

Design of a Web-Based Car Rental Application Using the Agile Scrum Method

[Perancangan Aplikasi Rental Mobil Berbasis website Menggunakan Metode Agile Scrum]

Taufik Darmawan¹⁾, Uce Indahyanti²⁾

¹⁾Program Studi Teknik Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia.

²⁾Program Studi Teknik Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia.

*Email Penulis Korespondensi : uceindahyanti@umsida.ac.id

Abstract. *In the digital era, the demand for easy and efficient access is increasing, including in the car rental industry. The development of a web-based car rental application has become an effective solution to meet consumers' needs for fast and practical services. This industry has experienced rapid growth, driven by increased mobility, economic growth, and the need for transportation flexibility. This study aims to develop a web-based car rental application using the Agile Scrum methodology, which is considered superior to previous methods. Laravel was chosen as the framework to support the efficient and structured development of the application. The combination of Agile Scrum and Laravel is expected to produce an application that simplifies administrative management, penalty implementation, and the booking of multiple cars by a single user account.*

Keywords - Information System, Rental Car, Agile, Scrum, Website, Laravel

Abstrak. *Pada era digital, kebutuhan akan akses yang mudah dan efisien semakin meningkat, termasuk dalam industri rental mobil. Pengembangan aplikasi rental mobil berbasis website menjadi solusi efektif untuk memenuhi kebutuhan konsumen akan layanan yang cepat dan praktis. Industri ini mengalami pertumbuhan pesat, didorong oleh peningkatan mobilitas, pertumbuhan ekonomi, dan kebutuhan akan fleksibilitas transportasi. Penelitian ini bertujuan untuk mengembangkan aplikasi rental mobil berbasis web menggunakan metode Agile Scrum, yang dianggap lebih unggul dibandingkan metode sebelumnya. Laravel dipilih sebagai framework untuk mendukung pengembangan aplikasi yang efisien dan terstruktur. Kombinasi Agile Scrum dan Laravel diharapkan dapat menghasilkan aplikasi yang mempermudah administrasi pengelolaan, penerapan denda, dan pemesanan mobil oleh satu akun pengguna.*

Kata Kunci - Sistem Informasi, Rental Mobil, Agile, Scrum, Website, Laravel

I. PENDAHULUAN

Dalam era digital yang terus berkembang, kebutuhan akan layanan yang cepat, efisien, dan mudah diakses menjadi semakin penting, terutama dalam industri penyewaan mobil [1]. Penyedia layanan rental mobil harus dapat mengakomodasi kebutuhan pelanggan dengan menawarkan pengalaman yang mudah dan menyenangkan, baik dalam proses pemesanan, pengelolaan armada, hingga pembayaran. Oleh karena itu, pengembangan aplikasi berbasis web menjadi solusi yang tepat untuk memenuhi kebutuhan tersebut [2].

Aplikasi rental mobil berbasis web memungkinkan pelanggan untuk memesan kendaraan dengan cepat tanpa harus datang langsung ke lokasi penyewaan [3]. Dengan hanya beberapa klik, pengguna dapat memilih jenis mobil yang diinginkan, menentukan waktu dan tempat penjemputan, serta menyelesaikan pembayaran secara online. Di sisi lain, penyedia layanan dapat mengelola armada kendaraan mereka secara lebih efektif, melacak ketersediaan, melakukan pemeliharaan rutin, serta memantau performa kendaraan secara real-time. Hal ini tidak hanya meningkatkan efisiensi operasional tetapi juga kualitas layanan yang diberikan kepada pelanggan [4].

Tinjauan literatur singkat menunjukkan bahwa sejumlah penelitian sebelumnya telah berupaya untuk mengatasi berbagai tantangan dalam pengelolaan layanan rental mobil berbasis web [5]. Namun, sebagian besar penelitian tersebut hanya berfokus pada aspek-aspek tertentu dari pengelolaan, seperti penambahan sanksi bagi pelanggan yang melanggar aturan, atau fitur yang memungkinkan satu pengguna untuk menyewa lebih dari satu kendaraan sekaligus. Pendekatan ini memang memberikan solusi terhadap masalah-masalah spesifik, tetapi seringkali mengabaikan kebutuhan untuk menciptakan sistem yang holistik dan terintegrasi [6].

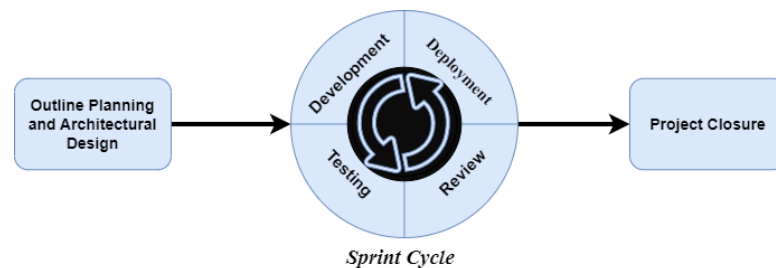
Oleh karena itu, penelitian ini bertujuan untuk mengevaluasi perancangan aplikasi rental mobil berbasis website menggunakan metode agile scrum sebagai solusi terhadap masalah-masalah spesifik untuk pengelolaan layanan rental mobil [7]. Aplikasi ini dirancang untuk menyediakan platform terpadu yang memungkinkan pemesanan kendaraan, pengelolaan armada, pemeliharaan, serta pengaturan pembayaran. Selain itu, penelitian ini memanfaatkan Framework Laravel yang digunakan untuk mengembangkan sistem ini [8].

Melalui penelitian ini, kami berharap untuk mencapai beberapa tujuan. Pertama, kami ingin meningkatkan efisiensi operasional dalam pengelolaan layanan rental mobil, termasuk dalam hal pemesanan kendaraan, pengelolaan armada, pemeliharaan, serta pengaturan pembayaran [9]. Kedua, kami bertujuan untuk meningkatkan transparansi dan aksesibilitas informasi terkait layanan rental mobil bagi semua pihak yang terlibat, baik penyedia jasa maupun pelanggan [10]. Terakhir, kami berharap untuk menyediakan platform yang memungkinkan evaluasi yang lebih baik terhadap performa armada dan kepuasan pelanggan secara keseluruhan [11]. Penelitian ini diharapkan dapat memberikan kontribusi yang signifikan dalam mengoptimalkan pengalaman pengguna, serta membuka peluang untuk pengembangan lebih lanjut dalam industri rental mobil [12].

II. METODE

Penelitian ini menerapkan metodologi Agile dengan model Scrum serta memanfaatkan framework Laravel untuk Pengembangan Aplikasi Rental Mobil Berbasis Website [13]. Agile dipilih karena kemampuannya untuk memberikan fleksibilitas dan adaptabilitas yang tinggi, sesuai untuk proyek-proyek perangkat lunak yang dinamis. Model Scrum memberikan struktur kerja yang terorganisir melalui pembagian tugas ke dalam sprint singkat, sehingga memfasilitasi kolaborasi tim dan perbaikan berkelanjutan [14]. Laravel dipilih sebagai framework karena fitur-fitur unggulnya dalam pembuatan aplikasi web, seperti routing, autentikasi, dan pengelolaan database, serta kemampuannya untuk mendukung skalabilitas dan pemeliharaan yang efektif [15].

Pada Model Scrum mengambil prinsip – prinsip umum yang ada pada metode Agile, akan tetapi lebih berfokus pada pengelolaan pengembangan berulang dari pada pendekatan teknis secara spesifik untuk rekayasa perangkat lunak dengan metode Agile [16]. Terdapat tiga fase dalam Scrum seperti yang ditunjukkan pada gambar 2.1.



Gambar 2.1 Proses Scrum

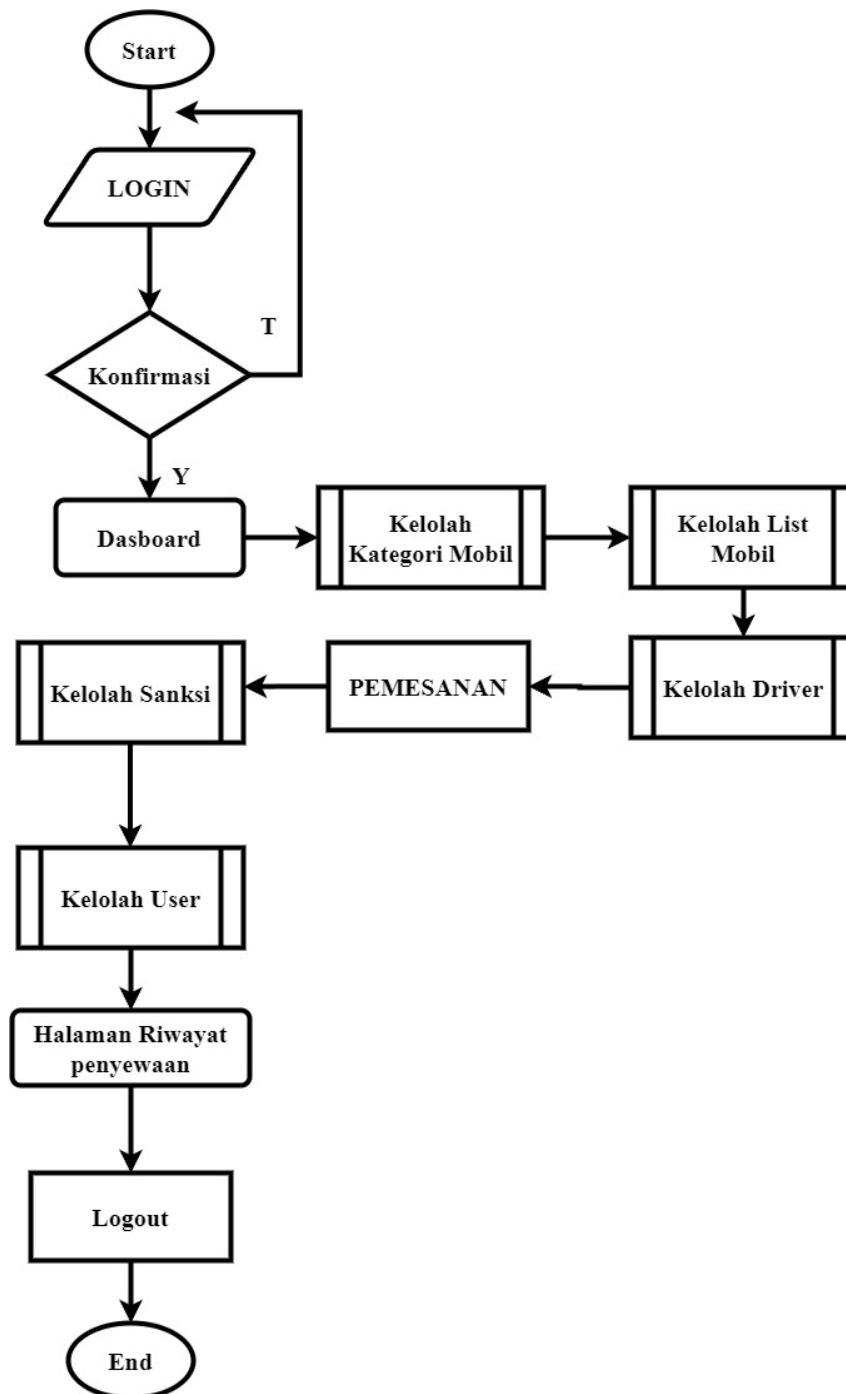
1. Fase Pertama adalah proses perencanaan yang digunakan untuk menetapkan tujuan project dan perancangan design perangkat lunak
2. Fase kedua adalah Sprint Cycle, yang merupakan tahap dimana setiap siklus berfokus kepada pengembangan tambahan dalam sistem. Pada fase Sprint dilakukan Development (pengembangan), Testing (pengujian), Review (Ulasan) dan Deployment (penyebaran).
3. Fase Ketiga adalah Fase penutupan project dengan membuat penjelasan cara menggunaan aplikasi

