

**PENGEMBANGAN SISTEM INFORMASI MANAJEMEN
TRANSPORTASI BERBASIS WEB DI PT SURVEYOR
INDONESIA MENGGUNAKAN *FRAMEWORK LARAVEL***

TUGAS AKHIR

Diajukan Oleh:

**TEUKU FASHA MIRAZA
NIM. 190705016
Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi**



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI AR-RANIRY
BANDA ACEH
2023 M /1444 H**

LEMBAR PERSETUJUAN

PENGEMBANGAN SISTEM INFORMASI MANAJEMEN TRANSPORTASI BERBASIS WEB DI PT SURVEYOR INDONESIA MENGGUNAKAN *FRAMEWORK LARAVEL*

TUGAS AKHIR

Diajukan kepada Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh
Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana (S1)
Dalam Ilmu Teknologi Informasi

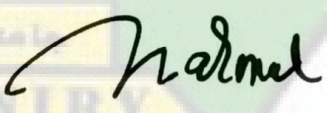
Oleh:
TEUKU FASHA MIRAZA
NIM. 190705016
Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi

Disetujui Untuk Dimunaqasyahkan Oleh:

Pembimbing I,


Mulkan Fadhli, S.T., M.T.
NIP.198811282020121006

Pembimbing II,


Nazaruddin Ahmad, M.T.
NIP.198206052014031002

Mengetahui,
Ketua Program Studi Teknologi Informasi


Ima Dwitawati, MBA
NIP. 198210132014032002

LEMBAR PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI MANAJEMEN TRANSPORTASI BERBASIS WEB DI PT SURVEYOR INDONESIA MENGGUNAKAN *FRAMEWORK LARAVEL*

TUGAS AKHIR

Telah Diuji Oleh Panitia Ujian Munaqasah Tugas Akhir
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan Lulus
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S-1)
Dalam Ilmu Teknologi Informasi

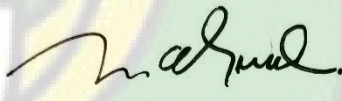
Pada Hari/Tanggal: Selasa, 21 November 2023
7 Jumadil Awal 1445 H
di Darussalam, Banda Aceh

Pantia Ujian Munaqasah Tugas Akhir

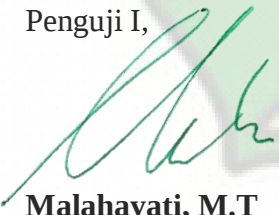
Ketua,


Mulkan Fadhli, S.T., M.T.
NIP. 198811282020121006

Sekretaris,


Nazaruddin Ahmad, M.T.
NIP. 198206052014031002

Penguji I,


Malahayati, M.T
NIP. 198301272015032003

Penguji II,


Khairan AR, M.Kom
NIP. 198607042014031001

Mengetahui
Dekan Fakultas Sains dan Teknologi
UIN Ar-Raniry Banda Aceh,


Dr. Ir. Muhammad Dirhamsyah, M.T., IPU
NIDN. 0002106203

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Teuku Fasha Miraza
NIM : 190705016
Program Studi : Teknologi Informasi
Fakultas : Sains dan Teknologi
Judul : Pengembangan Sistem Informasi Manajemen Transportasi
Berbasis Web Di PT Surveyor Indonesia Menggunakan
Framework Laravel

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah karya orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu bertanggungjawab atas karya ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 21 November 2023

Yang Menyatakan,



(Teuku Fasha Miraza)

ABSTRAK

Nama : Teuku Fasha Miraza
NIM : 190705016
Program Studi : Teknologi Informasi
Fakultas : Sains dan Teknologi
Judul : Pengembangan Sistem Informasi Manajemen Transportasi
Berbasis Web Di PT Surveyor Indonesia Menggunakan
Framework Laravel
Tanggal Sidang : 21 November 2023
Jumlah Halaman : 172 Halaman
Pembimbing I : Mulkan Fadhli, M.T.
Pembimbing II : Nazaruddin Ahmad, M.T.

Dalam era globalisasi dan transformasi digital yang sedang berkembang pesat, perusahaan-perusahaan kini semakin mengakui pentingnya pemanfaatan teknologi informasi untuk meningkatkan efisiensi operasional. Saat ini, PT Surveyor Indonesia masih menghadapi sejumlah tantangan terkait manajemen transportasinya, terutama terkait pencatatan dan pengelolaan mobil operasional kantor. Proses pencatatan yang masih dilakukan secara manual menimbulkan berbagai masalah, mulai dari ketidak-efisienan dalam sistem, kesulitan verifikasi terkait kebutuhan kendaraan operasional, hingga kurangnya ketelitian dalam mencatat peminjaman oleh karyawan.

Pendekatan teknologi yang akan digunakan dalam pengembangan ini adalah *framework Laravel*, sebuah alat pengembangan berbasis PHP yang telah teruji efektif dalam menciptakan aplikasi web yang canggih dan efisien. Sistem informasi manajemen transportasi (SIMTRANS) berbasis web menggunakan *framework Laravel* telah berhasil dikembangkan. Dalam aplikasi tersebut sudah dikembangkan fitur-fitur menunya yaitu menu keuangan, vendor, tanda terima kendaraan, merk kendaraan, jenis kendaraan, data kendaraan, permohonan mobil, *service*, driver, dan proyek yang berhubungan dengan permohonan mobil. Aplikasi SIMTRANS ini sudah berhasil dibuat

menggunakan *framework Laravel* pada PT Surveyor Indonesia, yang menunjukkan kemajuan dalam efisiensi pengelolaan data transportasi di perusahaan tersebut. Dengan penerapan aplikasi ini, PT Surveyor Indonesia dapat memperoleh manfaat dalam pengelolaan data transportasi, termasuk peningkatan efisiensi dan keterpaduan informasi terkait transportasi perusahaannya.

Kata kunci: PT Surveyor Indonesia, SIMTRANS, framework Laravel, efisiensi



ABSTRACT

Name : Teuku Fasha Miraza
NIM : 190705016
Study Program : Information Technology
Faculty : Science and Technology
Title : Development of Web-Based Transportation Management
Information System at PT Surveyor Indonesia Using
Laravel Framework
Date : 21 November 2023
Number of Pages : 172 Pages
Supervisor I : Mulkan Fadhli, M.T.
Supervisor II : Nazaruddin Ahmad, M.T.

In the rapidly developing era of globalization and digital transformation, companies are now increasingly recognizing the importance of utilizing information technology to increase operational efficiency. Currently, PT Surveyor Indonesia is still facing a number of challenges related to its transportation management, especially related to recording and managing office operational cars. The recording process which is still carried out manually causes various problems, ranging from inefficiencies in the system, difficulties in verifying operational vehicle requirements, to a lack of accuracy in recording loans by employees.

The technological approach that will be used in this development is the Laravel framework, a PHP-based development tool that has been proven effective in creating sophisticated and efficient web applications. A web-based transportation management information system (SIMTRANS) using the Laravel framework has been successfully developed. In this application, menu features have been developed, namely financial menus, vendors, vehicle receipts, vehicle brands, vehicle types, vehicle data, car requests, services, drivers, and projects related to car requests. This SIMTRANS application has been successfully created using the Laravel framework at PT Surveyor Indonesia, which shows progress in the

efficiency of managing transportation data at the company. By implementing this application, PT Surveyor Indonesia can gain benefits in managing transportation data, including increasing efficiency and integration of information related to the company's transportation.

Keywords: PT Surveyor Indonesia, SIMTRANS, Laravel framework, efficiency



KATA PENGANTAR

Puji syukur penulis panjatkan atas kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya, penulis dapat menyelesaikan proposal tugas akhir yang berjudul **“Pengembangan Sistem Informasi Manajemen Transportasi Berbasis Web Di PT Surveyor Indonesia Menggunakan Framework Laravel”**. Shalawat beserta salam kita sampaikan kepada Rasulullah SAW beserta keluarga dan sahabat beliau sekalian yang telah memperjuangkan umat Islam kepada jalan kebenaran dengan dibekali ilmu yang bermanfaat untuk dunia dan akhirat.

Penulisan proposal tugas akhir ini diajukan sebagai salah satu syarat untuk menyelesaikan skripsi pada program studi Teknologi Informasi, Fakultas Sains dan Teknologi, UIN Ar-Raniry. Dalam penulisan proposal tugas akhir ini, penulis dengan segala kerendahan hati ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Kedua orangtua dan keluarga tercinta yang selalu memberikan doa dan dukungan.
2. Bapak Dr. Ir. Muhammad Dirhamsyah, M.T., IPU. Selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Ar-Raniry Banda Aceh.
3. Ketua Prodi Teknologi Informasi, Ibu Ima Dwitawati, M.B.A yang telah memberikan bimbingan dan arahan dalam penyusunan proposal tugas akhir ini.
4. Dosen Pembimbing Proposal Tugas Akhir, Bapak Mulkan Fadhli, M.T yang senantiasa memberikan arahan dan bimbingan dalam penyusunan proposal tugas akhir ini.
5. Dosen Pembimbing I, Bapak Mulkan Fadhli, M.T dan Dosen Pembimbing II, Bapak Nazaruddin Ahmad, M.T yang senantiasa memberikan motivasi dalam penyelesaian Tugas Akhir ini.
6. Bapak dan Ibu dosen Program Studi Teknologi Informasi yang telah membekali penulis dengan ilmu pengetahuan di bidang Teknologi Informasi.

7. *Staff* Prodi Teknologi Informasi, Ibu Cut Ida Rahmadiana, S.Si yang telah membantu penulis dalam pemberkasan administrasi.
8. Sahabat dan teman-teman yang selalu memberikan dorongan dan semangat kepada penulis dalam menyelesaikan penyusunan proposal tugas akhir ini.
9. Pihak-pihak terkait lainnya yang membantu penulis dalam penyusunan proposal ini.

Banda Aceh, 5 Mei 2022

Penulis,

Teuku Fasha Miraza



DAFTAR ISI

LEMBAR PERSETUJUAN	i
LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN KEASLIAN	iii
ABSTRAK	iv
ABSTRACT	vi
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xv
Bab I Pendahuluan	1
I.1 Latar Belakang	1
I.2 Rumusan Masalah	2
I.3 Tujuan Penelitian	2
I.4 Manfaat Penelitian	3
I.5 Sistematika Penulisan	3
Bab II Tinjauan Pustaka	5
II.1 Penelitian Terdahulu	5
II.2 Kajian Teoritis	7
II.2.1 Sistem Informasi	7
II.2.2 <i>Laravel</i>	7
II.2.3 Sistem Informasi Manajemen	7
II.2.4 Transportasi	8
II.2.5 PHP	8
II.2.6 Visual Studio Code	9
II.2.7 Manajemen <i>User</i>	9
II.2.8 Laporan Keuangan	9
II.2.9 Vendor	9
II.2.10 <i>Website</i>	10
II.2.11 <i>Database</i>	10
II.2.12 Metode <i>Waterfall</i>	10
Bab III Metode Penelitian	13
III.1 Tahapan Penelitian	13
III.2 Metode Pengumpulan data	14
III.2.1 Observasi	14
III.2.2 Wawancara	14
III.3 Metode Pengembangan Aplikasi	15
III.3.1 Analisis Kebutuhan	15
III.3.2 Desain Sistem	16
III.3.3 Implementasi	85
III.3.4 Pengujian Sistem	86
III.4 Tempat Dan Waktu Penelitian	86
Bab IV Hasil dan Pembahasan	87
IV.1 Implementasi	87
IV.1.2. Hasil Metode Wawancara	87
IV.1.1 Fitur <i>Authentication</i>	89

Bab V Kesimpulan dan Saran.....	127
V.1 Kesimpulan.....	127
V.2 Saran	127
DAFTAR PUSTAKA	129
LAMPIRAN	131
RIWAYAT HIDUP	157



DAFTAR GAMBAR

Gambar II.2.14	Metode <i>Waterfall</i>	11
Gambar III.1	Diagram Alur Tahapan Penelitian	13
Gambar III.2	<i>Use Case Diagram</i> User	19
Gambar III.3	<i>Use Case Diagram</i> Approval	20
Gambar III.4	<i>Use Case Diagram</i> Admin	21
Gambar III.5	<i>Use Case Diagram</i> Vendor	23
Gambar III.6	<i>Activity Diagram</i> User Login	24
Gambar III.7	<i>Activity Diagram</i> User Register	25
Gambar III.8	<i>Activity Diagram</i> User Permohonan Mobil	26
Gambar III.9	<i>Activity Diagram</i> User Tanda Terima Kendaraan	27
Gambar III.10	<i>Activity Diagram</i> Approval Login	28
Gambar III.11	<i>Activity Diagram</i> Approval Register	29
Gambar III.12	<i>Activity Diagram</i> Approval Permohonan Mobil	30
Gambar III.13	<i>Activity Diagram</i> Approval Keuangan	31
Gambar III.14	<i>Activity Diagram</i> Admin Login	32
Gambar III.15	<i>Activity Diagram</i> Admin Register	33
Gambar III.16	<i>Activity Diagram</i> Admin Manajemen User	34
Gambar III.17	<i>Activity Diagram</i> Admin Merk Kendaraan	35
Gambar III.18	<i>Activity Diagram</i> Admin Jenis Kendaraan	36
Gambar III.19	<i>Activity Diagram</i> Admin Kendaraan	37
Gambar III.20	<i>Activity Diagram</i> Admin Service	38
Gambar III.21	<i>Activity Diagram</i> Admin Proyek	39
Gambar III.22	<i>Activity Diagram</i> Admin Tanda Terima Kendaraan	40
Gambar III.23	<i>Activity Diagram</i> Admin Driver	41
Gambar III.24	<i>Activity Diagram</i> Admin Vendor	42
Gambar III.25	<i>Activity Diagram</i> Admin Keuangan.....	43
Gambar III.26	<i>Activity Diagram</i> Vendor Login	44
Gambar III.27	<i>Activity Diagram</i> Vendor Register.....	45
Gambar III.28	<i>Activity Diagram</i> Vendor Kendaraan.....	46
Gambar III.29	<i>Activity Diagram</i> Vendor Keuangan.....	47
Gambar III.30	<i>Activity Diagram</i> Vendor Service	48
Gambar III.31	<i>Activity Diagram</i> Vendor Driver	49
Gambar III.32	<i>Activity Diagram</i> Vendor Tanda Terima Kendaraan	50
Gambar III.33	<i>Sequence Diagram</i> Login	51
Gambar III.34	<i>Sequence Diagram</i> Registrasi	52
Gambar III.35	<i>Sequence Diagram</i> Permohonan Mobil	53
Gambar III.36	<i>Sequence Diagram</i> Tanda Terima Kendaraan	54
Gambar III.37	<i>Sequence Diagram</i> Merk Kendaraan	55
Gambar III.38	<i>Sequence Diagram</i> Jenis Kendaraan	56
Gambar III.39	<i>Sequence Diagram</i> Kendaraan	57
Gambar III.40	<i>Sequence Diagram</i> Service	58
Gambar III.41	<i>Sequence Diagram</i> Proyek	59
Gambar III.42	<i>Sequence Diagram</i> Vendor	60
Gambar III.43	<i>Sequence Diagram</i> Keuangan	61

Gambar III.44	<i>Sequence Diagram</i> Manajemen User	62
Gambar III.45	Diagram Database	71
Gambar III.46	Tampilan Login	72
Gambar III.47	Tampilan Register	73
Gambar III.48	Tampilan Dashboard	74
Gambar III.49	Tampilan Manajemen User	75
Gambar III.50	Tampilan Merk Kendaraan	76
Gambar III.51	Tampilan Jenis Kendaraan	77
Gambar III.52	Tampilan Data Kendaraan	78
Gambar III.53	Tampilan Service	79
Gambar III.54	Tampilan Driver	80
Gambar III.55	Tampilan Vendor	81
Gambar III.56	Tampilan Keuangan	82
Gambar III.57	Tampilan Proyek	83
Gambar III.58	Tampilan Permohonan Mobil User	84
Gambar IV.1	Create Account laravel	90
Gambar IV.2	Hasil Halaman Login	91
Gambar IV.3	Hasil Halaman Register	91
Gambar IV.4	Hasil Menu Merk Kendaraan	92
Gambar IV.5	Algoritma dari fitur merk kendaraan	93
Gambar IV.6	Hasil Menu Jenis Kendaraan	94
Gambar IV.7	Algoritma dari fitur jenis kendaraan	96
Gambar IV.8	Hasil Menu Kendaraan	97
Gambar IV.9	Algoritma dari fitur kendaraan	99
Gambar IV.10	Hasil Menu Service Kendaraan	100
Gambar IV.11	Algoritma dari fitur service	102
Gambar IV.12	Hasil Menu Manajemen User	102
Gambar IV.13	Algoritma dari fitur manajemen user	104
Gambar IV.14	Hasil Menu Vendor	105
Gambar IV.15	Algoritma dari fitur vendor	107
Gambar IV.16	Hasil Menu Driver	108
Gambar IV.17	Algoritma dari fitur driver	109
Gambar IV.18	Hasil Menu Permohonan Mobil (user)	110
Gambar IV.19	Hasil Tambah Data Permohonan Mobil (user)	110
Gambar IV.20	Hasil Menu Permohonan Mobil (approval)	111
Gambar IV.21	Hasil Edit Permohonan Mobil (approval)	112
Gambar IV.22	Hasil Menu Permohonan Mobil (admin)	113
Gambar IV.23	Hasil Edit Permohonan Mobil (admin)	113
Gambar IV.24	Hasil Menu Permohonan Mobil (vendor)	114
Gambar IV.25	Hasil Edit Permohonan Mobil (vendor)	114
Gambar IV.26	Algoritma dari fitur permohonan mobil	116
Gambar IV.27	Hasil Menu Tanda Terima Kendaraan (admin)	117
Gambar IV.28	Hasil Tambah Tanda Terima Kendaraan (admin)	117
Gambar IV.29	Hasil Menu Tanda Terima Kendaraan (user)	118
Gambar IV.30	Hasil Menu Tanda Terima Kendaraan (vendor)	118
Gambar IV.31	Algoritma dari fitur tanda terima kendaraan	120
Gambar IV.32	Hasil Menu Proyek	121
Gambar IV.33	Algoritma dari fitur proyek	122

Gambar IV.34	Hasil Menu Keuangan (vendor)	123
Gambar IV.35	Hasil Menu Keuangan (<i>approval</i>)	124
Gambar IV.36	Hasil Menu Keuangan (vendor)	124
Gambar IV.37	Algoritma dari fitur keuangan	126



DAFTAR TABEL

Tabel III.2.2.1 Pertanyaan wawancara	14
Tabel III.3.2.1 Identifikasi <i>actor use case diagram</i>	17
Tabel IV.1.2.1 Hasil wawancara.....	87



BAB I

PENDAHULUAN

I.1 Latar Belakang

Dalam era globalisasi dan transformasi digital yang sedang berkembang pesat, perusahaan-perusahaan kini semakin mengakui pentingnya pemanfaatan teknologi informasi untuk meningkatkan efisiensi operasional. PT Surveyor Indonesia, sebagai perusahaan yang bergerak di bidang inspeksi dan pengujian, tidak terkecuali dari dinamika perkembangan ini. Salah satu aspek sangat penting dalam operasional suatu perusahaan adalah manajemen transportasi yang efektif dan efisien.

Pentingnya manajemen transportasi tidak hanya terkait dengan kelancaran logistik perusahaan, tetapi juga berdampak signifikan pada efisiensi waktu, biaya, dan kualitas layanan. Saat ini, PT Surveyor Indonesia masih menghadapi sejumlah tantangan terkait manajemen transportasinya, terutama terkait pencatatan dan pengelolaan mobil operasional kantor. Proses pencatatan yang masih dilakukan secara manual menimbulkan berbagai masalah, mulai dari ketidak-efisienan dalam sistem, kesulitan verifikasi terkait kebutuhan kendaraan operasional, hingga kurangnya ketelitian dalam mencatat peminjaman oleh karyawan.

Penelitian yang pernah dilakukan oleh Afril Caesar Muhammad Hanif pada tahun 2021 berjudul “Pembangunan Aplikasi Sistem Informasi Operasional Bus Berbasis Web Menggunakan *Framework Laravel*”. Masalah yang dibahas dalam penelitian ini adalah banyak operator bus yang kesulitan dalam memantau kondisi armadanya selama beroperasi, yang dapat mengakibatkan kerusakan bus yang tidak terdeteksi. (Hanif & Pakereng, 2021).

Dalam penelitian ini, metode penelitian yang digunakan adalah *Waterfall* dan menggunakan dalam mengembangkan sistem. Hasil dari penelitian ini adalah sebuah sistem informasi yang memudahkan operator bus untuk memantau kelayakan operasional armada bus.

Oleh karena itu, penelitian ini untuk mengembangkan Sistem Informasi Manajemen Transportasi berbasis web di PT Surveyor Indonesia. Pendekatan teknologi yang akan digunakan dalam pengembangan ini adalah *framework Laravel*, sebuah alat pengembangan berbasis PHP yang telah teruji efektif dalam menciptakan aplikasi web yang canggih dan efisien. *Framework* ini tidak hanya menawarkan keunggulan dalam aspek teknis, seperti *routing*, *templating*, dan integrasi *database*, tetapi juga memberikan fondasi kuat untuk pengembangan aplikasi berkelas dunia.

Pengembangan sistem informasi ini diharapkan mampu mengatasi tantangan yang dihadapi PT Surveyor Indonesia dalam manajemen transportasinya. Selain itu, penerapan teknologi ini diharapkan dapat memberikan kontribusi dalam peningkatan efisiensi operasional, pengurangan ketidak-efisienan pencatatan manual, dan peningkatan layanan kepada klien, pemerintah, dan mitra perusahaan. Dengan demikian, penelitian ini diarahkan untuk menjadi sebuah langkah strategis dan berkontribusi positif terhadap pertumbuhan dan kemajuan PT Surveyor Indonesia dalam menghadapi tuntutan era teknologi informasi saat ini.

I.2 Rumusan Masalah

Berdasarkan latar belakang yang telah penulis uraikan, maka penulis merumuskan masalah yaitu:

1. Bagaimana cara mengembangkan aplikasi sistem informasi manajemen transportasi berbasis web pada PT Surveyor Indonesia menggunakan *framework Laravel*?
2. Apakah aplikasi SIMTRANS dapat meningkatkan efisiensi pengelolaan data transportasi di PT Surveyor Indonesia?

1.3 Tujuan Penelitian

Berdasarkan perumusan masalah di atas maka tujuan penelitian ini adalah sebagai berikut:

1. Merancang dan mengembangkan aplikasi sistem informasi manajemen transportasi berbasis web pada PT Surveyor Indonesia menggunakan *framework Laravel*.
2. Menganalisis dan mengevaluasi sejauh mana aplikasi SIMTRANS dapat meningkatkan efisiensi pengelolaan data transportasi di PT Surveyor Indonesia.

