

**MATA-MATA DIGITAL DI AREA PARKIR:
PENGEMBANGAN SMART PARKING SYSTEM
BERBASIS AI DENGAN VISUALISASI
STATISTIK DAN SISTEM NOTIFIKASI**

PROYEK AKHIR

Disusun oleh:

Meizua Muhsana

3312301046

Disusun untuk memenuhi salah satu syarat memperoleh gelar Ahli Madya
Teknik Informatika



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM
2026**

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan laporan proyek akhir dengan judul “*Mata-Mata Digital di Area Parkir: Pengembangan Smart Parking System Berbasis AI dengan Visualisasi Statistik dan Sistem Notifikasi*”.

Ucapan terima kasih penulis sampaikan kepada dosen pembimbing, dosen penguji, serta seluruh pihak yang telah memberikan bimbingan, masukan, dan dukungan selama proses penyusunan laporan ini. Tidak lupa, penulis juga menyampaikan terima kasih kepada keluarga serta rekan-rekan yang senantiasa memberikan doa dan semangat.

Laporan ini disusun sebagai salah satu syarat akademik serta sebagai sarana untuk mengembangkan wawasan penulis dalam bidang teknologi berbasis AI. Penulis berharap laporan ini dapat memberikan manfaat bagi pembaca serta menjadi referensi untuk pengembangan penelitian selanjutnya.

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR TABEL	iv
DAFTAR GAMBAR	v
ABSTRAK	vi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan.....	3
1.4. Batasan Masalah.....	3
1.5. Manfaat	3
BAB II TINJAUAN PUSTAKA	4
2.1. Penelitian Terkait	4
2.2. Landasan Teori.....	6
2.2.1. Smart Parking System.....	6
2.2.2. Artificial Intelligence.....	7
2.2.3. YOLO (You Only Look Once)	7
2.2.4. Dataset	8
2.2.5. Proses Anotasi Dataset	9
2.2.6. Evaluasi Deteksi Objek.....	9
2.2.7. Real-Time Processing.....	10
2.2.8. Sistem Notifikasi	11
2.2.9. Laravel	11
2.3. Metode Pengembangan Sistem	12
BAB III METODOLOGI PENELITIAN.....	14
3.1. Analisis Kebutuhan	14
3.1.1. Proses Bisnis Berjalan	14
3.1.2. Gambaran Umum Sistem.....	14
3.1.3. Alur Kerja Sistem Deteksi YOLO.....	15
3.2. Pemodelan Sistem	16
3.2.1. Kebutuhan Fungsional	16
3.2.2. Kebutuhan Non-Fungsional	17
3.2.3. Use Case Diagram	18

3.2.4. ER Diagram.....	19
3.2.5. WireFrame (Sketch)	20
BAB IV IMPLEMENTASI DAN PENGUJIAN	27
4.1. Hasil Implementasi.....	27
4.1.1. Skema Relasional	28
4.1.2. Implementasi <i>User Interface</i> (UI)	28
4.2. Pengujian <i>User Acceptance Testing</i> (UAT)	34
4.2.1. Fungsional <i>Testing</i>	34
4.2.2. Non-Fungsional <i>Testing</i>	48
4.3 Keterbatasan Sistem	55
BAB V KESIMPULAN	56
5.1 Kesimpulan	56
5.2 Saran.....	57
DAFTAR PUSTAKA	58
DAFTAR LAMPIRAN	60

DAFTAR TABEL

Tabel 1. Perbandingan Penelitian Sebelumnya	4
Tabel 2. Kebutuhan Fungsional.....	16
Tabel 3. Kebutuhan Non-Fungsional.....	17
Tabel 4. <i>Black Box Testing Scenario Use Case</i> Registrasi Pengguna	34
Tabel 5. <i>Black Box Testing Scenario Use Case</i> Autentikasi Login	36
Tabel 6. <i>Black Box Testing Scenario Use Case</i> Monitoring Slot Parkir	38
Tabel 7. <i>Black Box Testing Scenario Use Case</i> Visualisasi Statistik Data Parkir	39
Tabel 8. <i>Black Box Testing Scenario Use Case</i> Manajemen Zona Parkir	40
Tabel 9. <i>Black Box Testing Scenario Use Case</i> Manajemen Sub-Zona Parkir	42
Tabel 10. <i>Black Box Testing Scenario Use Case</i> Manajemen Slot Parkir	44
Tabel 11. <i>Black Box Testing Scenario Use Case</i> Pengelolaan Pengguna.....	46
Tabel 12. <i>Black Box Testing Scenario Use Case</i> Notifikasi Slot Parkir.....	47
Tabel 13. <i>Black Box Testing Scenario Use Case</i> Logout	48
Tabel 14. <i>Performance Testing</i>	49
Tabel 15. <i>Security Testing</i>	52

DAFTAR GAMBAR

Gambar 1. Metode Waterfall	13
Gambar 2. Gambaran Umum Sistem	15
Gambar 3. Use Case Diagram	18
Gambar 4. ER Diagram.....	19
Gambar 5. Wireframe Halaman <i>Landing page</i>	20
Gambar 6. Wireframe Halaman Tentang Kami	21
Gambar 7. Wireframe fitur Login.....	21
Gambar 8. Wireframe fitur Registrasi	22
Gambar 9. Wireframe Halaman Beranda	22
Gambar 10. Wireframe fitur Pemantauan Real-time	23
Gambar 11. Wireframe fitur Analisis Data Penggunaan Area Parkir.....	23
Gambar 12. Wireframe Login Admin.....	24
Gambar 13. Wireframe Beranda Admin	24
Gambar 14. Wireframe Kelola Pengguna	25
Gambar 15. Wireframe Kelola Zona Parkir	25
Gambar 16. Wireframe fitur Analisis Data	26
Gambar 17. Wireframe Modal Zona dan Slot Parkir.....	26
Gambar 18. Skema Relasional	28
Gambar 19. Implementasi <i>Landing Page</i>	29
Gambar 20. Implementasi <i>Dashboard</i> Pengguna.....	29
Gambar 21. Implementasi <i>Real-Time</i>	30
Gambar 22. Implementasi Statistik.....	30
Gambar 23. Implementasi Pengaturan	31
Gambar 24. Implementasi <i>Landing Page</i> Admin	31
Gambar 25. Implementasi Manajemen Pengguna	32
Gambar 26. Implementasi Manajemen Zona	32
Gambar 27. Implementasi Manajemen Sub-Zona	33
Gambar 28. Implementasi Manajemen Slot	33
Gambar 29. Implementasi Analisis Data	34

ABSTRAK

Meningkatnya jumlah kendaraan di lingkungan kampus menimbulkan persoalan keterbatasan lahan parkir serta rendahnya efisiensi penggunaannya. Proses pencarian tempat parkir yang masih dilakukan secara manual kerap menyebabkan kemacetan dan pemborosan waktu. Penelitian ini bertujuan mengembangkan sistem parkir cerdas berbasis Artificial Intelligence (AI) dengan fitur analisis parkir serta notifikasi real-time. Metode pengembangan yang digunakan yaitu Waterfall, melalui tahapan berurutan dan sistematis mulai dari analisis kebutuhan hingga evaluasi. Sistem memanfaatkan algoritma YOLO dalam menganalisis area. Hasil deteksi ditampilkan melalui antarmuka web berbasis Laravel yang menampilkan status slot parkir dan mengirimkan notifikasi otomatis. Sistem ini diharapkan dapat membantu pengguna menemukan parkir lebih efisien sekaligus memberikan dukungan analitik bagi pengelola kampus dalam mewujudkan konsep smart campus berkelanjutan.

Kata kunci: *Artificial Intelligence (AI), Waterfall, Laravel, Notifikasi Real-Time, Smart Parking, YOLO.*

BAB I PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi digital mendorong berbagai sektor untuk beralih menuju sistem yang lebih cerdas dan efisien, termasuk pengelolaan fasilitas publik seperti area parkir. Lingkungan kampus merupakan salah satu kawasan yang paling terdampak oleh peningkatan jumlah kendaraan mahasiswa, dosen, dan staf. Kondisi ini menimbulkan persoalan seperti keterbatasan lahan parkir, penumpukan kendaraan pada jam-jam sibuk, serta rendahnya efisiensi penggunaan ruang. Banyak pengguna parkir yang harus berkeliling untuk menemukan slot kosong, yang tidak hanya membuang waktu tetapi juga menambah polusi suara serta emisi kendaraan Caicedo (2010). Proses pencarian manual juga dapat memicu kemacetan lokal. Kondisi ini menunjukkan perlunya sistem parkir modern berbasis teknologi.

Berdasarkan pemetaan area parkir Politeknik Negeri Batam yang dilakukan pada penelitian ini, area parkir kendaraan roda empat atau mobil terdiri atas lima zona dengan total sepuluh subzona dan kapasitas 340 slot parkir mobil. Zona 1 terdiri atas dua subzona dengan masing-masing 30 slot, Zona 2 terdiri atas dua subzona dengan masing-masing 40 slot, Zona 3 terdiri atas dua subzona dengan masing-masing 30 slot, Zona 4 terdiri atas dua subzona dengan masing-masing 30 slot, dan Zona 5 terdiri atas dua subzona dengan masing-masing 40 slot. Berdasarkan pengamatan pada area parkir mobil, tercatat sebanyak 287 kendaraan roda empat pada periode pengamatan. Kepadatan penggunaan parkir paling tinggi terjadi pada pukul 07.00–09.00, yang menjadi periode jam puncak aktivitas parkir mobil. Pada periode tersebut, pengguna membutuhkan rata-rata sekitar 6 menit untuk menemukan slot parkir mobil yang tersedia. Kondisi tersebut menunjukkan bahwa permasalahan utama pada area parkir mobil Politeknik Negeri Batam adalah keterbatasan informasi mengenai ketersediaan slot secara real-time. Pengguna masih perlu melakukan pencarian secara langsung pada area parkir, sehingga proses pencarian slot menjadi kurang efisien terutama pada periode jam puncak.

Smart Parking System berbasis Artificial Intelligence (AI) menawarkan solusi inovatif untuk memantau dan mengelola area parkir secara lebih efisien (Biyik et al., 2021). Sistem ini memanfaatkan kamera sebagai sensor visual yang diproses

oleh algoritma AI untuk mendeteksi kendaraan secara otomatis. Sistem mendeteksi kendaraan menggunakan bounding box, kemudian membandingkan posisi kendaraan dengan koordinat slot parkir untuk menentukan status keterisian slot (Al-Turjman & Malekloo, 2019). Informasi status parkir diperbarui secara real-time sehingga memudahkan pengguna dan membantu mengurangi waktu pencarian serta meningkatkan efisiensi lahan.

Beberapa penelitian telah dilakukan untuk mengetahui manfaat signifikan dari penerapan smart parking. Salah satu penelitian menunjukkan bahwa sistem smart parking mengurangi waktu pencarian parkir dan mengurangi dampak lingkungan akibat cruising (mobil berkeliling mencari parkir). Studi ini memberi bukti bahwa sekitar 30% dari lalu lintas perkotaan dapat terdiri dari kendaraan yang mencari parkir kosong, sehingga smart parking membantu mengurangi waktu dan emisi tersebut (Khan et al., 2020). Namun, sebagian besar sistem yang telah dikembangkan masih berfokus pada proses deteksi kendaraan dan visualisasi status parkir saja. Penelitian ini berfokus pada pengembangan sistem parkir cerdas berbasis AI yang dilengkapi visualisasi statistik parkir serta notifikasi real-time. Sistem diharapkan memberi pengalaman lebih interaktif dan mendukung efisiensi penggunaan lahan parkir di kampus.

Dengan memadukan teknologi pemrosesan AI serta sistem informasi terintegrasi, pengguna dapat memperoleh pengalaman parkir yang lebih efisien dan informatif. Implementasi sistem ini juga berpotensi membantu pihak kampus dalam mengatur distribusi kendaraan, memaksimalkan kapasitas lahan, serta mengurangi waktu tunggu dan kepadatan area tertentu. Oleh karena itu, penelitian ini diharapkan mampu memberikan kontribusi nyata dalam upaya modernisasi fasilitas kampus melalui pemanfaatan teknologi digital yang lebih maju.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah pada pengembangan sistem ini adalah sebagai berikut:

1. Bagaimana mengintegrasikan teknologi AI untuk mendeteksi serta memantau ketersediaan slot parkir secara real-time?
2. Bagaimana merancang sistem visualisasi statistik parkir berbasis data agar dapat menampilkan pola kepadatan area parkir?

3. Bagaimana mengembangkan sistem notifikasi yang dapat memberikan informasi langsung kepada pengguna mengenai ketersediaan slot parkir dan durasi parkir?

1.3. Tujuan

Tujuan dari pengembangan sistem ini adalah sebagai berikut:

1. Mengintegrasikan teknologi AI ke dalam sistem parkir untuk mendeteksi dan memantau ketersediaan slot parkir secara real-time.
2. Menyediakan fitur analisis parkir berbasis data yang mampu menampilkan pola kepadatan area parkir, sehingga pengguna maupun pengelola dapat mengetahui distribusi penggunaan area parkir secara akurat.
3. Menambahkan fitur sistem notifikasi yang memberikan informasi langsung kepada pengguna mengenai ketersediaan slot parkir dan perubahan status slot guna meningkatkan efisiensi dan kenyamanan penggunaan area parkir.

1.4. Batasan Masalah

Agar fokus dalam pengembangan sistem, maka batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Deteksi kendaraan dilakukan menggunakan algoritma AI berbasis YOLO, yang berfungsi untuk mengenali kendaraan serta menandai bounding box.
2. Pengembangan sistem dilakukan menggunakan framework Laravel dengan basis data MySQL.
3. Fitur notifikasi pada sistem dibatasi pada pengiriman notifikasi melalui email kepada pengguna.
4. Penerapan sistem dibatasi pada area parkir kampus Politeknik Negeri Batam, khususnya pada parkir off-street atau area dalam kampus.
5. Penelitian ini hanya berfokus kepada parkir kendaraan roda 4.

1.5. Manfaat

1. Bagi pengguna: Membantu pengguna menemukan slot parkir dengan lebih cepat melalui informasi status slot secara real-time serta notifikasi otomatis ketika slot tersedia, sehingga waktu pencarian dan aktivitas berkeliling dapat diminimalkan.

2. Bagi pengelola: Membantu pengelola memantau kondisi parkir secara real-time dan menganalisis pola penggunaan lahan. Data tersebut dapat digunakan untuk pengambilan keputusan, seperti penataan ulang area parkir, penambahan zona baru, atau pengaturan kapasitas pada jam-jam tertentu.
3. Bagi institusi: Institusi dapat memanfaatkan data dari sistem ini untuk mendukung perencanaan lingkungan kampus yang lebih teratur. Informasi kepadatan area parkir dapat menjadi dasar kerja sama lintas unit dalam pengembangan fasilitas kampus untuk mendukung konsep kampus hijau.

BAB II TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Perbandingan dilakukan dengan sistem yang dikembangkan sebelumnya, di mana sistem tersebut sudah menyediakan fitur registrasi, login, logout, serta pemantauan kendaraan. Namun, pada pengembangan ini menekankan pada efisiensi pengguna, keakuratan data real-time, serta fokus pada privasi dan keamanan data pengguna. Keunggulan utama SPARKING (Smart Parking) terletak pada pendekatan berbasis Artificial Intelligence (AI) dan antarmuka yang lebih responsif dan user-friendly.

Dalam rangka mendukung pengembangan ini, dilakukan studi perbandingan terhadap beberapa aplikasi parkir cerdas lainnya yang memiliki kesamaan dalam hal teknologi sensor, metode pemantauan, serta fitur yang ditawarkan. Studi ini bertujuan untuk mengidentifikasi keunggulan dan kelemahan relatif dari masing-masing sistem guna menjadi bahan evaluasi dan pengembangan lebih lanjut. Komparasi dilakukan terhadap tiga aplikasi berikut:

Tabel 1. Perbandingan Penelitian Sebelumnya

Peneliti/Tahun	Judul/Topik Penelitian	Metode/Teknologi	Kelebihan	Kekurangan
-----------------------	-------------------------------	-------------------------	------------------	-------------------

Abidin dan Pulungan (2020)	Cars Park of Engineering Diyala	Sensor ultrasonik	Akurasi tinggi	Biaya instalasi besar
Saputra et al. (2023)	Vehicle and Parking Space Detection for Smart Parking	YOLOv5 (deep learning / computer vision)	Akurasi deteksi objek tinggi, Langsung menghasilkan bounding box untuk tiap kendaraan/slot	Performa sangat bergantung pada kualitas kamera dan kondisi pencahayaan
Haslinda (2016)	Haphazard Parking	Kombinasi kamera dan RFID	Memiliki fitur reservasi	UI masih kurang intuitif
Meizua Muhsana (2026)	Smart Parking System Berbasis AI dengan Visualisasi Statistik dan Sistem Notifikasi	YOLO, Waterfall	Memiliki fitur analisis dan notifikasi	Jangkauan input AI terbatas, Belum mendukung multi-kamera / edge computing

Sistem smart parking berbasis machine vision yang dikaji oleh Abidin dan Pulungan (2020) menggunakan kamera dan teknik pengolahan citra untuk mendeteksi kendaraan serta ketersediaan slot parkir secara otomatis. Sistem ini mampu menampilkan informasi parkir secara real-time dengan visualisasi yang

kelas, sehingga mengurangi waktu pencarian parkir tanpa memerlukan sensor di setiap slot. Namun, kinerjanya dapat dipengaruhi oleh kondisi pencahayaan dan cuaca, serta membutuhkan komputasi dan pemeliharaan sistem yang relatif tinggi agar tetap akurat.