III. HASIL DAN PEMBAHASAN

A. Perancangan Sistem

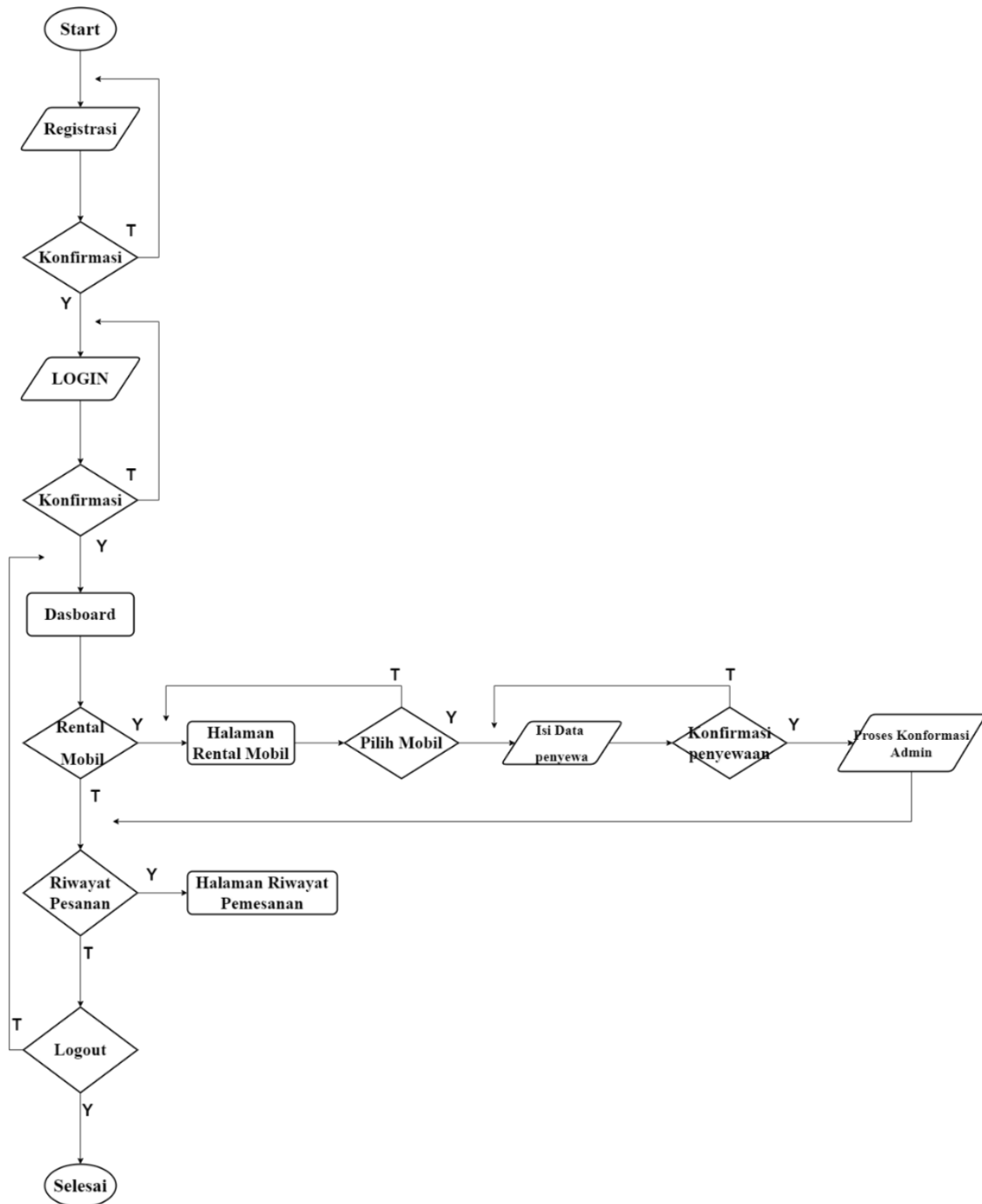
1. Flowchart

Flowchart digunakan untuk membantu memahami diagram yang menggambarkan alur proses dalam suatu sistem. Hasil penelitian ini mengidentifikasi dua level pengguna, yaitu Admin dan Penyewa, yang masing-masing memiliki alur proses yang berbeda.



Gambar 3. 1 Flowchart Admin

Gambar 3. 1 menjelaskan alur yang akan dilakukan oleh user Admin. Setelah berhasil login sebagai admin, akan muncul tampilan khusus untuk admin yang berisi berbagai aktivitas dalam sistem. Di dalam tampilan ini, admin dapat melakukan input, edit, dan delete karena admin bertanggung jawab untuk mengelola data yang diperlukan dalam website rental mobil.

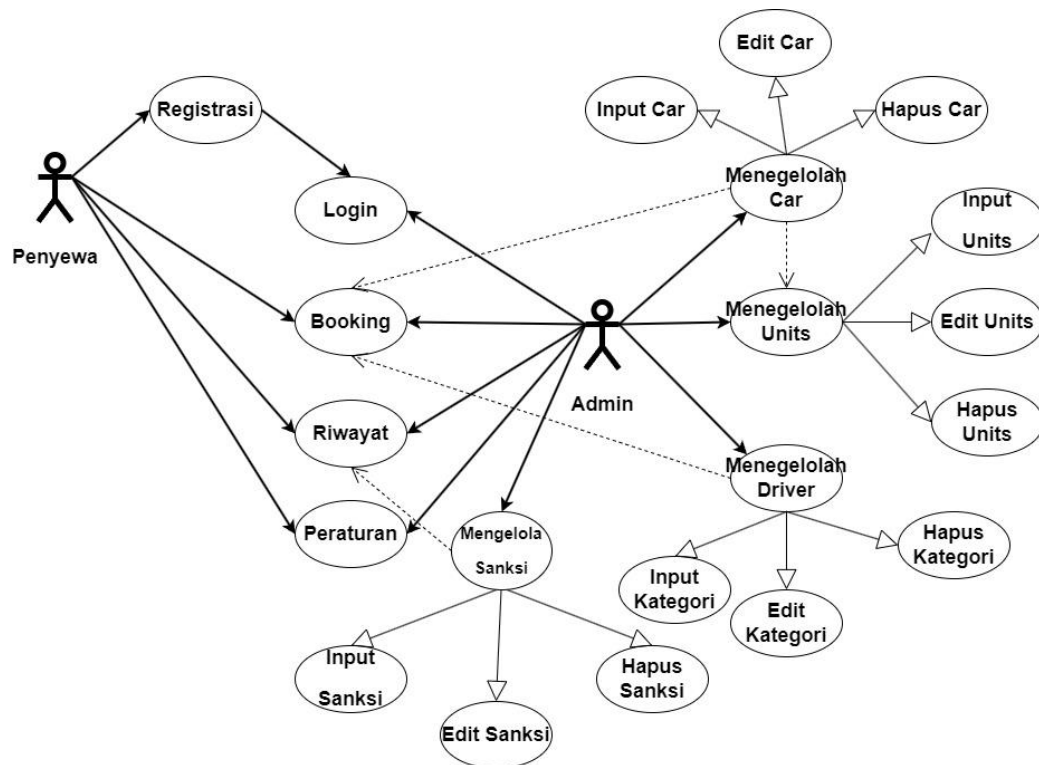


Gambar 3. 2 Flowchart Penyewa

Gambar 3. 2 menjelaskan alur yang akan dilakukan oleh user penyewa dapat dijelaskan bahwa Setelah login sebagai penyewa, akan muncul tampilan khusus untuk penyewa yang menampilkan berbagai aktivitas dalam sistem. Di dalam tampilan ini, penyewa dapat memilih mobil yang ingin dipesan dan melanjutkan proses pemesanan.

2. Use Case Diagram

Use case diagram adalah representasi pemodelan perilaku sistem yang akan dikembangkan. Setiap kasus pengguna membantu dalam menjelaskan hubungan umum antara pengguna suatu sistem dengan sistem itu sendiri melalui cerita tentang bagaimana sistem tersebut digunakan.