1.4 Manfaat Penelitian

Berdasarkan latar belakang, rumusan masalah, tujuan penelitian, batasan penelitian yang telah penulis uraikan, maka manfaat penelitian ini, yaitu:

1. Manfaat bagi si penulis : penulis dapat mengembangkan keterampilan teknis dalam penggunaan *framework Laravel*, pemodelan basis data, dan pengembangan aplikasi web.
2. Manfaat bagi insitusi : penelitian memberikan kontribusi pada bidang akademis melalui pengembangan pengetahuan dan penerapan teknologi terkini.
3. Manfaat bagi instasi : sistem informasi yang dikembangkan dapat membantu dalam peningkatan efisiensi manajemen transportasi di PT Surveyor Indonesia.

1.5 Sistematika Penulisan

Sistematika penulisan dalam penulisan skripsi terdiri dari 5 bab. Bab-bab ini akan dijabarkan secara singkat sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini, akan diuraikan tentang latar belakang, identifikasi masalah, rumusan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini, akan dijelaskan mengenai teori-teori yang berkaitan dengan topik penelitian, baik yang berkaitan dengan teori pengembangan sistem informasi, manajemen transportasi, maupun penggunaan *framework Laravel* dalam pengembangan aplikasi.

BAB III METODE PENELITIAN

Pada bab ini, akan dijelaskan tentang tahapan-tahapan yang akan dilakukan dalam penelitian. Setiap tahapan-tahapan penelitian tersebut akan dijelaskan secara rinci, dan akan dijelaskan juga tentang metode pengumpulan data metode pengembangan aplikasi.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini, akan diuraikan tentang hasil penelitian yang telah dilakukan, analisis data, serta pembahasan terkait temuan dan hasil penelitian. Selain itu, juga dijelaskan mengenai faktor-faktor yang mempengaruhi pengembangan sistem informasi manajemen transportasi pada PT Surveyor Indonesia.

BAB V KESIMPULAN DAN SARAN

Pada bab ini, akan diuraikan kesimpulan dari hasil penelitian yang telah dilakukan. Saran-saran yang diharapkan dapat digunakan pengembangan sistem informasi manajemen transportasi pada PT Surveyor Indonesia selanjutnya.

BAB II

TINJAUAN PUSTAKA

II.1 Penelitian Terdahulu

Beberapa penelitian terdahulu yang terkait dengan penelitian ini meliputi penelitian yang dilakukan oleh M. Yasmin Ramadhan M pada tahun 2018 dengan judul "Aplikasi Informasi Transportasi Angkutan Umum Antar Kota Pada Terminal Amplas Sumatra Utara". Penelitian tersebut membahas masalah yang dihadapi oleh banyak calon penumpang angkutan umum, yaitu ketidakpahaman dalam menanyakan arah dan tujuan angkutan kota, jam keberangkatan, jumlah mobil yang berangkat, dan ketersediaan bangku tempat duduk.

Tujuannya adalah untuk memberikan kemudahan bagi calon penumpang dalam melihat jadwal keberangkatan dan tujuan bis serta memilih kelas yang diinginkan secara lebih aman. Oleh karena itu, solusi yang disarankan adalah membuat aplikasi android yang akan memungkinkan data masuk dan dikonfirmasi melalui telepon ke penjaga loket atau toket bis. Pembayaran transaksi dapat dilakukan saat bis berjalan atau di terminal.

Penelitian ini pernah dilakukan oleh Sari Safitri pada tahun 2021, dengan judul "Perancangan Sistem Penyewaan Bus Berbasis *Laravel Framework* PT Sumber Jaya Trans Tangerang". Terdapat masalah operasional dalam sistem penyewaan bus yang masih menggunakan cara manual. Banyak kendala yang dihadapi dalam penyewaan bus tersebut, seperti proses bisnis yang belum terkomputerisasi dan efisiensi penyimpanan data yang kurang memadai. (Al Fatah et al., 2021).

Dalam penelitian ini, metode analisis sistem yang digunakan adalah *PIECES*, yang mencakup aspek-aspek seperti *Performance*, *Information*, *Economy*, *Control*, *Efficiency*, dan *Service*. Berdasarkan hasil penelitian, sistem penyewaan bus yang dikembangkan dengan *framework Laravel* dapat meningkatkan efektivitas dan efisiensi proses penyewaan bus yang padat.

Penelitian yang berjudul "Manajemen *E-scheduling* Transportasi dengan metode *Is Research*". Masalah yang teridentifikasi dalam penelitian ini adalah

bahwa proses penjadwalan peminjaman kendaraan dinas tidak optimal karena hanya mengandalkan pengarsipan formulir peminjaman kendaraan yang sudah ada. Selain itu, sering terjadi bentroknya jadwal peminjaman kendaraan karena kesalahan dalam menentukan tanggal peminjaman. (Januhari et al., 2019).

Untuk menyelesaikan masalah dalam penelitian ini, diperlukan suatu solusi untuk mempermudah proses penjadwalan peminjaman kendaraan operasional di bagian SARPRAS. Salah satu solusi yang diusulkan adalah dengan membuat sistem penjadwalan peminjaman kendaraan yang terkomputerisasi. Tujuannya adalah untuk mengurangi kesalahan dan meningkatkan efisiensi pengelolaan aset transportasi di SARPRAS. Oleh karena itu, dibangunlah sistem informasi manajemen transportasi berbasis desktop.

Hasil dari penelitian ini adalah sebuah sistem informasi Manajemen *E-Scheduling* Transportasi dengan metode *Is Research* yang berbasis Dekstop. Tujuan dari sistem ini adalah untuk memudahkan petugas di bagian SARPRAS, terutama ITB STIKOM Bali, dalam memberikan informasi dan layanan penjadwalan peminjaman kendaraan operasional kantor.

Penelitian yang pernah dilakukan oleh Ali Yusri pada tahun 2022 dengan judul “Sistem Informasi Lalu Lintas Harian Rata-Rata Berbasis Web Menggunakan *Framework Laravel*”. Masalah pada penelitian adalah ketidakseimbangan pembangunan jalan berbanding terbalik dengan meningkatnya jumlah kendaraan setiap tahun khususnya kendaraan roda dua dan mobil, sehingga berdampak pada kerusakan jalan dan kemacetan lalu lintas.

Hasil dari penelitian ini adalah sistem informasi lalu lintas harian rata-rata (LHR) berbasis web yang dapat digunakan untuk mempermudah penyimpanan data yang diperoleh dari tenaga surveyor dan sebagai media bagi tenaga surveyor guna menambah wawasan terkait LHR, alat apa saja yang digunakan dalam melakukan survey dan lain-lain di Dinas Perhubungan Kabupaten Cirebon.

II.2 Kajian Teoritis

II.2.1 Sistem Informasi

Sistem informasi merujuk pada serangkaian subsistem yang terintegrasi dan saling terhubung, yang meliputi masukan, proses, dan keluaran dalam pengolahan informasi. Data diolah sedemikian rupa sehingga bermanfaat bagi pengguna.

Sebuah sistem informasi dapat dianggap sebagai gabungan dari berbagai komponen yang terkait satu sama lain untuk menghasilkan informasi dalam domain yang sama. Secara keseluruhan, sistem informasi dapat diartikan sebagai suatu sistem yang menggabungkan pengetahuan sistem untuk memproses data dan menghasilkan informasi yang berguna bagi pengambilan keputusan. (Rajendra, 2013).

II.2.2 *Laravel*

Mulyadi menyatakan bahwa *Laravel Framework*, yang merupakan sebuah *framework* pemrograman *open-source*, telah menjadi pilihan utama banyak developer di seluruh dunia. Salah satu alasan mengapa *Laravel* populer dalam beberapa tahun terakhir adalah karena kemudahan penggunaannya dan dokumentasinya yang lengkap.

Laravel menggunakan arsitektur *Model-View-Controller (MVC)*, di mana aplikasi dibagi berdasarkan tugas-tugasnya seperti manajemen data, kontrol, dan *user interface*. Keuntungan dari menggunakan *Laravel* adalah kemudahan dalam pemeliharaan dan skalabilitas aplikasi yang dibuat.

II.2.3 Sistem Informasi Manajemen

Secara umum, sistem informasi manajemen adalah sistem yang memberikan informasi untuk mendukung operasi, manajemen, dan pengambilan keputusan dalam sebuah organisasi. Istilah lain yang digunakan untuk sistem ini meliputi "Sistem Informasi", "Sistem Manajemen Informasi", dan "Sistem Informasi dan Pengambilan Keputusan".

Tujuan dari sistem informasi manajemen adalah untuk mengumpulkan informasi dari berbagai departemen atau organisasi dan mengubahnya menjadi

informasi yang berguna untuk kebutuhan manajemen organisasi, dengan menerapkan prinsip-prinsip sistem.

Prinsip ini digunakan karena informasi yang dihasilkan dalam berbagai bentuk akan dikumpulkan, disimpan, dan diolah oleh tubuh yang memproduksinya, sehingga dapat menjadi pesan yang berguna.

II.2.4 Transportasi

Transportasi merupakan bagian penting dalam kehidupan masyarakat. Istilah "transportasi" berasal dari bahasa Latin, dimana "trans" berarti melewati atau ke seberang dan "*portare*" berarti mengangkut atau mengangkut. Menurut Salim (2000), transportasi adalah kegiatan memindahkan barang atau penumpang dari suatu daerah ke daerah lain. Transportasi dengan demikian dapat diartikan sebagai usaha dan tindakan untuk memindahkan atau mengangkut barang atau penumpang dari suatu daerah ke daerah lain.

II.2.5 PHP

PHP atau *Hypertext Preprocessor* adalah bahasa pemrograman untuk membangun aplikasi berbasis *web*. Seperti halnya aplikasi, situs *web* harus dinamis dan interaktif. Dinamis artinya sebuah *website* dapat menampilkan konten yang berbeda tergantung pada kondisi tertentu, misalnya menampilkan produk yang berbeda kepada setiap pengunjung.

Interaktif berarti bahwa situs *web* dapat memberikan jawaban atau umpan balik kepada pengguna, seperti tampilan hasil pencarian produk. PHP adalah bahasa pemrograman sisi server, sehingga server memprosesnya dan hasilnya dikirim ke browser. Oleh karena itu, sebelum menjalankan program PHP diperlukan server sebagai alat yang harus tersedia. (Perusahaan, 201).

Tim EMS menyadari bahwa salah satu kekuatan PHP adalah kemudahan konektivitas basis datanya. Berbeda dengan bahasa pemrograman Java, PHP secara langsung mendukung banyak basis data tanpa konfigurasi khusus.

Oleh karena itu, PHP sangat fleksibel dalam menangani *database* yang berbeda. MySQL adalah salah satu *database* PHP yang paling umum digunakan. Jika Anda ingin menghubungkan PHP ke *database* , yang perlu Anda ketahui

hanyalah nama dan lokasi *database* , serta nama pengguna dan kata sandi untuk mengakses *database* .

II.2.6 Visual Studio Code

Visual Code Studio (VCS) adalah alat untuk membangun aplikasi berbasis Android. Selain VCS, Android Studio juga digunakan sebagai emulator untuk menguji aplikasi setelah dibuat. Perangkat lunak pendukung lainnya adalah *React Native's Node js* untuk mengembangkan pustaka atau paket. Aplikasi sampul adalah aplikasi *Firebase* untuk API cloud dan layanan. Fitur *Firebase: Database* waktu nyata dan perlindungan data. (Hamzan et al., 2022)

II.2.7 Manajemen User

Manajemen *user* adalah proses pengelolaan yang digunakan untuk mengatur akses pengguna pada sistem dengan melakukan pengaturan pada proses *login*. Tujuan dari manajemen *user* adalah untuk mengelola hak akses pengguna pada objek-objek yang terdapat dalam *database* Oracle.

II.2.8 Laporan Keuangan

Kashmir (2016:7) mendefinisikan laporan keuangan sebagai laporan yang menyajikan keadaan keuangan perusahaan pada saat tertentu (neraca) atau untuk periode tertentu (laporan laba rugi). Kondisi perusahaan saat ini mengacu pada posisi keuangannya pada saat tertentu (neraca) atau selama periode tertentu (laporan laba rugi).

II.2.9 Vendor

Vendor adalah pihak yang bertanggung jawab untuk menyediakan bahan baku atau mentah yang dibutuhkan oleh perusahaan lain. Bahan mentah yang disediakan oleh vendor akan digunakan dalam proses produksi. Ada juga istilah lain seperti *supplier* atau *distributor* yang seringkali digunakan untuk menggantikan istilah *vendor*.

II.2.10 Website

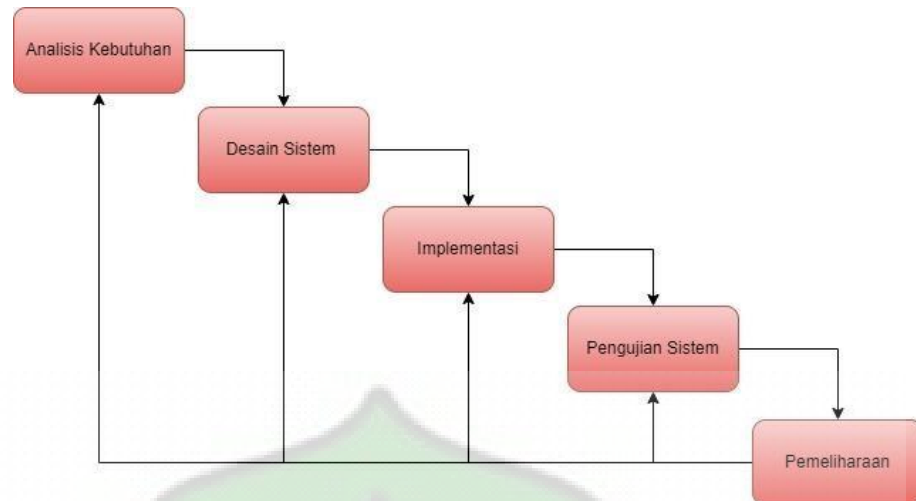
Website merupakan himpunan dari halaman-halaman *web* yang terhubung satu sama lain yang berisikan beragam konten seperti dokumen serta gambar yang tersimpan pada server *web*. Sedangkan aplikasi *web* merupakan jenis aplikasi yang berada pada server *web* dan dapat diakses oleh para pengguna dengan menggunakan browser. (Sebok, Vermat 2018).

II.2.11 Database

Basis data atau *database* adalah kumpulan informasi yang saling terkait dan tersusun secara teratur yang disimpan di dalam memori komputer. Data dalam *database* merepresentasikan fakta-fakta dalam dunia nyata dan sering kali diproses untuk menghasilkan informasi yang berguna. Dalam basis data, terdapat berbagai satuan data yang dikenal, salah satunya adalah file. File merupakan bentuk fisik dari basis data yang terkumpul dan terorganisir dengan baik untuk mempermudah akses dan pengolahan informasi (Edi, 2009).

II.2.12 Metode Waterfall

Metode *waterfall* adalah suatu metode yang terstruktur dalam pengembangan sistem yang memerlukan tindakan-tindakan yang dilakukan secara berurutan atau linier. Hal pokok dari metode *waterfall* adalah bahwa setiap tindakan harus dilakukan secara berurutan dan tindakan kedua tidak dapat dilakukan sebelum tindakan pertama selesai dilakukan. Begitu pula, tindakan ketiga tidak dapat dilakukan sebelum tindakan kedua selesai dilakukan, dan seterusnya. Dengan kata lain, tindakan ke-3 hanya dapat dilakukan setelah tindakan ke-1 dan ke-2 selesai dilakukan. “(Theodorus yagoyamu, 2020)”;



Gambar II.2.14 Metode *Waterfall*

Tahapan-tahapan yang dilakukan metode *waterfall*, yaitu:

1. **Analisis Kebutuhan.** Untuk memulai mengembangkan sistem informasi, langkah pertama yang dilakukan adalah melakukan analisis kebutuhan. Analisis kebutuhan dilakukan dengan mengumpulkan informasi dari berbagai sumber seperti pengguna sistem, pemangku kepentingan dan dokumen terkait. Ada dua jenis kebutuhan utama dalam suatu sistem, yaitu kebutuhan fungsional dan kebutuhan non-fungsional.
2. **Desain Sistem.** Tahapan proses pengembangan sistem informasi, yang tujuannya adalah merancang antarmuka pengguna secara detail dan struktur sistem berdasarkan hasil analisis kebutuhan yang telah dilakukan sebelumnya. Pada fase ini, basis data dirancang, alat dan teknologi yang akan digunakan dipilih, dan antarmuka yang mudah digunakan dibuat. Semua hal tersebut sangat penting agar sistem yang dibuat dapat memenuhi kebutuhan pengguna dan bekerja secara efektif dan efisien.
3. **Implementasi.** Dalam tahap implementasi, langkah-langkah yang diambil meliputi pembuatan aplikasi yang didasarkan pada desain dan analisis kebutuhan sebelumnya. Proses ini meliputi pembuatan *database*, integrasi fitur dan modul aplikasi, pengembangan front-end, back-end, dan pemrograman

secara keseluruhan. Tujuan dari implementasi adalah untuk menciptakan aplikasi yang sesuai dengan kebutuhan dan desain sebelumnya sehingga dapat berfungsi dengan baik dan memenuhi kebutuhan pengguna.

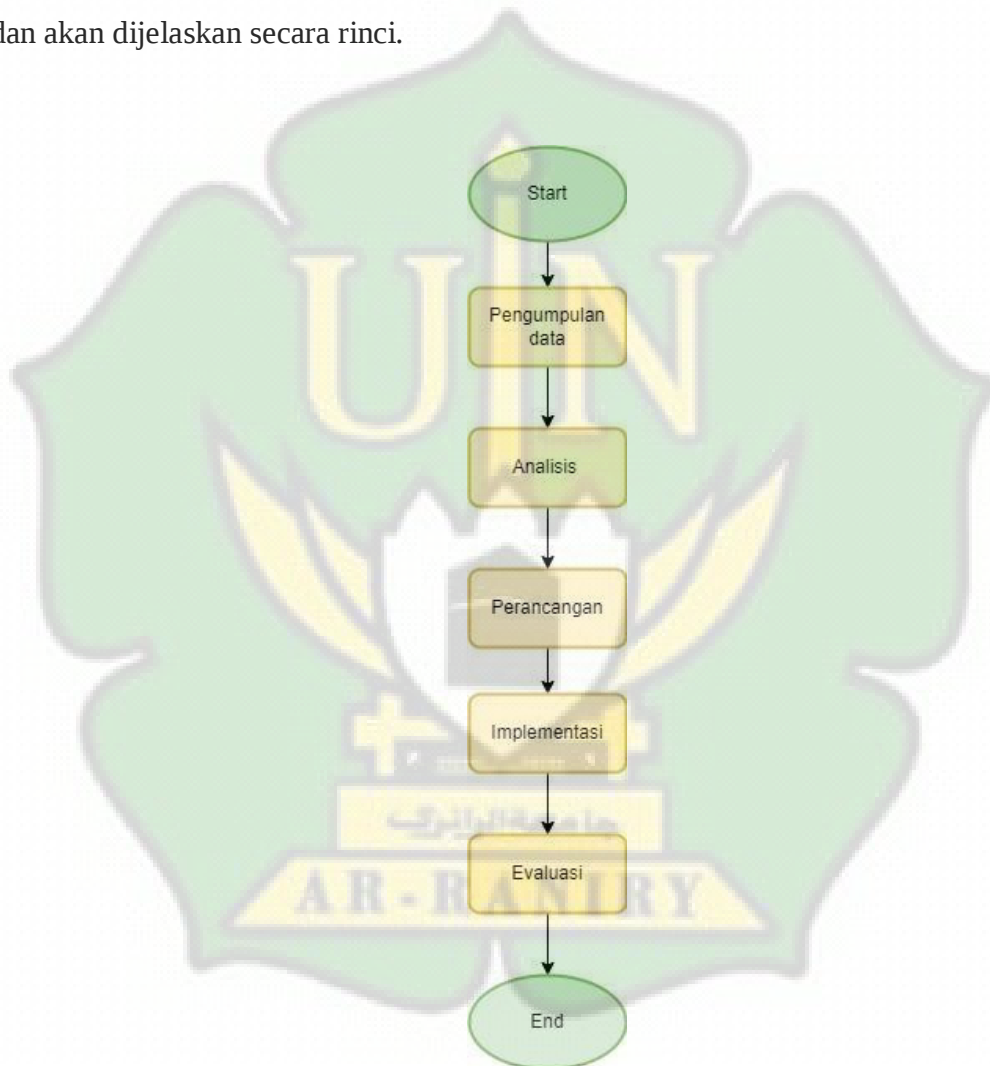
4. **Pengujian Sistem.** Pada tahap ini dilakukan serangkaian pengujian untuk memastikan bahwa sistem yang akan dibangun dapat bekerja dengan baik dan memenuhi kebutuhan pengguna. Tahap pengujian ini meliputi pengujian efisiensi, keamanan dan kehandalan sistem. Hasil pengujian digunakan untuk mengevaluasi dan memperbaiki sistem sebelum diterapkan secara penuh.
5. **Pemeliharaan.** Tujuan dari pemeliharaan adalah memastikan sistem dapat berjalan dengan baik dan tetap memenuhi kebutuhan penggunanya. Pemeliharaan sistem meliputi kegiatan perbaikan bug, perbaikan kinerja, penambahan fitur baru, dan pengaturan keamanan. Dalam pemeliharaan sistem, diperlukan tim yang terlatih dan terampil dalam mengelola sistem untuk menjamin kelancaran dan keberhasilan operasional sistem.

BAB III

METODE PENELITIAN

III.1 Tahapan Penelitian

Penelitian ini menggambarkan langkah-langkah yang diambil oleh penulis dalam melakukan penelitian. Langkah-langkah tersebut tertera pada Gambar III.1 dan akan dijelaskan secara rinci.



Gambar III.1 Diagram Alur Tahapan Penelitian

III.2 Metode Pengumpulan Data

III.2.1 Observasi

Pada tahapan ini, penulis akan melakukan pengamatan langsung pada objek atau subjek yang sedang diteliti, seperti mengamati penggunaan aplikasi sistem informasi manajemen transportasi pada pengguna.

III.2.2 Wawancara

Pada tahapan ini melibatkan proses wawancara yang dilakukan kepada admin PT Surveyor Indonesia yang nantinya akan menggunakan sistem yang telah dirancang. Pada PT Surveyor Indonesia memiliki satu orang admin untuk mengelola sistem tersebut. Hasil dari wawancara akan disusun dan digunakan dalam perancangan sistem informasi manajemen transportasi berbasis web.