Penelitian oleh Saputra et al. (2023) mengembangkan sistem deteksi kendaraan dan slot parkir menggunakan algoritma YOLOv5 berbasis computer vision. Sistem ini mampu mengenali kendaraan dan menandai slot parkir melalui bounding box, sehingga informasi ketersediaan dapat ditampilkan secara real-time dan mudah diintegrasikan ke antarmuka pengguna. Kelebihan utama metode ini adalah akurasi deteksi yang tinggi serta kemampuan bekerja secara cepat pada perangkat yang memadai. Namun, performanya sangat dipengaruhi kualitas kamera dan pencahayaan, serta membutuhkan proses pelatihan model yang cukup kompleks. Selain itu, kebutuhan komputasi yang tinggi dapat menjadi kendala jika diterapkan pada perangkat dengan sumber daya terbatas.

Haslinda et al. (2016) melakukan penelitian dengan sistem yang menggunakan kamera untuk mendeteksi ketersediaan slot parkir serta mengontrol akses masuk dan keluar kendaraan secara otomatis bernama Haphazard Parking. Sistem ini memberikan informasi secara real-time, sehingga pengemudi dapat langsung mengetahui slot kosong yang tersedia. Selain itu, keamanan sistem cukup baik karena mampu mengidentifikasi kendaraan berdasarkan plat nomor atau kartu RFID yang terdaftar, mencegah akses oleh kendaraan yang tidak diizinkan. Fitur pemesanan tempat parkir juga tersedia untuk meningkatkan kenyamanan pengguna. Namun demikian, antarmuka halaman login masih kurang intuitif sehingga dapat membingungkan pengguna dalam mengakses layanan.

Berdasarkan perbandingan tersebut, penelitian ini mengambil keunggulan dari metode YOLO dalam akurasi deteksi serta menambahkan fitur visualisasi statistik dan notifikasi yang belum banyak dikaji sebelumnya.

2.2. Landasan Teori

2.2.1. Smart Parking System

Smart Parking System merupakan sistem berbasis teknologi yang dirancang untuk memantau, mengelola, dan mengoptimalkan penggunaan area parkir dengan

memanfaatkan integrasi perangkat keras, perangkat lunak, jaringan, dan teknologi pengolahan data. Sistem ini dapat digunakan untuk memperoleh informasi mengenai kondisi area parkir secara otomatis dan menyajikan informasi tersebut kepada pengguna.

Penerapan Smart Parking System dapat membantu meningkatkan efisiensi penggunaan area parkir karena informasi mengenai ketersediaan slot dapat diperoleh secara lebih cepat. Sistem parkir cerdas juga dapat mengurangi ketergantungan terhadap pengawasan manual serta membantu mengurangi waktu yang dibutuhkan pengguna dalam mencari slot parkir. Dalam penelitian ini, Smart Parking System diterapkan pada area parkir kendaraan roda empat di lingkungan kampus Politeknik Negeri Batam dengan memanfaatkan teknologi Artificial Intelligence untuk mendeteksi kendaraan dan menentukan status slot parkir secara otomatis.

2.2.2. Artificial Intelligence

Artificial Intelligence (AI) merupakan bidang ilmu komputer yang berfokus pada pengembangan sistem yang mampu meniru kemampuan kecerdasan manusia, seperti pengenalan pola, pengambilan keputusan, dan pemrosesan informasi. AI dapat mempelajari pola dari data dan menghasilkan prediksi melalui penerapan algoritma pembelajaran mesin dan deep learning (Russell dan Norvig, 2021).

Dalam sistem parkir cerdas, AI dapat dimanfaatkan untuk menganalisis citra area parkir dan mengenali objek kendaraan secara otomatis. Hasil pengenalan tersebut kemudian dapat digunakan untuk menentukan kondisi keterisian slot parkir. Pada penelitian ini, AI digunakan sebagai teknologi inti dalam proses deteksi kendaraan roda empat menggunakan algoritma YOLOv8.

2.2.3. YOLO (You Only Look Once)

YOLO (You Only Look Once) merupakan algoritma object detection berbasis deep learning yang digunakan untuk mendeteksi objek pada citra. YOLO memiliki karakteristik utama berupa kemampuan melakukan proses deteksi dalam satu kali pemrosesan atau single pass. Berbeda dengan metode deteksi yang melakukan proses pencarian objek secara bertahap, YOLO memproses citra secara menyeluruh untuk menghasilkan prediksi objek.

Hasil deteksi YOLO terdiri atas kelas objek, lokasi objek, dan nilai confidence. Lokasi objek direpresentasikan dalam bentuk bounding box yang digunakan untuk menandai posisi objek pada citra. Nilai confidence menunjukkan tingkat keyakinan model terhadap hasil deteksi yang dihasilkan.

Pada penelitian ini, YOLOv8 digunakan untuk mendeteksi kendaraan roda empat pada area parkir. Citra area parkir diproses oleh model YOLOv8 untuk mengenali keberadaan kendaraan. Setiap kendaraan yang terdeteksi akan ditandai menggunakan bounding box. Hasil deteksi tersebut kemudian digunakan sebagai dasar untuk menentukan status keterisian slot parkir.

YOLOv8 dipilih karena memiliki kemampuan deteksi objek yang cepat dan mendukung kebutuhan sistem yang memerlukan pemrosesan data secara real-time. Kecepatan proses deteksi memungkinkan perubahan kondisi kendaraan pada area parkir dapat diproses dan digunakan untuk memperbarui informasi ketersediaan slot parkir.

2.2.4. Dataset

Dataset merupakan kumpulan data yang digunakan sebagai bahan dalam proses pelatihan dan pengujian model. Dalam sistem deteksi objek, dataset digunakan untuk memberikan contoh kepada model mengenai objek yang harus dikenali. Kualitas dan keberagaman dataset dapat memengaruhi kemampuan model dalam mengenali objek pada kondisi yang berbeda.

Pada penelitian ini, dataset digunakan untuk melatih model YOLOv8 agar mampu mengenali kendaraan roda empat pada area parkir. Dataset berupa citra area parkir yang diperoleh melalui kamera CCTV dan kamera smartphone. Citra yang digunakan menggambarkan kondisi area parkir dengan objek kendaraan yang menjadi fokus deteksi.

Sebelum digunakan dalam proses pelatihan, dataset dipersiapkan melalui proses anotasi dan pembagian data. Dataset dibagi menjadi tiga bagian, yaitu data training, data validation, dan data testing. Data training digunakan untuk melatih model, data validation digunakan untuk memantau performa model selama proses pelatihan, sedangkan data testing digunakan untuk mengukur kemampuan model dalam mendeteksi objek pada data yang tidak digunakan dalam proses pelatihan.

Pembagian dataset pada penelitian ini menggunakan komposisi 70% data training, 20% data validation, dan 10% data testing. Pembagian tersebut dilakukan agar model memiliki data yang cukup untuk proses pembelajaran serta tersedia data terpisah untuk proses validasi dan evaluasi.

2.2.5. Proses Anotasi Dataset

Anotasi dataset merupakan proses pemberian informasi atau label pada objek yang terdapat dalam citra. Pada proses deteksi objek, anotasi digunakan untuk menunjukkan lokasi objek dan kelas objek yang harus dipelajari oleh model. Informasi anotasi menjadi acuan bagi model dalam memahami hubungan antara citra dan objek yang terdapat di dalamnya.

Pada penelitian ini, proses anotasi dilakukan dengan memberikan bounding box pada objek kendaraan roda empat. Bounding box digunakan untuk menandai area kendaraan pada citra. Setiap citra yang telah diberi anotasi memiliki informasi mengenai posisi objek yang menjadi target deteksi.

Proses anotasi dilakukan menggunakan LabelImg dan Roboflow. Hasil anotasi kemudian digunakan sebagai bagian dari dataset untuk melatih model YOLOv8. Proses ini penting karena kesalahan dalam menentukan posisi bounding box atau kelas objek dapat memengaruhi proses pembelajaran model dan hasil deteksi yang dihasilkan.

Setelah proses anotasi selesai, data diperiksa dan dipersiapkan sebelum digunakan dalam proses pelatihan. Pemeriksaan dilakukan untuk memastikan bahwa objek kendaraan telah ditandai dengan benar dan data dapat digunakan oleh model YOLOv8.

2.2.6. Evaluasi Deteksi Objek

Evaluasi deteksi objek dilakukan untuk mengukur kemampuan model dalam mengenali objek pada citra. Evaluasi diperlukan untuk mengetahui tingkat ketepatan hasil deteksi dan kemampuan model dalam menemukan objek yang terdapat pada data pengujian.

Pada penelitian ini, evaluasi model YOLOv8 menggunakan metrik Precision dan Recall. Precision digunakan untuk mengukur tingkat ketepatan hasil deteksi model. Nilai Precision menunjukkan perbandingan antara deteksi objek

yang benar dengan seluruh objek yang diprediksi oleh model. Semakin tinggi nilai Precision, semakin kecil kemungkinan model menghasilkan deteksi objek yang salah.

Recall digunakan untuk mengukur kemampuan model dalam menemukan objek yang seharusnya terdeteksi. Nilai Recall menunjukkan perbandingan antara objek yang berhasil dideteksi dengan seluruh objek yang terdapat pada data sebenarnya. Semakin tinggi nilai Recall, semakin baik kemampuan model dalam menemukan objek pada citra.

Evaluasi Precision dan Recall penting diterapkan pada sistem parkir karena hasil deteksi kendaraan digunakan untuk menentukan status slot parkir. Kesalahan deteksi dapat menyebabkan slot yang terisi dianggap kosong atau slot yang kosong dianggap terisi. Oleh karena itu, hasil evaluasi deteksi menjadi salah satu indikator dalam menilai kemampuan model YOLOv8 yang digunakan pada sistem.

2.2.7. Real-Time Processing

Real-time processing merupakan proses pengolahan data yang dilakukan secara langsung dengan waktu pemrosesan yang relatif cepat sehingga informasi dapat diterima atau ditampilkan tanpa jeda yang signifikan. Pada sistem parkir, pemrosesan real-time diperlukan agar perubahan kondisi kendaraan dan slot parkir dapat diketahui dengan cepat.

Dalam penelitian ini, proses real-time dimulai ketika kamera mengambil citra area parkir secara berkala. Citra tersebut kemudian diproses oleh model YOLOv8 yang berjalan pada Raspberry Pi 5. Model melakukan deteksi terhadap kendaraan roda empat dan menghasilkan informasi mengenai objek yang terdeteksi.

Hasil deteksi selanjutnya diproses oleh layanan Flask dan digunakan untuk menentukan status slot parkir berdasarkan koordinat slot yang telah ditentukan. Data hasil deteksi kemudian diteruskan ke sistem dan ditampilkan melalui aplikasi web berbasis Laravel. Alur tersebut memungkinkan informasi ketersediaan slot parkir diperbarui secara berkala.

Penerapan real-time processing pada penelitian ini bertujuan untuk mengurangi ketergantungan terhadap pengawasan manual. Dengan adanya pemrosesan secara langsung, pengguna dapat memperoleh informasi mengenai

kondisi slot parkir dan pengelola dapat memantau perubahan penggunaan area parkir secara lebih efisien.

2.2.8. Sistem Notifikasi

Sistem notifikasi merupakan mekanisme penyampaian informasi secara otomatis kepada pengguna berdasarkan kondisi atau peristiwa tertentu yang terjadi pada sistem. Dalam sistem parkir, notifikasi dapat digunakan untuk memberikan informasi mengenai perubahan status atau ketersediaan slot parkir.

Integrasi API dapat digunakan untuk mengirimkan pesan secara otomatis melalui server yang terhubung sehingga penyampaian informasi dapat dilakukan dengan lebih cepat dan efektif (Hossain et al., 2020). Pada penelitian ini, sistem notifikasi digunakan untuk memberikan informasi kepada pengguna ketika terdapat slot parkir yang tersedia pada zona yang telah dipilih.

Mekanisme notifikasi dimulai ketika sistem mendeteksi perubahan status slot parkir. Apabila sistem mendeteksi adanya slot yang tersedia dan pengguna telah mendaftarkan pemberitahuan pada zona tersebut, data pengguna akan diproses oleh sistem. Selanjutnya, sistem mengirimkan notifikasi melalui email kepada pengguna yang terdaftar.

Pengiriman notifikasi pada penelitian ini menggunakan integrasi Laravel dengan layanan Brevo sebagai penyedia layanan email. Dengan mekanisme tersebut, pengguna tidak perlu melakukan pemantauan secara terus-menerus terhadap halaman sistem untuk mengetahui perubahan ketersediaan slot parkir. Notifikasi dapat diterima secara otomatis ketika kondisi yang sesuai dengan pengaturan pemberitahuan pengguna terpenuhi.

2.2.9. Laravel

Laravel merupakan framework pengembangan aplikasi web berbasis PHP yang menerapkan arsitektur Model-View-Controller (MVC). Laravel menyediakan berbagai fitur pendukung pengembangan aplikasi seperti routing, pengelolaan basis data, autentikasi, serta REST API (Otwell, 2020).

Pada penelitian ini, Laravel digunakan sebagai framework utama dalam pengembangan sistem SPARKING. Laravel berfungsi untuk mengelola proses

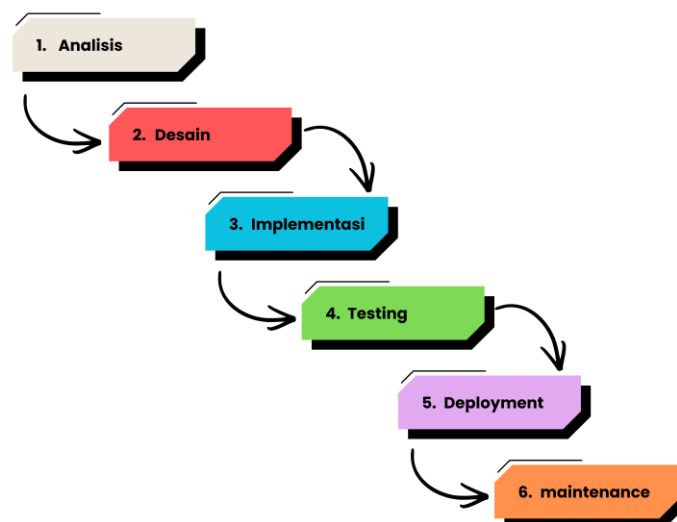
backend, autentikasi pengguna, pengelolaan data zona, subzona, dan slot parkir, serta integrasi hasil deteksi AI dengan sistem web.

Arsitektur MVC pada Laravel memisahkan proses pengelolaan data, tampilan, dan logika aplikasi. Model digunakan untuk mengelola data dan interaksi dengan basis data, View digunakan untuk menampilkan antarmuka kepada pengguna, sedangkan Controller digunakan untuk mengatur alur proses aplikasi. Dengan penerapan Laravel, pengembangan sistem dapat dilakukan secara lebih terstruktur.

2.3. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode Waterfall. Metode ini merupakan model pengembangan perangkat lunak yang menerapkan tahapan pengembangan secara berurutan dan sistematis, sehingga setiap tahap harus diselesaikan sebelum melanjutkan ke tahap berikutnya (Bassil, 2012). Pendekatan ini memungkinkan proses pengembangan dilakukan secara terstruktur dan terdokumentasi dengan baik sehingga memudahkan pengendalian setiap tahapan pengembangan sistem.

Metode Waterfall dianggap sesuai untuk penelitian ini karena pengembangan sistem parkir berbasis Artificial Intelligence memerlukan tahapan analisis, perancangan, implementasi, dan pengujian yang jelas sebelum sistem diterapkan. Dengan pendekatan ini, setiap proses pengembangan dapat terdokumentasi dengan baik dan meminimalkan kesalahan pada tahap selanjutnya.



Gambar 1. Metode Waterfall

1. Analisis

Mengidentifikasi kebutuhan pengguna melalui observasi lapangan dan wawancara terkait proses parkir, kendala lapangan, serta informasi apa saja yang perlu ditampilkan dalam sistem. Hasil analisis kebutuhan ini menjadi dasar dalam merancang fitur utama sistem.

2. Desain

Merancang struktur dan alur sistem berdasarkan kebutuhan yang telah dikumpulkan. Perancangan meliputi pembuatan diagram use case, ERD, serta desain antarmuka sistem. Tahapan ini bertujuan memberikan gambaran mengenai bagaimana sistem akan bekerja dan bagaimana data diproses di dalam sistem.

3. Implementasi

Mengembangkan sistem berdasarkan hasil perancangan yang telah dibuat pada tahap sebelumnya. Pada tahap ini dilakukan proses pengkodean program, integrasi sistem dengan perangkat kamera dan modul deteksi kendaraan menggunakan YOLO, serta pembangunan antarmuka web untuk menampilkan informasi slot parkir.

4. Testing

Melakukan pengujian sistem menggunakan metode Black Box Testing untuk memastikan setiap fitur berjalan sesuai dengan kebutuhan fungsional. Selain itu, dilakukan User Acceptance Testing (UAT) dengan melibatkan pengguna untuk menilai kesesuaian sistem terhadap kebutuhan pengguna.

5. Deployment

Setelah sistem dinyatakan berjalan dengan baik, sistem kemudian diterapkan pada lingkungan penelitian, yaitu area parkir kampus. Pada tahap ini pengguna dapat mengakses sistem untuk melihat informasi ketersediaan slot parkir secara langsung.