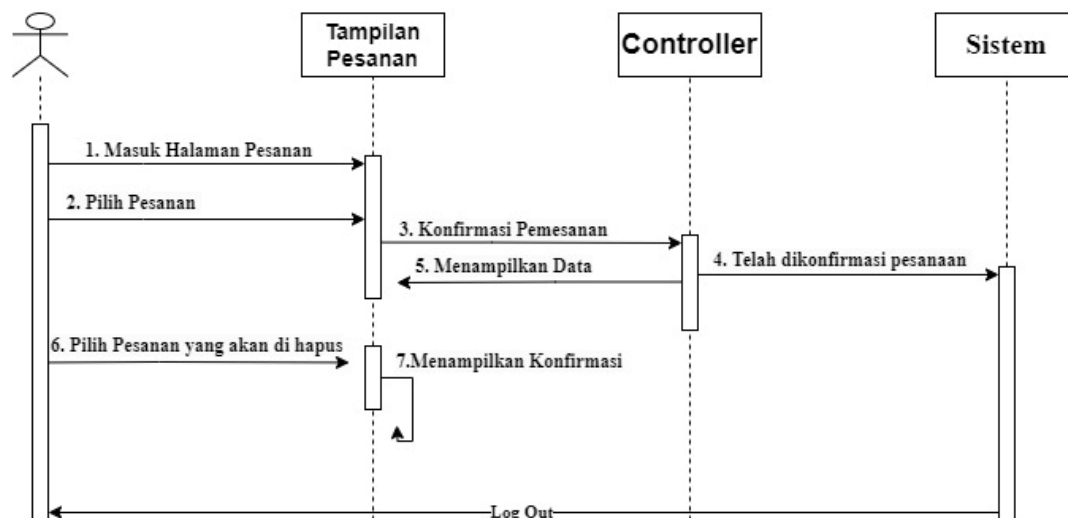


Gambar 3. 3 Use Case Diagram

Gambar 3. 3 menunjukkan user penyewa yang dapat melakukan registrasi dan login, serta dapat membuat pesanan dan melihat riwayat pemesanan. Sementara itu, user admin mengelola data Kategori, Mobil, Sopir, dan mengelola sanksi. Data yang dikelola oleh admin akan masuk ke dalam data pemesanan.

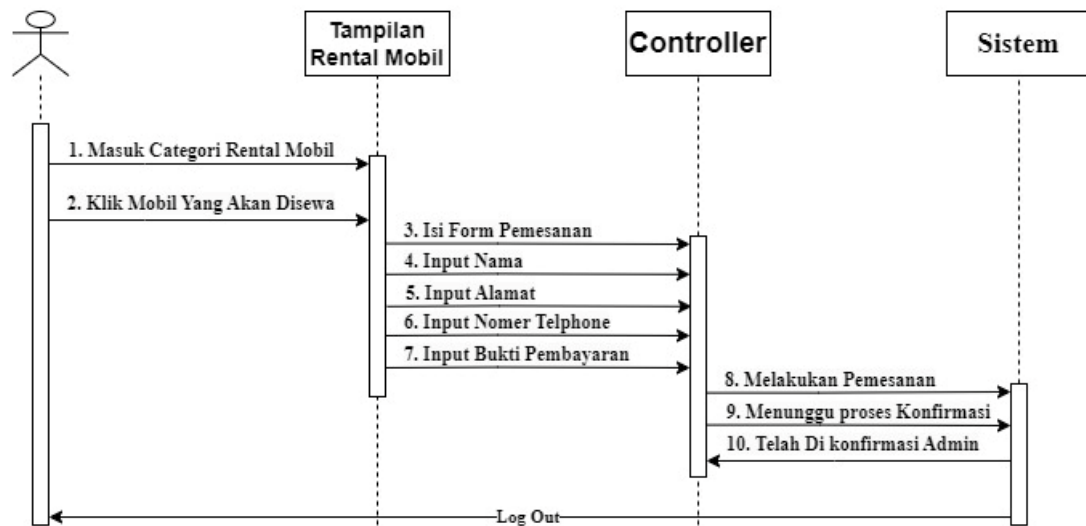
3. Sequence diagrams

Sequence diagrams (diagram urutan) adalah jenis diagram interaksi dalam pemodelan sistem yang menunjukkan bagaimana obyek-obyek berinteraksi dalam suatu skenario tertentu dari waktu ke waktu. Berikut ini adalah Sequence Diagram yang dilakukan oleh user.



Gambar 3.4 Sequence Admin Mengkonfirmasi Pesanan

Gambar 3.4 menunjukkan sequence diagram dari Admin yang akan melakukan konfirmasi pemesanan bahwa, penyewa telah mengirimkan bukti pembayaran dan disana admin bisa menghapus data pemesanan yang dilakukan penyewa.



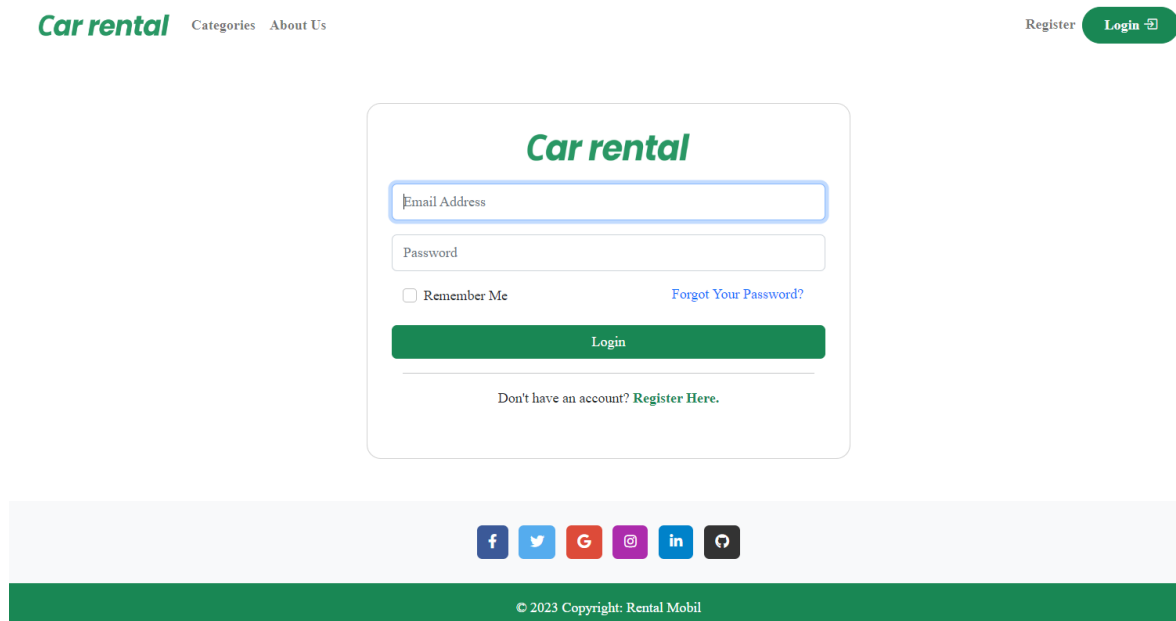
Gambar 3.5 Sequence Diagram Penyewa Melakukan Pemesanan

Gambar 3.5 adalah Sequence Diagram yang digunakan user Penyewa untuk melakukan pemesanan mobil, sesudah melakukan pesanan maka user penyewa harus mengirim bukti pembayaran setelah itu menunggu konfirmasi oleh admin.

A. Design Interface

Aplikasi untuk user

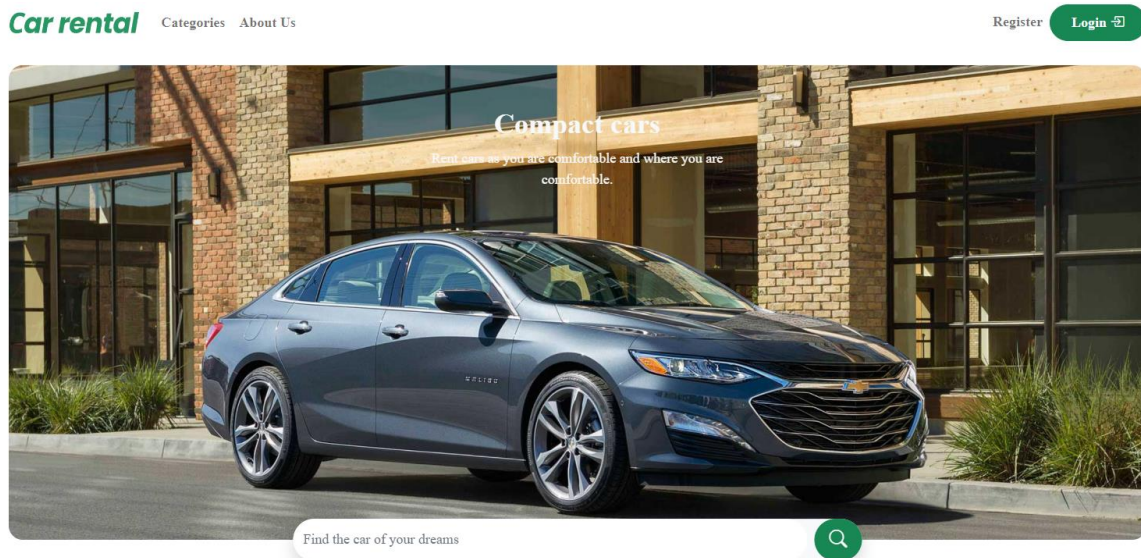
1. Halaman Login



Gambar. 3.6 Halaman Login

Pada gambar 3.6 halaman ini digunakan untuk Admin dan User login supaya bisa mengakses website.