Tabel III.2.2.1 Pertanyaan Wawancara

No	Pertanyaan
1	Bagaimana proses peminjaman mobil yang dilakukan oleh karyawan SI selama ini?
2	Apakah selama ini PT Surveyor Indonesia (SI) bekerjasama dengan beberapa vendor dalam penyediaan mobil atau hanya mengandalkan ketersediaan mobil kantor saja?
3	Bagaimana cara SI melakukan pemeliharaan terhadap mobil operasional?
4	Apakah kantor SI sering melakukan kegiatan dinas luar kota yang mengharuskan penggunaan mobil operasional?
5	Bagaimana cara pihak SI membayar vendor setelah menyewa mobil?
6	Apakah selama ini SI mencatat setiap peminjaman mobil operasional?
7	Menurut anda, apakah aplikasi SIMTRANS ini bisa mengefisiensikan proses peminjaman mobil?

III.3 Metode Pengembangan Aplikasi

Dalam metode pengembangan aplikasi pada sistem informasi manajemen transportasi berbasis *web* dengan menggunakan *framework laravel*, terdiri dari 4 macam: analisis kebutuhan, desain sistem, implementasi, pengujian sistem.

III.3.1 Analisis Kebutuhan

Analisis Kebutuhan Perangkat Lunak adalah langkah (tahap) awal dari siklus hidup pengembangan perangkat lunak, tahap ini akan menghasilkan spesifikasi / kebutuhan / kondisi yang akan dimiliki oleh perangkat lunak yang akan dikembangkan dengan mempertimbangkan semua kebutuhan dari seluruh *stakeholder*.

1. Kebutuhan Fungsional

Dalam sistem informasi, kebutuhan fungsional mengacu pada kebutuhan akan fasilitas dan aktivitas yang harus dilakukan oleh sistem secara keseluruhan. Ini mencakup proses yang akan dilakukan oleh sistem dan informasi yang harus dihasilkan oleh sistem. (Lila Setiyani, 2021).

Beberapa kebutuhan fungsional yang diperlukan oleh sistem informasi manajemen transportasi berbasis *web* dengan menggunakan *framework Laravel* adalah sebagai berikut:

- a. Registrasi pengguna
- b. *Login* dan *logout* pengguna
- c. Manajemen pemesanan mobil
- d. Manajemen kendaraan
- e. Manajemen *driver*
- f. Laporan keuangan
- g. Sistem vendor
- h. Manajemen

2. Kebutuhan Non-Fungsional

Kebutuhan Non Fungsional merujuk pada aspek perilaku sistem yang harus diperhatikan, dan sering kali menekankan pembatasan dalam layanan atau fungsi sistem, seperti batasan waktu, pembatasan dalam pengembangan, standarisasi, dan

lain-lain. Kebutuhan ini juga dikenal dengan istilah pembatasan layanan atau fungsi yang diberikan oleh sistem.

Berikut ini beberapa contoh kebutuhan non-fungsional yang harus dipenuhi dalam sistem informasi manajemen transportasi berbasis *web* dengan menggunakan *framework Laravel*:

- a. **Keamanan:** Sistem informasi harus memiliki tingkat keamanan yang memadai untuk melindungi informasi sensitif seperti data pribadi pelanggan dan informasi pembayaran.
- b. **Performa:** Sistem harus mampu memberikan kinerja yang baik dan responsif dalam melakukan tugas-tugasnya. Hal ini dapat dicapai dengan melakukan optimasi pada kode dan *database* , serta menggunakan teknologi *caching*.
- c. **Skalabilitas:** Sistem harus mampu menangani peningkatan jumlah pengguna dan data yang masuk ke dalam sistem. Ini dapat dicapai dengan menerapkan teknologi yang memungkinkan sistem untuk ditingkatkan atau diturunkan.
- d. **Ketersediaan:** Sistem harus selalu tersedia dan dapat diakses oleh pengguna kapanpun mereka membutuhkan. Hal ini dapat dicapai dengan mengimplementasikan teknologi yang memungkinkan pemantauan sistem secara *real-time* dan penanganan error yang cepat.
- e. **User Experience:** Sistem harus memiliki antarmuka yang mudah digunakan dan menarik bagi pengguna, sehingga dapat meningkatkan pengalaman pengguna dalam menggunakan sistem tersebut. Hal ini dapat dicapai dengan mengimplementasikan teknologi *front-end* yang modern dan *user-friendly*.

III.3.2 Desain Sistem

Setelah analisis kebutuhan selesai, langkah selanjutnya adalah merancang desain sistem. Desain ini mencakup desain arsitektur sistem, desain *database* , desain model data, dan desain antarmuka pengguna. Desain sistem pada sistem informasi manajemen transportasi (SIMTRANS) berbasis *web* akan mencakup struktur *database* yang tepat, entitas-entitas yang terlibat dalam sistem, serta tampilan dan navigasi antarmuka pengguna yang sesuai dengan kebutuhan pengguna dan tujuan sistem.

Dalam desain sistem aplikasi SIMTRANS terdapat 2 macam, yaitu: rancangan sistem dan rancangan *user interface*.

1. Rancangan Sistem

Metode *Unified Model Language* (UML) digunakan untuk merancang sistem informasi manajemen transportasi berbasis *web*. Model UML yang digunakan meliputi *use case diagram* dan *activity diagram*.

A. Use Case Diagram

Setelah menganalisis kebutuhan sistem, dibuatlah suatu diagram *use case* yang bertujuan untuk mengembangkan sistem informasi manajemen transportasi berbasis *web*. Diagram *use case* tersebut terdiri dari empat bagian, yakni *use case* untuk pengguna, *use case* untuk persetujuan, *use case* untuk admin, dan *use case* untuk vendor. Sebelum menggambarkan *use case diagram* maka harus ada identifikasi *actor*.

Tabel III.3.2.1 Identifikasi *actor use case diagram*

Actor	Deskripsi
Pegawai (<i>user</i>)	Orang yang menggunakan sistem untuk melakukan permohonan mobil, memilih jenis kendaraan, serta memantau status permohonan.
Admin	Bertanggung jawab mengelola dan memantau sistem, termasuk mengelola data kendaraan, <i>driver</i> , vendor, keuangan, dan lain-lain.
<i>Approval</i>	Memberikan persetujuan terhadap permohonan mobil dari pegawai.
Vendor	Menyediakan kendaraan yang disewakan melalui sistem.

1) Use case diagram pegawai (*user*)

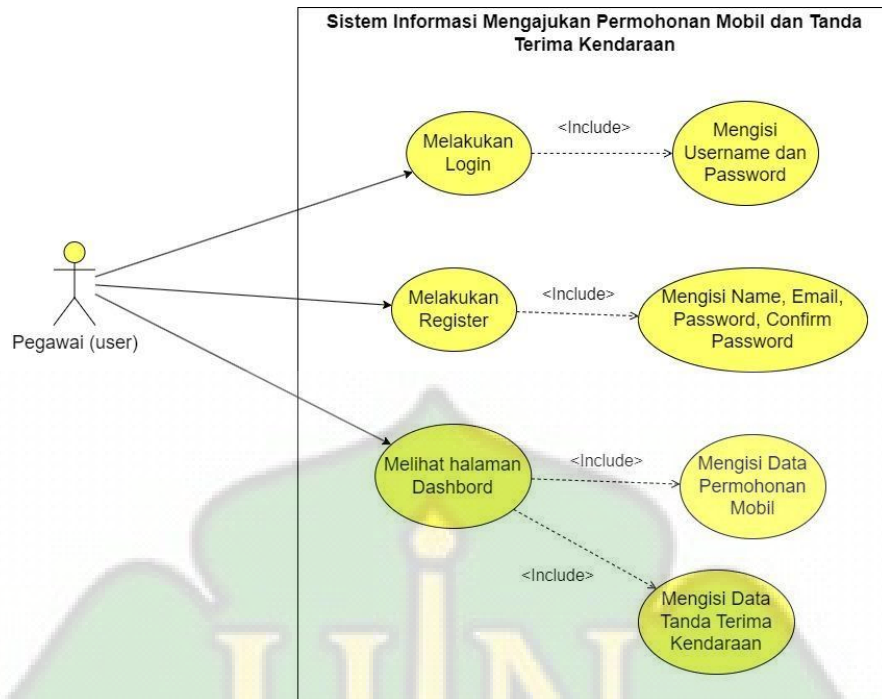
Actor pegawai merujuk pada karyawan atau staf PT Surveyor Indonesia yang menggunakan sistem informasi manajemen transportasi

berbasis *web* dengan menggunakan *framework Laravel*. Fungsi dari *actor* pegawai ini adalah:

- Mengajukan permohonan peminjaman mobil untuk keperluan tugas.
- Melakukan konfirmasi terkait persetujuan peminjaman mobil oleh pihak *approval*.
- Mengetahui status dan detail informasi terkait peminjaman mobil yang telah disetujui oleh pihak *approval*.
- Melakukan pembaruan informasi terkait peminjaman mobil (jika ada perubahan jadwal atau kendaraan yang akan digunakan).
- Melakukan konfirmasi pengembalian mobil setelah digunakan untuk keperluan tugas.
- Melaporkan kerusakan atau masalah pada kendaraan setelah penggunaan.
- Melihat riwayat peminjaman mobil yang pernah dilakukan.

Tugas pegawai dalam *use case diagram* sistem informasi manajemen transportasi berbasis *web* dengan menggunakan *framework Laravel* di PT Surveyor Indonesia dapat meliputi beberapa hal seperti:

- Melakukan pendaftaran pengajuan transportasi.
- Melakukan pengisian detail permintaan transportasi.
- Melakukan pengisian laporan perjalanan.
- Melakukan pemantauan status permintaan transportasi yang diajukan.
- Memberikan konfirmasi penerimaan transportasi dari vendor.
- Melakukan konfirmasi penerimaan laporan perjalanan.
- Melakukan pemantauan dan pengarsipan data-data transportasi dan laporan perjalanan.



Gambar III.2 Use Case Diagram User

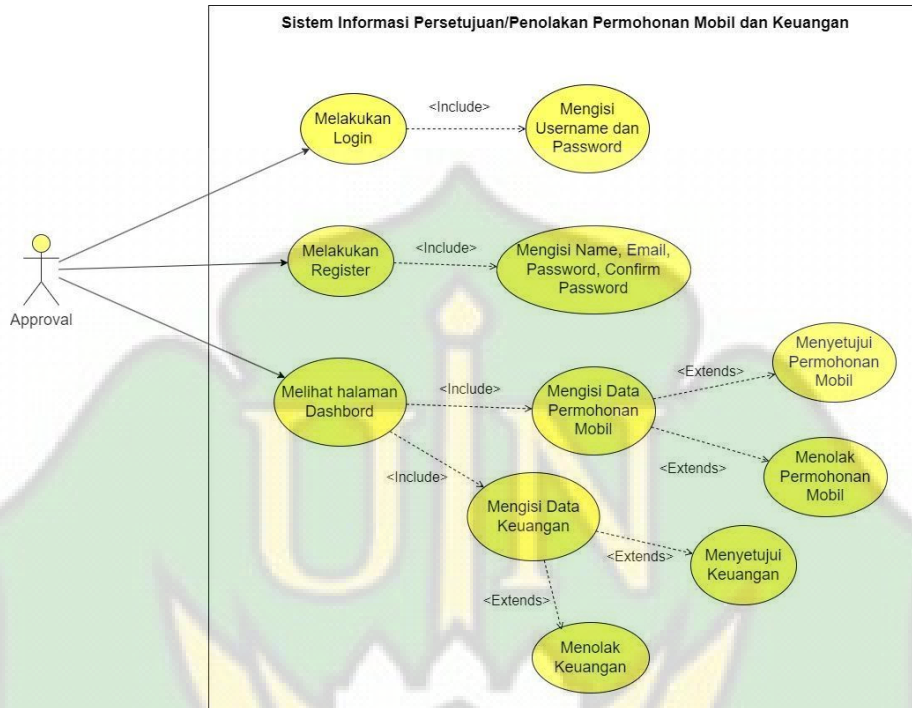
User/pegawai hendak masuk ke dalam sistem, dimana user diminta memasukkan alamat *email* dan *password* terlebih dahulu. Jika user belum mempunyai akun, maka user akan diberikan opsi untuk mendaftar melalui menu registrasi di mana user harus mengisi data nama, *email*, *password*, dan *confirm password*.

Setelah berhasil mendaftar, user dapat masuk ke dalam sistem dan dapat melihat halaman *dashboard* dan mengakses fitur menu permohonan mobil dan tanda terima kendaraan. Pegawai bisa mengisi data permohonan mobil dan tanda terima kendaraan. Sehingga dapat dilihat pada gambar III.2.

2) Use case diagram approval

Actor approval dapat berupa atasan atau pihak yang memiliki kewenangan untuk memberikan persetujuan dalam proses bisnis di PT Surveyor Indonesia. Fungsi dari actor "*approval*" dalam *use case diagram* sistem informasi manajemen transportasi berbasis web dengan menggunakan *framework Laravel* di PT Surveyor Indonesia adalah melakukan persetujuan atau *approval* terhadap permintaan transportasi

yang diajukan oleh pegawai. Dalam hal ini, *approval* berperan sebagai pihak yang memberikan keputusan akhir atas permintaan transportasi yang telah diajukan.



Gambar III.3 Use Case Diagram Approval

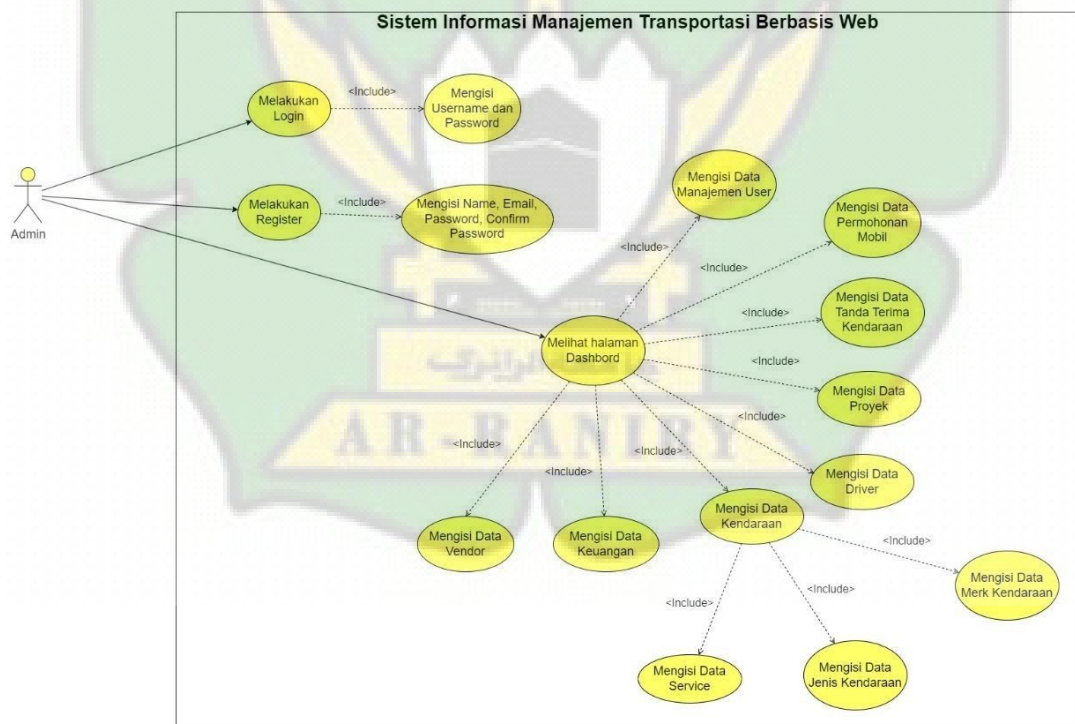
Seorang *approval* akan melakukan *login* ke dalam sistem dengan memasukkan *email* dan *password* terlebih dahulu. Jika *approval* belum memiliki akun, maka akan diarahkan ke menu registrasi di mana harus mengisi data nama, *email*, *password*, *confirm password*.

Setelah berhasil registrasi, *approval* dapat *login* ke dalam sistem dan masuk untuk melihat halaman *dashboard*. Di halaman *dashboard* dapat diakses fitur menu permohonan mobil dan keuangan. Kemudian *approval* dapat mengisi data tersebut dengan cara menyetujui/menolak data permohonan mobil dan keuangan. Sehingga dapat dilihat pada gambar III.3.

3) Use case diagram admin

Actor admin merujuk kepada orang yang bertanggung jawab untuk mengelola data-data dari sistem *web* tersebut. Tugas admin pada *use case diagram* dari sistem informasi manajemen transportasi berbasis *web* dengan menggunakan *framework Laravel* di PT Surveyor Indonesia dapat meliputi beberapa hal seperti:

- Mengelola data pegawai, *approval*, dan vendor dalam sistem.
- Menambahkan, mengubah, atau menghapus data transportasi yang ada dalam sistem.
- Memantau dan melihat laporan penggunaan transportasi.
- Melakukan verifikasi data yang diinput oleh pegawai atau *approval*.
- Memberikan hak akses kepada pegawai dan *approval* terkait dengan penggunaan sistem.
- Menjaga keamanan dan kerahasiaan data dalam sistem.



Gambar III.4 Use Case Diagram Admin

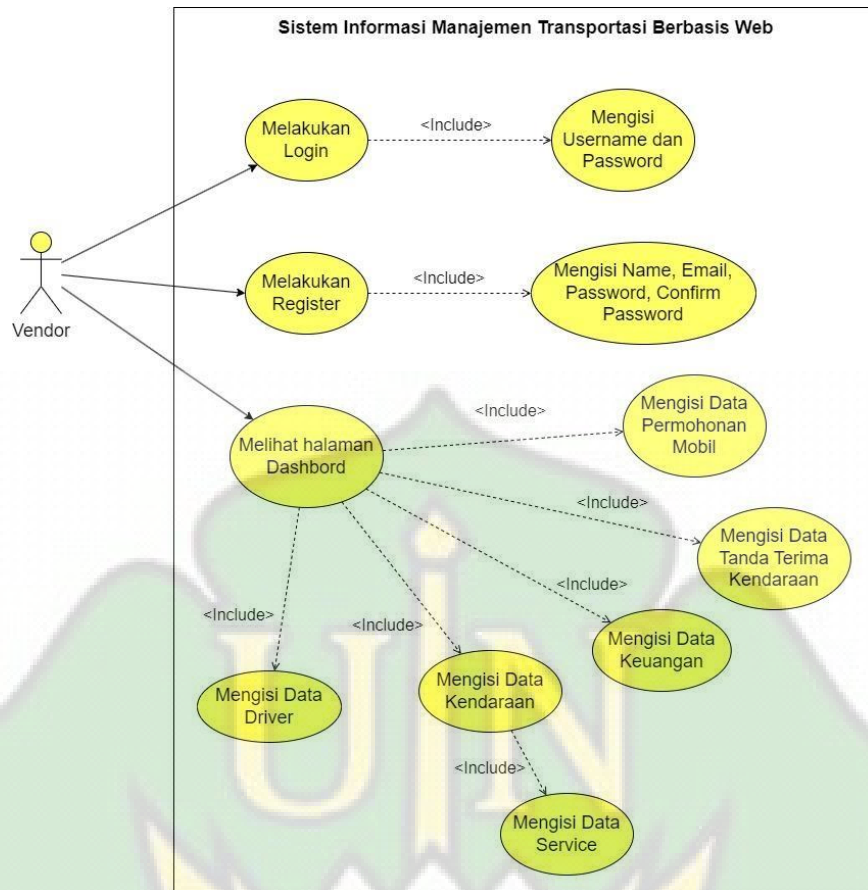
Langkah pertama yang dilakukan adalah admin melakukan *login* ke dalam sistem dengan memasukkan *email* dan *password*. Jika admin belum memiliki akun, maka akan diarahkan ke halaman registrasi, di mana admin diminta untuk mengisi nama, *email*, *password*, dan *confirm password*.

Setelah berhasil melakukan registrasi, admin dapat masuk ke halaman *login*. Setelah berhasil *login*, admin bisa mengakses melihat halaman *dashboard*, menu permohonan mobil, *driver*, vendor, keuangan, tanda terima kendaraan, proyek, kendaraan (merk, jenis, dan data), dan manajemen *user*. Admin dapat mengisi data permohonan mobil, *driver*, vendor, keuangan, tanda terima kendaraan, proyek, kendaraan (merk, jenis, dan data), dan manajemen *user*. Sehingga dapat dilihat pada gambar III.4.

4) *Use case diagram vendor*

Actor vendor merupakan perusahaan *third party* atau pihak ketiga yang memberikan kendaraan untuk disewakan kepada PT Surveyor Indonesia. Tugas vendor dalam *use case diagram* dari sistem informasi manajemen transportasi berbasis *web* dengan menggunakan *framework Laravel* di PT Surveyor Indonesia biasanya meliputi:

- Menambahkan data transportasi ke dalam sistem.
- Mengupdate data transportasi yang sudah ada dalam sistem.
- Melihat laporan penggunaan transportasi yang terkait dengan vendor tersebut.
- Melihat status pemesanan yang dilakukan oleh pelanggan.
- Memproses permintaan pemesanan transportasi dari pelanggan.
- Menambahkan informasi faktur dan melakukan konfirmasi pembayaran dari pelanggan.



Gambar III.5 Use Case Diagram Vendor

Vendor melakukan *login* menggunakan *email* dan *password*. Jika vendor belum memiliki akun, maka akan diarahkan ke menu registrasi. Vendor diwajibkan mengisi data seperti nama, *email*, *password*, *confirm password* pada halaman registrasi.