6. Maintenance

Tahap terakhir adalah melakukan evaluasi dan pemeliharaan sistem. Evaluasi dilakukan berdasarkan akurasi deteksi kendaraan, kecepatan sistem dalam menampilkan informasi parkir, serta stabilitas sistem selama digunakan.

BAB III METODOLOGI PENELITIAN

3.1. Analisis Kebutuhan

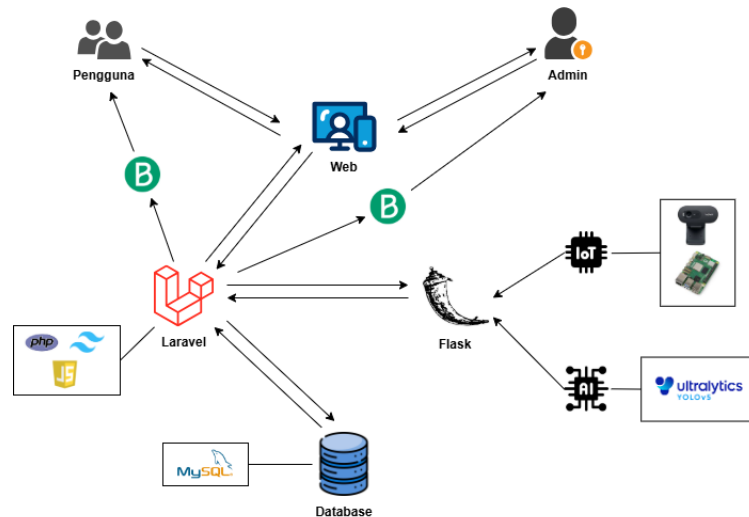
3.1.1. Proses Bisnis Berjalan

Proses bisnis yang diangkat mencakup pengelolaan parkir kendaraan pada area parkir kampus. Sistem ini memantau slot parkir yang tersedia, menampilkan informasi secara real-time kepada pengguna, dan membantu pengelola untuk menganalisis kepadatan serta penggunaan area parkir secara efisien.

3.1.2. Gambaran Umum Sistem

Sistem parkir di kampus yang masih mengandalkan pengawasan manual mengakibatkan pengguna kesulitan memperoleh informasi ketersediaan parkir secara langsung. Kondisi tersebut menimbulkan waktu tunggu lama dan kemacetan di area tertentu. Untuk mengatasi hal tersebut, penelitian ini mengembangkan aplikasi web berbasis AI yang mampu memantau area parkir secara otomatis. Melalui algoritma YOLO (You Only Look Once), data hasil deteksi dikirim ke server, disimpan dalam basis data MySQL, dan divisualisasikan pada dashboard Laravel secara real-time. Sistem juga dilengkapi notifikasi email otomatis saat terdapat slot kosong serta dashboard analitik untuk memantau statistik kepadatan dan penggunaan area parkir.

Pada penelitian ini, sistem masih menggunakan satu alur pemrosesan deteksi AI yang terdiri atas kamera, Raspberry Pi 5, YOLOv8, dan layanan Flask. Sistem belum menyediakan konfigurasi multi-kamera yang memungkinkan setiap zona atau sub-zona menggunakan kamera yang berbeda secara independen. Dengan demikian, apabila terdapat dua zona dengan sumber kamera yang berbeda, sistem belum memiliki pengaturan untuk menentukan dan menghubungkan kamera tertentu pada masing-masing zona. Keterbatasan ini menjadi batasan pada pengembangan sistem yang dilakukan.



Gambar 2. Gambaran Umum Sistem

3.1.3. Alur Kerja Sistem Deteksi YOLO

Proses deteksi kendaraan dimulai ketika kamera mengambil citra area parkir secara berkala. Citra yang diperoleh kemudian diproses oleh model YOLOv8 yang berjalan pada perangkat Raspberry Pi 5 untuk mendeteksi keberadaan kendaraan roda empat pada area parkir secara *real-time*. Setiap kendaraan yang berhasil dikenali akan ditandai dengan *bounding box* sehingga posisi kendaraan dapat diketahui.

Hasil deteksi kendaraan tersebut selanjutnya dibandingkan dengan koordinat masing-masing slot parkir yang telah ditentukan sebelumnya. Apabila pada area suatu slot terdeteksi terdapat kendaraan, maka status slot akan ditandai sebagai terisi. Sebaliknya, apabila tidak terdapat kendaraan pada area slot tersebut, maka status slot akan ditandai sebagai kosong.

Informasi status slot yang diperoleh kemudian diproses oleh layanan Flask yang berjalan pada Raspberry Pi 5. Agar layanan Flask dapat diakses dari internet, digunakan Cloudflare Tunnel yang menyediakan domain publik secara aman. Domain tersebut selanjutnya digunakan oleh *backend* Laravel untuk berkomunikasi dengan layanan deteksi pada Raspberry Pi dan memperoleh informasi terbaru mengenai status slot parkir.

Data status slot yang diterima kemudian disimpan ke dalam basis data MySQL dan ditampilkan secara *real-time* pada aplikasi web. Selain digunakan

untuk menampilkan informasi ketersediaan slot parkir, data tersebut juga dimanfaatkan untuk menghasilkan analisis penggunaan area parkir serta mendukung pengiriman notifikasi kepada pengguna yang telah berlangganan notifikasi. Dengan demikian, sistem yang dikembangkan mampu menyediakan informasi ketersediaan slot parkir secara cepat dan terintegrasi.

3.2. Pemodelan Sistem

3.2.1. Kebutuhan Fungsional

Tabel 2 menjelaskan kebutuhan fungsional sistem (Functional Requirements) yang harus diimplementasikan:

Tabel 2. Kebutuhan Fungsional

Kode	Kebutuhan Fungsional	Deskripsi
FR-01	Registrasi pengguna	Sistem memungkinkan pengguna membuat akun baru untuk mengakses fitur aplikasi.
FR-02	Autentikasi login	Sistem menyediakan mekanisme login bagi pengguna dan admin.
FR-03	Monitoring slot parkir	Sistem menampilkan status ketersediaan slot parkir.
FR-04	Visualisasi Statistik data parkir	Pengguna dapat melihat laporan statistik penggunaan lahan parkir (kepadatan, durasi, waktu puncak).
FR-05	Manajemen zona parkir	Admin dapat menambah, mengubah, dan menghapus zona.
FR-06	Manajemen sub-zona parkir	Admin dapat menambah, mengubah, dan menghapus sub-zona.
FR-07	Manajemen slot parkir	Admin dapat menambah, mengubah, dan menghapus slot.

FR-08	Pengelolaan pengguna	Admin dapat mengelola data pengguna yang terdaftar.
FR-09	Sistem notifikasi slot kosong	Sistem mengirimkan notifikasi otomatis kepada pengguna yang telah mendaftar pada fitur pemberitahuan slot kosong ketika area parkir penuh. Notifikasi akan dikirim ketika sistem mendeteksi adanya slot parkir yang tersedia untuk digunakan.
FR-10	Logout	Sistem menyediakan fitur keluar dari aplikasi dengan aman.

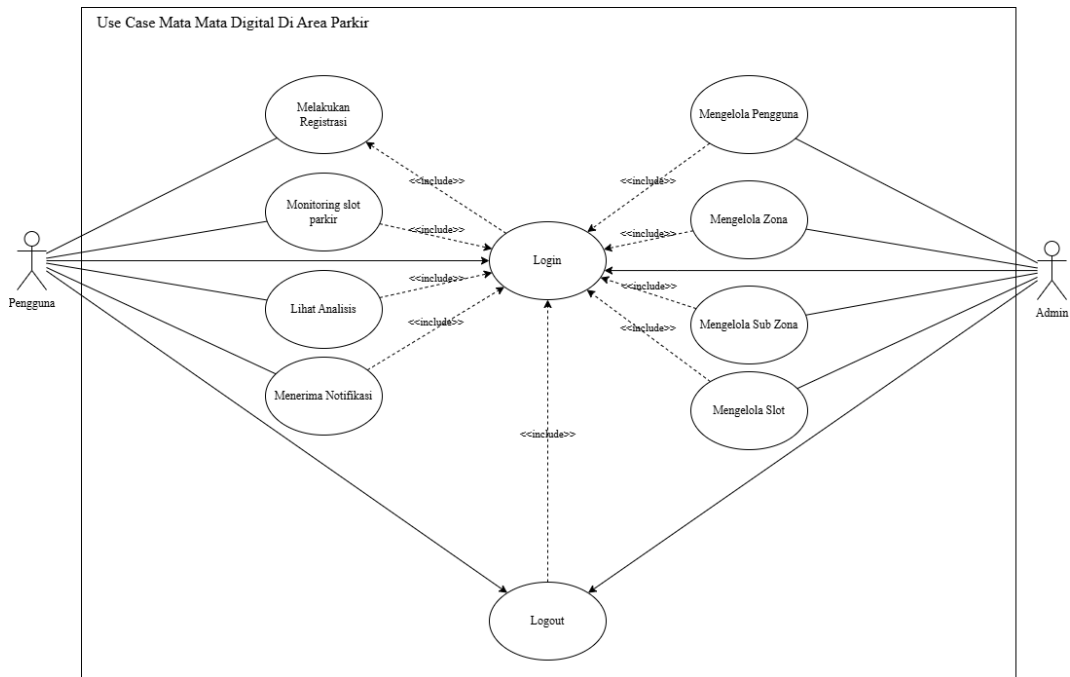
3.2.2. Kebutuhan Non-Fungsional

Tabel 3 menunjukkan kebutuhan non-fungsional sistem yang harus diimplementasikan:

Tabel 3. Kebutuhan Non-Fungsional

Kode	Kebutuhan Non-Fungsional	Deskripsi
NFR-01	Keamanan Data	Kata sandi pengguna disimpan secara terenkripsi dan akses data dilakukan melalui mekanisme autentikasi yang aman.
NFR-02	Performance	Sistem harus mampu menampilkan halaman dashboard dan streaming kamera dengan waktu respon maksimal 5 detik.

3.2.3. Use Case Diagram



Gambar 3. Use Case Diagram

Use case diagram di atas menggambarkan interaksi antara pengguna dan sistem pada aplikasi Mata Mata Digital di Area Parkir. Terdapat dua aktor yang terlibat dalam sistem ini, yaitu Pengguna dan Admin. Pengguna merupakan pengguna umum yang memanfaatkan sistem untuk memantau kondisi parkir, sedangkan Admin adalah pihak yang bertanggung jawab dalam mengelola seluruh data yang ada di dalam sistem.

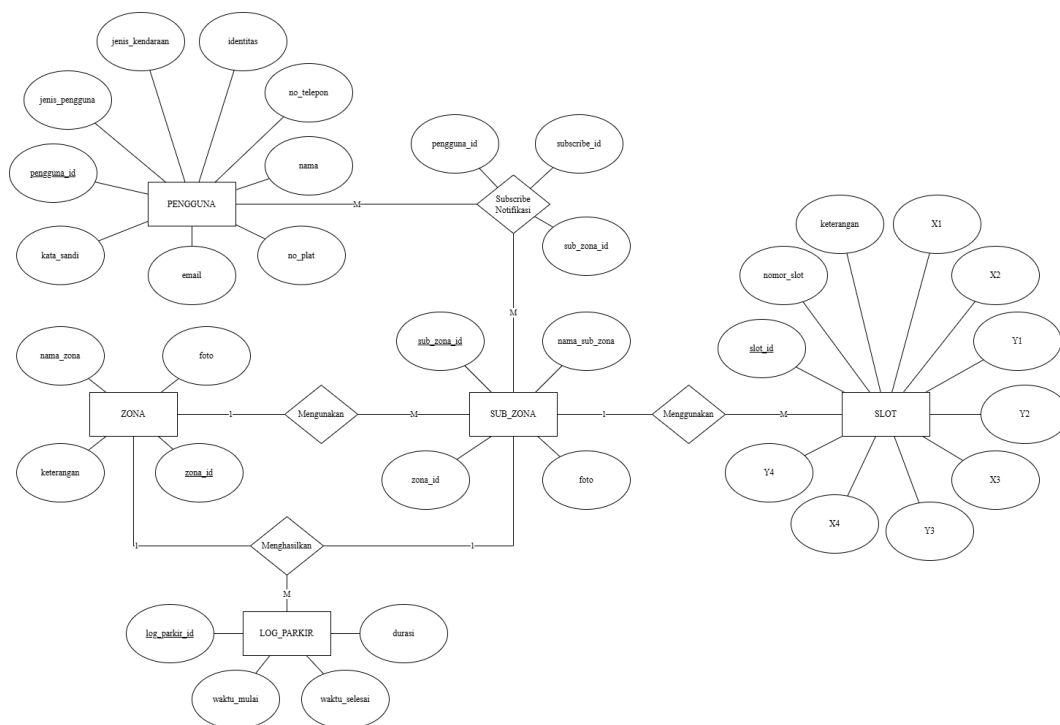
Pada sisi Pengguna, terdapat empat use case utama yang dapat diakses, yaitu Melakukan Registrasi, Monitoring Slot Parkir, Lihat Analisis, dan Menerima Notifikasi. Use case Melakukan Registrasi dapat diakses langsung tanpa perlu login terlebih dahulu karena merupakan tahap awal pembuatan akun. Sementara itu, use case Monitoring Slot Parkir, Lihat Analisis, dan Menerima Notifikasi memiliki relasi *include* terhadap use case Login, yang berarti ketiga fitur tersebut hanya dapat digunakan setelah pengguna berhasil melakukan autentikasi ke dalam sistem.

Pada sisi Admin, terdapat empat use case yang dapat dilakukan, yaitu Mengelola Pengguna, Zona, Sub-Zona, dan Slot. Keempat use case tersebut juga memiliki relasi *include* terhadap use case Login, sehingga Admin diwajibkan untuk

login terlebih dahulu sebelum dapat mengakses fitur pengelolaan data tersebut. Fitur pengelolaan ini mencakup proses menambah, mengubah, dan menghapus data zona, sub-zona, maupun slot parkir yang tersedia dalam sistem.

Selain itu, terdapat use case Logout yang dapat dilakukan oleh kedua aktor, yaitu Pengguna maupun Admin. Use case Logout memiliki relasi *include* dengan use case Login, yang menandakan bahwa proses logout hanya dapat dilakukan ketika sesi login sedang aktif. Dengan adanya mekanisme Login dan Logout ini, sistem dapat menjamin keamanan akses sehingga hanya pengguna yang terautentikasi yang dapat menggunakan fitur-fitur yang tersedia.

3.2.4. ER Diagram



Gambar 4. ER Diagram

1. PENGGUNA

Menyimpan data seluruh pengguna sistem parkir, dengan atribut pengguna_id (PK), nama, email, kata_sandi, no_telepon, no_plat, identitas, role, jenis_pengguna, jenis_kendaraan, foto_pengguna, foto_kendaraan.

2. ZONA

Merepresentasikan area parkir tingkat pertama (zona utama), dengan atribut zona_id (pk), nama_zona, foto, keterangan.

3. SUB_ZONA

Merepresentasikan pembagian lebih detail dari setiap zona, dengan atribut sub_zona_id (pk), zona_id (fk), nama_sub_zona, foto.

4. SLOT

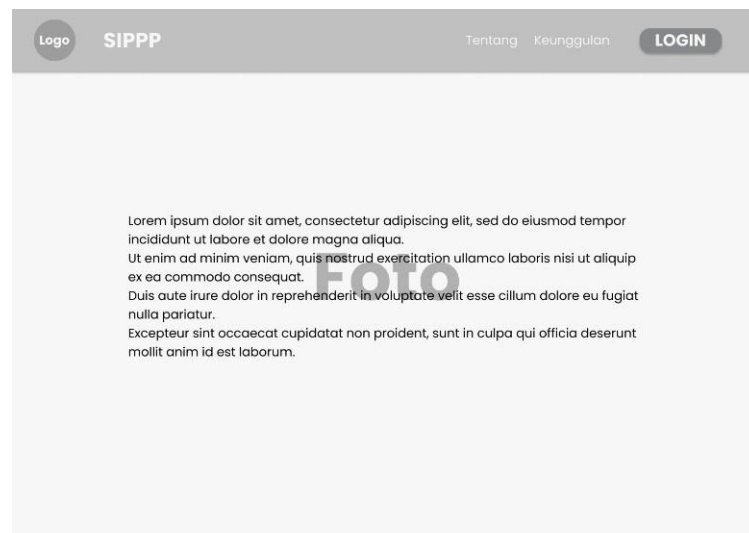
Merepresentasikan slot/tempat parkir individual di dalam sub-zona, dengan atribut slot_id (pk), nomor_slot, keterangan, X1, X2, X3, X4, Y1, Y2, Y3, Y4. Atribut X dan Y merepresentasikan koordinat posisi slot pada peta/denah parkir.

5. LOG_PARKIR

Menyimpan riwayat aktivitas parkir, dengan atribut log_parkir_id (pk), waktu_mulai, waktu_selesai, durasi.