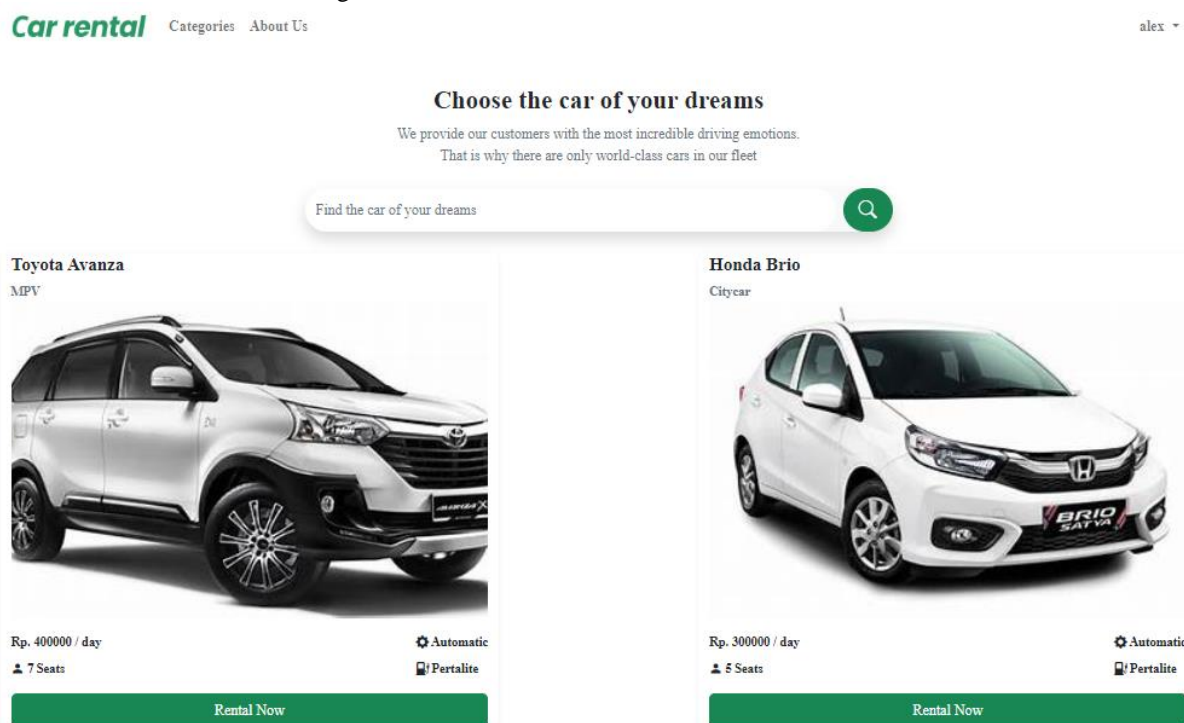
2. Halaman Utama



Gambar 3.7 Halamanan Utama

Pada gambar 3.7 merupakan tampilan dari halaman utama yang digunakan untuk mencari mobil yang akan disewa.

3. Halaman Categories



Gambar 3.8 Halaman Categories

Pada Gambar 3.8 merupakan Halaman Categories yang berisi informasi mobil yang bisa di sewa, dan user bisa memilih mobil yang ingin disewa.

4. Halaman Order


Car rental

Categories About Us

alex

Peraturan		Sanksi	
No	Nama Peraturan	No	Nama Sanksi
1	Penyewa harus berusia minimal 21 tahun dan memiliki SIM (Surat Izin Mengemudi) yang masih berlaku sesuai dengan jenis kendaraan yang akan disewa.	1	Penyewa yang terlambat mengembalikan mobil akan dikenakan biaya tambahan sebesar Rp 50,000 per jam.
2	Penyewa harus mengisi formulir penyewaan yang tersedia di kantor atau melalui aplikasi website rental.	2	Jika keterlambatan melebihi 4 jam, penyewa akan dikenakan biaya tambahan setara dengan 1 hari tarif sewa.
3	Pembayaran sewa dilakukan di muka saat pengambilan mobil. Pembayaran bisa dilakukan melalui transfer bank	3	Untuk kerusakan ringan (misalnya goresan kecil atau kerusakan interior), penyewa akan dikenakan denda sebesar Rp 500,000 atau sesuai dengan biaya perbaikan, mana yang lebih tinggi.
4	Penyewa wajib mengembalikan mobil dalam kondisi yang sama seperti saat disewa (bersih dan dengan bahan bakar penuh).	4	Untuk kerusakan sedang hingga berat (misalnya kerusakan bodi atau mesin), penyewa akan dikenakan denda sesuai dengan estimasi biaya perbaikan di bengkel resmi, ditambah biaya administrasi sebesar
5	Mobil hanya boleh digunakan untuk tujuan pribadi dan tidak untuk disewakan kembali atau digunakan untuk aktivitas ilegal.	5	Kehilangan aksesoris mobil (misalnya ban cadangan, kunci, atau peralatan standar lainnya) akan dikenakan denda sesuai dengan harga barang yang hilang ditambah biaya administrasi sebesar Rp 200,000 per item.
6	Penyewa bertanggung jawab penuh atas mobil selama masa sewa, termasuk kerusakan atau kehilangan.	6	Jika mobil dikembalikan dalam kondisi sangat kotor (misalnya noda, bau tidak sedap, sampah berserakan),
7	Pembatalan sewa setelah pembayaran akan dikenakan biaya pembatalan sesuai dengan ketentuan yang berlaku.		
8	Jika mobil tidak dikembalikan sesuai dengan waktu yang telah disepakati, maka penyewa akan dikenakan biaya tambahan per jam keterlambatan.		
9	Penyewa bertanggung jawab atas segala pelanggaran lalu lintas atau denda yang terjadi selama masa sewa.		
10	Mobil harus dikembalikan sesuai dengan waktu dan lokasi yang telah disepakati.		

Toyota Avanza
MPV



Rp. 400000 / day
7 Seats
Toyota

Automatic
Pteralic

Isi Lengkap Formulir Dibawah Ini!

Nama Hari


Alamat

Total Biaya

Bukti Transfer Pembayaran Tidak ada file yang dipilih

No Rek : 201000200054

Pesan Sekarang



© 2023 Copyright: Rental Mobil

Gambar 3.9 Halaman Order

Pada Gambar 3.9 merupakan tampilan halaman order yang akan menampilkan Peraturan dan sanksi untuk penyewa, dan digunakan untuk melengkapi formulir penyewaan.

5. Halaman Riwayat pemesanan



Car rental Categories About Us alex ▾

alex	Dengan Sopir: Tidak	Rp. 800000
HP: 333728	Hari Sewa: 2 Hari	
sda		

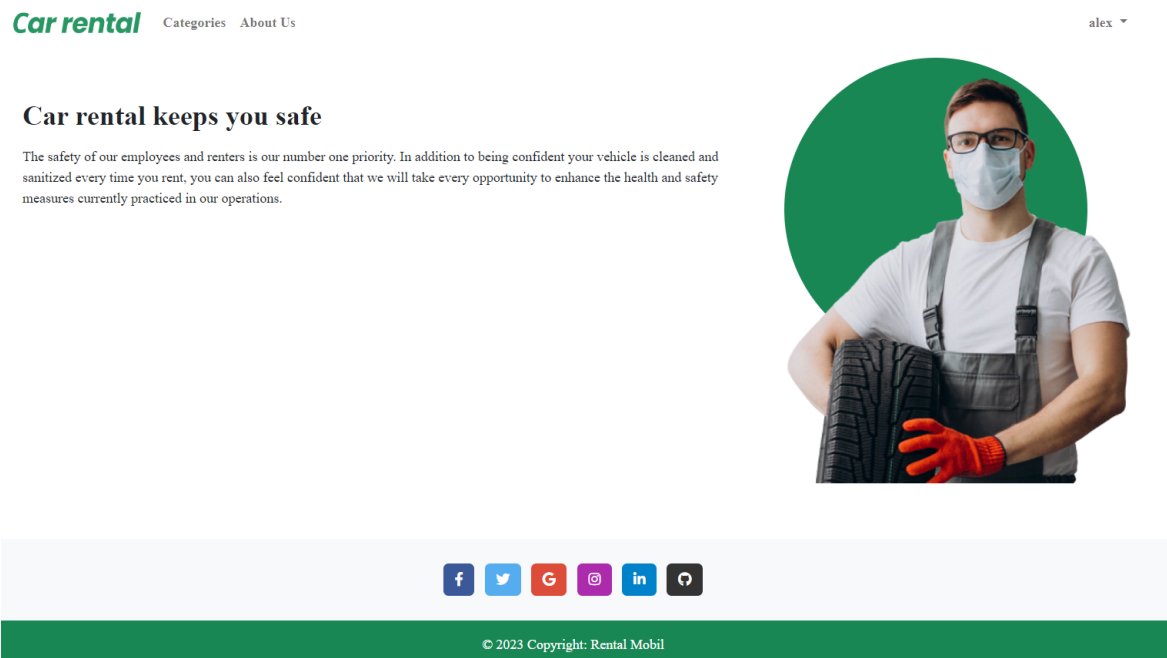
2024-08-27 06:49:39 Pending

© 2023 Copyright: Rental Mobil

Gambar 3.10 Halaman Riwayat pemesanan

Pada gambar 3.10 merupakan tampilan dari halaman transaksi berisi detail pesanan dan total transaksi pesanan.

6. Halaman About us



Car rental Categories About Us alex ▾

Car rental keeps you safe

The safety of our employees and renters is our number one priority. In addition to being confident your vehicle is cleaned and sanitized every time you rent, you can also feel confident that we will take every opportunity to enhance the health and safety measures currently practiced in our operations.