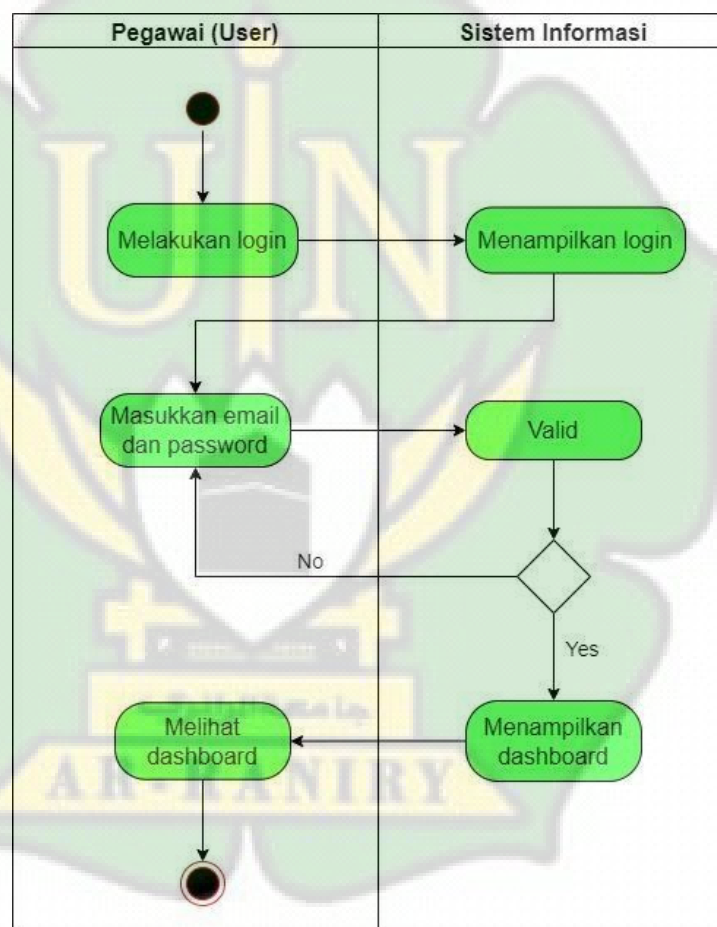
Setelah berhasil terdaftar, vendor dapat melakukan *login* dan bisa mengakses ke halaman *dashboard*, menu permohonan mobil, kendaraan (data), *service*, *driver*, keuangan, serta tanda terima kendaraan. Vendor dapat mengisi data permohonan mobil, kendaraan (data), *service*, *driver*, keuangan, serta tanda terima kendaraan. Untuk detail lebih lengkap, dapat dilihat pada gambar III.5.

B. Activity Diagram

Dari *use case* yang telah disusun sebelumnya, dibuatlah *activity diagram* untuk sistem informasi manajemen transportasi berbasis *web*. *Activity diagram* ini terdiri dari empat bagian yaitu *activity diagram user*, *activity diagram approval*, *activity diagram admin*, dan *activity diagram vendor*.

a. Activity diagram user

1. Activity diagram login



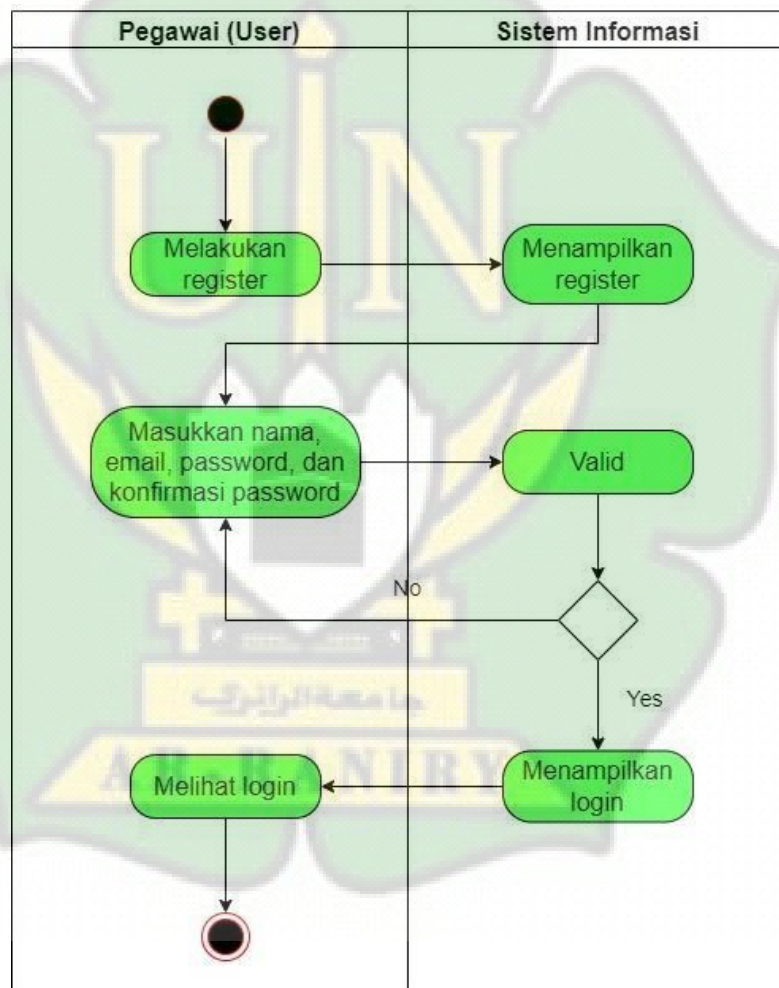
Gambar III.6 Activity Diagram User Login

Diagram aktivitas untuk *login* pengguna menjelaskan bahwa saat pengguna ingin masuk ke dalam sistem, sistem akan menampilkan menu *login* dan meminta pengguna untuk memasukkan *email* dan *password*.

Kemudian, sistem akan memeriksa keabsahan data yang dimasukkan oleh pengguna.

Jika data yang dimasukkan salah, pengguna akan diminta untuk memasukkan kembali *email* dan *password*. Tetapi, jika data yang dimasukkan benar, maka pengguna akan diarahkan ke halaman *dashboard*. Informasi ini terlihat pada gambar III.6.

2. Activity diagram register



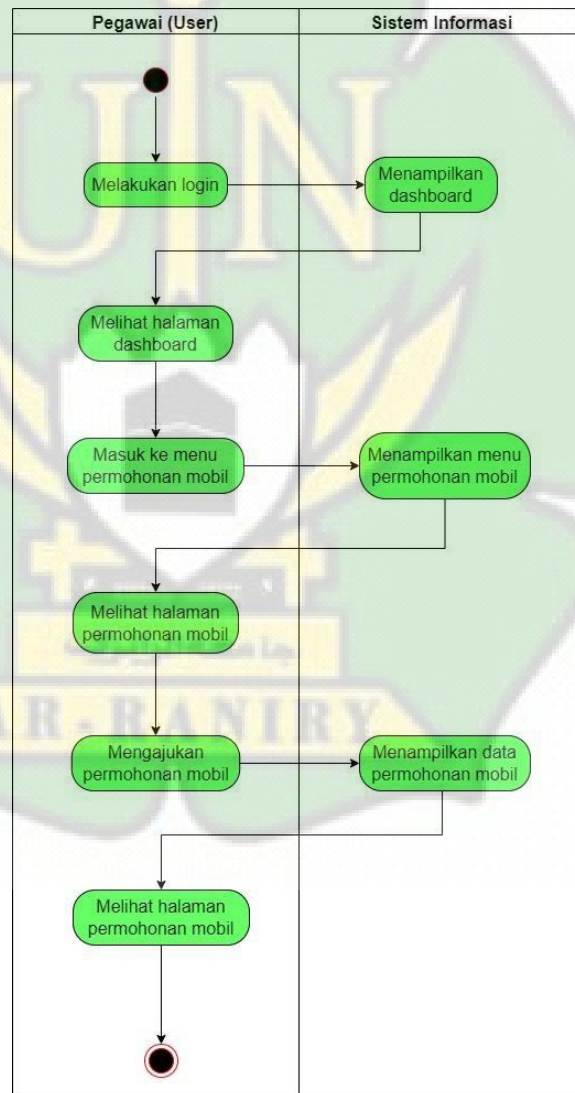
Gambar III.7 Activity Diagram User Register

Diagram activity user registrasi menjelaskan bahwa ketika pengguna ingin melakukan registrasi, sistem akan menampilkan menu registrasi dan

meminta pengguna untuk mengisi data seperti nama, *email*, *password*, dan konfirmasi *password*.

Selanjutnya, sistem akan memvalidasi data yang dimasukkan oleh pengguna. Jika data yang dimasukkan salah, pengguna akan diminta untuk memasukkan ulang data tersebut. Namun jika data yang dimasukkan benar, maka pengguna akan diarahkan ke menu *login*. Informasi ini dapat dilihat pada gambar III.7.

3. Activity diagram permohonan mobil

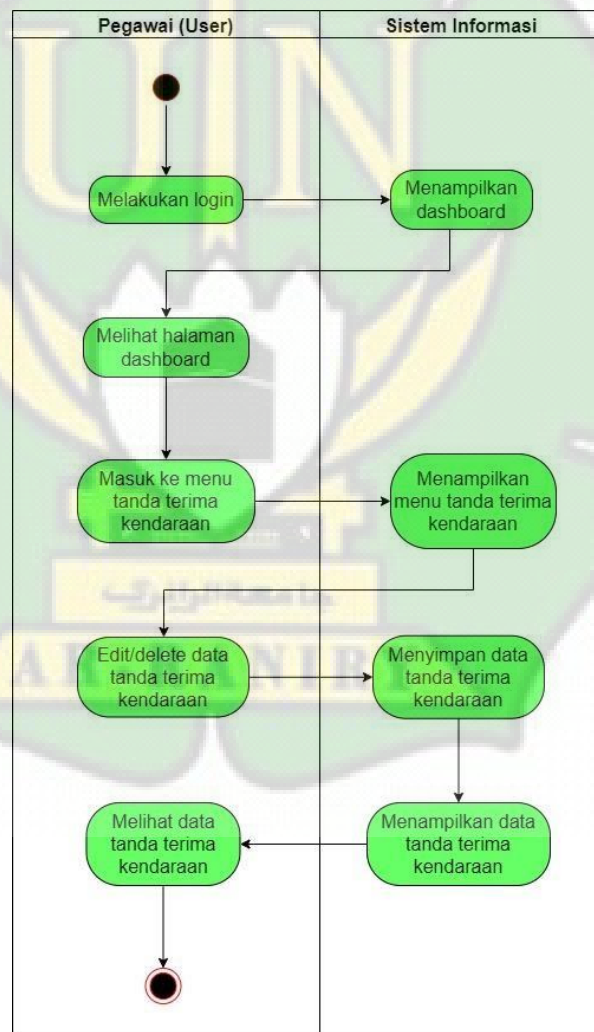


Gambar III.8 Activity Diagram User Permohonan Mobil

Diagram activity permohonan mobil pengguna menjelaskan bahwa ketika pengguna berhasil *login*, sistem akan menampilkan halaman *dashboard*. Pengguna dapat memilih menu permohonan mobil di halaman tersebut, dan sistem akan menampilkan menu tersebut.

Selanjutnya, pengguna akan melihat halaman permohonan mobil dan mengajukan permohonan tersebut. Setelah itu, sistem akan menampilkan data permohonan mobil dan pengguna akan melihat halaman permohonan mobil tersebut. *Diagram activity* tersebut dapat dilihat pada gambar III.8.

4. *Activity diagram* tanda terima kendaraan



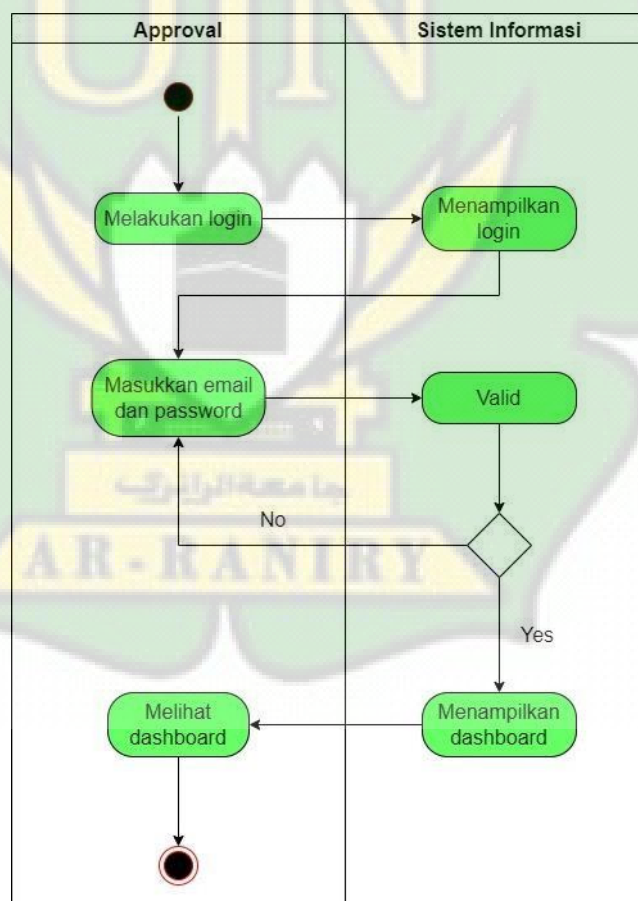
Gambar III.9 *Activity Diagram User* Tanda Terima Kendaraan

Diagram activity tanda terima kendaraan pengguna menjelaskan bahwa ketika pengguna berhasil *login*, sistem akan menampilkan halaman *dashboard*. Pengguna dapat memilih menu tanda terima kendaraan di halaman tersebut, dan sistem akan menampilkan menu tersebut.

Selanjutnya, pengguna akan melihat halaman tanda terima kendaraan dan mengisi data tersebut. Setelah itu, sistem akan menampilkan data tanda terima kendaraan dan pengguna akan melihat halaman tanda terima kendaraan tersebut. *Diagram activity* tersebut dapat dilihat pada gambar III.9.

b. Activity diagram approval

1. Activity diagram login

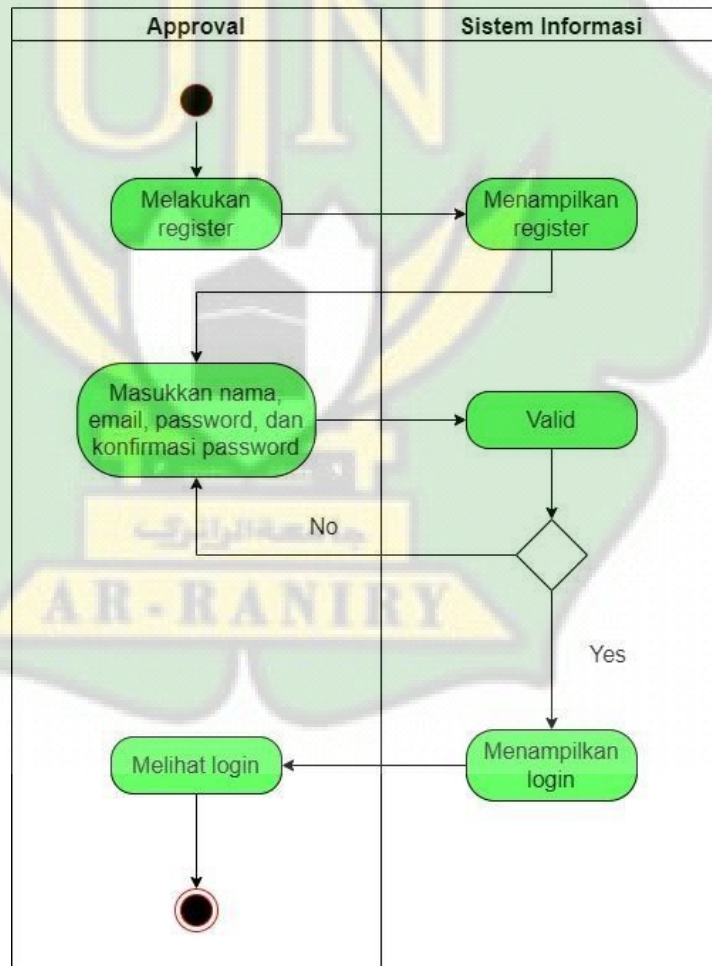


Gambar III.10 Activity Diagram Approval Login

Diagram activity login approval menjelaskan bahwa ketika pengguna *approval* hendak masuk ke sistem, maka sistem akan menampilkan halaman *login*. Kemudian pengguna *approval* diminta untuk memasukkan *email* dan *password* ke dalam sistem. Selanjutnya, sistem akan melakukan validasi data yang dimasukkan oleh pengguna *approval*.

Jika data yang dimasukkan salah, pengguna *approval* akan diminta untuk memasukkan ulang *email* dan *password* yang benar. Namun jika data yang dimasukkan sudah benar, pengguna *approval* akan berhasil masuk ke dalam *dashboard* sistem. Lebih rinciannya dapat dilihat pada gambar III.10.

2. Activity diagram register

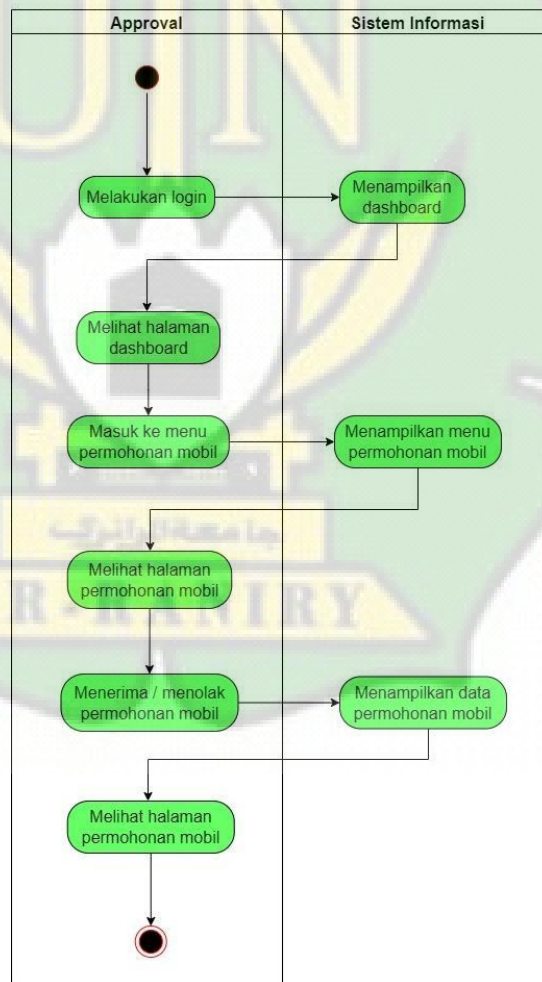


Gambar III.11 Activity Diagram Approval Register

Diagram activity approval registrasi menjelaskan bahwa ketika *approval* ingin mendaftar, maka sistem akan menampilkan menu pendaftaran. *Approval* harus memasukkan nama, *email*, *password*, dan konfirmasi *password*, kemudian sistem akan memeriksa validitas data yang dimasukkan.

Jika ada kesalahan, *approval* akan diminta untuk memperbaiki data yang tidak valid. Namun, jika data yang dimasukkan valid, *approval* akan dapat masuk ke halaman *login*. Gambaran lengkap dari diagram ini dapat dilihat pada gambar III.11.

3. Activity diagram permohonan mobil

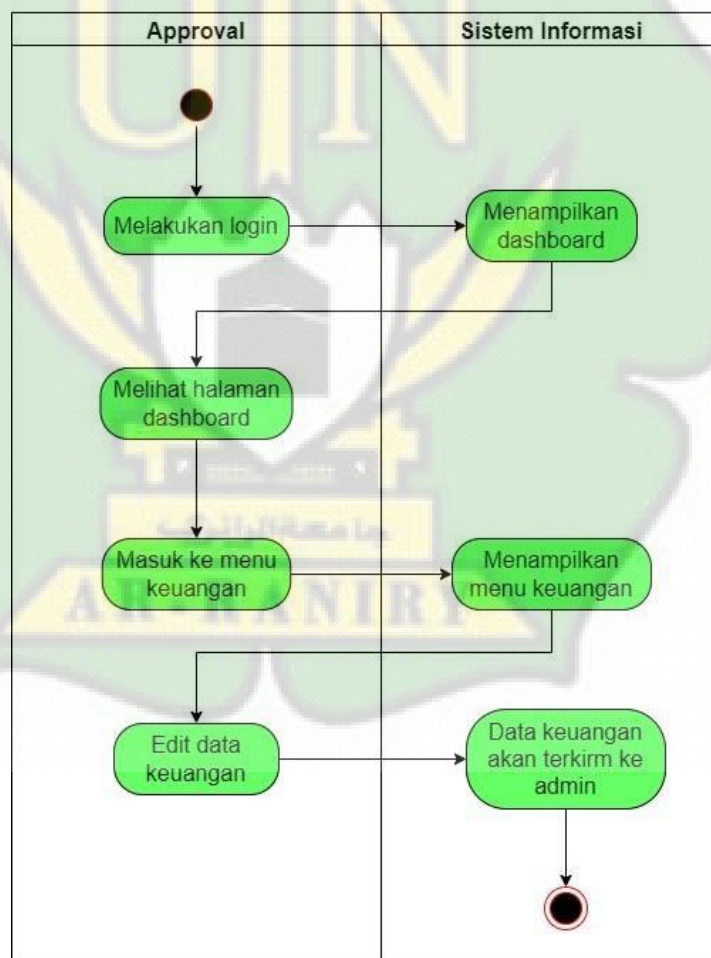


Gambar III.12 Activity Diagram Approval
Permohonan Mobil

Diagram activity approval permohonan mobil menjelaskan bahwa ketika *approval* melakukan *login* maka sistem akan menampilkan halaman *dashboard*. *Approval* akan melihat halaman dashbord dan akan diminta masuk ke menu permohonan mobil maka sistem akan menampilkan menu permohonan mobil.

Kemudian *approval* akan melihat halaman permohonan mobil dan akan menerima / menolak permohonan mobil, lalu sistem akan menampilkan data permohonan mobil. Terakhir, *approval* akan melihat halaman permohonan mobil tersebut, sehingga dapat dilihat pada gambar III.12.