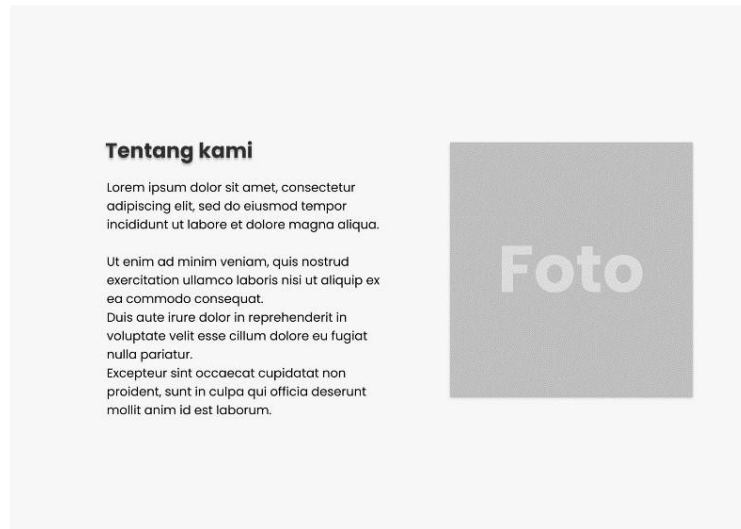
3.2.5. WireFrame (Sketch)

a. Pengguna



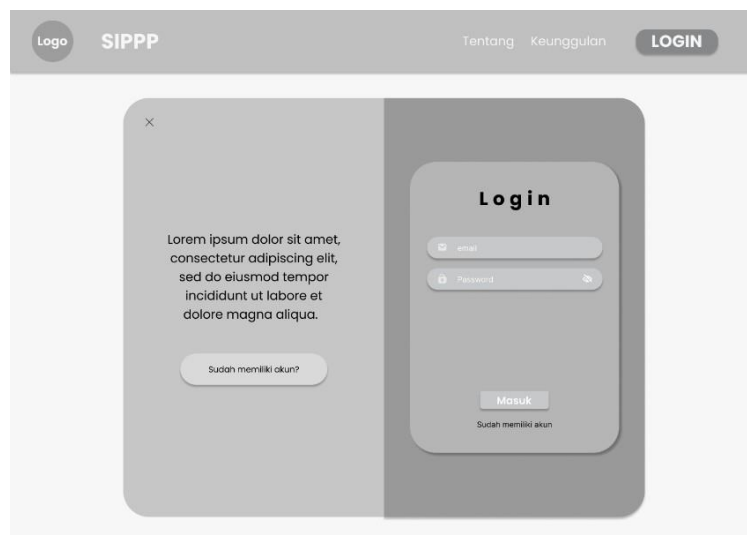
Gambar 5. Wireframe Halaman *Landing page*

Gambar di atas merupakan sketsa halaman *landing page*. Pada sketsa ini terdapat logo, beserta foto dengan teks berada di tengah halaman.



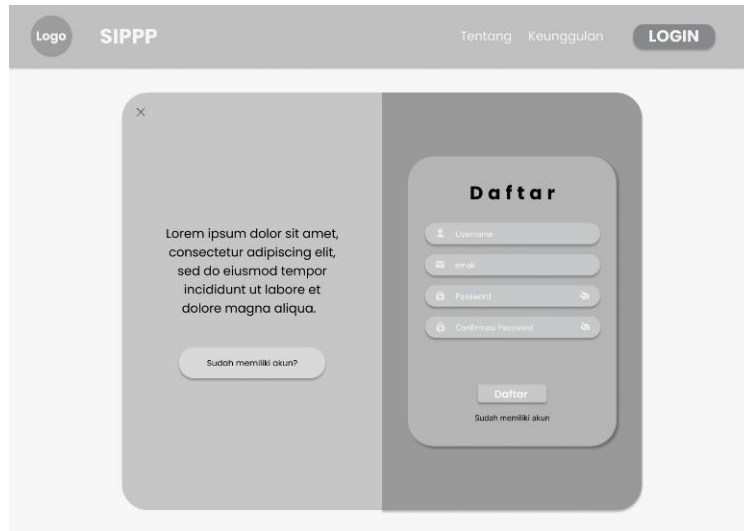
Gambar 6. Wireframe Halaman Tentang Kami

Gambar di atas merupakan sketsa halaman tentang kami. Pada sketsa ini terdapat foto, beserta teks yang berisi tentang dari aplikasi SPARKING.



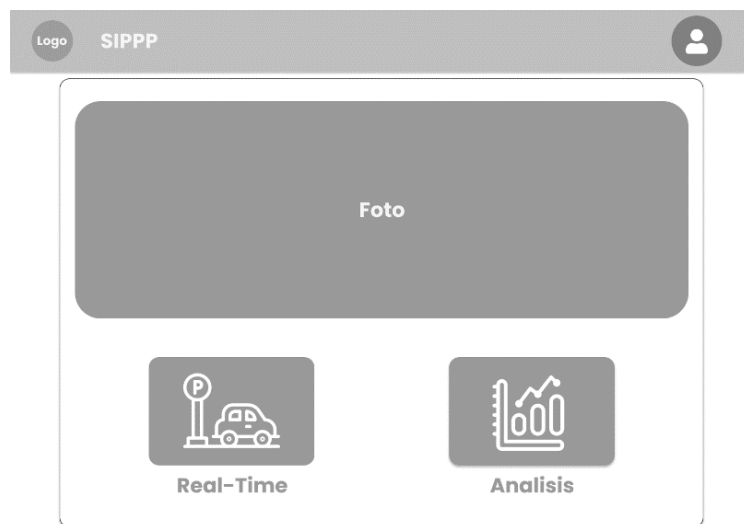
Gambar 7. Wireframe fitur Login

Gambar di atas merupakan sketsa Fitur login. Pada sketsa ini terdapat tombol *slide* ke form registrasi dan form untuk memasukkan data berupa email dan kata sandi.



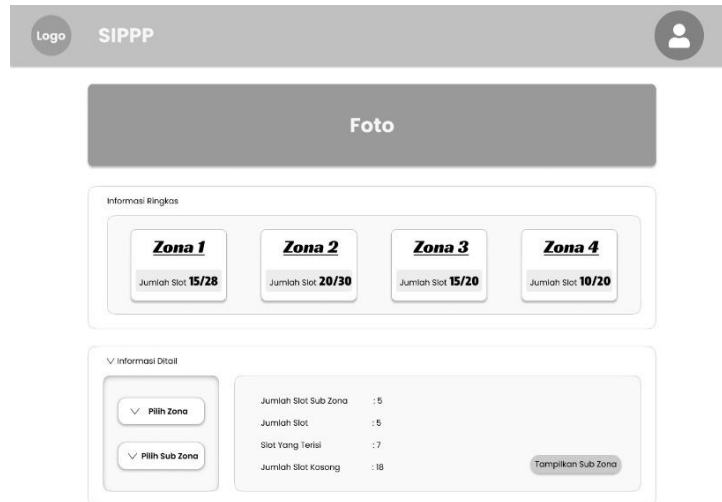
Gambar 8. Wireframe fitur Registrasi

Gambar di atas merupakan sketsa Fitur registrasi. Pada sketsa ini terdapat tombol *slide* ke form login, beserta tempat untuk memasukkan data berupa nama pengguna, email, kata sandi, konfirmasi kata sandi, dan tombol login.



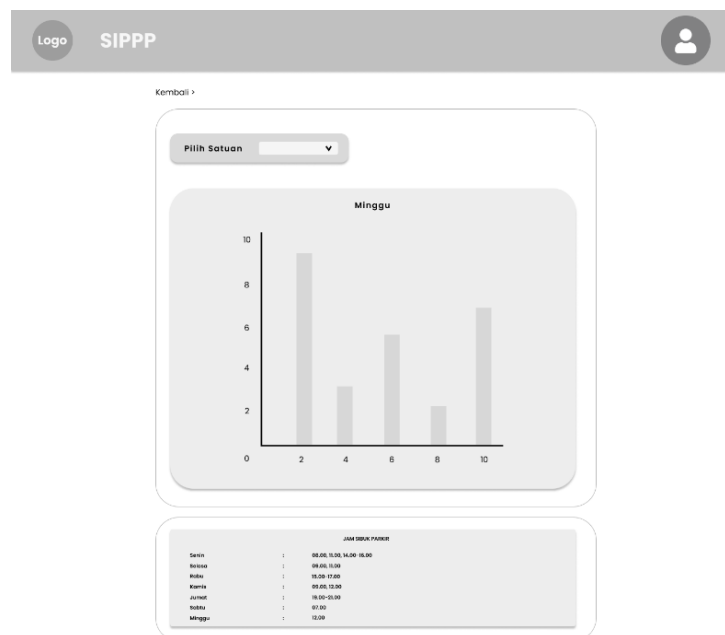
Gambar 9. Wireframe Halaman Beranda

Gambar di atas merupakan sketsa halaman beranda pengguna. Pada sketsa ini terdapat kolom yang nantinya akan diisi untuk fitur, beserta tombol beranda, dan profil.



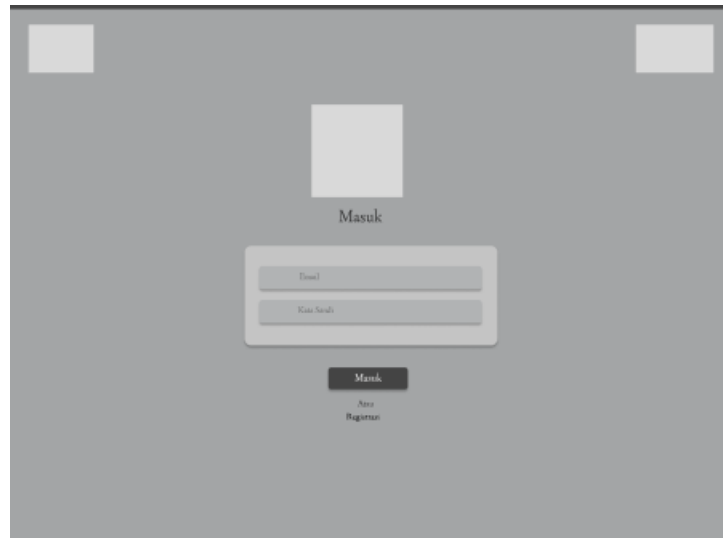
Gambar 10. Wireframe fitur Pemantauan Real-time

Gambar di atas merupakan sketsa fitur pemantauan real-time pada pengguna.



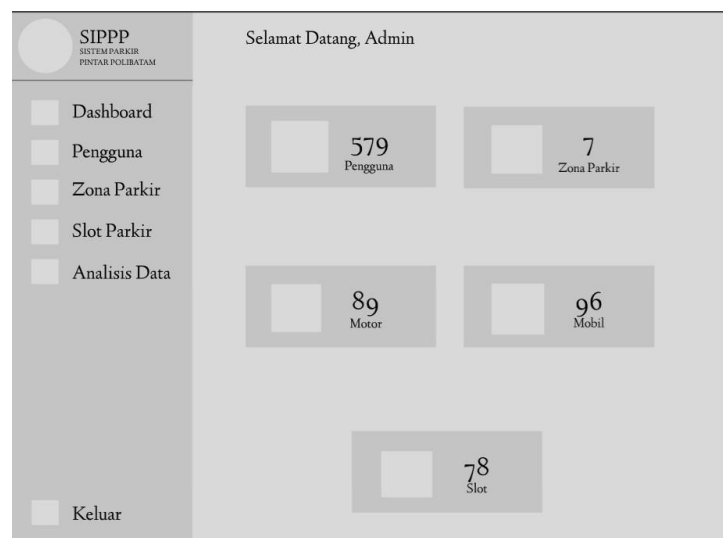
Gambar 11. Wireframe fitur Analisis Data Penggunaan Area Parkir

Gambar di atas merupakan sketsa dari fitur analisis data bagian pengguna.



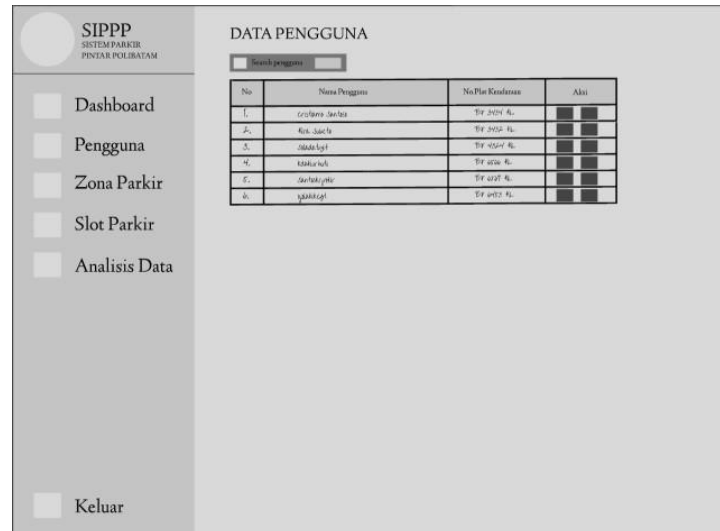
Gambar 12. Wireframe Login Admin

Gambar di atas merupakan sketsa fitur login. Pada sketsa ini terdapat logo, beserta tempat untuk memasukkan data berupa email dan kata sandi, serta terdapat pula tombol registrasi.



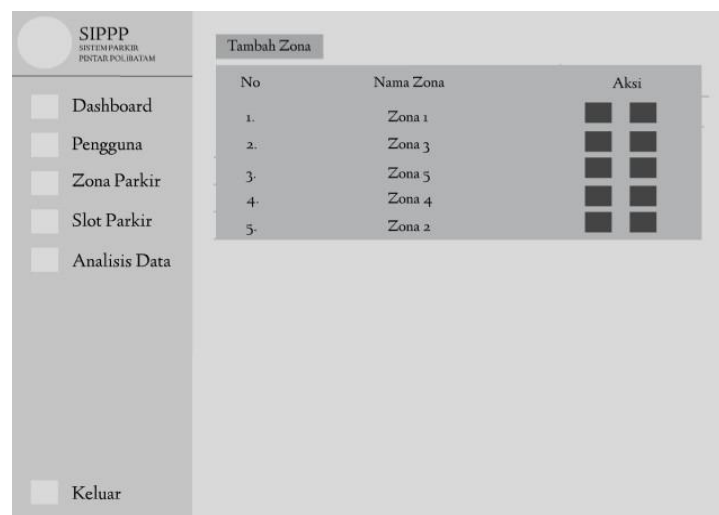
Gambar 13. Wireframe Beranda Admin

Gambar di atas merupakan *sketch* beranda admin. Pada halaman ini admin dapat memilih ingin mengelola pengguna, zona / slot, serta melihat analisis data.



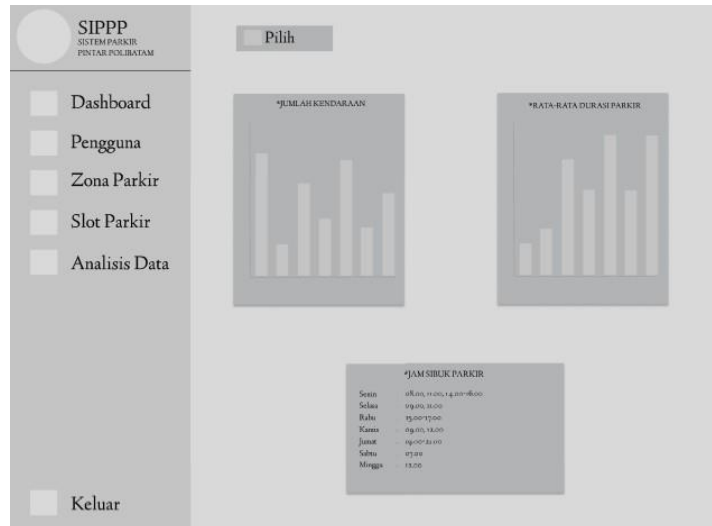
Gambar 14. Wireframe Kelola Pengguna

Gambar di atas merupakan *sketch* kelola pengguna. Pada halaman ini admin dapat melihat data pengguna, serta dapat menghapus data pengguna.



Gambar 15. Wireframe Kelola Zona Parkir

Gambar di atas merupakan *sketch* kelola zona parkir. Pada halaman ini admin dapat melihat zona pada masing – masing area, serta dapat melakukan aksi tambah, edit, ataupun hapus.



Gambar 16. Wireframe fitur Analisis Data

Gambar di atas merupakan *sketch* analisis data pada admin. Pada halaman ini admin dapat memilih kendaraan kemudian dapat melihat grafik jumlah kendaraan, rata-rata durasi parkir, dan jam sibuk parkir selama 7 hari.



Gambar 17. Wireframe Modal Zona dan Slot Parkir

Wireframe ini di rancang untuk mendukung manajemen zona dan slot parkir, memungkinkan pengguna untuk menambahkan, mengedit, atau mengelola data dengan mudah melalui tampilan modal CRUD yang sederhana

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1. Hasil Implementasi

Proses implementasi pada proyek akhir ini dilakukan setelah tahap analisis dan perancangan sistem selesai dilaksanakan. Implementasi dilakukan dengan menyiapkan framework serta lingkungan pengembangan yang diperlukan, baik untuk sisi backend, frontend, maupun sistem deteksi.

Pada sisi backend, Laravel digunakan untuk membangun API sekaligus mengelola logika bisnis aplikasi. Laravel beserta database MySQL di-deploy pada platform Railway sehingga mendapatkan domain publik yang dapat diakses secara online tanpa perlu mengelola server sendiri. Proses deployment dilakukan dengan mengonfigurasi environment variable pada Railway, mencakup koneksi database, konfigurasi aplikasi, serta menjalankan migrasi tabel secara otomatis saat proses deployment berlangsung. Database yang digunakan adalah MySQL yang juga disediakan oleh Railway sebagai layanan terpisah, sehingga koneksi antara aplikasi Laravel dan database dapat dilakukan secara langsung melalui internal network Railway.