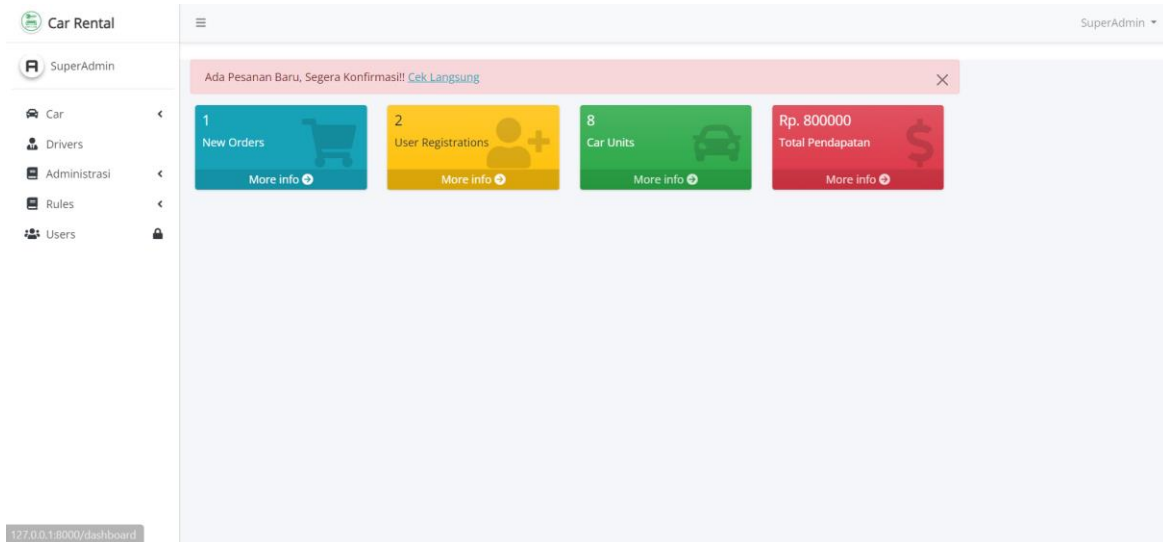
© 2023 Copyright: Rental Mobil

Gambar 3.11 Halaman Pengaturan Aplikasi

Pada gambar 3.11 merupakan halaman About Us yang menjelaskan informasi esensial tentang Perusahaan.

Aplikasi untuk Admin

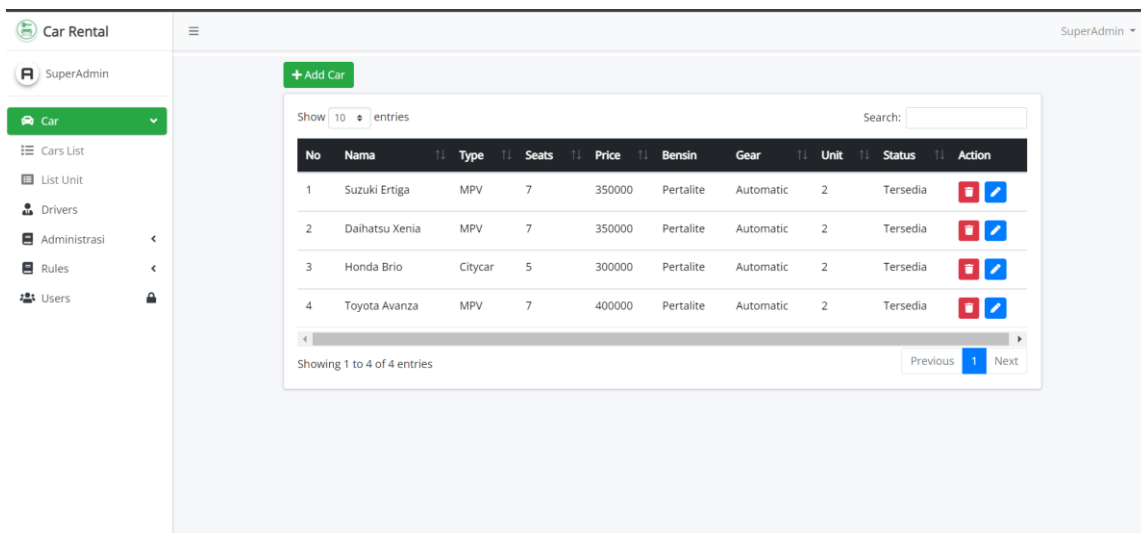
1. Halaman Dashboard Admin



Gambar 3.12 Halaman Dashboard Admin

Pada Gambar 3.12 tampilan halaman dashboard menampilkan data dari order masuk, data user, data units cars, dan total pendapatan.

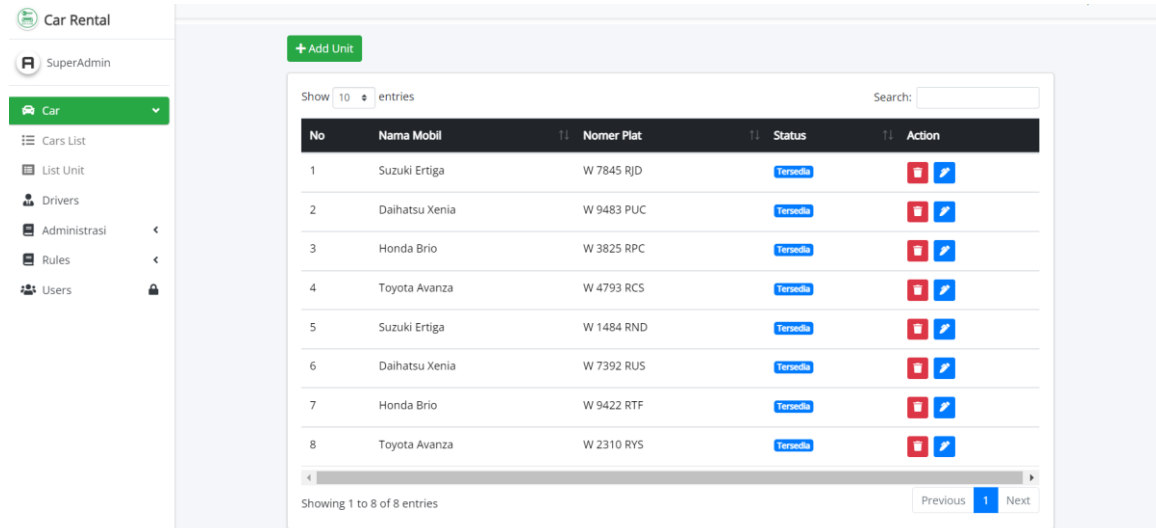
2. Halaman Car list



Gambar 3.13 Halaman Car list

Pada gambar 3.13 halaman Car list yang digunakan Admin mengelola jenis Mobil disini admin bisa melakukan CRUD (Create, Update, dan Delete).

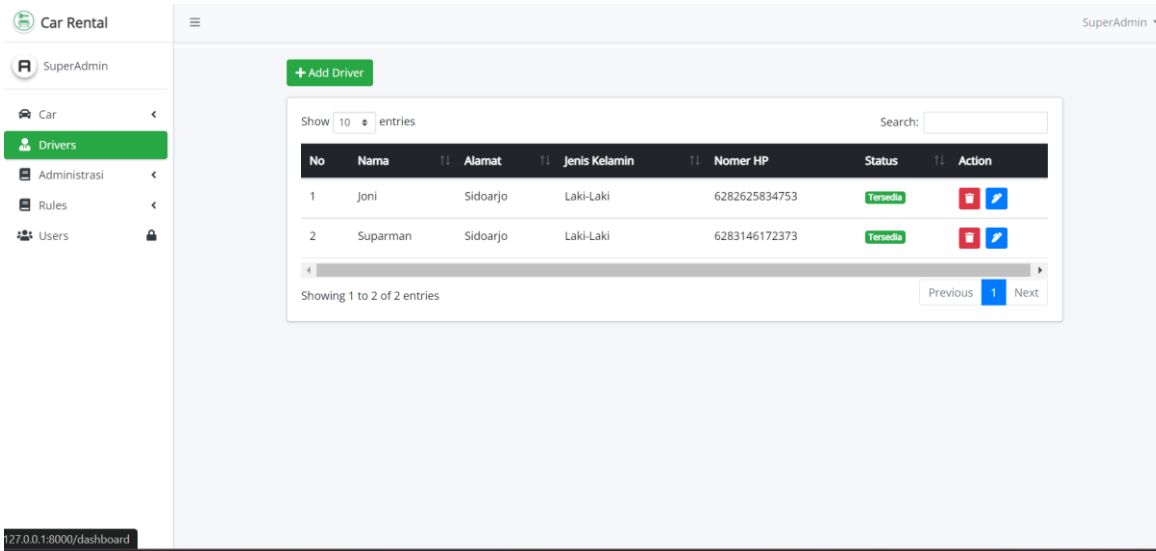
3. Halaman List unit



Gambar 2.14 Halaman List unit

Pada gambar 3.14 halaman List Unit yang digunakan Admin mengelolah Unit dari Mobil disini admin bisa melakukan CRUD (Create, Update, dan Delete).

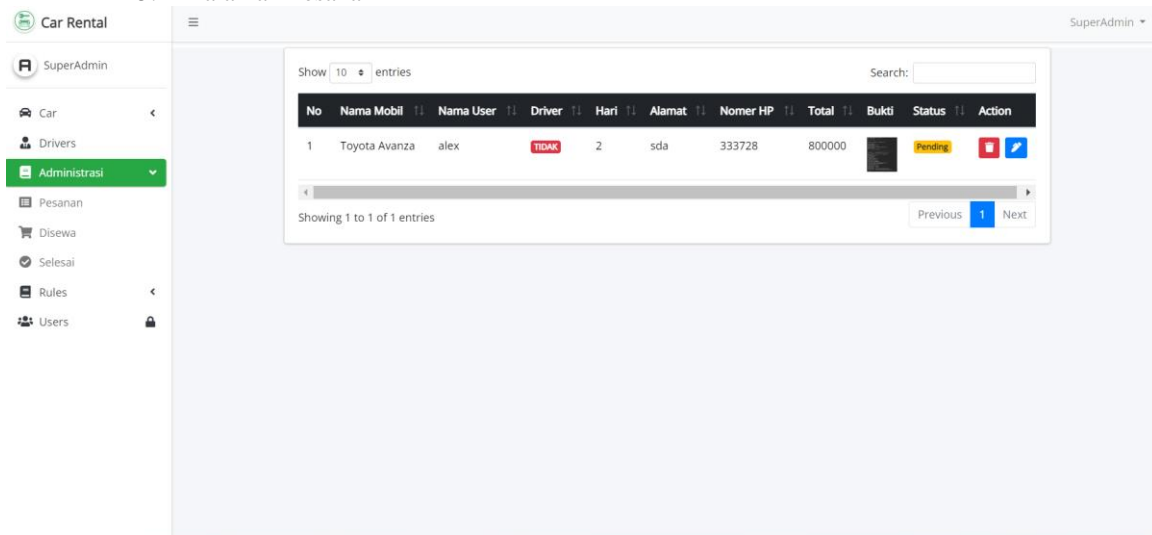
4. Halaman Driver



Gambar 3.15 Halaman Driver

Pada gambar 3.15 halaman Driver yang digunakan admin untuk mengelolah Driver, disini admin bisa melakukan CRUD (Create, Update, dan Delete).