4. Activity diagram keuangan

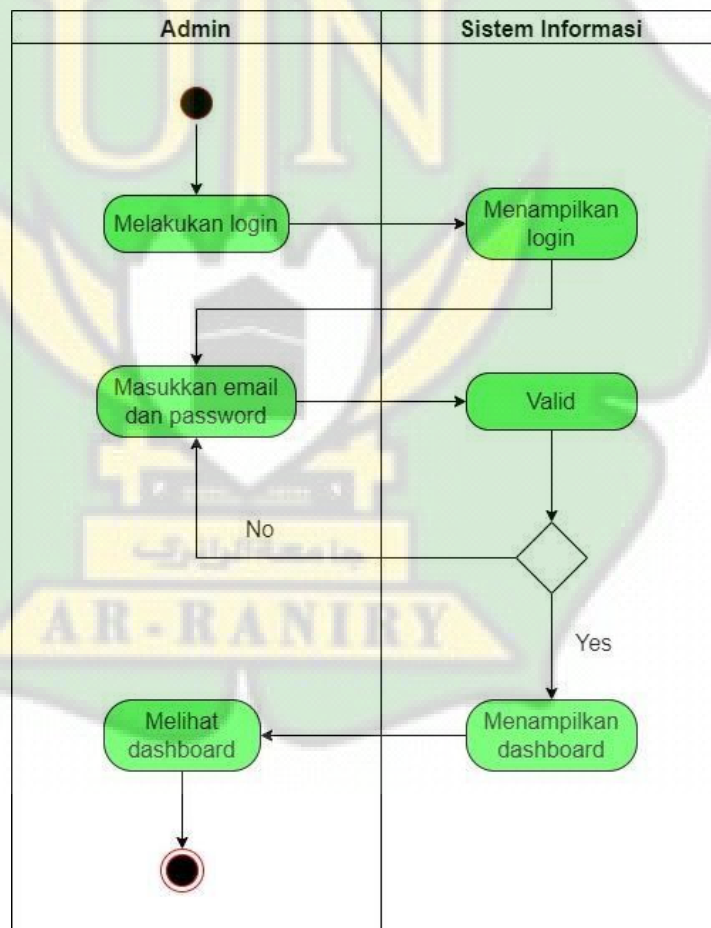


Gambar III.13 Activity Diagram Approval Keuangan

Diagram activity approval keuangan menjelaskan bahwa ketika *approval* melakukan *login* maka sistem akan menampilkan halaman *dashboard*. *Approval* akan melihat halaman dashbord dan akan diminta masuk ke menu keuangan maka sistem akan menampilkan menu keuangan. Kemudian *approval* akan melihat halaman keuangan dan akan menerima / menolak keuangan, lalu sistem akan mengirim data keuangan ke admin. Sehingga dapat dilihat pada gambar III.13.

c. Activity diagram admin

1. Activity diagram login

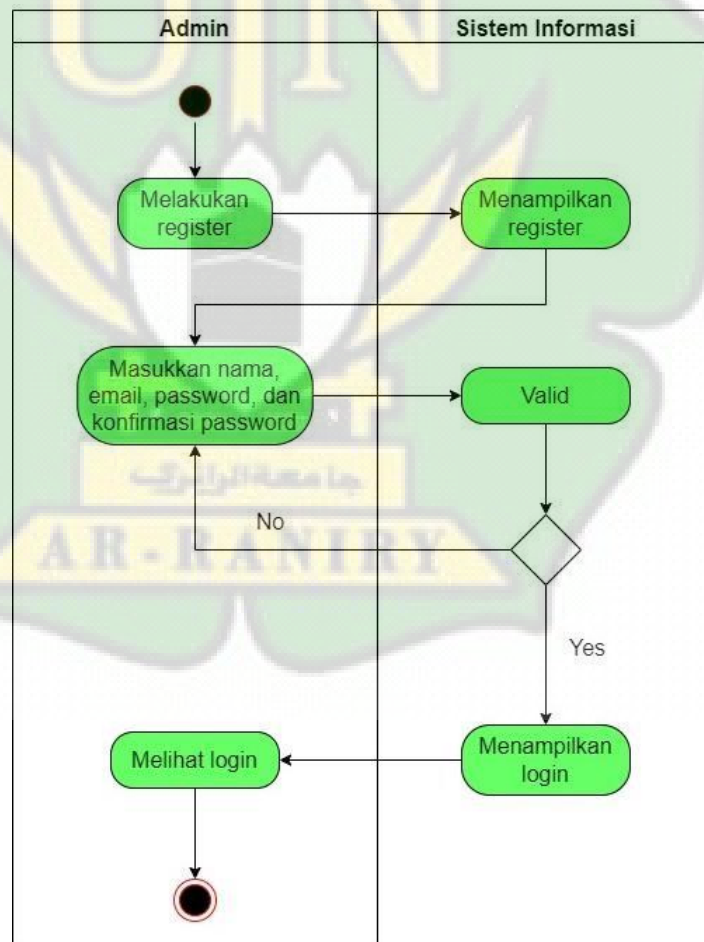


Gambar III.14 Activity Diagram Admin Login

Diagram activity login admin menjelaskan bahwa pada saat admin ingin masuk ke dalam sistem, maka sistem akan menampilkan halaman *login* terlebih dahulu. Kemudian, admin diminta untuk memasukkan *email* dan *password* ke dalam sistem. Setelahnya, sistem akan memvalidasi data yang dimasukkan oleh admin.

Jika terdapat kesalahan dalam memasukkan data, maka admin akan diminta untuk memasukkan *email* dan *password* kembali dengan benar. Namun, jika data yang dimasukkan sudah benar, maka admin dapat berhasil masuk ke dalam halaman *dashboard* sistem. Lebih detail mengenai hal ini dapat dilihat pada gambar III.14.

2. Activity diagram register

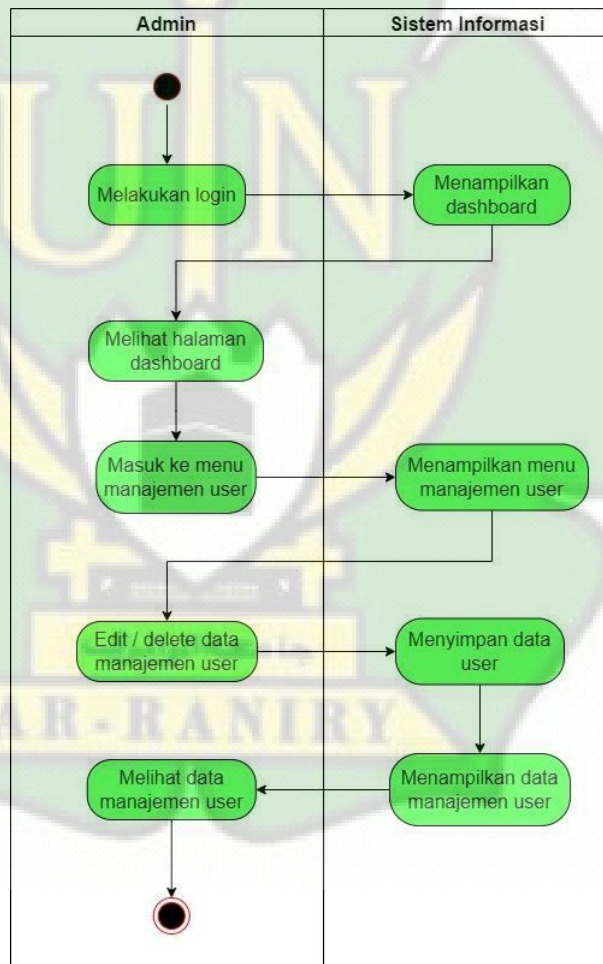


Gambar III.15 Activity Diagram Admin Register

Diagram activity admin registrasi menjelaskan bahwa ketika admin ingin melakukan register maka sistem akan menampilkan register. Admin akan diminta untuk memasukkan nama, *email*, *password*, dan konfirmasi *password*, kemudian sistem akan memvalidasi data.

Jika salah maka admin akan diminta memasukan ulang nama, *email*, *password* dan konfirmasi *password*, jika benar admin akan masuk ke *login*, sehingga dapat dilihat pada gambar III.15.

3. Activity diagram manajemen user



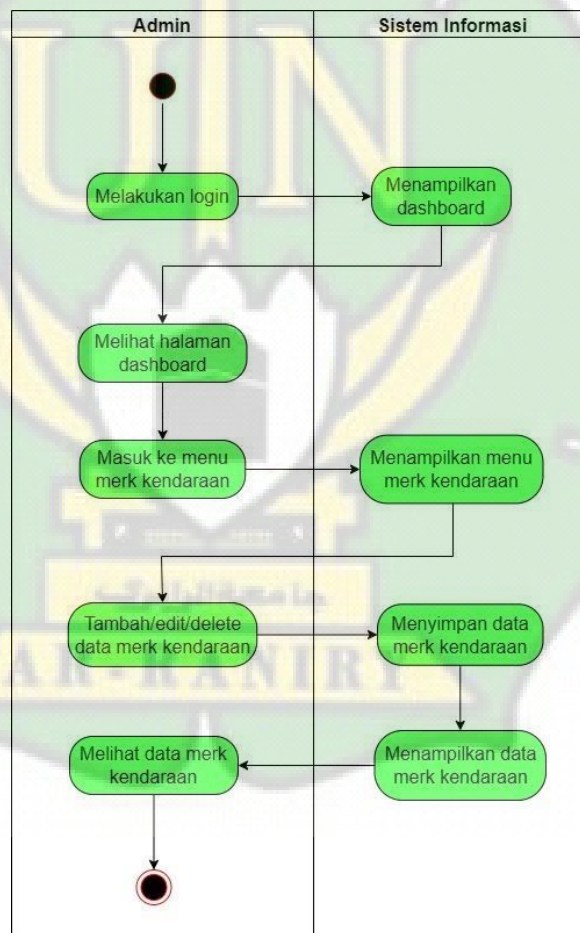
Gambar III.16 Activity Diagram Admin Manajemen User

Diagram activity manajemen user oleh admin menjelaskan bahwa setelah admin berhasil melakukan *login* pada sistem, maka sistem akan

menampilkan *dashboard*. Selanjutnya, admin akan mengakses menu manajemen *user* pada *dashboard*, sehingga sistem akan menampilkan menu tersebut.

Admin kemudian akan melakukan edit atau hapus pada data manajemen *user*, dan sistem akan menyimpan data tersebut. Setelah melakukan edit atau hapus, sistem akan menampilkan data manajemen *user*, sehingga admin dapat melihat data tersebut. Detail lengkapnya dapat dilihat pada gambar III.16.

4. Activity diagram merk kendaraan



Gambar III.17 Activity Diagram Admin Merk Kendaraan

Dalam *diagram activity* merk kendaraan admin, dijelaskan bahwa ketika admin masuk ke dalam sistem, maka sistem akan menampilkan

halaman *dashboard*. Admin akan melihat halaman *dashboard* dan memilih menu merk kendaraan, lalu sistem akan menampilkan menu merk kendaraan.

Selanjutnya, admin akan diminta untuk menambahkan, mengedit, atau menghapus data merk kendaraan, dan sistem akan menampilkan data merk kendaraan yang sesuai dengan permintaan admin. Akhirnya, admin akan melihat data merk kendaraan tersebut yang dapat dilihat pada gambar III.17.

5. Activity diagram jenis kendaraan



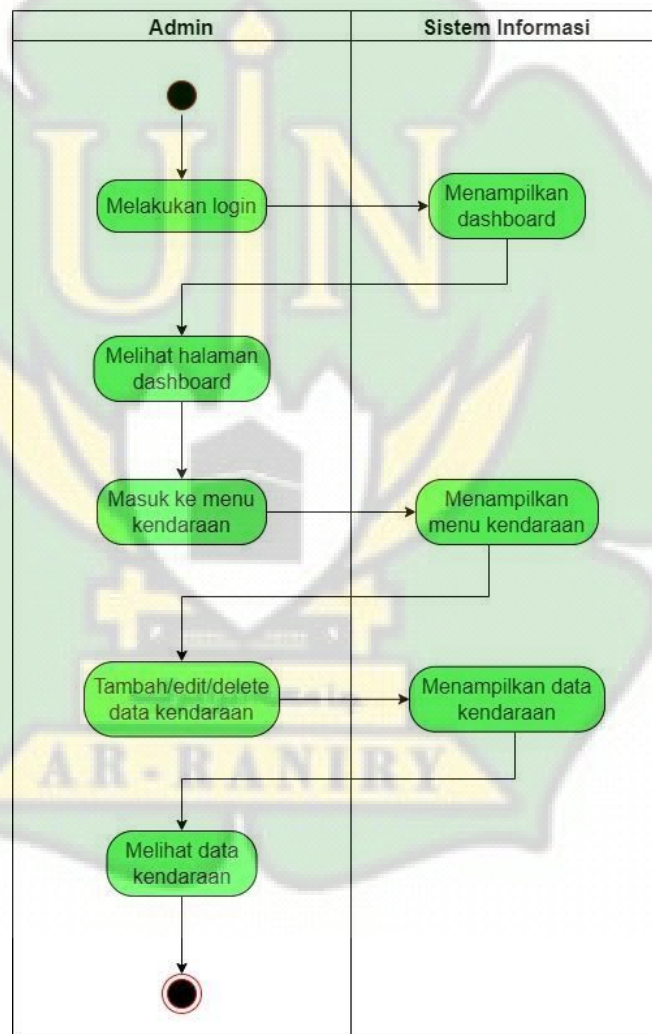
Gambar III.18 Activity Diagram Admin Jenis Kendaraan

Dalam *diagram activity* jenis kendaraan admin, dijelaskan bahwa ketika admin masuk ke dalam sistem, maka sistem akan menampilkan

halaman *dashboard*. Admin akan melihat halaman *dashboard* dan memilih menu jenis kendaraan, lalu sistem akan menampilkan menu jenis kendaraan.

Selanjutnya, admin akan diminta untuk menambahkan, mengedit, atau menghapus data jenis kendaraan, dan sistem akan menampilkan data jenis kendaraan yang sesuai dengan permintaan admin. Akhirnya, admin akan melihat data jenis kendaraan tersebut yang dapat dilihat pada gambar III.18.

6. Activity diagram data kendaraan



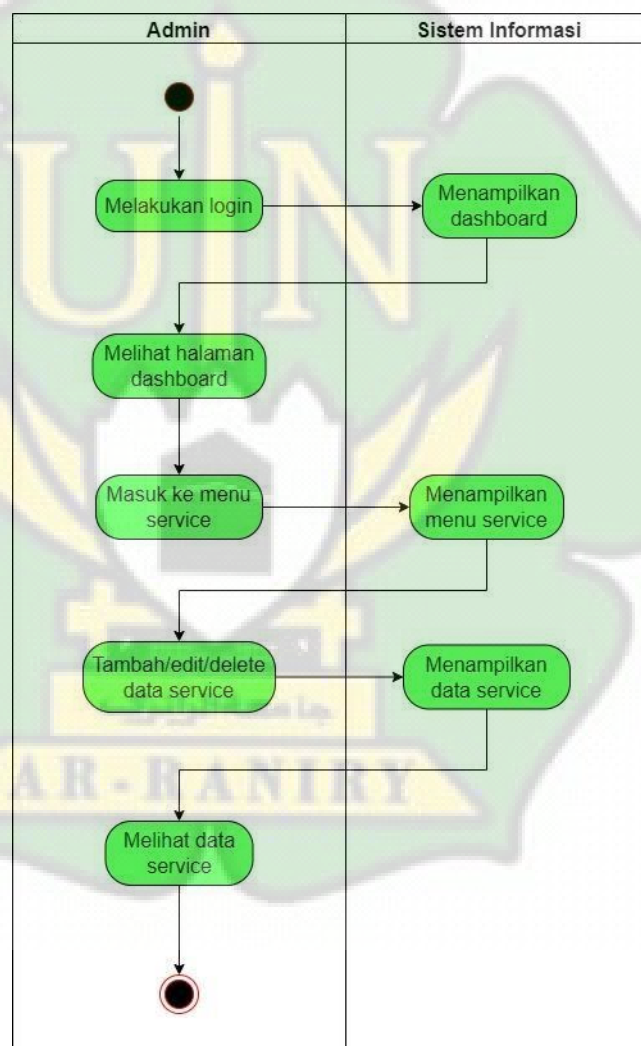
Gambar III.19 Activity Diagram Admin Kendaraan

Dalam *diagram activity* kendaraan admin, dijelaskan bahwa ketika admin masuk ke dalam sistem, maka sistem akan menampilkan halaman

dashboard. Admin akan melihat halaman *dashboard* dan memilih menu data kendaraan, lalu sistem akan menampilkan menu data kendaraan.

Selanjutnya, admin akan diminta untuk menambahkan, mengedit, atau menghapus data kendaraan, dan sistem akan menampilkan data kendaraan yang sesuai dengan permintaan admin. Akhirnya, admin akan melihat data kendaraan tersebut yang dapat dilihat pada gambar III.19.

7. Activity diagram data service

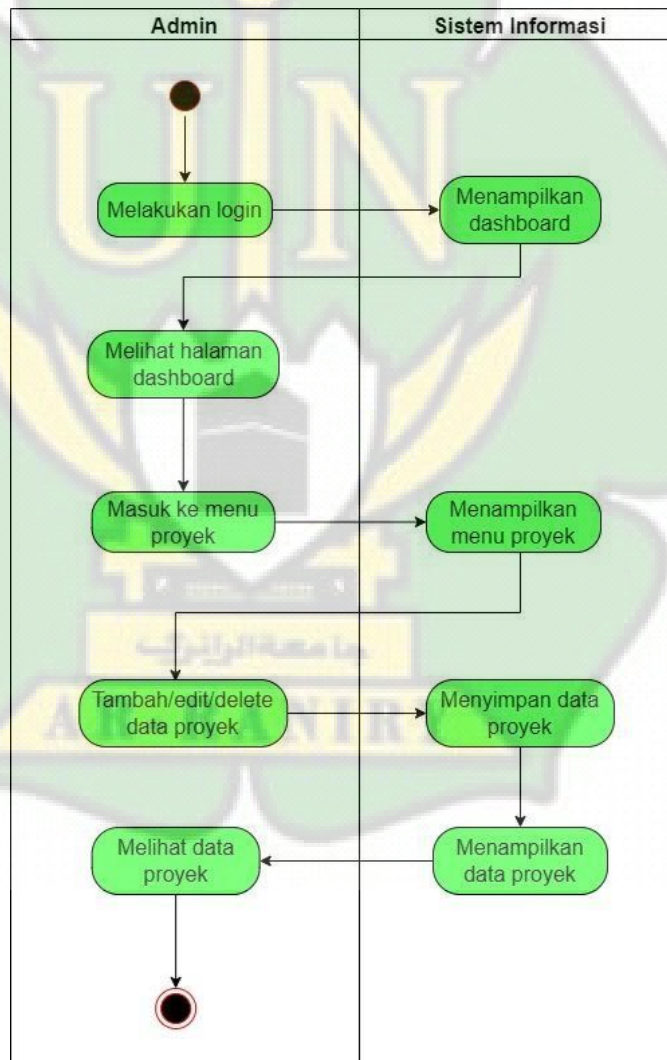


Gambar III.20 Activity Diagram Admin Service

Diagram activity admin menjelaskan bahwa ketika admin *login* ke aplikasi, sistem akan menampilkan halaman *dashboard*. Setelah itu, admin akan memilih menu dan sistem akan menampilkan menu tersebut.

Selanjutnya, admin akan diminta untuk menambahkan, mengedit, atau menghapus data, dan sistem akan menampilkan data yang relevan. Terakhir, admin akan melihat data tersebut di halaman yang sesuai. Semua langkah ini dapat dilihat secara detail pada gambar III.20.

8. Activity diagram proyek

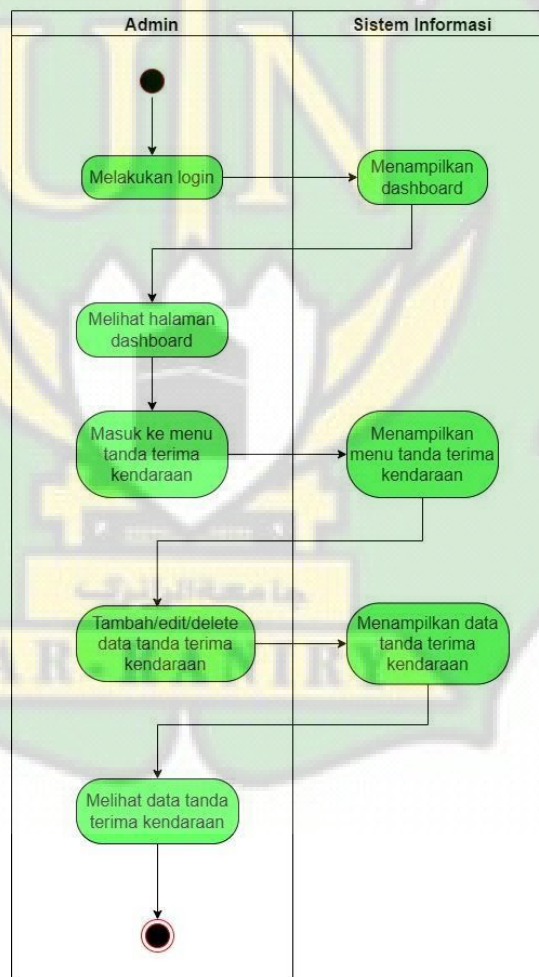


Gambar III.21 Activity Diagram Admin Proyek

Diagram activity admin proyek menjelaskan bahwa ketika admin *login* ke aplikasi, sistem akan menampilkan halaman *dashboard*. Setelah itu, admin akan memilih menu proyek dan sistem akan menampilkan menu tersebut.

Selanjutnya, admin akan diminta untuk menambahkan, mengedit, atau menghapus data proyek, dan sistem akan menampilkan data proyek yang relevan. Terakhir, admin akan melihat data proyek tersebut di halaman yang sesuai. Semua langkah ini dapat dilihat secara detail pada gambar III.21.

9. *Activity diagram* tanda terima kendaraan

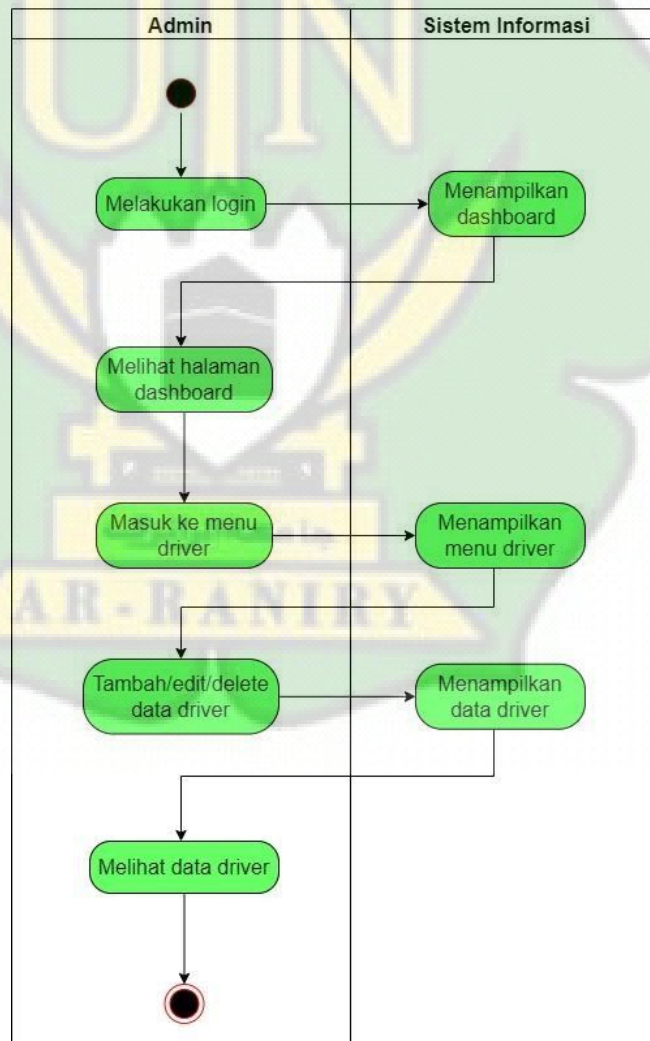


Gambar III.22 *Activity Diagram* Admin
Tanda Terima Kendaraan

Diagram activity admin tanda terima kendaraan menjelaskan bahwa ketika admin *login* ke aplikasi, sistem akan menampilkan halaman *dashboard*. Setelah itu, admin akan memilih menu tanda terima kendaraan dan sistem akan menampilkan menu tersebut.