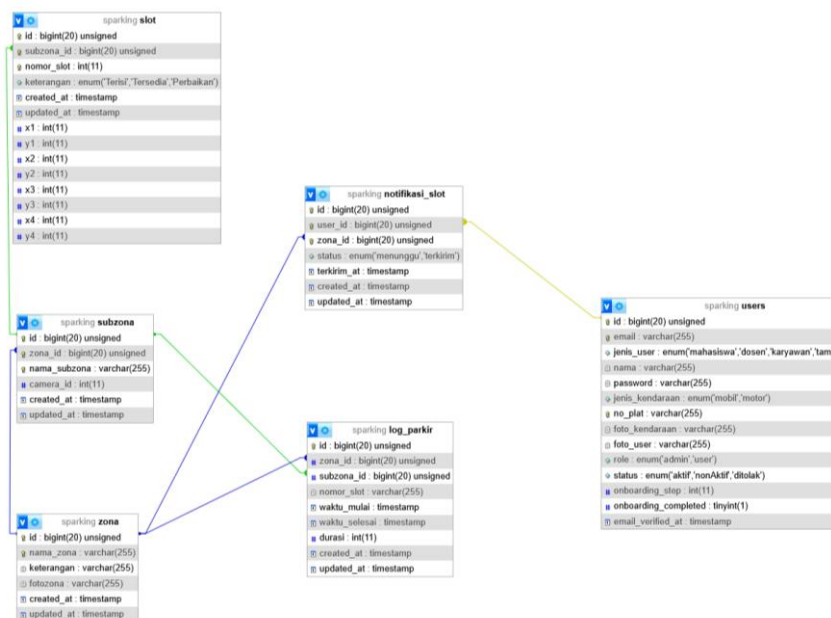
Pada sisi frontend, digunakan Laravel Blade sebagai template engine untuk membangun tampilan antarmuka pengguna. Blade dipilih karena sudah terintegrasi langsung dengan Laravel, sehingga data dari backend dapat diteruskan ke tampilan tanpa memerlukan pemanggilan API secara terpisah. Antarmuka yang dibangun mencakup halaman manajemen slot, monitoring status, halaman notifikasi bagi pengguna dan halaman lainnya.

Untuk keperluan notifikasi, Laravel mengintegrasikan layanan Brevo sebagai penyedia pengiriman email. Notifikasi dikirimkan secara otomatis kepada pengguna ketika terjadi perubahan status slot menjadi tersedia. Integrasi dilakukan

melalui konfigurasi SMTP Brevo pada file konfigurasi mail Laravel, dengan kredensial yang disimpan pada environment variable Railway agar tetap aman.

Pada sisi sistem deteksi, digunakan Raspberry Pi 5 sebagai perangkat edge computing yang menjalankan proses deteksi secara lokal. Di dalam Raspberry Pi, dibangun sebuah layanan berbasis Python menggunakan framework Flask yang bertugas menerima permintaan dan mengembalikan hasil deteksi. Model yang digunakan untuk proses deteksi adalah YOLOv8, yang dijalankan secara langsung di atas Raspberry Pi untuk memproses data secara real-time. Agar layanan Flask yang berjalan secara lokal di Raspberry Pi dapat diakses dari internet, digunakan Cloudflare Tunnel sebagai solusi tunneling. Cloudflare Tunnel memungkinkan perangkat lokal mendapatkan domain publik yang aman. Domain yang dihasilkan dari Cloudflare Tunnel kemudian digunakan oleh backend Laravel untuk berkomunikasi dengan layanan deteksi di Raspberry Pi.

4.1.1. Skema Relasional



Gambar 18. Skema Relasional

4.1.2. Implementasi User Interface (UI)

1. Landing Page

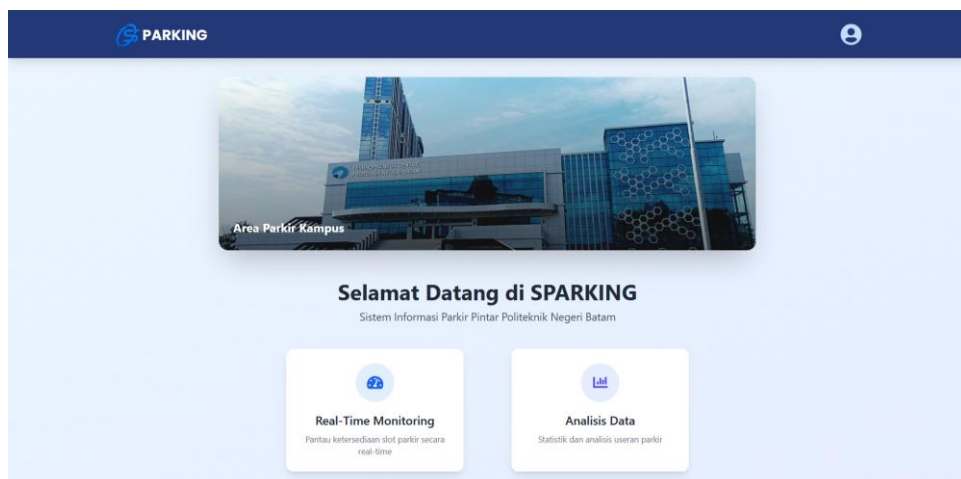
Gambar di bawah merupakan tampilan awal ketika sistem dibuka, dan terdapat informasi mengenai SPARKING (Smart Parking) beserta keunggulannya.



Gambar 19. Implementasi *Landing Page*

2. *Dashboard Pengguna*

Gambar di bawah merupakan halaman utama pada sistem. Pada halaman ini terdapat dua fitur utama yang dikembangkan, yaitu monitoring ketersediaan slot parkir secara real-time dan analisis data.

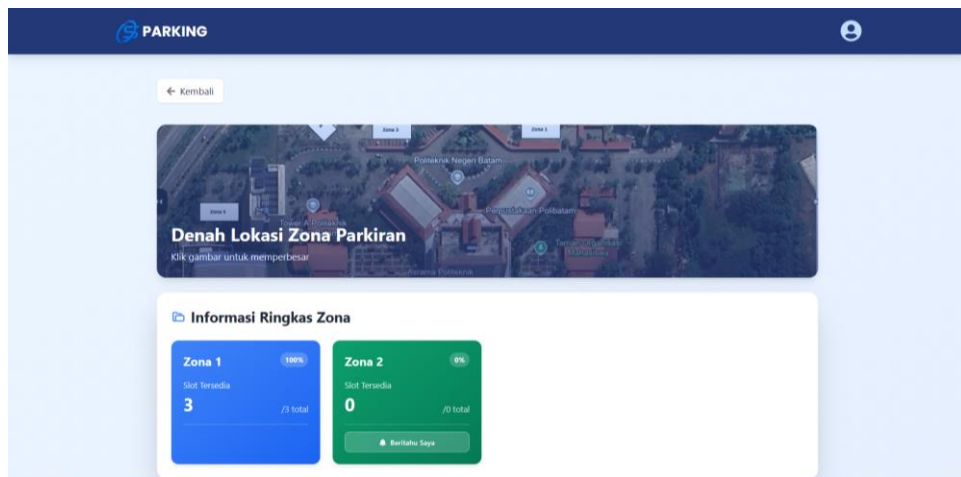


Gambar 20. Implementasi *Dashboard Pengguna*

3. *Real-Time*

Gambar di bawah merupakan halaman yang menampilkan fitur untuk melihat peta parkir, serta ketersediaan slot parkir pada tiap zona parkir yang ada di kampus Politeknik Negeri Batam. Selain dapat melihat area sub-

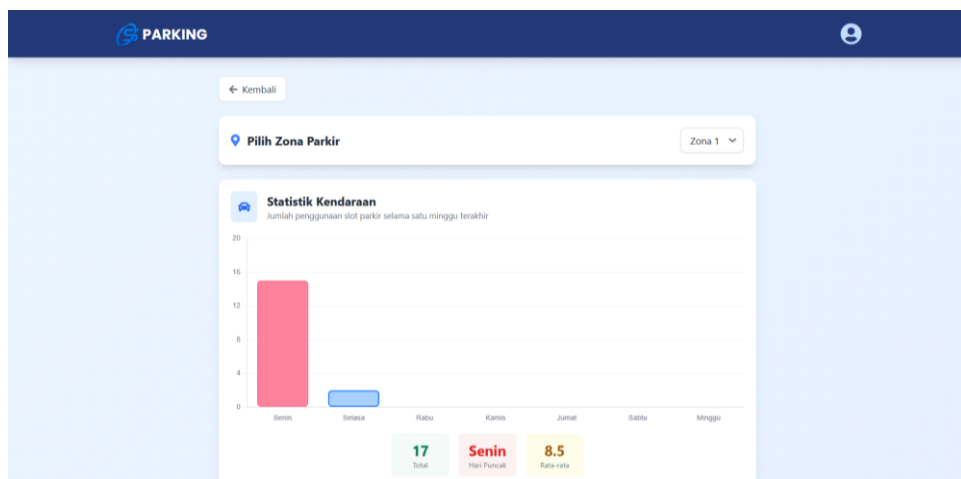
zona, pada halaman ini pengguna juga dapat melakukan *subscribe* notifikasi zona.



Gambar 21. Implementasi *Real-Time*

4. Analisis Data

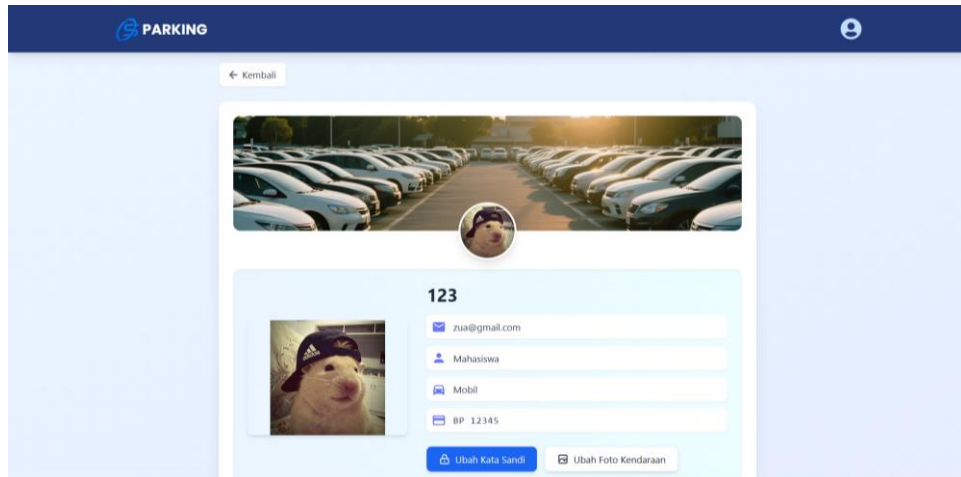
Gambar di bawah menampilkan hasil analisis data yang terdiri dari data jumlah kendaraan dalam bentuk grafik batang, data durasi parkir berbentuk grafik, serta data jam sibuk parkir dalam kurun waktu seminggu.



Gambar 22. Implementasi Statistik

5. Pengaturan

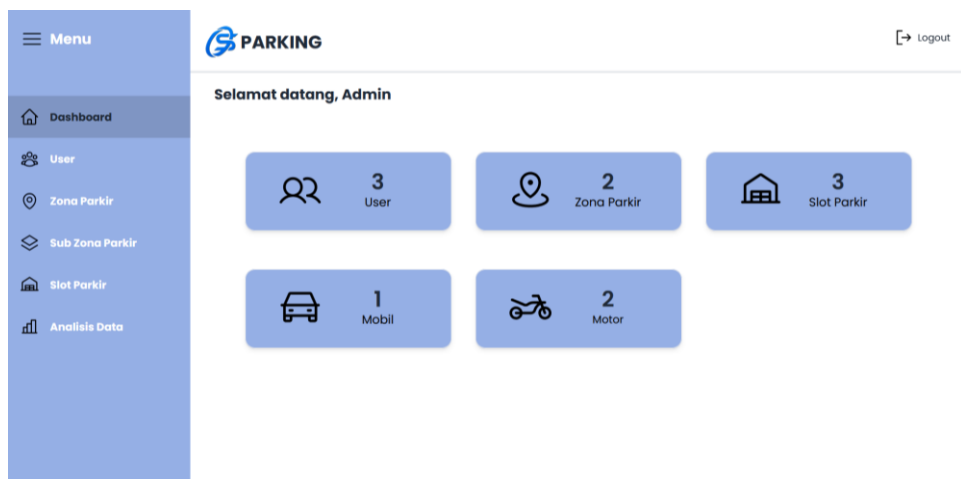
Gambar di bawah merupakan tampilan profil. Pengguna dapat mengubah isi data diri yang telah didaftarkan sebelumnya, seperti mengganti kata sandi dan mengubah foto kendaraan yang dimiliki.



Gambar 23. Implementasi Pengaturan

6. *Landing Page Admin*

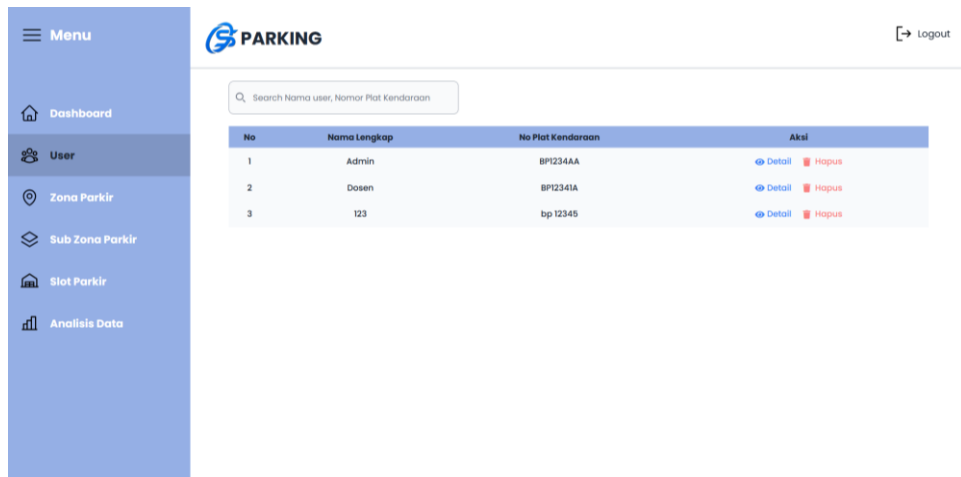
Halaman ini menampilkan fitur khusus untuk admin, seperti melihat jumlah pengguna website, mengelola persetujuan, zona, dan slot parkir, serta memantau jumlah kendaraan yang terdaftar.



Gambar 24. Implementasi *Landing Page Admin*

7. Manajemen Pengguna

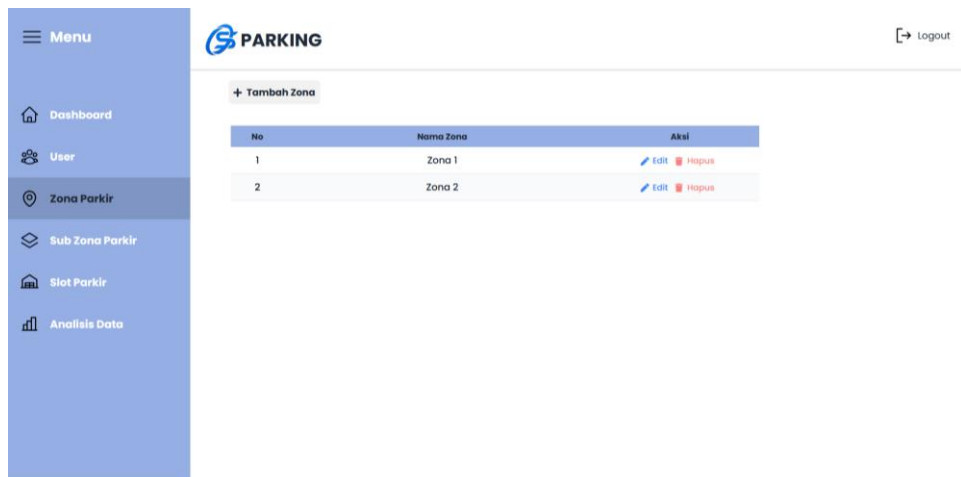
Halaman ini memudahkan admin dalam melihat detail dan menghapus data pengguna. Dilengkapi fitur pencarian untuk mempermudah mencari pengguna sehingga meningkatkan efisiensi aplikasi.



Gambar 25. Implementasi Manajemen Pengguna

8. Manajemen Zona

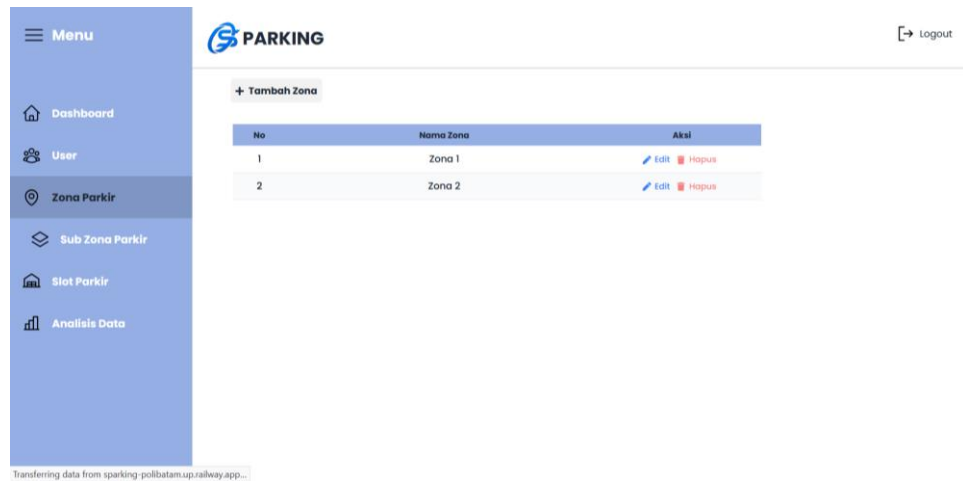
Gambar di bawah merupakan tampilan fitur Kelola Zona Parkir. Memungkinkan admin untuk mengelola zona parkir, aksi yang dapat dilakukan admin ialah menambahkan zona, mengedit zona, serta menghapus zona.