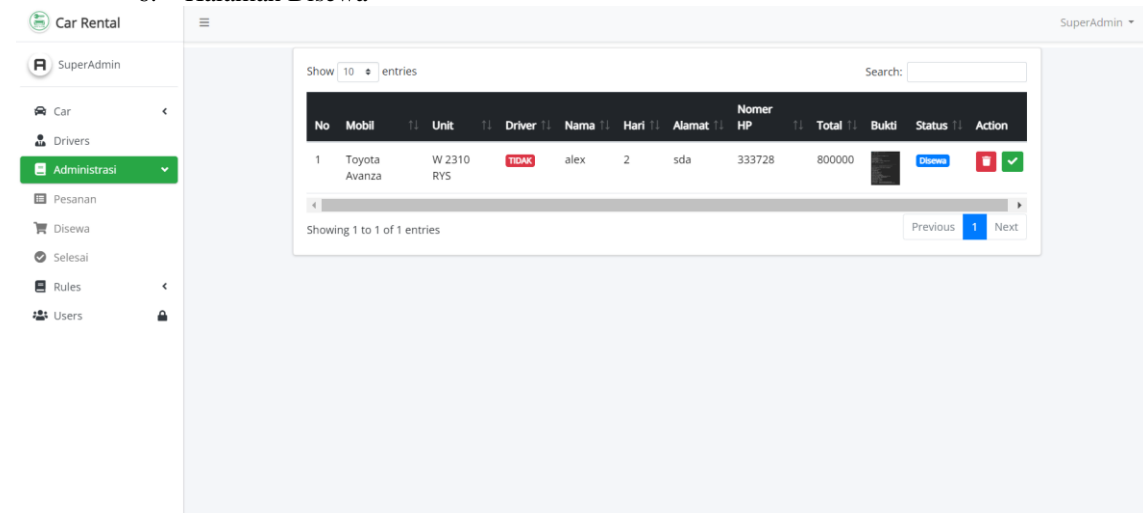
5. Halaman Pesanan



Gambar 3.16 Halaman pesanan admin

Pada gambar 3.16 Halaman pesanan admin menampilkan detail pesanan dan digunakan untuk mengkonfirmasi pesanan dari Penyewa.

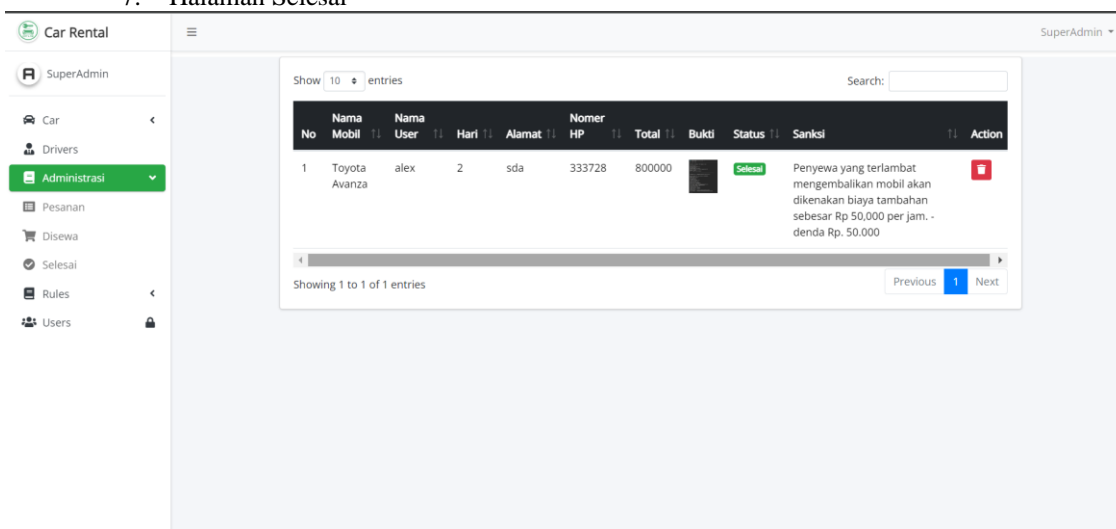
6. Halaman Disewa



Gambar 3.17 Halaman Disewa

Pada gambar 3.17 halaman Disewa ini akan menampilkan dari pesanan yang masih menyewa atau masih dalam proses penyewaan, dan admin akan melakukan konfirmasi jika pesanan sudah selesai jika mobil dikembalikan. Dan jika terjadi pelanggaran peraturan maka admin akan menginputkan sanksi di halaman tersebut.

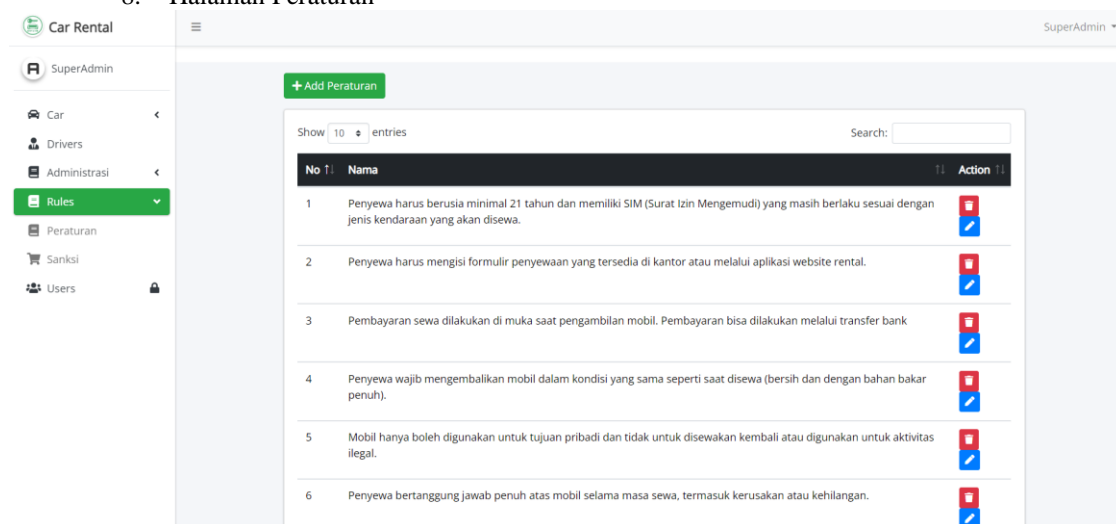
7. Halaman Selesai



Gambar 3.18 Halaman Selesai

Pada gambar 3.18 halaman Selesai ini akan menampilkan dari pesanan yang telah selesai disewa.

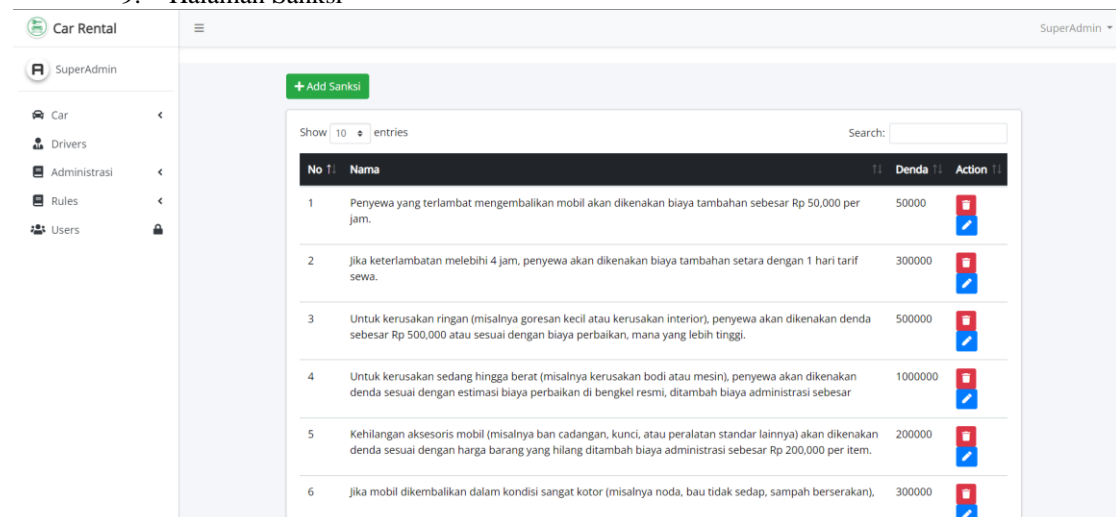
8. Halaman Peraturan



Gambar 3.19 Halaman Peraturan

Pada gambar 3.19 halaman Peraturan yang digunakan Admin mengelola Peraturan disini admin bisa melakukan CRUD (Create, Update, dan Delete).