Selanjutnya, admin akan diminta untuk menambahkan, mengedit, atau menghapus data tanda terima kendaraan, dan sistem akan menampilkan data tanda terima kendaraan yang relevan. Terakhir, admin akan melihat data tanda terima kendaraan tersebut di halaman yang sesuai. Semua langkah ini dapat dilihat secara detail pada gambar III.22.

10. Activity diagram driver

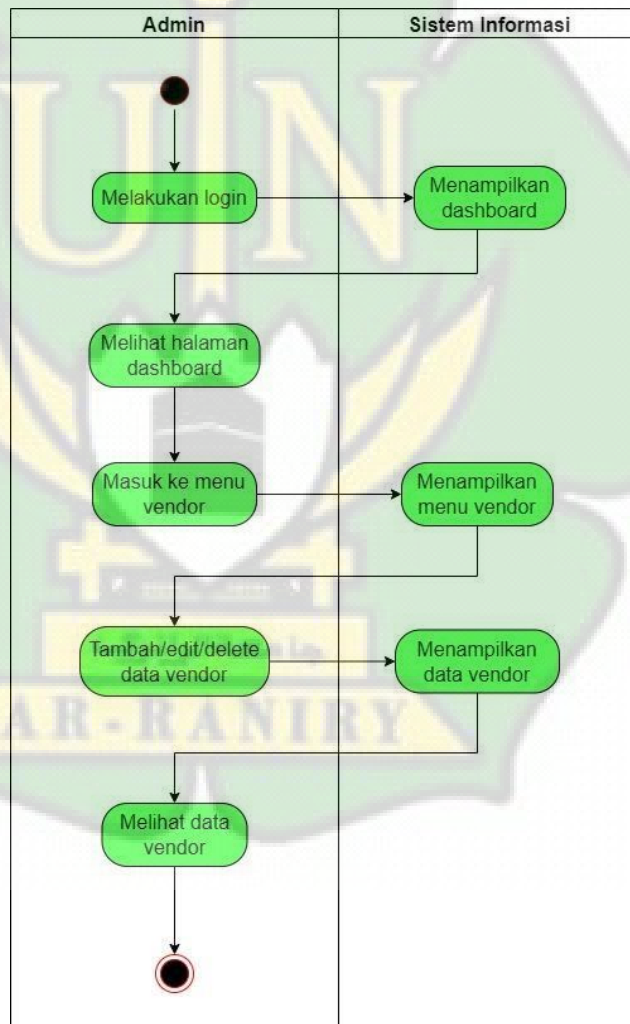


Gambar III.23 Activity Diagram Admin Driver

Dalam *diagram activity admin driver*, dijelaskan bahwa ketika admin *login* ke aplikasi, sistem akan menampilkan halaman *dashboard*. Selanjutnya, admin akan melihat menu *driver* pada halaman tersebut dan akan memilih untuk menambahkan, mengedit, atau menghapus data *driver*.

Sistem akan menampilkan data *driver* yang diminta oleh admin. Terakhir, admin akan dapat melihat data *driver* tersebut yang dapat dilihat pada gambar III.23.

11. Activity diagram vendor

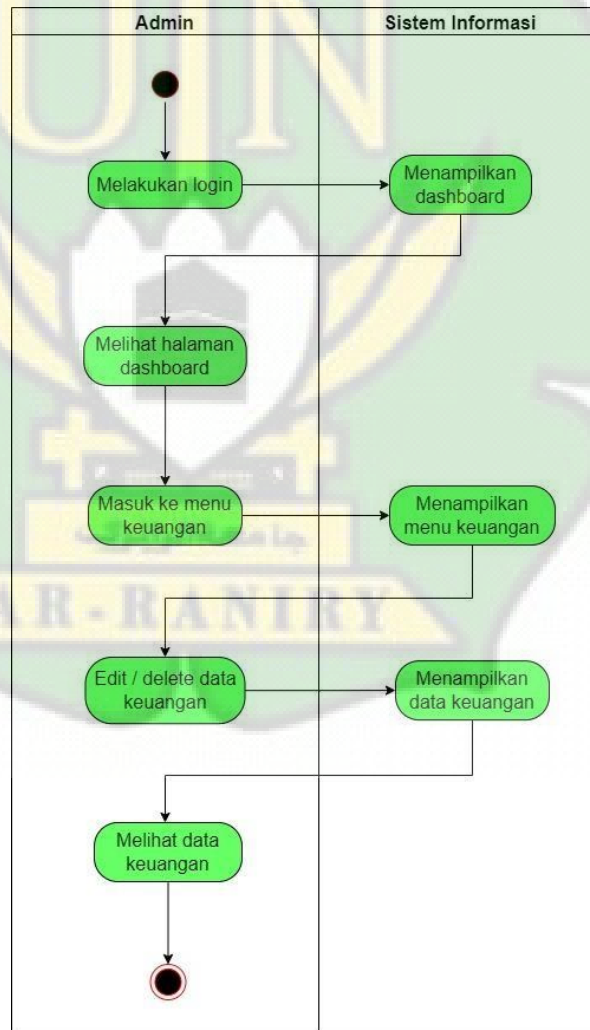


Gambar III.24 Activity Diagram Admin Vendor

Diagram activity untuk admin vendor menggambarkan bagaimana admin dapat mengakses dan mengelola data vendor dalam sistem. Ketika admin berhasil melakukan *login* ke aplikasi, maka akan ditampilkan halaman *dashboard*.

Dari sana, admin dapat memilih menu vendor dan sistem akan menampilkan opsi-opsi yang tersedia pada menu tersebut. Admin dapat menambah, mengedit, atau menghapus data vendor dan sistem akan menampilkan data vendor yang sesuai. Semua proses tersebut terlihat jelas pada gambar III.24.

12. Activity diagram keuangan



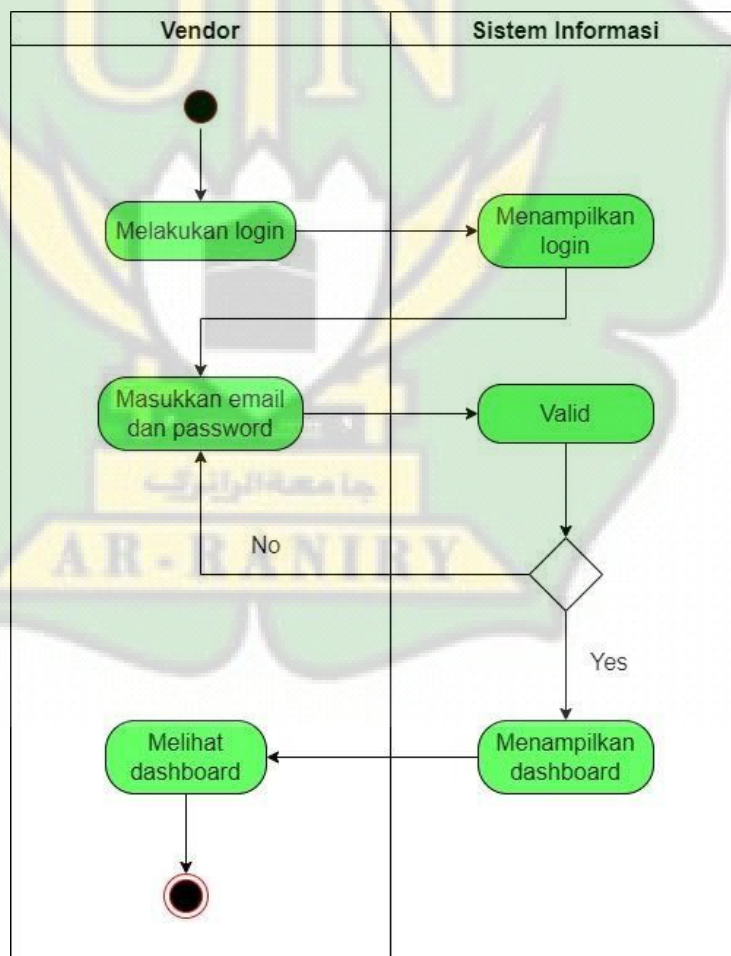
Gambar III.25 Activity Diagram Admin Keuangan

Dalam *diagram activity* admin keuangan dijelaskan bahwa ketika admin melakukan *login* ke aplikasi, maka sistem akan menampilkan *dashboard*. Selanjutnya, admin akan melihat halaman *dashboard* dan memilih menu keuangan, dan sistem akan menampilkan menu keuangan.

Setelah itu, admin akan diminta untuk mengedit, atau menghapus data keuangan, dan sistem akan menampilkan data keuangan yang diminta. Terakhir, admin akan melihat data keuangan tersebut, yang dapat dilihat pada gambar III.25.

d. *Activity diagram vendor*

1. *Activity diagram login*

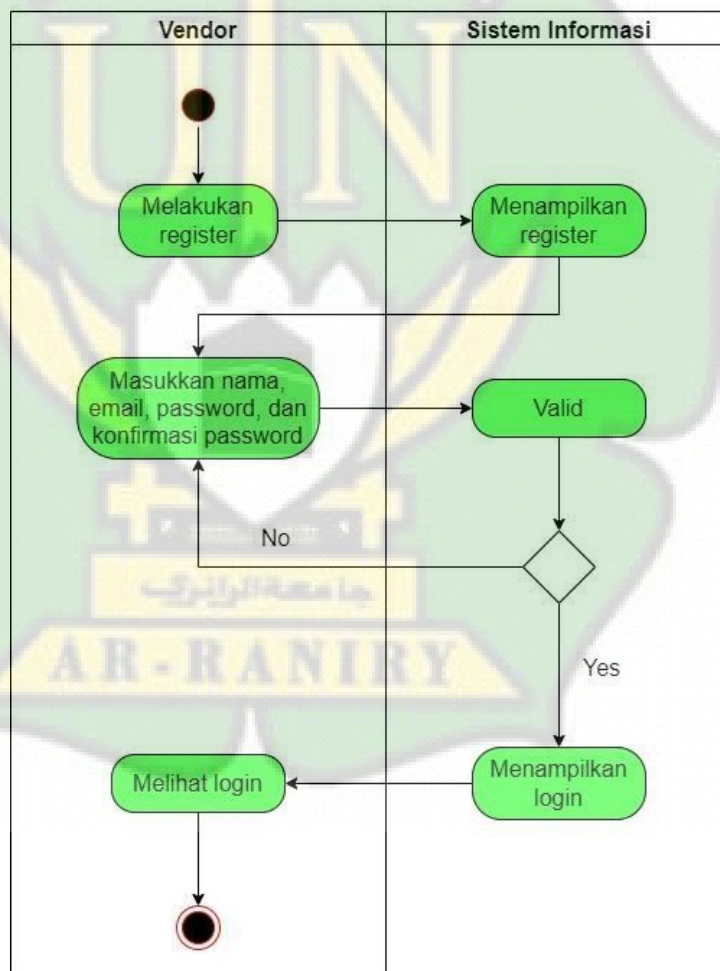


Gambar III.26 *Activity Diagram Vendor Login*

Diagram activity login vendor menjelaskan bahwa ketika vendor ingin masuk, sistem akan menampilkan menu *login*. Kemudian, vendor akan diminta untuk memasukkan *email* dan *password* mereka. Setelah itu, sistem akan memvalidasi data yang dimasukkan oleh vendor.

Jika data yang dimasukkan salah, vendor akan diminta untuk memasukkannya kembali. Namun, jika data yang dimasukkan benar, vendor akan diarahkan ke *dashboard*. Semua proses ini dapat dilihat pada gambar III.26.

2. Activity diagram register

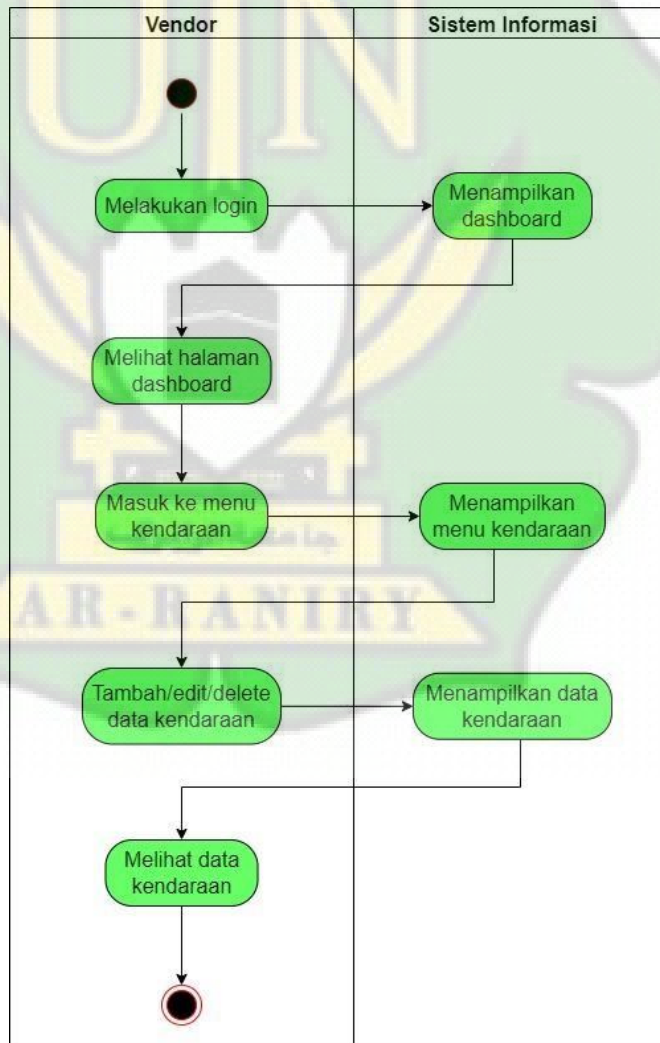


Gambar III.27 Activity Diagram Vendor Register

Diagram activity vendor registrasi menggambarkan proses registrasi vendor pada aplikasi. Ketika seorang vendor ingin melakukan registrasi, sistem akan menampilkan formulir registrasi. Vendor akan diminta untuk mengisi nama, *email*, *password*, dan konfirmasi *password*, kemudian sistem akan memvalidasi data yang diinput.

Jika data yang dimasukkan tidak sesuai, maka sistem akan meminta vendor untuk mengulang proses pengisian formulir registrasi. Namun, jika data yang dimasukkan benar, maka vendor akan diarahkan ke halaman *login*. Diagram ini diilustrasikan pada gambar III.27.

3. *Activity diagram* kendaraan

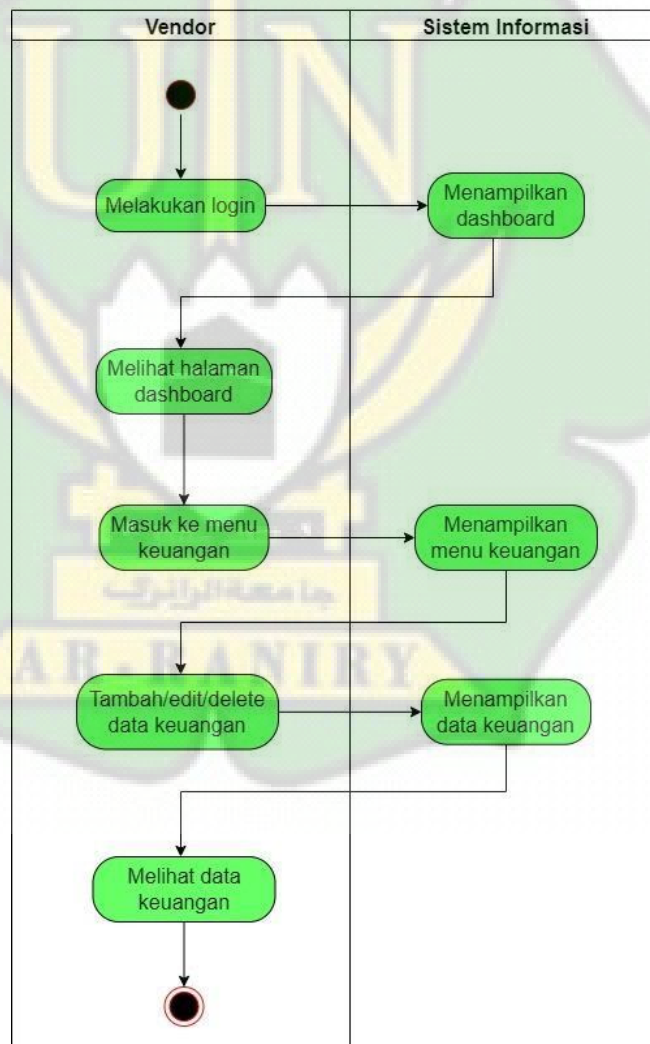


Gambar III.28 *Activity Diagram* Vendor Kendaraan

Diagram activity vendor kendaraan menjelaskan bahwa ketika vendor melakukan *login* maka sistem akan menampilkan *dashboard*. Vendor akan melihat halaman *dashboard* dan akan masuk ke menu kendaraan dan sistem akan menampilkan menu kendaraan.

Kemudian vendor akan diminta untuk menambahkan /mengedit /menghapus data kendaraan maka sistem akan menampilkan data kendaraan. Terakhir, vendor akan melihat data kendaraan, sehingga dapat dilihat pada gambar III.28.

4. *Activity diagram* keuangan

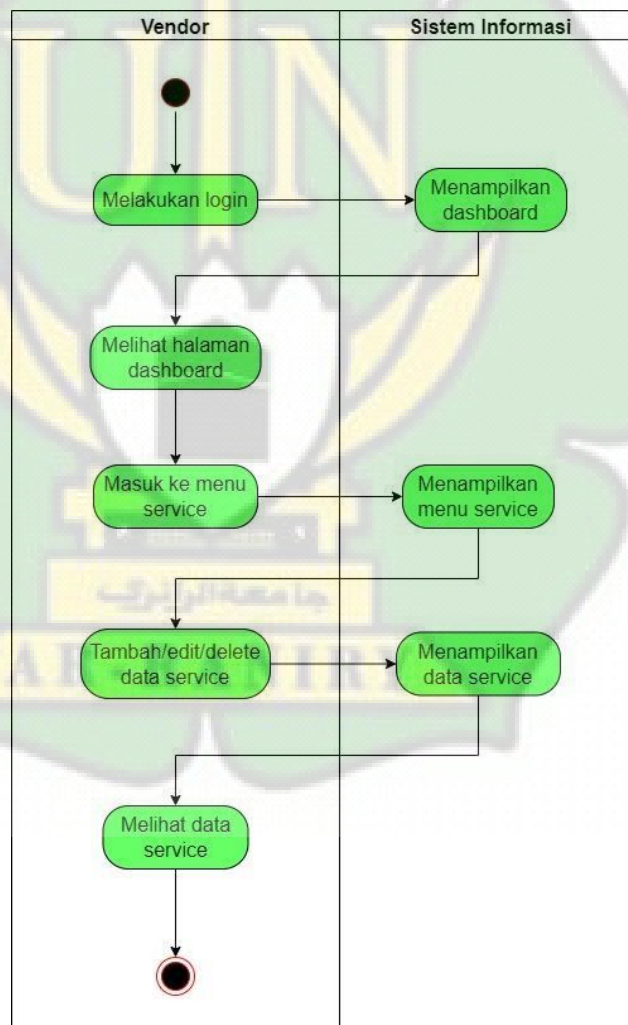


Gambar III.29 *Activity Diagram* Vendor Keuangan

Diagram activity vendor keuangan menjelaskan bahwa ketika vendor melakukan *login* maka sistem akan menampilkan *dashboard*. Vendor akan melihat halaman *dashboard* dan akan masuk ke menu keuangan dan sistem akan menampilkan menu keuangan.

Kemudian vendor akan diminta untuk menambahkan /mengedit /menghapus data keuangan maka sistem akan menampilkan data keuangan. Terakhir, sehingga vendor akan melihat data keuangan, dapat dilihat pada gambar III.29.

5. Activity diagram service



Gambar III.30 Activity Diagram Vendor Service

Diagram activity vendor menjelaskan bahwa ketika vendor melakukan *login* aplikasi maka sistem akan menampilkan *dashboard*. Vendor akan melihat halaman *dashboard* dan akan masuk ke menu , lalu sistem akan menampilkan menu.

Setelah itu, vendor akan diminta untuk menambahkan/mengedit/menghapus data maka sistem akan menampilkan data. Kemudian vendor akan melihat data, sehingga dapat dilihat pada gambar III.30.

6. *Activity diagram driver*

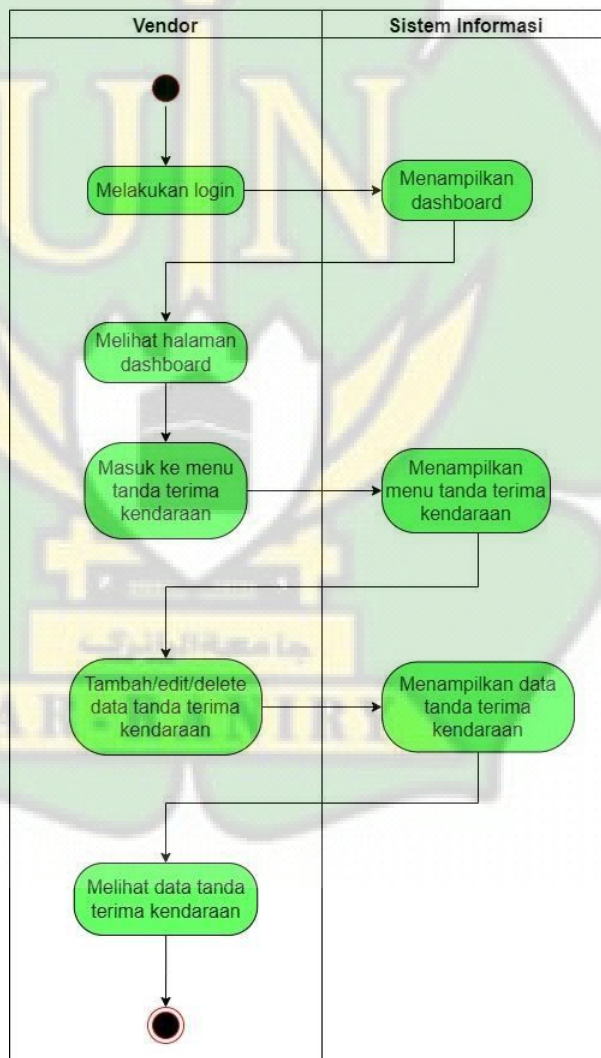


Gambar III.31 *Activity Diagram Vendor Driver*

Diagram activity vendor menjelaskan bahwa ketika vendor melakukan *login* aplikasi maka sistem akan menampilkan *dashboard*. Vendor akan melihat halaman *dashboard* dan akan masuk ke menu, lalu sistem akan menampilkan menu.