Gambar 26. Implementasi Manajemen Zona

9. Manajemen Sub-Zona

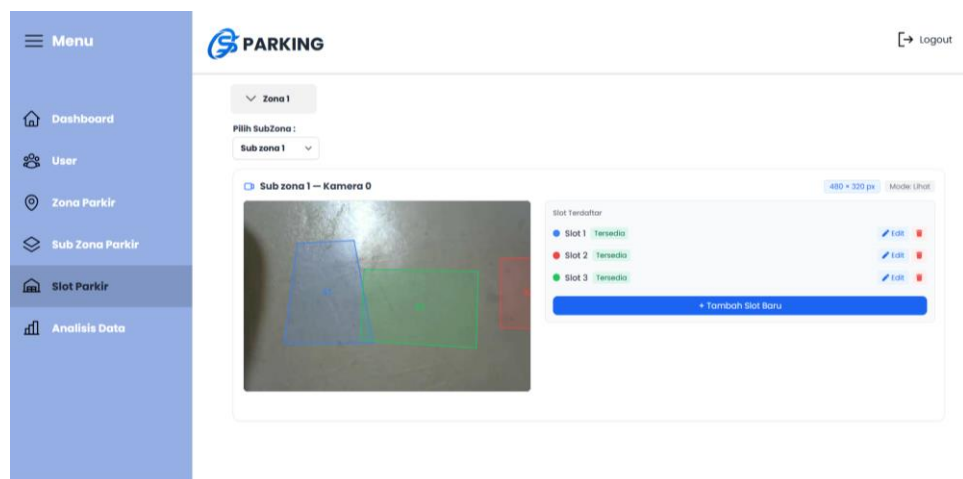
Pada halaman ini memungkinkan admin untuk mengelola sub-zona parkir, aksi yang dapat dilakukan admin ialah menambahkan sub-zona, mengedit sub-zona, serta menghapus sub-zona.



Gambar 27. Implementasi Manajemen Sub-Zona

10. Manajemen Slot

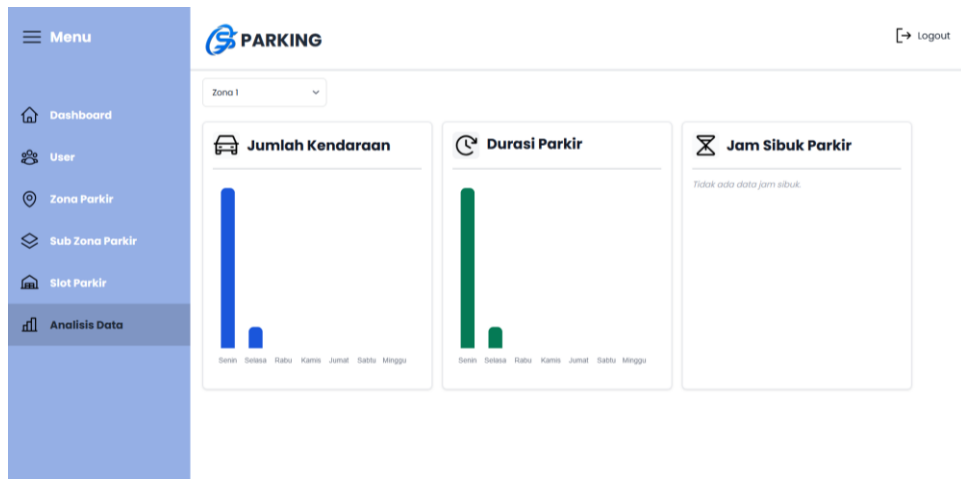
Pada halaman ini dapat menampilkan zona, subzona serta keterangan pada masing-masing slot parkir, dimana admin dapat melakukan aksi seperti melihat, mengedit, serta menghapus slot parkir tersebut.



Gambar 28. Implementasi Manajemen Slot

11. Analisis Data

Pada halaman ini, admin hanya dapat melihat analisis data sistem dalam kurun waktu tertentu, data yang ditunjukkan ialah analisis dari data jumlah kendaraan, durasi parkir, serta jam sibuk parkir.



Gambar 29. Implementasi Analisis Data

4.2. Pengujian *User Acceptance Testing* (UAT)

4.2.1. *Fungsional Testing*

1. *Black Box Testing Scenario Use Case Registrasi Pengguna*

Black box testing ini berfokus pada pengujian fungsional menggunakan *black box testing*.

Tabel 4. *Black Box Testing Scenario Use Case Registrasi Pengguna*

Tes Case ID	Skenario Pengujian	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01-F01	Mengisi form Registrasi dengan ketentuan: 1. Data yang diisi sesuai dengan format yang dibawah ini: • Nama Lengkap (Huruf) • Email (username@gmail.com) • Kata Sandi • Konfirmasi Sandi 2. Data yang terisi pada	Akun berhasil dibuat dan diarahkan ke halaman beranda. Lalu, muncul pesan: 'Periksa email anda untuk melakukan verifikasi sebelum dapat login'.	Akun berhasil dibuat dan diarahkan ke halaman beranda. Lalu, muncul pesan: 'Periksa email anda untuk melakukan verifikasi sebelum dapat login'.	✓	

	Email tidak bisa sama dengan yang sudah ada. 3. Semua data terisi.				
TC02-F01	Data yang diisi tidak sesuai dengan format yang dibawah ini: • Nama Lengkap (Huruf) • Email (username@gmail.com) • Kata Sandi • Konfirmasi Sandi	Tidak Berhasil membuat akun dan memunculkan pesan kesalahan pada setiap kolom data yang formatnya tidak sesuai.	Tidak Berhasil membuat akun dan memunculkan pesan kesalahan pada setiap kolom data yang formatnya tidak sesuai.	✓	
TC03-F01	Data yang terisi pada Email sudah pernah terdaftar.	Tidak Berhasil membuat akun dan memunculkan pesan pada kolom data tersebut bahwa duplikasi data.	Tidak Berhasil membuat akun dan memunculkan pesan pada kolom data tersebut bahwa duplikasi data.	✓	
TC04-F01	Mengecek data pendaftaran ketika pendaftaran berhasil. Mengecek data pendaftaran ketika pendaftaran gagal.	Data yang di input masuk ke dalam database. Data yang di input tidak masuk ke dalam database.	Data yang di input masuk ke dalam database. Data yang di input tidak masuk ke dalam database.	✓	
TC05-F01	Mengecek data pendaftaran ketika pendaftaran berhasil.	Data yang di input masuk ke dalam database.	Data yang di input masuk ke dalam database.	✓	

Hasil analisis:

1) *Persentase test case* yang lolos = $\frac{5}{5} = 100\%$

2) *Persentase test case* yang gagal = $\frac{0}{5} = 0\%$

3) *Persentase test case* yang telah diuji = $5 = 100\%$

4) *Persentase test case* yang belum diuji = $0 = 0\%$

Jumlah total *test case* dalam pengujian = 5

2. Black Box Testing Scenario Use Case Autentikasi Login

Black box testing ini berfokus pada pengujian fungsional menggunakan *black box testing*.

Tabel 5. Black Box Testing Scenario Use Case Autentikasi Login

Tes Case ID	Skenario Pengujian	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01-F02	Email terisi dengan benar dan kata sandi terisi dengan benar.	Login berhasil divalidasi dan berpindah ke halaman utama.	Login berhasil divalidasi dan berpindah ke halaman utama.	✓	
TC02-F02	Email terisi dengan benar dan kata sandi terisi dengan salah.	Login tidak berhasil divalidasi sehingga masih berada pada halaman beranda dan memunculkan pesan “Email atau Password salah”.	Login tidak berhasil divalidasi sehingga masih berada pada halaman beranda dan memunculkan pesan “Email atau Password salah”.	✓	
TC03-F02	Email terisi dengan benar dan kata sandi tidak terisi.	Login tidak berhasil divalidasi sehingga masih berada pada halaman beranda dan memunculkan pesan “Email atau Password salah”.	Login tidak berhasil divalidasi sehingga masih berada pada halaman beranda dan memunculkan pesan “Email atau Password salah”.	✓	
TC04-F02	Email terisi dengan salah dan kata sandi terisi dengan benar.	Login tidak berhasil divalidasi sehingga masih berada pada halaman beranda dan memunculkan pesan “Email atau Password salah”.	Login tidak berhasil divalidasi sehingga masih berada pada halaman beranda dan memunculkan pesan “Email atau Password salah”.	✓	
TC05-F02	Email terisi dengan salah dan kata sandi terisi dengan salah.	Login tidak berhasil divalidasi sehingga masih berada pada halaman beranda dan memunculkan pesan “Email	Login tidak berhasil divalidasi sehingga masih berada pada halaman beranda dan memunculkan	✓	

		atau Password salah”.	pesan “Email atau Password salah”.		
TC06-F02	Email terisi dengan salah dan kata sandi tidak terisi.	Login tidak berhasil divalidasi sehingga masih berada pada halaman beranda dan memunculkan pesan “Gagal masuk”.	Login tidak berhasil divalidasi sehingga masih berada pada halaman beranda dan memunculkan pesan “Gagal masuk”.	✓	
TC07-F02	Email tidak terisi dan kata sandi terisi dengan benar.	Login tidak berhasil divalidasi sehingga masih berada pada halaman beranda dan memunculkan pesan “Gagal masuk”.	Login tidak berhasil divalidasi sehingga masih berada pada halaman beranda dan memunculkan pesan “Gagal masuk”.	✓	
TC08-F02	Email tidak terisi dan kata sandi terisi dengan salah.	Login tidak berhasil divalidasi sehingga masih berada pada halaman beranda dan memunculkan pesan “Gagal masuk”.	Login tidak berhasil divalidasi sehingga masih berada pada halaman beranda dan memunculkan pesan “Gagal masuk”.	✓	
TC09-F02	Email tidak terisi dan kata sandi tidak terisi.	Login tidak berhasil divalidasi sehingga masih berada pada halaman beranda dan memunculkan pesan “Gagal masuk”.	Login tidak berhasil divalidasi sehingga masih berada pada halaman login dan memunculkan pesan “Gagal masuk”.	✓	
TC10-F02	Email terisi dengan benar dan kata sandi terisi dengan benar. Namun, pengguna belum melakukan aktivasi.	Login tidak berhasil divalidasi sehingga masih berada pada halaman beranda dan memunculkan pesan “Akun belum diaktivasi”.	Login tidak berhasil divalidasi sehingga masih berada pada halaman beranda dan memunculkan pesan “Akun belum diaktivasi”.	✓	

Hasil analisis:

$$1) \text{ Persentase test case yang lolos} = \frac{10}{10} = 100\%$$

$$2) \text{ Persentase test case yang gagal} = \frac{0}{10} = 0\%$$

3) *Persentase test case* yang telah diuji = 10 = 100%

4) *Persentase test case* yang belum diuji = 0 = 0%

Jumlah total *test case* dalam pengujian = 10

3. *Black Box Testing Scenario Use Case* Monitoring Slot Parkir

Black box testing ini berfokus pada pengujian fungsional menggunakan *black box testing*.

Tabel 6. *Black Box Testing Scenario Use Case* Monitoring Slot Parkir

Tes Case ID	Skenario Pengujian	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01-F03	Menekan menu pemantauan real-time ketersediaan slot parkir.	Menampilkan halaman yang berisi Dropdown dan Peta Parkir.	Menampilkan halaman yang berisi Dropdown dan Peta Parkir.	✓	
TC02-F03	Memilih Zona yang ingin dilihat Sub-Zona dan Slot nya melalui select dropdown.	Menampilkan informasi real-time mengenai ketersediaan slot parkir di suatu Sub-Zona area parkir Polibatam berdasarkan Zona yang dipilih.	Menampilkan informasi real-time mengenai ketersediaan slot parkir di suatu Sub-Zona area parkir Polibatam berdasarkan Zona yang dipilih.	✓	
TC03-F03	Menekan tombol “Tampilkan subzona”.	Menampilkan modal yang berisi video streaming dan informasi real-time mengenai status ketersediaan slot.	Menampilkan modal yang berisi video streaming dan informasi real-time mengenai status ketersediaan slot.	✓	
TC04-F03	Menekan tombol untuk memperbesar tampilan peta parkir	Menampilkan gambar nyata dari area parkir Polibatam berhasil ditampilkan secara jelas dan informatif.	Menampilkan gambar nyata dari area parkir Polibatam berhasil ditampilkan secara jelas dan informatif.	✓	

TC05-F03	Webcam dapat mendeteksi kendaraan yang ada pada setiap Slot parkir.	Aplikasi menerima data dari webcam yang telah diproses oleh perangkat Raspberry Pi.	Aplikasi menerima data dari webcam yang telah diproses oleh perangkat Raspberry Pi.	✓	
TC06-F03	Mengecek sinkronisasi antara status ketersediaan slot parkir di aplikasi (setiap sub-zona) dengan di lapangan	Status ketersediaan slot di lapangan sesuai dengan yang ada di aplikasi.	Status ketersediaan slot di lapangan sesuai dengan yang ada di aplikasi.	✓	
TC07-F03	Mengecek kesesuaian modal informasi zona pada aplikasi dengan yang ada di lapangan.	Informasi zona yang terdapat pada aplikasi sesuai dengan yang ada di lapangan.	Informasi zona yang terdapat pada aplikasi sesuai dengan yang ada di lapangan.	✓	

Hasil analisis:

- 1) *Persentase test case* yang lolos = $\frac{7}{7} = 100\%$
 - 2) *Persentase test case* yang gagal = $\frac{0}{7} = 0\%$
 - 3) *Persentase test case* yang telah diuji = 7 = 100%
 - 4) *Persentase test case* yang belum diuji = 0 = 0%
- Jumlah total *test case* dalam pengujian = 7

4. *Black Box Testing Scenario Use Case Visualisasi Statistik Data Parkir*

Black box testing ini berfokus pada pengujian fungsional menggunakan *black box testing*.

Tabel 7. *Black Box Testing Scenario Use Case Visualisasi Statistik Data Parkir*

Tes Case ID	Skenario Pengujian	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01-F04	Menekan menu Analisis data penggunaan area parkir	Menampilkan halaman dengan dropdown pilihan zona, namun grafik belum tersedia atau masih kosong.	Menampilkan halaman dengan dropdown pilihan zona, namun grafik belum tersedia atau masih kosong.	✓	

TC02-F04	Memilih Zona yang ingin dilihat datanya melalui select dropdown.	Menampilkan halaman yang menyajikan grafik analisis penggunaan area parkir berdasarkan zona yang dipilih, termasuk informasi waktu penggunaan dan hari tersibuk setiap minggunya.	Menampilkan halaman yang menyajikan grafik analisis penggunaan area parkir berdasarkan zona yang dipilih termasuk informasi waktu penggunaan dan hari tersibuk setiap minggunya.	✓	
TC03-F04	Mengecek kesesuaian antara hasil analisis data penggunaan area parkir di aplikasi dengan kondisi aktual di lapangan	Hasil analisis data yang ditampilkan aplikasi sesuai dengan kondisi penggunaan area parkir secara nyata di zona yang bersangkutan.	Hasil analisis data yang ditampilkan aplikasi sesuai dengan kondisi penggunaan area parkir secara nyata di zona yang bersangkutan.	✓	

Hasil analisis:

1) *Persentase test case* yang lolos = $\frac{3}{3} = 100\%$

2) *Persentase test case* yang gagal = $\frac{0}{3} = 0\%$

3) *Persentase test case* yang telah diuji = 3 = 100%

4) *Persentase test case* yang belum diuji = 0 = 0%

Jumlah total *test case* dalam pengujian = 3

5. *Black Box Testing Scenario Use Case* Manajemen Zona Parkir

Black box testing ini berfokus pada pengujian fungsional menggunakan *black box testing*.