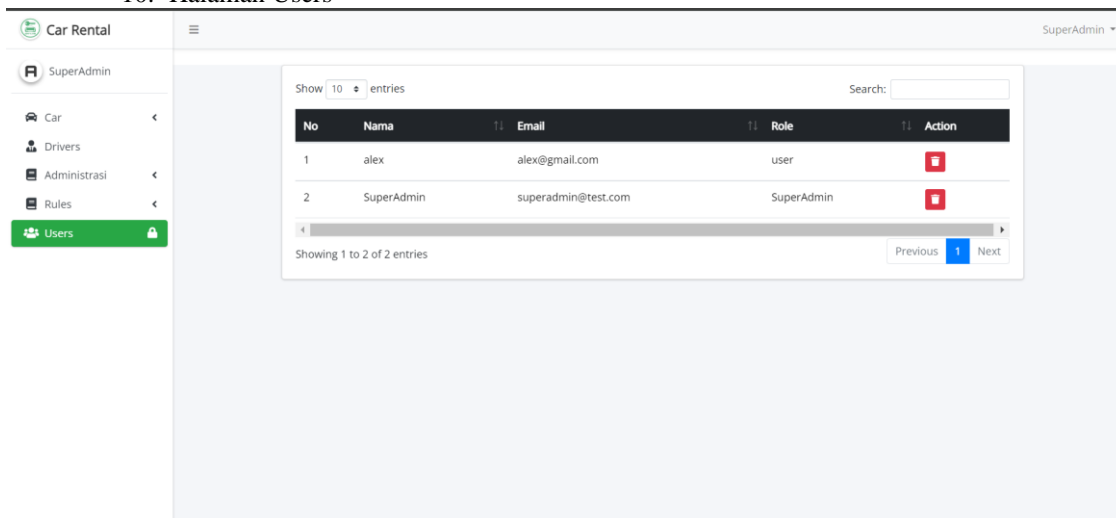
9. Halaman Sanksi



Gambar 3.20 Halaman Sanksi

Pada gambar 3.20 halaman Sanksi yang digunakan Admin mengelola Sanksi dan Denda disini admin bisa melakukan CRUD (Create, Update, dan Delete).

10. Halaman Users

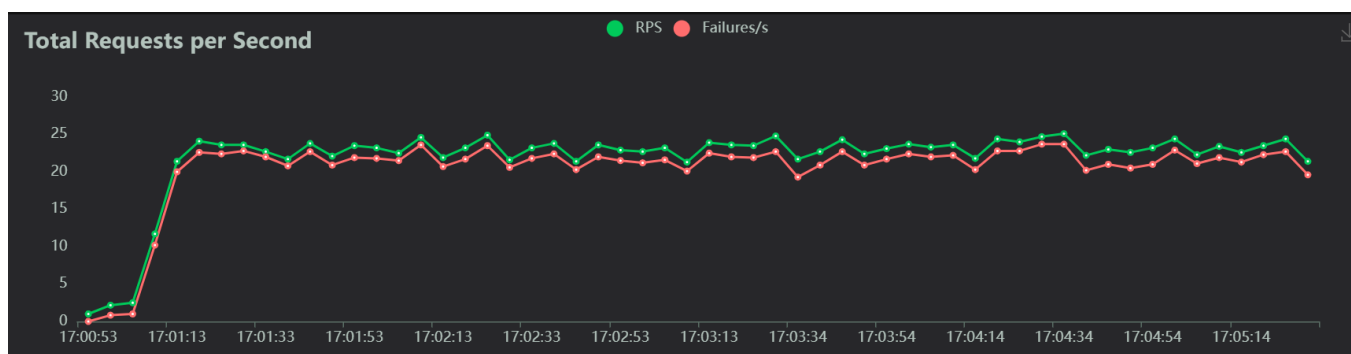


Gambar 3.21 Halaman Users

Pada gambar 3.21 halaman User yang dimana untuk menampilkan data user yang sudah terdaftar pada website ini.

B. Hasil dari Acceptance Testing

Tahap testing pengujian ini peneliti bisa menggunakan Acceptance testing yang biasanya melibatkan pengujian skenario "end-to-end" yang mencakup berbagai fitur dan fungsi yang relevan untuk menguji fungsional system yang sudah selesai dikembangkan dan melakukan Load testing menggunakan locust.io untuk memastikan bahwa sistem atau aplikasi dapat menangani jumlah pengguna dan transaksi yang diharapkan tanpa mengalami kegagalan atau penurunan kinerja yang signifikan.



Gambar 3.22 Grafik RPS

Berdasarkan Gambar 3.22 hasil uji coba load testing, terlihat bahwa jumlah "Requests per Second" (RPS) mencapai stabilitas setelah lonjakan awal, dengan rata-rata sekitar 20-25 RPS. Meskipun demikian, grafik juga menunjukkan adanya tingkat kegagalan permintaan yang relatif konsisten sepanjang pengujian. Lonjakan awal dari 0 hingga sekitar 20 RPS menunjukkan bahwa sistem mampu menangani peningkatan beban dengan cepat. Setelah itu, baik RPS maupun tingkat kegagalan tetap stabil dengan hanya sedikit fluktuasi. Namun, jumlah kegagalan yang mendekati jumlah permintaan per detik mengindikasikan adanya proporsi kegagalan yang cukup signifikan, yang perlu ditinjau lebih lanjut untuk menemukan penyebabnya. Secara keseluruhan, hasil ini menunjukkan sistem dapat menangani beban dengan stabil, namun memerlukan optimalisasi pada aspek kegagalan permintaan.. Berikut adalah hasil Grafik RPS Bisa dilihat di table 3.1 *Request Statistics*.

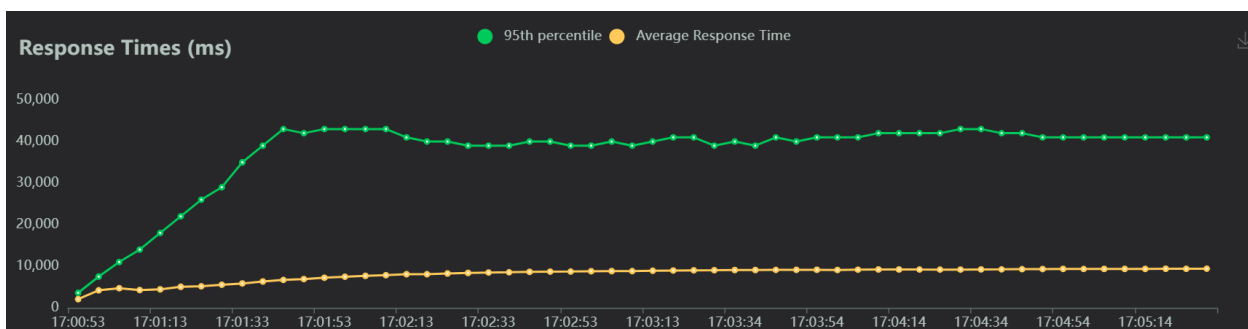
Type	Name	# Requests	# Fails	Median (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/about	545	433	2000	8706.51	266	43888	2071.69	2.8	2.1
GET	/booking/index	485	467	2008.86	9480.86	2009	82387	506.86	2.2	2.2
GET	/booking/list	585	562	2014.86	10530.84	2015	81704	536.94	1.6	1.5
GET	/booking/list/selesai	575	561	2003.69	9191.25	2004	84403	332.52	2.4	2.3
GET	/booking/list/sewa	537	517	2009.14	9792.76	2009	83208	508.64	1.3	1.3
GET	/car/index	589	554	2008.56	10533.25	2009	83011	811.54	2.4	2.2
GET	/car/list	526	508	2007.73	10130.31	2008	81304	467.35	2.1	2.1
GET	/cars	555	454	2000	7986.98	1291	43433	4131.9	1.9	1.5
GET	/dashboard	535	519	2013	9805.21	2013	80787	408.43	2.1	2
GET	/user/index	538	538	2005.44	9415.58	2005	43315	248626.66	2.2	2.2
GET	/user/list	550	550	2005.73	8779.83	2006	43045	225407.17	2.2	2.2
Aggregated	-	6301	5893	2000	9424.52	266	84403	41939.2	23.9	22.1

Tabel 3.1 Request Statistics

Tabel 3.1 *Request Statistics* menampilkan hasil tes beban yang dijalankan pada server menggunakan Locust.io. Tes ini berhasil menjalankan simulasi beban hingga 6301 pengguna bersamaan, tetapi ada beberapa permintaan yang gagal. Berikut yang bisa kita lihat dari

- Beberapa endpoint memiliki jumlah kegagalan yang sangat signifikan, seperti */booking/list* dengan 562 kegagalan, dan */car/index* dengan 554 kegagalan.
- Waktu respons rata-rata untuk sebagian besar endpoint berkisar antara 7700 ms hingga 10500 ms, dengan beberapa permintaan seperti */car/index* dan */booking/list* mencapai waktu respons lebih dari 10 detik. Ini menunjukkan adanya keterlambatan besar dalam menangani permintaan.
- Ukuran rata-rata respons bervariasi, dengan permintaan seperti */cars* menghasilkan ukuran respons yang sangat besar (4131.9 byte) dibandingkan dengan endpoint lainnya. Namun, */user/index* dan */user/list* juga menunjukkan ukuran data yang besar (248626.66 byte dan 225407.17 byte), yang mungkin berkontribusi pada lambatnya respons.

Secara keseluruhan, hasil tes beban menunjukkan bahwa sistem memiliki masalah performa yang signifikan, terutama dalam hal tingginya tingkat kegagalan permintaan dan lambatnya waktu respons pada sebagian besar endpoint. Dibutuhkan optimasi pada server atau aplikasi untuk menangani beban yang lebih baik serta mengurangi waktu respons dan tingkat kegagalan.