Setelah itu, vendor akan diminta untuk menambahkan /mengedit /menghapus data maka sistem akan menampilkan data. Kemudian vendor akan melihat data, sehingga dapat dilihat pada gambar III.31.

7. Activity diagram tanda terima kendaraan



Gambar III.32 Activity Diagram Vendor
Tanda Terima Kendaraan

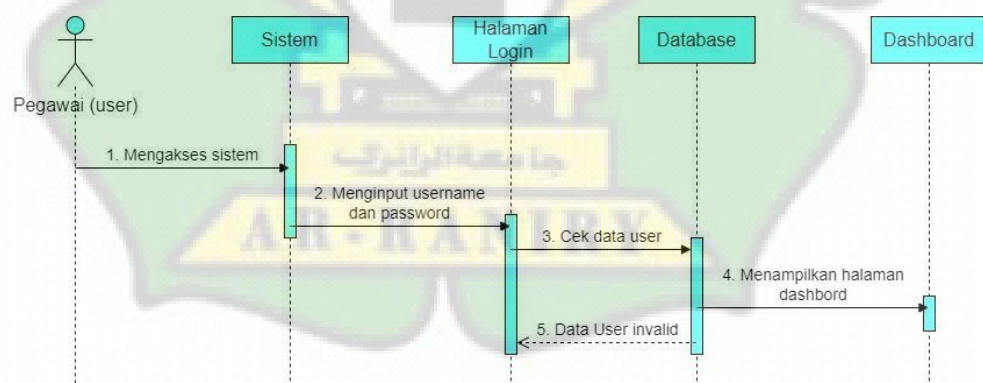
Diagram activity vendor tanda terima kendaraan menjelaskan bahwa ketika vendor melakukan *login* maka sistem akan menampilkan *dashboard*. Vendor akan melihat halaman *dashboard* dan akan masuk ke menu tanda terima kendaraan dan sistem akan menampilkan menu tanda terima kendaraan.

Kemudian vendor akan diminta untuk menambahkan /mengedit /menghapus data tanda terima kendaraan maka sistem akan menampilkan data tanda terima kendaraan. Terakhir, vendor akan melihat data tanda terima kendaraan, sehingga dapat dilihat pada gambar III.32.

C. Sequence diagram

Setelah mengidentifikasi aktivitas dan objek dalam Activity Diagram, dibuat *Sequence Diagram* untuk menggambarkan interaksi antar objek dalam waktu nyata selama eksekusi use case atau aktivitas tertentu pada sistem informasi manajemen transportasi berbasis web.

1. Sequence diagram login

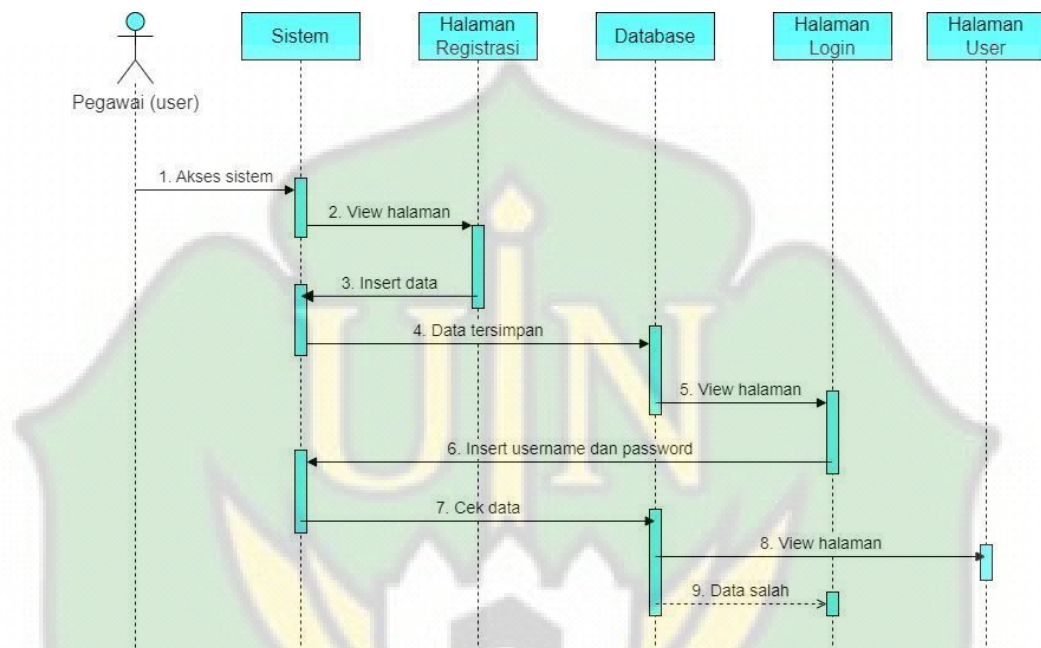


Gambar III.33 Sequence Diagram Login

User melakukan akses terhadap sistem, lalu user akan melakukan login sistem dengan menginput *username* dan *password*. Kemudian sistem akan melakukan cek data terhadap *database*. Jika data valid, maka sistem akan langsung mengakses ke halaman dashboard, jika data invalid, maka user akan

kembali ke halaman login untuk melakukan login kembali. Sehingga dapat dilihat pada gambar III.33.

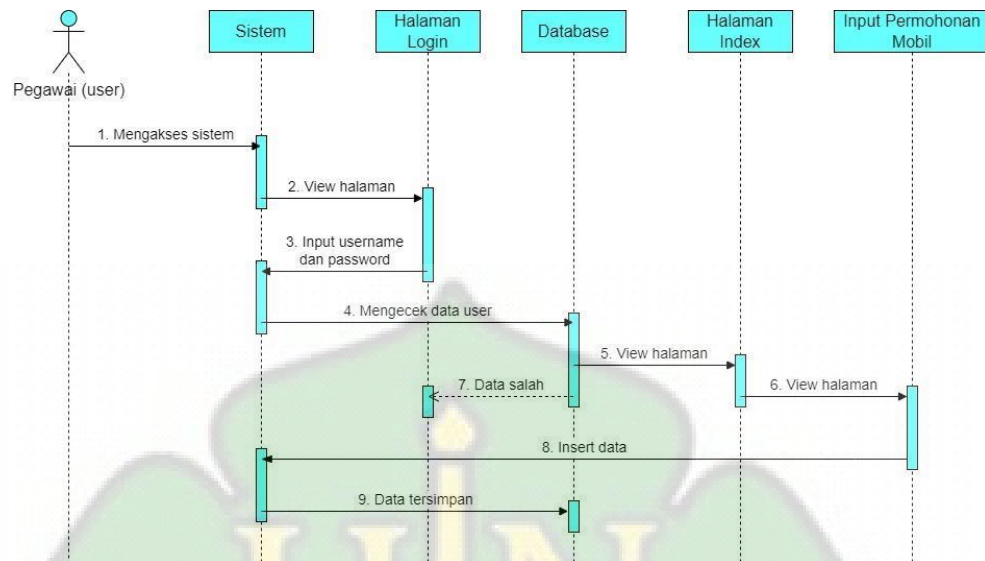
2. Sequence diagram register



Gambar III.34 Sequence Diagram Registrasi

User melakukan akses terhadap sistem, lalu user akan memilih halaman registrasi dan melakukan input terhadap sistem dengan mengisi nama, email, password, dan confirm password. Kemudian sistem menyimpan data ke database. Jika berhasil, maka akan otomatis mengakses ke halaman login, jika gagal, maka sistem meminta user melakukan input data yang benar kembali. Sehingga dapat dilihat pada gambar III.34.

3. Sequence diagram permohonan mobil

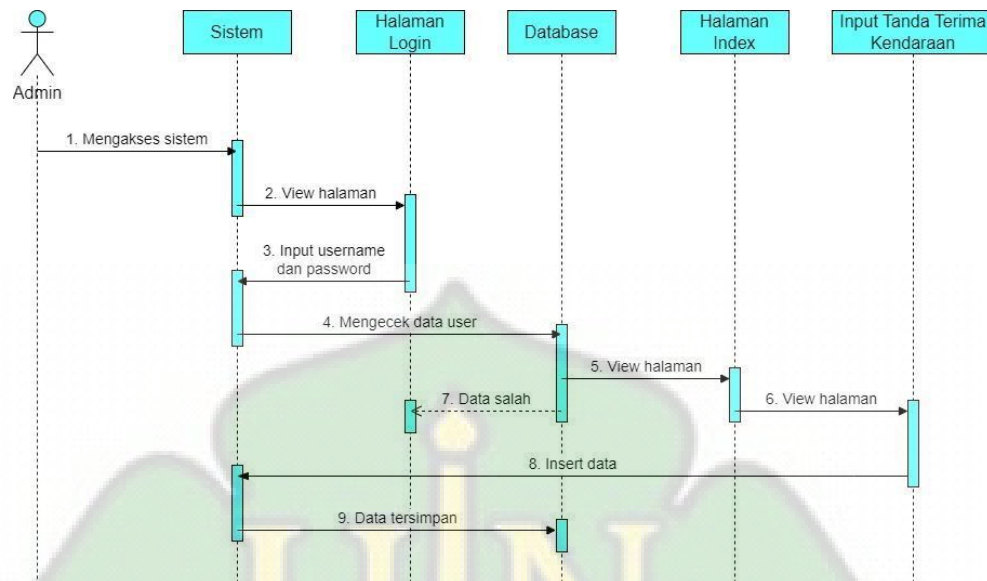


Gambar III.35 Sequence Diagram Permohonan Mobil

User melakukan akses terhadap sistem, lalu user akan melakukan login sistem. Kemudian sistem akan melakukan cek data terhadap database. Jika data benar, maka sistem akan langsung mengakses ke halaman index, jika data salah, maka user diminta untuk login kembali.

Setelah masuk ke halaman index, user akan memilih halaman input permohonan mobil. Di halaman input permohonan mobil, user akan melakukan pengisian data permohonan mobil. Lalu sistem menyimpan data ke database dan user bisa melihat data permohonan mobil di halaman index. Sehingga dapat dilihat pada gambar III.35.

4. *Sequence diagram* tanda terima kendaraan

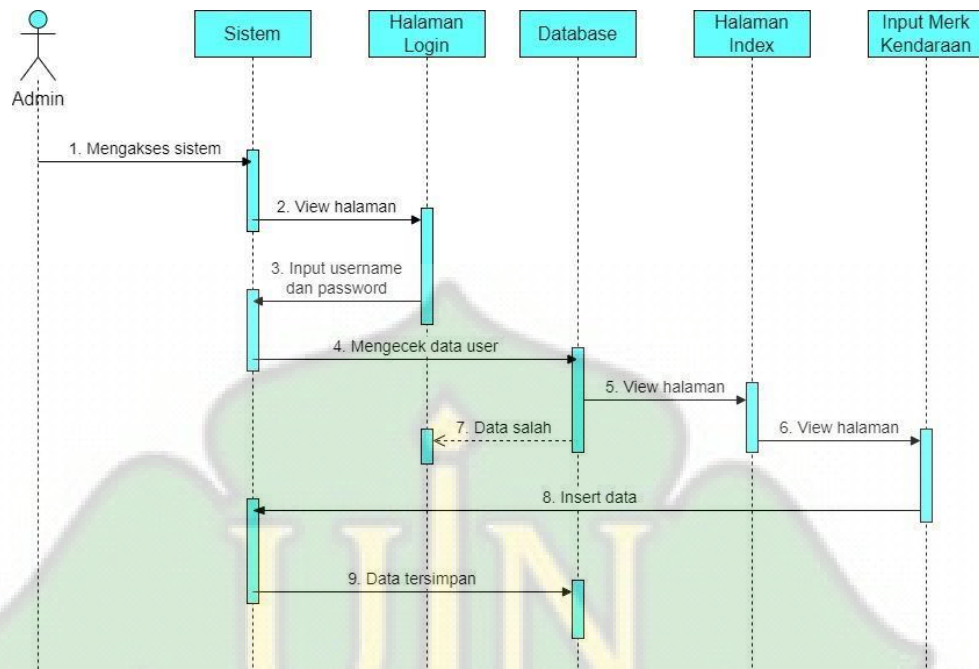


Gambar III.36 *Sequence Diagram* Tanda Terima Kendaraan

Admin melakukan akses terhadap sistem, lalu admin akan melakukan login sistem. Kemudian sistem akan melakukan cek data terhadap *database*. Jika data benar, maka sistem akan langsung mengakses ke halaman *index*, jika data salah, maka admin diminta untuk login kembali.

Setelah masuk ke halaman *index*, admin akan memilih halaman input tanda terima kendaraan. Di halaman input tanda terima kendaraan, admin akan melakukan pengisian data tanda terima kendaraan. Lalu sistem menyimpan data ke *database* dan admin bisa melihat data tanda terima kendaraan di halaman *index*. Sehingga dapat dilihat pada gambar III.36.

5. Sequence diagram merk kendaraan

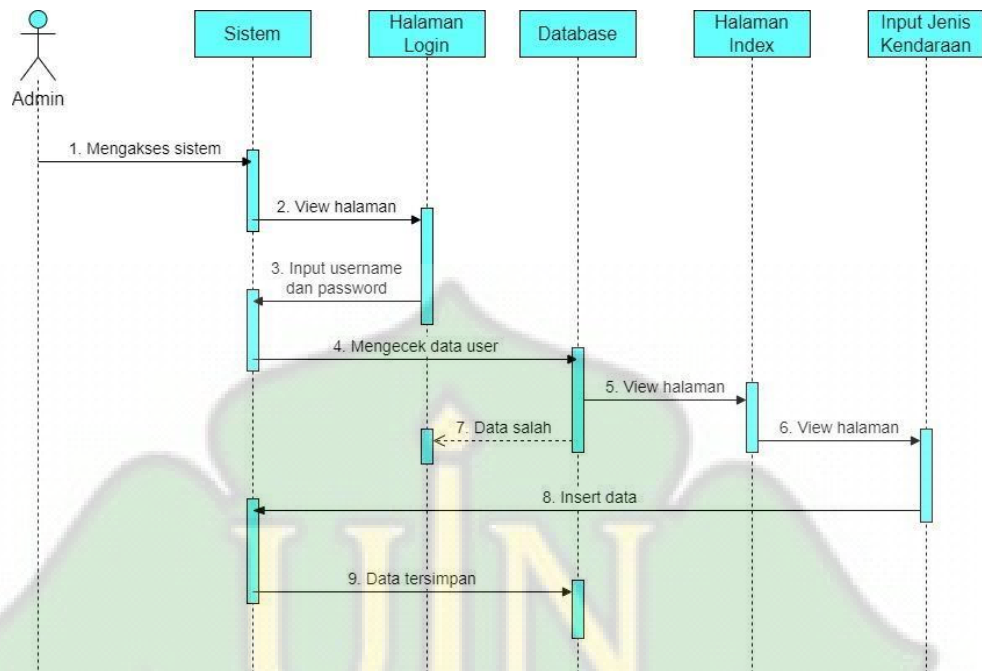


Gambar III.37 Sequence Diagram Merk Kendaraan

Admin melakukan akses terhadap sistem, lalu admin akan melakukan login sistem. Kemudian sistem akan melakukan cek data terhadap *database*. Jika data benar, maka sistem akan langsung mengakses ke halaman *index*, jika data salah, maka admin diminta untuk login kembali.

Setelah masuk ke halaman *index*, admin akan memilih halaman input merk kendaraan. Di halaman input merk kendaraan, admin akan melakukan pengisian data merk kendaraan. Lalu sistem menyimpan data ke *database* dan admin bisa melihat data merk kendaraan di halaman *index*. Sehingga dapat dilihat pada gambar III.37.

6. Sequence diagram jenis kendaraan

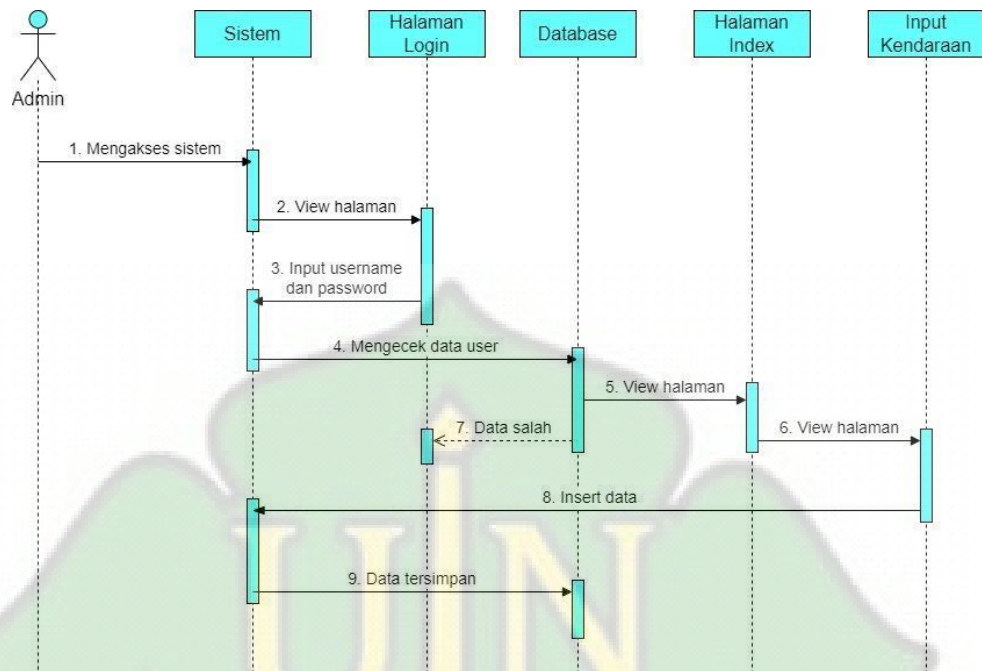


Gambar III.38 Sequence Diagram Jenis Kendaraan

Admin melakukan akses terhadap sistem, lalu admin akan melakukan login sistem. Kemudian sistem akan melakukan cek data terhadap *database*. Jika data benar, maka sistem akan langsung mengakses ke halaman *index*, jika data salah, maka admin diminta untuk login kembali.

Setelah masuk ke halaman *index*, admin akan memilih halaman input jenis kendaraan. Di halaman input merk kendaraan, admin akan melakukan pengisian data jenis kendaraan. Lalu sistem menyimpan data ke *database* dan admin bisa melihat data jenis kendaraan di halaman *index*. Sehingga dapat dilihat pada gambar III.38.

7. Sequence diagram kendaraan

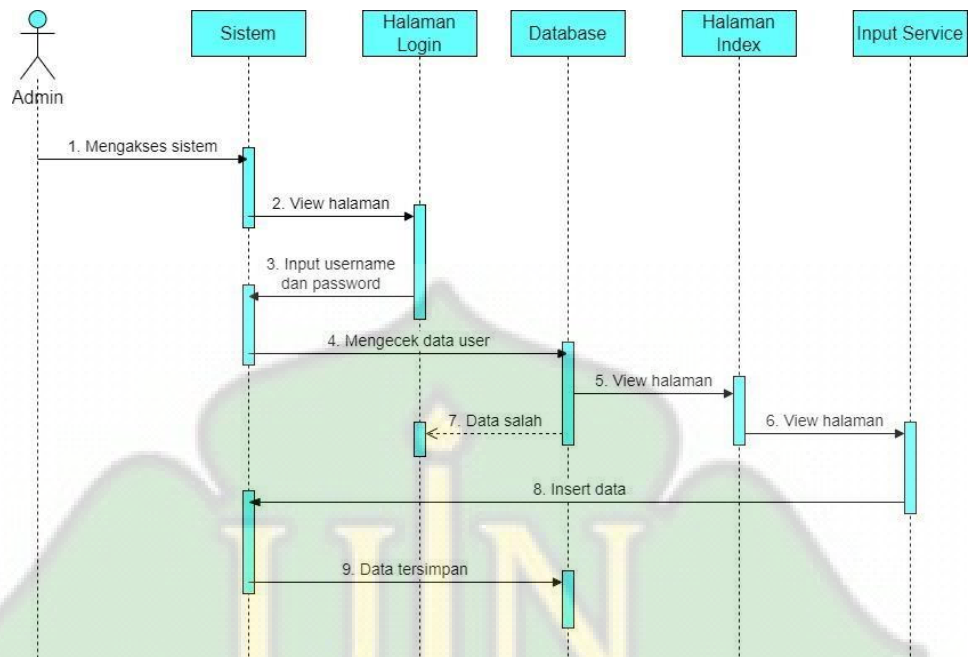


Gambar III.39 Sequence Diagram Kendaraan

Admin melakukan akses terhadap sistem, lalu admin akan melakukan login sistem. Kemudian sistem akan melakukan cek data terhadap *database*. Jika data benar, maka sistem akan langsung mengakses ke halaman *index*, jika data salah, maka admin diminta untuk login kembali.

Setelah masuk ke halaman *index*, admin akan memilih halaman input kendaraan. Di halaman input kendaraan, admin akan melakukan pengisian data kendaraan. Lalu sistem menyimpan data ke *database* dan admin bisa melihat data kendaraan di halaman *index*. Sehingga dapat dilihat pada gambar III.39.