Tabel 8. *Black Box Testing Scenario Use Case* Manajemen Zona Parkir

Tes Case ID	Skenario Pengujian	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01-F5	Menekan menu “Zona”	Menampilkan halaman “Kelola Zona dan Sub- Zona Parkir”.	Menampilkan halaman “Kelola Zona dan Sub- Zona Parkir”.	✓	
TC02-F5	Menekan button “Tambah Zona”.	Menampilkan modal “Tambah Zona” yang berisi Nama Zona, Keterangan	Menampilkan modal “Tambah Zona” yang berisi Nama Zona, Keterangan	✓	

		Zona, dan Foto Area Zona.	Zona, dan Foto Area Zona.		
TC03-F5	Mengisi form “Tambah Zona” dengan ketentuan: 1. Data yang diisi sesuai dengan format yang dibawah ini: ▪ Nama Zona (huruf/angka) ▪ Keterangan Zona (huruf/angka)) 2. Data yang terisi pada Nama Zona tidak bisa sama dengan yang sudah ada. 3. Semua data terisi.	Berhasil menambahkan zona dengan menampilkannya pada tabel Zona dan menyimpannya pada database.	Berhasil menambahkan zona dengan menampilkannya pada tabel Zona dan menyimpannya pada database.	✓	
TC04-F5	Mengisi form “Tambah Zona”, tetapi tidak sesuai dengan ketentuan.	Tidak Berhasil menambahkan zona dan memberikan pesan kesalahan terkait pengisian pada form “Tambah Zona”.	Tidak Berhasil menambahkan zona dan memberikan pesan kesalahan terkait pengisian pada form “Tambah Zona”.	✓	
TC05-F5	Menekan Aksi “Edit Zona”.	Menampilkan modal “Edit Zona” yang berisi Nama Zona, Keterangan Zona, dan Foto Area Zona. Tetapi, hanya Keterangan Zona dan Foto Area Zona yang dapat diedit.	Menampilkan modal “Edit Zona” yang berisi Nama Zona, Keterangan Zona, dan Foto Area Zona. Tetapi, hanya Keterangan Zona dan Foto Area Zona yang dapat diedit.	✓	
TC06-F5	Mengedit form “Edit Zona” dengan ketentuan : 1. Data yang diisi sesuai dengan format yang dibawah ini: ▪ Keterangan Zona (huruf/angka))	Berhasil mengedit zona dengan menampilkannya pada tabel Zona dan menyimpannya pada database.	Berhasil mengedit zona dengan menampilkannya pada tabel Zona dan menyimpannya pada database.	✓	

	2. Semua data terisi.				
TC07-F5	Mengedit form “Edit Zona”, tetapi tidak sesuai dengan ketentuan.	Tidak Berhasil mengedit zona dan memberikan pesan kesalahan terkait pengisian pada form “Edit Zona”.	Tidak Berhasil mengedit zona dan memberikan pesan kesalahan terkait pengisian pada form “Edit Zona”.	✓	
TC08-F5	Menekan Aksi “Hapus” pada salah satu Zona.	Berhasil menghapus Zona dari tampilan tabel dan database.	Berhasil menghapus Zona dari tampilan tabel dan database.	✓	

Hasil analisis:

1) *Persentase test case* yang lolos = $\frac{8}{8} = 100\%$

2) *Persentase test case* yang gagal = $\frac{0}{8} = 0\%$

3) *Persentase test case* yang telah diuji = $8 = 100\%$

4) *Persentase test case* yang belum diuji = $0 = 0\%$

Jumlah total *test case* dalam pengujian = 8

6. *Black Box Testing Scenario Use Case* Manajemen Sub-Zona Parkir

Black box testing ini berfokus pada pengujian fungsional menggunakan *black box testing*.

Tabel 9. *Black Box Testing Scenario Use Case* Manajemen Sub-Zona Parkir

Tes Case ID	Skenario Pengujian	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01-F06	Memilih Zona yang ingin dilihat Sub-Zona nya pada dropdown.	Menampilkan tabel Sub- Zona sesuai dengan Zona yang dipilih.	Menampilkan tabel Sub-Zona sesuai dengan Zona yang dipilih.	✓	
TC02-F06	Menekan button “Tambah Sub- Zona”.	Menampilkan modal “Tambah Sub-Zona” yang berisi Nama Zona, Nama Sub-Zona, dan Foto Area Sub-Zona.	Menampilkan modal “Tambah Sub-Zona” yang berisi Nama Zona, Nama Sub-Zona, dan Foto Area Sub-Zona.	✓	

TC03 - F06	Mengisi form “Tambah Sub-Zona” dengan ketentuan : 1. Data yang diisi sesuai dengan format yang dibawah ini : ▪ Nama Sub-Zona (huruf/angka) ▪ Nama Sub-Zona (huruf/angka)) 2. Data yang terisi pada Nama Sub-Zona tidak bisa sama dengan yang sudah ada. 3. Semua data terisi.	Berhasil menambahkan Sub-Zona dengan menampilkannya pada tabel Sub-Zona dan menyimpannya pada database.	Berhasil menambahkan Sub-Zona dengan menampilkannya pada tabel Sub-Zona dan menyimpannya pada database.	✓	
TC04 - F06	Mengisi form “Tambah Sub-Zona”, tetapi tidak sesuai dengan ketentuan.	Tidak Berhasil menambahkan Sub- Zona dan memberikan pesan kesalahan terkait pengisian pada form “Tambah Sub-Zona”.	Tidak Berhasil menambahkan Sub- Zona dan memberikan pesan kesalahan terkait pengisian pada form “Tambah Sub-Zona”.	✓	
TC05 - F06	Menekan Aksi “Edit Sub-Zona”.	Menampilkan modal “Edit Sub-Zona” yang berisi Nama Zona, Nama Sub-Zona, dan Foto Area Sub-Zona. Tetapi, hanya Foto Area Sub-Zona yang dapat diedit.	Menampilkan modal “Edit Sub-Zona” yang berisi Nama Zona, Nama Sub-Zona, dan Foto Area Sub-Zona. Tetapi, hanya Foto Area Sub-Zona yang dapat diedit.	✓	
TC06 - F06	Mengedit form “Edit Sub-Zona” dengan ketentuan : Data yang diisi sesuai.	Berhasil mengedit Sub- Zona dengan menampilkannya pada tabel Sub-Zona dan menyimpannya pada database.	Berhasil mengedit Sub- Zona dengan menampilkannya pada tabel Sub-Zona dan menyimpannya pada	✓	

			database.		
TC07 - F06	Mengedit form “Edit Sub-Zona”, tetapi tidak sesuai dengan ketentuan.	Tidak Berhasil mengedit Sub-Zona dan memberikan pesan kesalahan terkait pengisian pada form “Edit Sub-Zona”.	Tidak Berhasil mengedit Sub-Zona dan memberikan pesan kesalahan terkait pengisian pada form “Edit Sub-Zona”.	✓	
TC08 - F06	Menekan Aksi “Hapus” pada salah satu Sub-Zona.	Berhasil menghapus Sub-Zona dari database.	Berhasil menghapus Sub-Zona dari database.	✓	

Hasil analisis:

1) *Persentase test case* yang lolos = $\frac{8}{8} = 100\%$

2) *Persentase test case* yang gagal = $\frac{0}{8} = 0\%$

3) *Persentase test case* yang telah diuji = 8 = 100%

4) *Persentase test case* yang belum diuji = 0 = 0%

Jumlah total *test case* dalam pengujian = 8

7. Black Box Testing Scenario Use Case Manajemen Slot Parkir

Black box testing ini berfokus pada pengujian fungsional menggunakan *black box testing*.

Tabel 10. *Black Box Testing Scenario Use Case* Manajemen Slot Parkir

Tes Case ID	Skenario Pengujian	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01-F07	Memilih Zona yang ingin ditampilkan Sub-Zona nya pada dropdown.	Menampilkan Sub- Zona yang berkaitan dengan Zona pada dropdown.	Menampilkan Sub-Zona yang berkaitan dengan Zona pada dropdown.	✓	
TC02-F07	Memilih Sub-Zona yang ingin dilihat Slot nya pada dropdown.	Menampilkan table Slot sesuai dengan Sub-Zona yang dipilih.	Menampilkan table Slot sesuai dengan Sub-Zona yang dipilih.	✓	
TC03-F07	Menekan button “Tambah Slot”.	Menampilkan modal “Tambah Slot” yang berisi Nama Sub-Zona, Nomor Slot, dan Keterangan.	Menampilkan modal “Tambah Slot” yang berisi Nama Sub-Zona, Nomor Slot, dan Keterangan.	✓	

TC04-F07	Mengisi form “Tambah Slot” dengan ketentuan: 1. Data yang diisi sesuai dengan format yang dibawah ini: ▪ Sub-Zona ▪ Nomor Slot (angka) ▪ Keterangan Koordinat 2. Data yang terisi pada Nomor Slot tidak bisa sama dengan yang sudah ada. 3. Semua data terisi.	Berhasil menambahkan Slot dengan menampilkannya pada tabel Slot dan menyimpannya pada database.	Berhasil menambahkan Slot dengan menampilkannya pada tabel Slot dan menyimpannya pada database.	✓	
TC05-F07	Mengisi form “Tambah Slot”, tetapi tidak sesuai dengan ketentuan.	Tidak Berhasil menambahkan Slot dan memberikan pesan kesalahan terkait pengisian pada form “Tambah Slot”.	Tidak Berhasil menambahkan Slot dan memberikan pesan kesalahan terkait pengisian pada form “Tambah Slot”.	✓	
TC06-F07	Menekan Aksi “Edit Slot”.	Menampilkan modal “Edit Slot” yang berisi Nama Sub-Zona, Nomor Slot, dan Keterangan. Tetapi, hanya Nomor Slot dan Keterangan yang dapat diedit.	Menampilkan modal “Edit Slot” yang berisi Nama Sub-Zona, Nomor Slot, dan Keterangan. Tetapi, hanya Nomor Slot dan Keterangan yang dapat diedit.	✓	
TC07-F07	Mengedit form “Edit Slot” dengan ketentuan: 1. Data yang diisi sesuai dengan format yang dibawah ini: ▪ Nomor Slot (angka) ▪ Keterangan Koordinat	Berhasil mengedit Slot dengan menampilkannya pada tabel Slot dan menyimpannya pada database.	Berhasil mengedit Slot dengan menampilkannya pada tabel Slot dan menyimpannya pada database.	✓	

	2. Data yang terisi pada Nomor Slot tidak bisa sama dengan yang sudah ada. 3. Semua data terisi.				
TC08-F07	Mengedit form “Edit Slot”, tetapi tidak sesuai dengan ketentuan.	Tidak Berhasil mengedit Slot dan memberikan pesan kesalahan terkait pengisian pada form “Edit Slot”.	Tidak Berhasil mengedit Slot dan memberikan pesan kesalahan terkait pengisian pada form “Edit Slot”.	✓	
TC09-F07	Menekan Aksi “Hapus” pada salah satu Slot.	Berhasil menghapus Slot dari database.	Berhasil menghapus Slot dari database.	✓	

Hasil analisis:

1) *Persentase test case yang lolos* = $\frac{9}{9} = 100\%$

2) *Persentase test case yang gagal* = $\frac{0}{9} = 0\%$

3) *Persentase test case yang telah diuji* = 9 = 100%

4) *Persentase test case yang belum diuji* = 0 = 0%

Jumlah total *test case* dalam pengujian = 9

8. *Black Box Testing Scenario Use Case* Pengelolaan Pengguna

Black box testing ini berfokus pada pengujian fungsional menggunakan *black box testing*.

Tabel 11. *Black Box Testing Scenario Use Case* Pengelolaan Pengguna

Tes Case ID	Skenario Pengujian	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01-F08	Menekan menu “Pengguna”.	Menampilkan halaman “Kelola data Pengguna”.	Menampilkan halaman “Kelola data Pengguna”.	✓	
TC02-F08	Menggunakan fitur search dengan mengisi Nama lengkap/Nomor Plat Kendaraan.	Melakukan filter pada tabel, sehingga hanya menampilkan data yang berkaitan dengan yang diisi pada fitur search.	Melakukan filter pada tabel, sehingga hanya menampilkan data yang berkaitan dengan yang diisi pada fitur search.	✓	

TC03 - F08	Menekan Aksi “Detail” Pada salah satu data Pengguna.	Menampilkan modal yang berisi keseluruhan data diri dan kendaraan Pengguna tersebut.	Menampilkan modal yang berisi keseluruhan data diri dan kendaraan Pengguna tersebut.	✓	
TC04 - F08	Menekan Aksi “Hapus” pada salah satu data pengguna.	Berhasil menghapus akun Pengguna dari database, sehingga Pengguna tidak bisa menggunakan akunnya untuk mengakses Aplikasi SPARKING (Smart Parking).	Berhasil menghapus akun Pengguna dari database, sehingga Pengguna tidak bisa menggunakan akunnya untuk mengakses Aplikasi SPARKING (Smart Parking).	✓	

Hasil analisis:

1) *Persentase test case yang lolos* = $\frac{4}{4} = 100\%$

2) *Persentase test case yang gagal* = $\frac{0}{4} = 0\%$

3) *Persentase test case yang telah diuji* = 4 = 100%

4) *Persentase test case yang belum diuji* = 0 = 0%

Jumlah total *test case* dalam pengujian = 4

9. *Black Box Testing Scenario Use Case* Notifikasi Slot Parkir

Black box testing ini berfokus pada pengujian fungsional menggunakan *black box testing*.

Tabel 12. *Black Box Testing Scenario Use Case* Notifikasi Slot Parkir

Tes Case ID	Skenario Pengujian	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01- F09	Menekan tombol tambah notifikasi	Berhasil menambahkan antrian notifikasi	Berhasil menambahkan antrian notifikasi	✓	
TC02- F09	Terdapat slot yang kosong pada zona yang diinginkan pengguna	Sistem mengirimkan notifikasi email pada email pengguna yang terdaftar	Sistem mengirimkan notifikasi email pada email pengguna yang terdaftar	✓	
TC03- F09	Menekan tombol batal notifikasi	Berhasil menghapus antrian notifikasi	Berhasil menghapus antrian notifikasi	✓	

Hasil analisis:

- 1) *Persentase test case* yang lolos = $\frac{3}{3} = 100\%$
 - 2) *Persentase test case* yang gagal = $\frac{0}{3} = 0\%$
 - 3) *Persentase test case* yang telah diuji = 3 = 100%
 - 4) *Persentase test case* yang belum diuji = 0 = 0%
- Jumlah total *test case* dalam pengujian = 3

10. Black Box Testing Scenario Use Case Logout

Black box testing ini berfokus pada pengujian fungsional menggunakan *black box testing*.

Tabel 13. *Black Box Testing Scenario Use Case Logout*

Tes Case ID	Skenario Pengujian	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01-F10	Menekan tombol konfirmasi keluar	Logout berhasil dan menuju ke halaman login	Logout berhasil dan menuju ke halaman login	✓	
TC02-F10	Menekan tombol batal keluar	Logout gagal dan tetap berada di halaman profil	Logout gagal dan tetap berada di halaman profil	✓	

Hasil analisis:

- 1) *Persentase test case* yang lolos = $\frac{2}{2} = 100\%$
 - 2) *Persentase test case* yang gagal = $\frac{0}{2} = 0\%$
 - 3) *Persentase test case* yang telah diuji = 2 = 100%
 - 4) *Persentase test case* yang belum diuji = 0 = 0%
- Jumlah total *test case* dalam pengujian = 2

4.2.2. Non-Fungsional Testing

1. Performance Testing

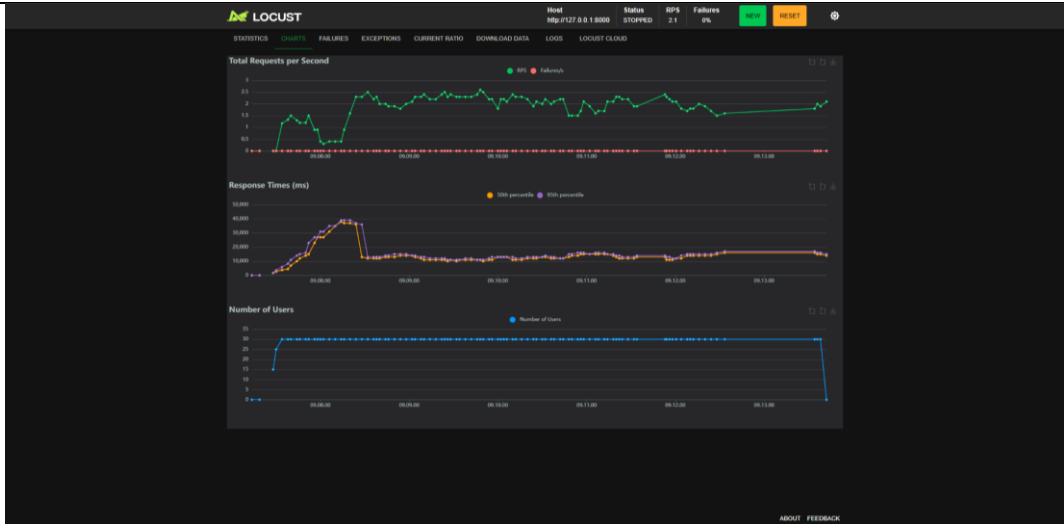
Pengujian non-fungsional pada penelitian ini dilakukan untuk mengevaluasi performa sistem dalam menangani sejumlah permintaan secara bersamaan. Pengujian menggunakan Locust dengan metode load testing. Parameter pengujian meliputi jumlah pengguna, laju penambahan pengguna, durasi pengujian, jumlah

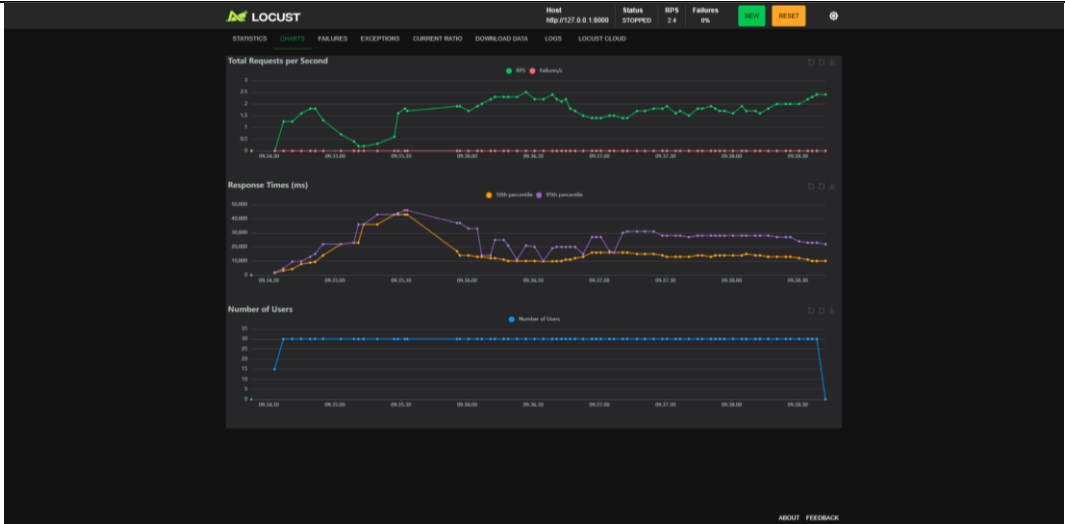
request, waktu respons, dan tingkat kegagalan request. Pengujian dilakukan dengan mensimulasikan 30 pengguna dengan ramp-up sebanyak 5 pengguna per detik selama 5 menit.