Gambar 3.23 Grafik Response Times

Berdasarkan Gambar 3.23 Grafik Response Times menunjukkan dua metrik utama, yaitu 95th Percentile Response Time dan Average Response Time. Pada garis 95th percentile (hijau), terjadi peningkatan signifikan dari sekitar 10.000 ms pada pukul 17:00:53 hingga mencapai puncak di sekitar 40.000 ms pada pukul 17:02:33. Setelah itu, waktu respons stabil di angka yang cukup tinggi hingga akhir grafik pada 17:05:14. Sementara itu, garis Average Response Time (kuning) relatif stabil di sekitar 10.000 ms sepanjang waktu, tanpa menunjukkan fluktuasi besar. Berikut adalah hasil Grafik Response Times yang dilihat di table 3.2 *Response Time Statistics*

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	/about	2000	2000	2000	9400	37000	39000	42000	44000
GET	/booking/index	2000	2000	2000	2100	39000	43000	74000	82000
GET	/booking/list	2000	2000	2000	27000	39000	42000	75000	82000
GET	/booking/list/selesai	2000	2000	2000	2100	39000	41000	75000	84000
GET	/booking/list/sewa	2000	2000	2000	2100	38000	42000	72000	83000
GET	/car/index	2000	2000	2000	2100	40000	66000	79000	83000
GET	/car/list	2000	2000	2000	2100	40000	42000	77000	81000
GET	/cars	2000	2000	2000	2100	36000	39000	42000	43000
GET	/dashboard	2000	2000	2000	2100	39000	42000	70000	81000
GET	/user/index	2000	2000	2100	26000	38000	40000	42000	43000
GET	/user/list	2000	2000	2000	7000	38000	40000	42000	43000
Aggregated	-	2000	2000	2000	2100	38000	41000	72000	84000

tabel 3.2 Response Time Statistics

Berdasarkan hasil load testing rental card yang menggunakan Locust seperti pada Tabel 3.2 Response Time Statistics, dapat disimpulkan beberapa hal berikut:

- Untuk semua endpoint, waktu respons pada persentil 50%, 60%, dan 70% cenderung stabil di sekitar 2000 ms. Hal ini menunjukkan bahwa sebagian besar permintaan memiliki waktu respons yang cukup cepat.
- Pada persentil 90% ke atas, terdapat lonjakan waktu respons yang cukup besar. Misalnya, endpoint */booking/index* memiliki waktu respons 43.000 ms di persentil 95%, dan mencapai hingga 82.000 ms di persentil 100%. Lonjakan ini menandakan adanya outlier atau permintaan yang memerlukan waktu jauh lebih lama dari rata-rata untuk diproses.

- Endpoint terkait *booking*, seperti */booking/index*, */booking/list*, dan variasinya (*/selesai*, */sewa*), memiliki waktu respons terlama, terutama pada persentil 95% hingga 100%. Endpoint */booking/list/selesai* mencapai hingga 84.000 ms pada persentil 100%, yang merupakan salah satu yang tertinggi dalam tabel.
- Endpoint */about* dan */cars* memiliki waktu respons yang relatif lebih rendah dibandingkan endpoint lain, dengan waktu 44.000 ms dan 43.000 ms masing-masing di persentil 100%.
- Data agregat menunjukkan pola yang mirip dengan beberapa endpoint lainnya, dengan waktu respons 41.000 ms pada persentil 95% dan hingga 84.000 ms pada persentil 100%. Hal ini mencerminkan gambaran umum bahwa sebagian besar permintaan diproses cepat, namun ada permintaan yang memerlukan waktu jauh lebih lama.

Secara keseluruhan, meskipun sebagian besar permintaan di semua endpoint diproses dalam waktu yang wajar (2000 ms untuk 50% hingga 70% permintaan), terdapat outlier yang cukup signifikan di persentil 90% ke atas, terutama untuk endpoint yang berkaitan dengan *booking*. Ini bisa menjadi indikasi masalah performa atau beban yang harus diperhatikan pada beberapa endpoint tertentu.

IV. KESIMPULAN

Penelitian ini berhasil mengembangkan dan mengimplementasikan Perancangan Aplikasi Rental Mobil Berbasis website menggunakan metode Agile Scrum dengan framework Laravel. Aplikasi ini bertujuan untuk meningkatkan efisiensi dan kemudahan dalam pengelolaan layanan penyewaan mobil. Hasil penelitian menunjukkan bahwa aplikasi berhasil meningkatkan efisiensi operasional dengan mempercepat proses pemesanan, pengelolaan kendaraan, penambahan sanksi bagi pelanggan yang melanggar aturan, atau fitur yang memungkinkan satu pengguna untuk menyewa lebih dari satu kendaraan sekaligus. Selain itu, transparansi informasi terkait ketersediaan mobil dan riwayat penyewaan juga meningkat melalui antarmuka yang ramah pengguna yang memudahkan akses bagi admin dan pengguna. Hasil load testing menunjukkan bahwa aplikasi mampu menangani sejumlah besar permintaan, meskipun terdapat beberapa area yang memerlukan optimalisasi, seperti kecepatan respon dan tingkat kegagalan pada beberapa endpoint. Secara keseluruhan, aplikasi ini memberikan kontribusi yang signifikan dalam meningkatkan pengalaman pengguna dan efisiensi operasional dalam bisnis rental mobil melalui pengelolaan yang lebih cepat, transparan, dan terorganisir.

UCAPAN TERIMA KASIH

Semoga artikel ini dapat memberikan wawasan dan pemahaman yang bermanfaat bagi para pembaca. Terima kasih kepada semua pihak yang telah berperan serta dalam proses penulisan dan penyusunan artikel ini. Bantuan, dukungan, serta kontribusi yang diberikan selama tahap penelitian hingga penulisan sangatlah berharga. Tanpa keterlibatan mereka, artikel ini tidak akan dapat terselesaikan dengan baik..

REFERENSI

- [1] A. A. F. Amarta and I. G. Anugrah, "Implementasi Agile Scrum Dengan Menggunakan Trello Sebagai Manajemen Proyek Di PT Andromedia," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 4, no. 6, pp. 528–534, 2021, doi: 10.32672/jnkti.v4i6.3702.
- [2] N. Yunita and R. Rosmawati, "Sistem Informasi Rental Mobil Berbasis Web Pada PT Karya Mobil," *Simpatik J. Sist. Inf. dan Inform.*, vol. 1, no. 1, pp. 53–62, 2021, doi: 10.31294/simpatik.v1i1.410.
- [3] W. A. Prabowo and C. Wiguna, "Sistem Informasi UMKM Bengkel Berbasis Web Menggunakan Metode SCRUM," *J. Media Inform. Budidarma*, vol. 5, no. 1, p. 149, 2021, doi: 10.30865/mib.v5i1.2604.
- [4] A. Andipradana and K. Dwi Hartomo, "Rancang Bangun Aplikasi Penjualan Online Berbasis Web Menggunakan Metode Scrum," *J. Algoritm.*, vol. 18, no. 1, pp. 161–172, 2021, doi: 10.33364/algoritma/v.18-1.869.
- [5] N. L. Mufidah and M. S. Mauluddin, "Sistem Penyewaan Mobil Berbasis Web (Studi Kasus Sastro Rent Car)," *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 3, no. 2, p. 131, 2021, doi: 10.36499/jinrpl.v3i2.4606.
- [6] A. P. Aji and E. Supriyanto, "Model Sistem Informasi Penyewaan Mobil Wibi Rent Car Berbasis Web Mobile," *Model Sist. Inf. Penyewaan Mob. Wibi Rent Car Berbas. Web Mob.*, vol. 12, no. 1, pp. 11–20, 2023.
- [7] Y. M. Kristania, "Sistem Informasi Rental Mobil (Si Robi) Berbasis Web Pada Sewa Mobil Sahabat Purwokerto," *Indones. J. Softw. Eng.*, vol. 8, no. 2, pp. 131–137, 2022, [Online]. Available: <https://ejournal.bsi.ac.id/ejournal/index.php/ijse/article/view/12989>
- [8] Y. Fitriasia and R. Oktari Sakti, "Rancang Bangun Sistem Informasi Rental Mobil Bebasis Web (Studi Kasus: Zelta Rent Car)," *J. Komput. Terap.*, vol. 8, no. 1, pp. 12–23, 2022, doi: 10.35143/jkt.v8i1.5275.
- [9] M. H. Romadhon, Y. Yudhistira, and M. Mukrodin, "Sistem Informasi Rental Mobil Berbasis Android Dan Website Menggunakan Framework Codeigniter 3 Studi Kasus : CV Kopja Mandiri," *J. Sist. Inf. dan Teknol. Perad.*, vol. 2, no. 1, pp. 30–36, 2021.

- [10] D. Humaedi, A. Pratama Putra, and A. Saifudin, "Pengembangan Aplikasi Rental Mobil Berbasis Web Menggunakan Model Agile di CV. Endang Jaya," *J. Inform. Univ. Pamulang*, vol. 6, no. 3, pp. 2622–4615, 2021, [Online]. Available: <http://openjournal.unpam.ac.id/index.php/informatika>
- [11] P. Agus *et al.*, "Penyewaan Kendaraan (Sewadisini . Com)," *J. Teknol. Inf. dan Komput.*, vol. 6, no. 1, pp. 66–77, 2020, [Online]. Available: <https://www.jurnal.undhirabali.ac.id/index.php/jutik/article/view/1005>
- [12] C. Kesuma, I. Soleh Marifati, and P. Studi Teknologi Komputer Kabupaten Banyumas, "Rancang Bangun Sistem Informasi Penyewaan Mobil Berbasis Web," *J. ICT Inf. Commun. Technol.*, vol. 23, no. 1, pp. 124–128, 2023.
- [13] E. S. Eriana, "Implementasi Scrum Pada Framework Sistem Informasi Penjualan Dan Pembelian Cat Pada Toko Cad," *Sainstech J. Penelit. Dan Pengkaj. Sains Dan Teknol.*, vol. 32, no. 1, pp. 49–55, 2022, doi: 10.37277/stch.v32i1.1246.
- [14] G. Indah and D. Jollyta, "Penerapan Metode Agile dalam Pengembangan Sistem Informasi Rental Mobil Berbasis Android," *J. Mhs. Apl. Teknol. Komput. dan Inf.*, vol. 4, no. 3, pp. 113–122, 2022.
- [15] A. Susanto, "Penerapan Agile Development Methods Pada Perancangan Sistem Inforamsi Rental Mobil Barokah," *JUSTIFY J. Sist. Inf. Ibrahimy*, vol. 1, no. 2, pp. 126–132, 2023.
- [16] N. Haqqizar, T. W. Widyaningsih, and M. A. Dewi, "Agile Scrum Model for Development of e-Customer Relationship Management to Support Warehouse Rental Services," *J. SISKOM-KB (Sistem Komput. dan Kecerdasan Buatan)*, vol. 6, no. 2, pp. 118–124, 2023, doi: 10.47970/siskom-kb.v6i2.374.

Conflict of Interest Statement:

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.