Macmillan Publishing Co., Inc., 1991.
- [20] D. Aivaliotis, *Mastering Nginx*. Birmingham, UK: Packt Publishing, 2013.
- [21] V. Kholodkov, *Nginx Essentials*. Birmingham, UK: Packt Publishing Ltd., 2015.
- [22] J. D. Rockefeller, *Web Hosting Guide for Beginners*. J.D. Rockefeller, 2016.

- [23] A. Hidayatullah and D. Yusuf, "Pengembangan Backend Sistem Kearsipan Dokumen Menggunakan Framework Laravel di CV. Nakula Sadewa," *Softw. Dev. Digit. Bus. Intell. Comput. Eng.*, vol. 2, no. 02, pp. 61–68, Mar. 2024.
- [24] W. D. Fitriani, A. Hidayat, and I. W. Suardinata, "Perancangan Backend Pada Sistem Informasi Penerimaan Peserta Didik Baru (PPDB) SMK Bina Mandiri Al Qodiriyah Srono," *Jikom J. Inform. Dan Komput.*, vol. 14, no. 2, pp. 146–156, Oct. 2024.
- [25] A. A. Bimantara and R. D. Gunawan, "Sistem Monitoring Produksi Menggunakan Laravel Dan Cork-Bootstrap," *J. Inf. Technol. Softw. Eng. Comput. Sci. ITSECS*, vol. 1, no. 4, pp. 143–153, Oct. 2023.
- [26] I. R. Syaefulloh, "Pembuatan Backend Sistem Informasi Kepegawaian pada PT Erka Service Solusindo untuk Pengguna Admin Berbasis Website Menggunakan Kerangka Kerja Laravel," *Jurnal Teknik Komputer*, vol. 2, no. 2, pp. 123–131, 2023.
- [27] I. Nurjaman, F. S. Utomo, and N. Hermanto, "Penerapan REST API Laravel sebagai Fondasi Back-end Aplikasi G-MOOC 4D," *Journal of Informatics and Interactive Technology*, vol. 1, no. 1, pp. 9–18, 2024.
- [28] F. Fatmawati, A. Qashlim, and A. Kahpi, "Sistem Informasi Lahan Pertanian Kabupaten Polewali Mandar Berbasis Gis," *J. Pegguruang Conf. Ser.*, vol. 5, no. 1, p. 171, May 2023.
- [29] Sekolah Tinggi Teknologi Garut, R. E. G. Rahayu, and A. Fauzi, "Rancang Bangun Sistem Informasi Geografis Hasil Pertanian di Kabupaten Garut," *J. Algoritma*, vol. 17, no. 2, pp. 515–524, Feb. 2021.
- [30] Y. A. Susetyo, "Geospatial API Architecture with Laravel for Agricultural Land Suitability Detection System," *Sistemasi: Jurnal Sistem Informasi*, vol. 12, no. 3, pp. 726–738, 2023..
- [31] L. S. Saputra and H. F. Herdiyatkoko, "Implementasi Rest Api Pada Sistem Pendataan Kapal Menggunakan Laravel 9 Dan Nuxtjs," *Journal Computer and Technology*, vol. 2, no. 1, pp. 10–21, 2024.

- [32] K. Gowell, "Perancangan Web Service REST API Menggunakan PHP dan Framework Laravel di Tenta Tour Salatiga," Doctoral dissertation, 2023.
- [33] D. Winarti, L. S. Lesmana, and T. Pratiwi, "Implementation of Web-Based Geographical Information System Mapping Recipients of Social Assistance in Giri Purno Village," *Bit-Tech*, vol. 6, no. 1, pp. 59–65, Aug. 2023.
- [34] T. K. A and H. Sulistiani, "Geographic Information System for Mapping Agricultural Technology Dissemination," *Proceeding Int. Conf. Relig. Health Educ. Sci. Technol.*, vol. 1, no. 1, pp. 305–311, Aug. 2024.
- [35] I. Yanuarsyah, S. Ahmad, and N. Khalid, "The Designing of Geoinformation Backend-Frontend to Improve Landslide Mitigation Application," *J. Adv. Res. Comput. Appl.*, vol. 37, no. 1, pp. 21–33, Dec. 2024.
- [36] H. Lyu, L. Miao, D. Jiao, and Q. Zhong, "The design of scalable web GIS microservice framework for undergraduate education," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 10, pp. 45–49, 2023.
- [37] M. Fahri, "Melihat Peta Penyebaran Pasien Covid-19 Dengan Kombinasi QGIS Dan Framework Laravel," *J. Teknol. Terpadu*, vol. 6, no. 1, pp. 25–30, July 2020.
- [38] S. T. S. De Silva, R. P. De Silva, and N. D. K. Dayawansa, "An Approach for Common Web GIS Platform for Agriculture Sector Modernization in Sri Lanka," *Trop. Agric. Res.*, vol. 36, no. 1, pp. 39–53, Jan. 2025.
- [39] Forest Management, Polytechnic of Agricultural, Samarinda, Indonesia, A. Juli Kardika, A. Khoirunnita, Salman, Saharuddin, and I. Muliana, "Development Web-GIS of Commodity Information System for Agriculture, Establishment and Forestry in Marangkayu District," *Int. J. Educ. Manag. Eng.*, vol. 12, no. 5, pp. 1–8, Oct. 2022.
- [40] R. Septiani, Y. Y. Joeffie, R. Ardiansyah, S. A. Pratama, and R. Laila, "Geographical Information System Shortes Path Delivery Of Goods Using The Bellman-Ford And Dijkstra Algorithm (Case Study J&T Palu City)," *JUPI*

J. Ilm. Penelit. Dan Pembelajaran Inform., vol. 10, no. 2, pp. 1427–1435, Mar. 2025.

- [41] CCSICT, Isabela State University, San Fabian, Echague, Isabela, Philippines *et al.*, “Drone-based Geographical Information System (GIS) Mapping of Cassava Pythoplasma Disease (CPD) for Precision Agriculture,” *Int. J. Emerg. Technol. Adv. Eng.*, vol. 12, no. 2, pp. 1–9, Feb. 2022.
- [42] R. I. Tarmizi, M. N. Hamidah, and A. Arizal, “Webgis Application For Searching Nu Mosque In East Java Using The Haversine Method,” *JEECS J. Electr. Eng. Comput. Sci.*, vol. 7, no. 2, pp. 1311–1320, Jan. 2023.