Threshold keberhasilan ditetapkan berdasarkan kebutuhan non-fungsional sistem, yaitu waktu respons maksimal sebesar 5 detik atau 5.000 ms. Selain itu, sistem dinyatakan berhasil apabila tidak terjadi kegagalan request atau error rate sebesar 0%. Dengan demikian, hasil pengujian dinyatakan memenuhi kriteria apabila nilai response time berada di bawah 5.000 ms dan seluruh request berhasil diproses.

Tabel 14. *Performance Testing*

Komponen	Rincian
Tes Case ID	TC01-NF01
Type	<i>Performance Testing</i>
Judul	Sistem harus mampu menangani peningkatan jumlah permintaan secara bertahap tanpa mengalami penurunan kinerja yang signifikan.
Tujuan	Memastikan sistem dapat beroperasi dengan baik di bawah berbagai beban kerja tanpa penurunan performa.
Lingkungan	Server pengujian menggunakan spesifikasi menyerupai lingkungan produksi, termasuk CPU, RAM, disk, sistem operasi, dan konfigurasi jaringan. Database serta web server dikonfigurasi identik dengan produksi, sementara koneksi jaringan disimulasikan dengan kecepatan, <i>latensi</i> , dan jitter realistis untuk mencerminkan kondisi pengguna akhir.
Alat	Locust
Metode	<i>Load Testing</i>
Ekspektasi	Sistem dapat merespons permintaan dalam waktu yang konsisten, dengan konsumsi sumber daya yang efisien dan tanpa <i>crash</i> saat mencapai beban puncak.
Hasil Uji	
ID	LOT01
Nama Pengujian	Pengujian performa fitur utama (Login, Dashboard, Statistik, Real-time, Settings)

Deskripsi Pengujian	Pengujian performa fitur utama dengan simulasi 30 pengguna mengakses login hingga berbagai halaman secara bersamaan	
Method	GET & POST	
Skenario Uji	Total Users	30
	Ramp Up	5 pengguna/detik
	Time (durasi)	5 menit
Response Time (ms)	Min (ms)	1.713 ms
	Average (ms)	13.703 ms
	Max (ms)	39.363 ms
	95%ile (ms)	18.000 ms
Requests	Total Requests	702
	Success Requests	702
	Failed Requests	0
Error rate (%)	0%	
Chart Pengujian		
Hasil Uji		
ID	LOT02	
Nama Pengujian	Pengujian performa fitur real-time parkir	
Deskripsi Pengujian	Pengujian performa halaman real-time dengan simulasi 30 pengguna melakukan streaming, filter zona, filter subzona, dan pengecekan notifikasi slot parkir secara bersamaan	

<i>Method</i>	GET & POST	
<i>Skenario Uji</i>	Total Users	30
	Ramp Up	5 pengguna/detik
	Time (durasi)	5 menit
<i>Response Time (ms)</i>	Min (ms)	428 ms
	Average (ms)	15.954 ms
	Max (ms)	46.436 ms
	95%ile (ms)	43.000 ms
<i>Requests</i>	Total Requests	408
	Success Requests	408
	Failed Requests	0
Error rate (%)	0%	
Chart Pengujian		

Berdasarkan hasil pengujian LOT01, sebanyak 702 request berhasil diproses dari total 702 request dengan tingkat kegagalan sebesar 0%. Waktu respons rata-rata yang diperoleh adalah 13,703 ms, sedangkan waktu respons maksimum sebesar 39,363 ms. Hasil tersebut berada jauh di bawah threshold 5.000 ms. Dengan demikian, performa fitur utama sistem pada skenario LOT01 dinyatakan memenuhi kriteria keberhasilan.

Berdasarkan hasil pengujian LOT02, sebanyak 408 request berhasil diproses dari total 408 request dengan tingkat kegagalan sebesar 0%. Waktu respons rata-

rata yang diperoleh adalah 15,954 ms, sedangkan waktu respons maksimum sebesar 46,436 ms. Hasil tersebut berada di bawah threshold 5.000 ms. Dengan demikian, performa fitur real-time parkir pada skenario LOT02 dinyatakan memenuhi kriteria keberhasilan.

2. Security Testing

Tabel 15. Security Testing

Komponen		Rincian	
Tes Case ID		TC02-NF02	
Type		Security Testing	
Tujuan		Memastikan sistem terlindungi dari ancaman keamanan seperti injeksi SQL, <i>cross-site scripting</i> , dan akses tidak sah.	
Lingkungan		Pengujian pada server pengembangan dengan akses terbatas untuk mencegah kebocoran data.	
Alat		OWASP ZAP	
Metode		1. Melakukan pengujian penetrasi (<i>penetration testing</i>) terhadap fitur aplikasi. 2. Mengidentifikasi kerentanan dan melakukan <i>eksploitasi</i> terkontrol. 3. Menganalisis keamanan data saat transit dan penyimpanan.	
Ekspektasi		Sistem bebas dari kerentanan utama, data terenkripsi dengan baik, dan akses hanya dapat dilakukan oleh pihak yang berwenang.	
Hasil Uji			
Jumlah Alert		13	
No	Temuan	Tingkat Risiko	Penjelasan
1	Vulnerable JS Library	Tinggi	Library Swiper versi 8.4.7 yang digunakan memiliki kerentanan yang tercatat pada CVE-2026-27212. Penggunaan library dengan versi usang berisiko dieksploitasi oleh penyerang melalui celah keamanan yang sudah diketahui publik. Solusi: upgrade ke versi 12.1.2 atau lebih baru.

2	Content Security Policy (CSP) Header Not Set	Menengah	Header Content-Security-Policy tidak dikonfigurasi pada aplikasi. Ketiadaan CSP meningkatkan risiko serangan Cross Site Scripting (XSS) dan injeksi data karena browser tidak mengetahui sumber konten mana yang diizinkan untuk dimuat.
3	Cross-Domain Misconfiguration	Menengah	Konfigurasi CORS pada resource pihak ketiga (jsdelivr.net, cdnjs.cloudflare.com, unpkg.com) menggunakan Access-Control-Allow-Origin: * yang terlalu permisif. Hal ini membuka potensi akses data oleh domain yang tidak terpercaya.
4	Missing Anti-clickjacking Header	Menengah	Halaman aplikasi tidak memiliki perlindungan terhadap serangan Clickjacking karena tidak menyertakan header X-Frame-Options maupun direktif frame-ancestors pada CSP. Penyerang dapat menyematkan halaman dalam iframe untuk mengelabui pengguna agar melakukan aksi yang tidak disengaja.
5	Cookie No HttpOnly Flag	Rendah	Cookie XSRF-TOKEN disetel tanpa flag HttpOnly sehingga dapat diakses melalui JavaScript. Apabila terdapat celah XSS pada aplikasi, penyerang dapat mencuri cookie tersebut dan berpotensi melakukan pemalsuan permintaan lintas situs (CSRF).
6	Authentication Request Identified	Informasional	ZAP mendeteksi adanya permintaan autentikasi pada endpoint /login-proses

			dan /registrasi-proses. Alert ini bersifat informasional dan menandakan ZAP berhasil mengidentifikasi form autentikasi beserta parameter email, password, dan CSRF token yang digunakan aplikasi.
7	Modern Web Application	Informasional	Aplikasi teridentifikasi sebagai modern web application karena menggunakan JavaScript framework (Alpine.js) dan elemen interaktif tanpa atribut href tradisional. Alert ini bersifat informasional dan menyarankan penggunaan Ajax Spider untuk eksplorasi aplikasi yang lebih menyeluruh.
8	Retrieved from Cache	Informasional	Beberapa resource seperti file JavaScript dari cdn.jsdelivr.net dan cdnjs.cloudflare.com diambil dari cache server. Alert ini bersifat informasional dan menginformasikan bahwa konten di-cache oleh server proxy, yang berpotensi menampilkan data lama kepada pengguna berbeda.
9	Session Management Response Identified	Informasional	ZAP mendeteksi token manajemen sesi berupa cookie laravel_session dan XSRF-TOKEN pada respons aplikasi. Alert ini bersifat informasional dan menunjukkan mekanisme sesi Laravel berjalan sebagaimana mestinya, namun perlu dipastikan konfigurasi keamanan cookie sudah memadai.
Screenshot			

Generated on Fri, 5 Jun 2026 09:26:16

ZAP Version: 2.17.0

ZAP by [Checkmarx](#)

Summary of Alerts

Risk Level	Number of Alerts
High	1
Medium	3
Low	1
Informational	4

Alerts

Name	Risk Level	Number of Instances
Vulnerable JS Library	High	1
Content Security Policy (CSP) Header Not Set	Medium	Systemic
Cross-Domain Misconfiguration	Medium	Systemic
Missing Anti-clickjacking Header	Medium	2
Cookie No HttpOnly Flag	Low	Systemic
Authentication Request Identified	Informational	2
Modern Web Application	Informational	Systemic
Retrieved from Cache	Informational	Systemic

4.3 Keterbatasan Sistem

Berdasarkan hasil implementasi dan pengujian, sistem SPARKING masih memiliki keterbatasan pada pengelolaan sumber kamera. Sistem belum menyediakan fitur konfigurasi multi-kamera untuk menghubungkan kamera yang berbeda pada setiap zona atau sub-zona. Dengan demikian, skenario dua zona yang menggunakan kamera berbeda belum dapat dikonfigurasi secara independen melalui sistem. Keterbatasan ini disebabkan oleh arsitektur deteksi yang masih menggunakan satu alur pemrosesan AI melalui Raspberry Pi 5, YOLOv8, dan layanan Flask. Pengembangan selanjutnya dapat menambahkan manajemen kamera serta relasi antara kamera dan zona atau sub-zona agar sistem dapat mendukung pemantauan multi-kamera.

BAB V KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil perancangan, pengembangan, dan pengujian aplikasi Smart Parking berbasis Kecerdasan Buatan (AI) di lingkungan area parkir Politeknik Negeri Batam, dapat disimpulkan bahwa:

1. Integrasi teknologi AI dalam sistem parkir berhasil diterapkan untuk mendeteksi kendaraan roda empat dan memantau status ketersediaan slot parkir secara real-time. Hasil pengujian fungsional monitoring slot parkir menunjukkan 100% test case berhasil, dengan seluruh 7 test case dinyatakan lolos dan 0% test case gagal.
2. Fitur analisis parkir berbasis data berhasil diimplementasikan untuk menampilkan informasi penggunaan area parkir berdasarkan zona. Sistem mampu menyajikan visualisasi analisis penggunaan parkir, termasuk informasi waktu penggunaan dan hari tersibuk berdasarkan data yang tersedia pada sistem.
3. Sistem notifikasi parkir berhasil diimplementasikan menggunakan mekanisme notifikasi email. Berdasarkan hasil Black Box Testing, 3 dari 3 test case notifikasi berhasil dijalankan dengan persentase keberhasilan 100% dan tingkat kegagalan 0%. Pengujian mencakup proses penambahan antrian notifikasi, pengiriman email ketika slot parkir tersedia pada zona yang dipilih pengguna, serta pembatalan notifikasi.
4. Berdasarkan pengujian performa menggunakan Locust, sistem mampu memproses 702 request pada pengujian fitur utama dan 408 request pada pengujian fitur real-time parkir dengan error rate sebesar 0% pada kedua skenario. Waktu respons maksimum yang diperoleh masing-masing sebesar 39,363 ms dan 46,436 ms, sehingga masih berada di bawah batas kebutuhan non-fungsional sebesar 5 detik. Dengan demikian, sistem SPARKING telah berhasil menyediakan fungsi monitoring slot parkir, analisis penggunaan area parkir, dan notifikasi ketersediaan slot.

5.2 Saran

Berdasarkan hasil pengembangan yang telah dilakukan, terdapat beberapa saran yang dapat digunakan untuk pengembangan sistem pada penelitian selanjutnya, yaitu:

1. Mengembangkan fitur manajemen dan konfigurasi multi-kamera dengan menyediakan pengaturan hubungan antara kamera, zona, dan sub-zona sehingga setiap area parkir dapat menggunakan sumber kamera yang berbeda secara independen.
2. Meningkatkan aspek keamanan sistem dengan memperbaiki konfigurasi keamanan aplikasi serta memperbarui library yang digunakan agar terhindar dari kerentanan keamanan.
3. Menambahkan fitur notifikasi melalui media lain, seperti WhatsApp atau Telegram, sehingga pengguna dapat menerima informasi ketersediaan slot parkir dengan lebih cepat.
4. Melakukan pengujian sistem dengan jumlah pengguna yang lebih besar dan pada kondisi penggunaan yang lebih beragam untuk memperoleh hasil evaluasi yang lebih komprehensif.

DAFTAR PUSTAKA

- Abidin, M. Z., & Pulungan, R. (2020). A systematic review of machine-vision-based smart parking systems. *Scientific Journal of Informatics*, 7(2), 213–227.
- Al-Turjman, F., & Malekloo, A. (2019). Smart parking systems: A comprehensive review. *IEEE Access*, 7, 105532–105553. <https://doi.org/10.1109/ACCESS.2019.2932194>
- Biyik, C., Allam, Z., Pieri, G., Moroni, D., O’Fraifer, M., O’Connell, E., Olariu, S., & Khalid, M. (2021). Smart Parking Systems: Reviewing the Literature, Architecture and Ways Forward. *Smart Cities*, 4(2), 623–642. <https://doi.org/10.3390/smartcities4020032>
- Bassil, Y. (2012). A simulation model for the waterfall software development life cycle. *International Journal of Engineering & Technology (IJET)*, 2(5), 1–7.
- Caicedo, F. (2010). Real-time parking information management to reduce search time, vehicle displacement and emissions. *Transportation Research Part D: Transport and Environment*, 15(4), 228–234. <https://doi.org/10.1016/j.trd.2010.02.008>
- Haslinda, A., Ng, R., & Mohamed, R. (2016). Factors influencing haphazard parking in the Klang Valley, Malaysia. *International Journal of Academic Research in Business and Social Sciences*, 6(9), 159–181. <https://doi.org/10.6007/IJARBSS/v6-i9/2303>
- Hossain, M. S., Muhammad, G., & Rahman, S. M. (2020). Cyber–physical smart parking system using edge computing. *Future Generation Computer Systems*, 110, 566–574. <https://doi.org/10.1016/j.future.2019.10.013>
- Khan, A., Kulkarni, P., Shergold, I., Jones, M., Dogramadzi, M., Carnelli, P., & Sooriyabandara, M. (2020). Reducing parking space search time and environmental impacts: A technology-driven smart parking case study. *IEEE Technology and Society Magazine*, 39(3), 62–75. <https://doi.org/10.1109/MTS.2020.3012329>
- Otwell, T. (2020). *Laravel: Up & running* (2nd ed.). O’Reilly Media.

- Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. *arXiv:1804.02767*. <https://arxiv.org/abs/1804.02767>
- Russell, S., & Norvig, P. (2021). *Artificial intelligence: A modern approach* (4th ed.). Pearson.
- Saputra, A. E., Giawa, B. K. S., & Khana, R. (2023). Vehicle and parking space detection for smart parking systems using the YOLOv5 method. *In Proceedings of the 4th International Seminar and Call for Paper (ISCP UTA '45 JAKARTA 2023)* (pp. 458–466). SCITEPRESS – Science and Technology Publications. <https://doi.org/10.5220/0012584700003821>
- Saravanan, G., Salomon, B., S., Cibi, S., & P., Hari Krishnan. (2024). Smart Parking System with Computer Vision. *International Research Journal on Advanced Engineering Hub (IRJAEH)*, 2(12), 2816-2820. <https://doi.org/10.47392/IRJAEH.2024.0389>
- Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.

DAFTAR LAMPIRAN

Lampiran A. Dokumen Proses Pengumpulan Requirement

Dokumentasi Work Process

https://drive.google.com/drive/folders/1lE5w8ktaJIXHLoLNYKdjf-IvEO5s_9c?usp=drive_link

Lampiran B. Link Produk

Repository GitHub

<https://github.com/mzuamhsn/Sparking.git>

Lampiran C. Dokumen Pengujian UAT

Security Testing

https://drive.google.com/file/d/1QOnZ9oenymxFELbvW812nT4FoW8VTvx/view?usp=drive_link

Dokumen User Acceptance Testing (UAT)

https://drive.google.com/drive/folders/1ThqtY_OBwMvqde83NPChE27OyOapJPZT?usp=drive_link