8. Sequence diagram service

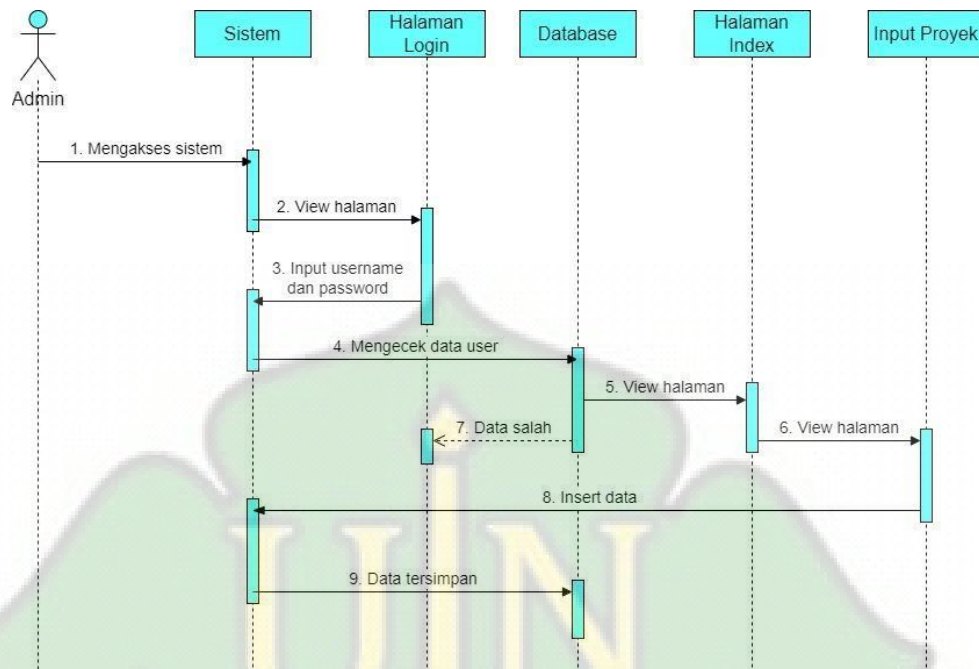


Gambar III.40 Sequence Diagram Service

Admin melakukan akses terhadap sistem, lalu admin akan melakukan login sistem. Kemudian sistem akan melakukan cek data terhadap *database*. Jika data benar, maka sistem akan langsung mengakses ke halaman *index*, jika data salah, maka admin diminta untuk login kembali.

Setelah masuk ke halaman *index*, admin akan memilih halaman input *service*. Di halaman input *service*, admin akan melakukan pengisian data *service*. Lalu sistem menyimpan data ke *database* dan admin bisa melihat data *service* di halaman *index*. Sehingga dapat dilihat pada gambar III.40.

9. Sequence diagram proyek

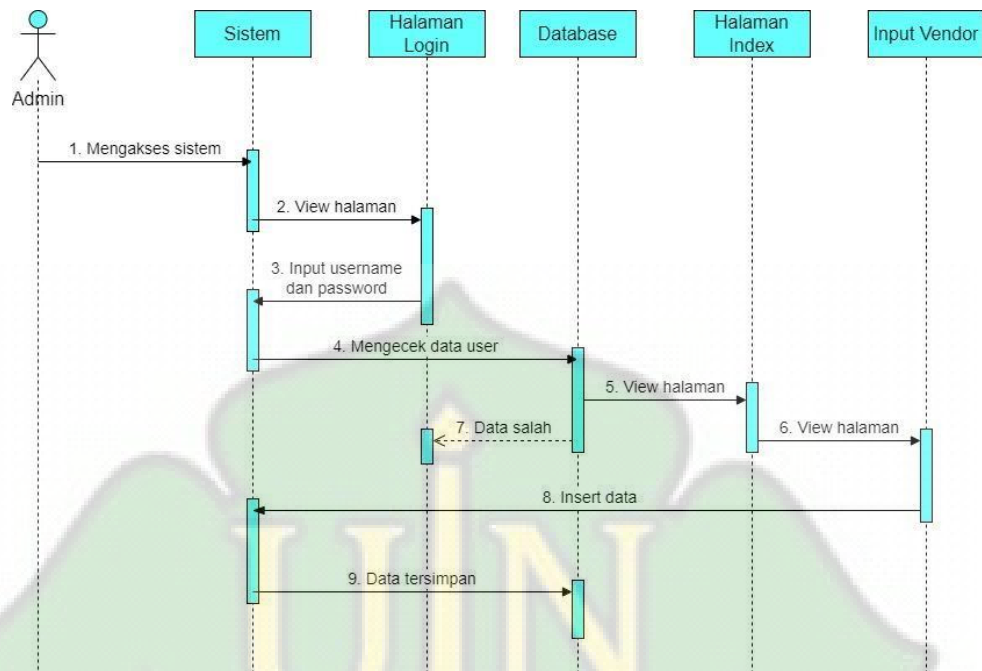


Gambar III.41 Sequence Diagram Proyek

Admin melakukan akses terhadap sistem, lalu admin akan melakukan login sistem. Kemudian sistem akan melakukan cek data terhadap *database*. Jika data benar, maka sistem akan langsung mengakses ke halaman *index*, jika data salah, maka admin diminta untuk login kembali.

Setelah masuk ke halaman *index*, admin akan memilih halaman input proyek. Di halaman input proyek, admin akan melakukan pengisian data proyek. Lalu sistem menyimpan data ke *database* dan admin bisa melihat data proyek di halaman *index*. Sehingga dapat dilihat pada gambar III.41.

10. Sequence diagram vendor

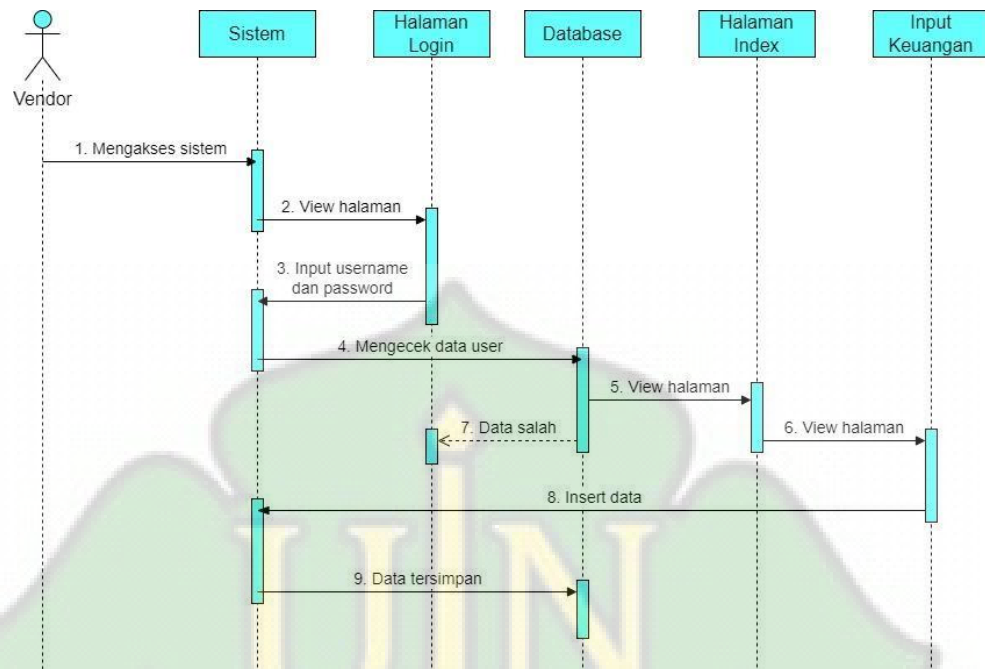


Gambar III.42 Sequence Diagram Vendor

Admin melakukan akses terhadap sistem, lalu admin akan melakukan login sistem. Kemudian sistem akan melakukan cek data terhadap *database*. Jika data benar, maka sistem akan langsung mengakses ke halaman *index*, jika data salah, maka admin diminta untuk login kembali.

Setelah masuk ke halaman *index*, admin akan memilih halaman input vendor. Di halaman input vendor, admin akan melakukan pengisian data vendor. Lalu sistem menyimpan data ke *database* dan admin bisa melihat data vendor di halaman *index*. Sehingga dapat dilihat pada gambar III.42.

11. Sequence diagram keuangan

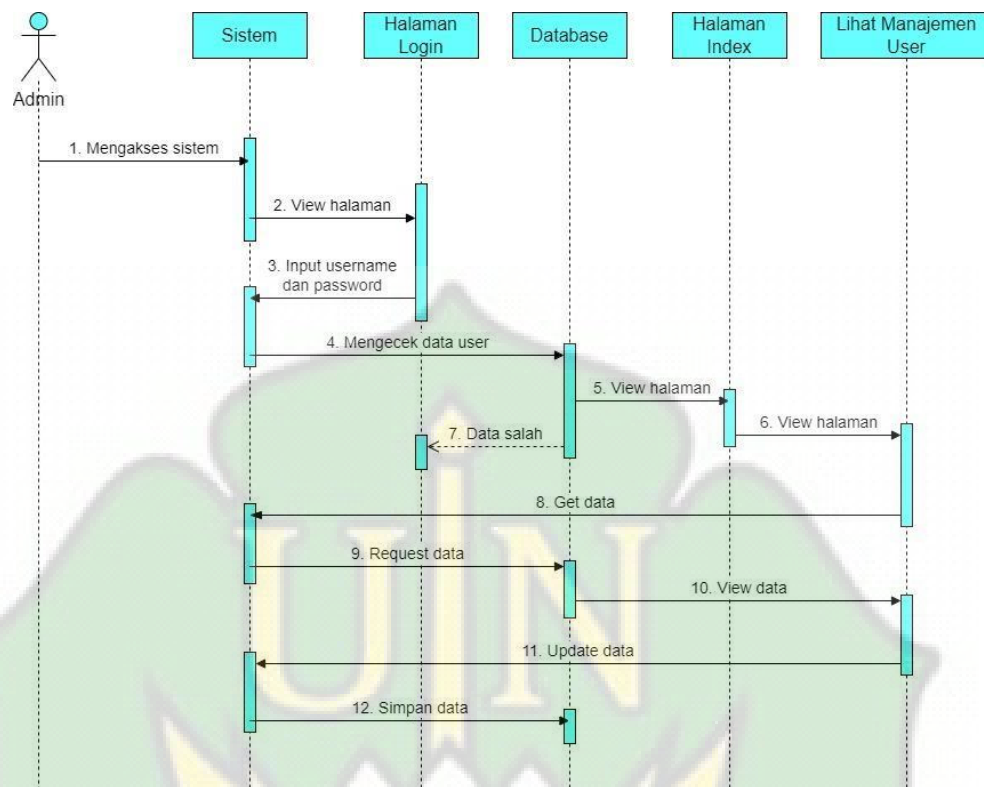


Gambar III.43 Sequence Diagram Keuangan

Vendor melakukan akses terhadap sistem, lalu vendor akan melakukan login sistem. Kemudian sistem akan melakukan cek data terhadap *database*. Jika data benar, maka sistem akan langsung mengakses ke halaman *index*, jika data salah, maka vendor diminta untuk login kembali.

Setelah masuk ke halaman *index*, admin akan memilih halaman input keuangan. Di halaman input keuangan, vendor akan melakukan pengisian data keuangan. Lalu sistem menyimpan data ke *database* dan vendor bisa melihat data keuangan di halaman *index*. Sehingga dapat dilihat pada gambar III.43.

12. Sequence diagram manajemen user



Gambar III.44 Sequence Diagram Manajemen User

Admin melakukan akses terhadap sistem, lalu vendor akan melakukan login sistem. Kemudian sistem akan melakukan cek data terhadap *database*. Jika data benar, maka sistem akan langsung mengakses ke halaman *index*, jika data salah, maka admin diminta untuk login kembali.

Setelah masuk ke halaman *index*, admin akan memilih halaman input manajemen *user*. Di halaman lihat manajemen *user*, admin dapat melihat data manajemen *user* yang mana *user* telah melakukan login. Serta admin dapat mengedit level *user* manajemen *user* di halaman tersebut dengan menekan *button* edit.

Setelah menekan *button* edit, admin akan masuk ke halaman edit dan bisa merubah level *user* manajemen *user* jika ada yang salah dalam proses penginputan data *user* yang telah login. Setelah merubah data, sistem menyimpan data ke *database*, dan admin bisa melihat data manajemen *user* yang telah di edit di halaman *index*. Sehingga dapat dilihat pada gambar III.44.

D. Skema Basis Data

1. Tabel *Service*

Nama : tabel *service*

Deskripsi isi : untuk menyimpan data master *service* mobil

Primary key : *id_service*

Foreign key : *id_kendaraan*, *id_vendor*

Kolom	Tipe data	Null	Key	Relasi ke
id	Int (20)	No	PK	
id_kendaraan	Int (10)	No	FK	Tabel kendaraan → id
tgl_service_mobil	Date	No		
perbaiki	Varchar (255)	No		
biaya_service	Varchar (255)	No		
id_vendor	Int (10)	No	FK	Tabel vendor → id

2. Tabel Merk Kendaraan

Nama : tabel merkkendaraan

Deskripsi isi : untuk menyimpan data merk mobil

Primary key : *id_merk*

Foreign key : -

Kolom	Tipe data	Null	Key	Relasi ke
id	Int (20)	No	PK	Tabel jenis kendaraan → id_merk
merk	Varchar (255)	No		

3. Tabel Jenis Kendaraan

Nama : tabel jeniskendaraan

Deskripsi isi : untuk menyimpan data jenis mobil

Primary key : id_jenis

Foreign key : id_merk

Kolom	Tipe data	Null	Key	Relasi ke
id	Int (10)	No	PK	Tabel kendaraan → id_jenis
				Tabel permohonan mobil → id_jenis
id_merk	Int (10)	No	FK	Tabel merk kendaraan → id

4. Tabel Kendaraan

Nama : tabel kendaraan

Deskripsi isi : untuk menyimpan data kendaraan

Primary key : id_kendaraan

Foreign key : id_jenis dan id_vendor

Kolom	Tipe data	Null	Key	Relasi ke
id	Int (10)	No	PK	Tabel <i>service</i> → id_kendaraan
				Tabel permohonan mobil → no_polisi
platmobil	Varchar (8)	No		
id_jenis	Int (10)	No	FK	Tabel jenis kendaraan → id
id_vendor	Int (10)	No	FK	Tabel vendor → id
bbm	Varchar (20)	No		

Harga	Varchar (25)	No		
Tahun	Int (4)	No		
Pajak	Varchar (20)	No		
Status	Varchar (10)	No		

5. Tabel *User*

Nama : tabel *user*

Deskripsi isi : untuk menyimpan data *user* (pegawai)

Primary key : *id_user*

Foreign key : *user*

Kolom	Tipe data	Null	Key	Relasi ke
id	Int (20)	No	PK	
name	Varchar (255)	No	FK	Tabel permohonan mobil → nama_pemohon
level	Varchar (255)	No		
homebase	Int (10)	No		
email	Varchar (255)	No		
email_verified_at	timestamp	Yes		
password	Varchar (255)	No		
Remember_token	Varchar (100)	Yes		

6. Tabel Permohonan Mobil

Nama : tabel permohonan mobil

Deskripsi isi : untuk menyimpan informasi terkait permohonan
peminjaman kendaraan dari *user*

Primary key : id_permohonanmobil

Foreign key : nama_pemohon, id_jenis, id_proyek, no_polisi, driver,
no_telp_driver, id_vendor, dan tandaterima.

Kolom	Tipe data	Null	Key	Relasi ke
id	Int (20)	No	PK	Tabel tanda terima → id_permohonanmobil
nama_pemohon	Varchar (255)	No	FK	Tabel user → name
id_jenis	Int (10)	No	FK	Tabel jenis kendaraan → id
Jumlah_orang	Varchar (255)	No		
id_proyek	Int (10)	No	FK	Tabel proyek → id
Tujuan	Varchar (255)	No		
barang	Varchar (255)	Yes		
Tgl_berangkat	Date	Yes		
Tgl_pulang	Date	Yes		
Status	Varchar (12)	No		
Keterangan	Varchar (255)	Yes		
no_polisi	Int (10)	Yes	FK	Tabel kendaraan → id
Driver	Int (20)	Yes	FK	Tabel driver → id
No_telp_driver	Int (10)	Yes	FK	Tabel driver → id

Id_vendor	Int (20)	No	FK	Tabel vendor → id
Approve_vendor	Varchar (25)	Yes		
Tandaterima	Varchar (25)	Yes	FK	Tabel tanda terima → no_tandaterima
Invoice	Varchar (25)	Yes		

7. Tabel Driver

Nama : tabel driver

Deskripsi isi : untuk menyimpan data driver

Primary key : id_driver

Foreign key : id_vendor

Kolom	Tipe data	Null	Key	Relasi ke
id	Int (20)	No	PK	Tabel permohnan mobil → driver
				Tabel permohnan mobil → no_telp_driver
nama	Varchar (255)	No		
no_telp	Int (15)	No		
Alamat	Varchar (255)	No		
Id_vendor	Int (10)	No	FK	Tabel vendor → id

8. Tabel Proyek

Nama : tabel proyek

Deskripsi isi : untuk menyimpan data-data proyek yang memerlukan transportasi untuk melaksanakan tugas-tugasnya

Primary key : id_proyek

Foreign key : -

Kolom	Tipe data	Null	Key	Relasi ke
id	Int (20)	No	PK	Tabel permohonan mobil → id_proyek
kode_proyek	Varchar (20)	No		
Nama_proyek	Varchar (255)	No		
klien	Varchar (100)	No		

9. Tabel Vendor

Nama : tabel vendor

Deskripsi isi : untuk menyimpan informasi tentang vendor yang terlibat dengan SIMTRANS

Primary key : id_vendor

Foreign key : -

Kolom	Tipe data	Null	Key	Relasi ke
id	Int (20)	No	PK	Tabel kendaraan → id_vendor
				Tabel keuangan → id_vendor
				Tabel tanda terima → id_vendor
				Tabel driver → id_vendor
				Tabel service → id_vendor

				Tabel permohonan mobil → id_vendor
Vendor	Varchar (100)	No		
Direktur	Varchar (100)	No		
Alamat	Varchar (255)	No		
No_telp	Varchar (255)	No		

10. Tabel Keuangan

Nama : tabel keuangan

Deskripsi isi : untuk melacak semua transaksi dan catatan keuangan yang berkaitan dengan operasi transportasi perusahaan atau vendor

Primary key : id_keuangan

Foreign key : no_invoice, id_vendor, dan id_tt

Kolom	Tipe data	Null	Key	Relasi ke
id	Int (20)	No	PK	
No_invoice	Varchar (25)	No	FK	Tabel tanda terima → invoice
Id_vendor	Int (10)	No	FK	Tabel vendor → id
Id_tt	Int (10)	No	FK	Tabel tanda terima → id
Jml_hari	Int (10)	No		
Harga	Varchar (10)	No		
Total_harga	Varchar (10)	No		

Status	Varchar (10)	No		
--------	-----------------	----	--	--

11. Tabel Tanda Terima

Nama : tabel tanda terima

Deskripsi isi : untuk melacak tanda terima atau invoice yang dikeluarkan kepada pemohon kendaraan, pihak vendor, atau pihak lain yang terkait dalam pengelolaan transportasi

Primary key : id_tandaterima

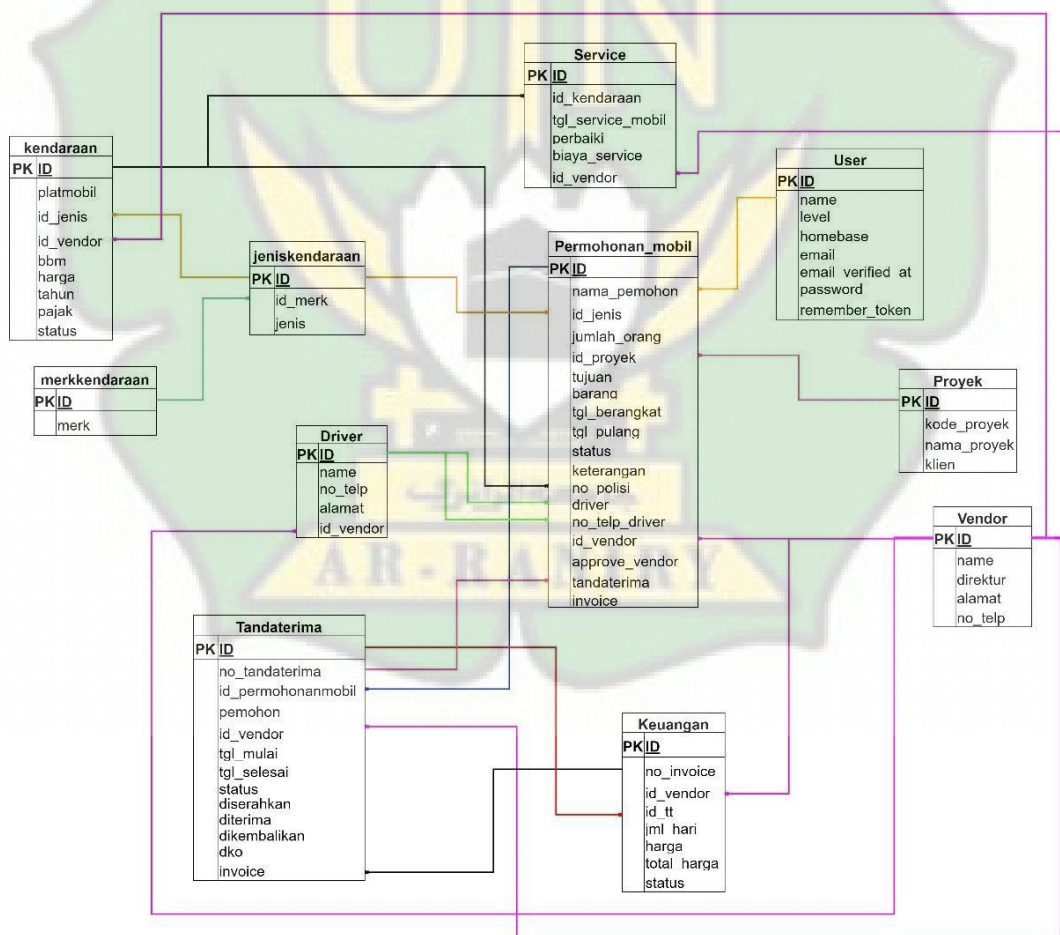
Foreign key : no_tandaterima, id_permohonanmobil, id_vendor, invoice

Kolom	Tipe data	Null	Key	Relasi ke
id	Int (10)	No	PK	Tabel keuangan → id_tt
No_tandaterima	Varchar (25)	No	FK	Tabel permohonan mobil → tandaterima
Id_permohonan mobil	Int (20)	No	FK	Tabel permohonan mobil → id
Pemohon	Varchar (25)	No		
Id_vendor	Int (10)	No	FK	Tabel vendor → id
Tgl_mulai	Date	Yes		
Tgl_selesai	Date	Yes		
Status	Varchar (25)	No		
Diserahkan	Varchar (25)	Yes		
Diterima	Varchar (25)	Yes		
Dikembalikan	Varchar (25)	Yes		

Dko	Varchar (25)	Yes		
Invoice	Varchar (25)	Yes	FK	Tabel keuangan → no_invoice

E. Diagram Database

Diagram *database* atau diagram tabel-tabel pada *database* menggambarkan hubungan atau relasi antar tabel yang ada. Relasi antar tabel tersebut menunjukkan hubungan data yang terjadi pada tabel-tabel yang terdapat di dalam *database*. Untuk *database* SIMTRANS ini, relasi antar tabel yang terjadi dapat dilihat pada gambar III.33.

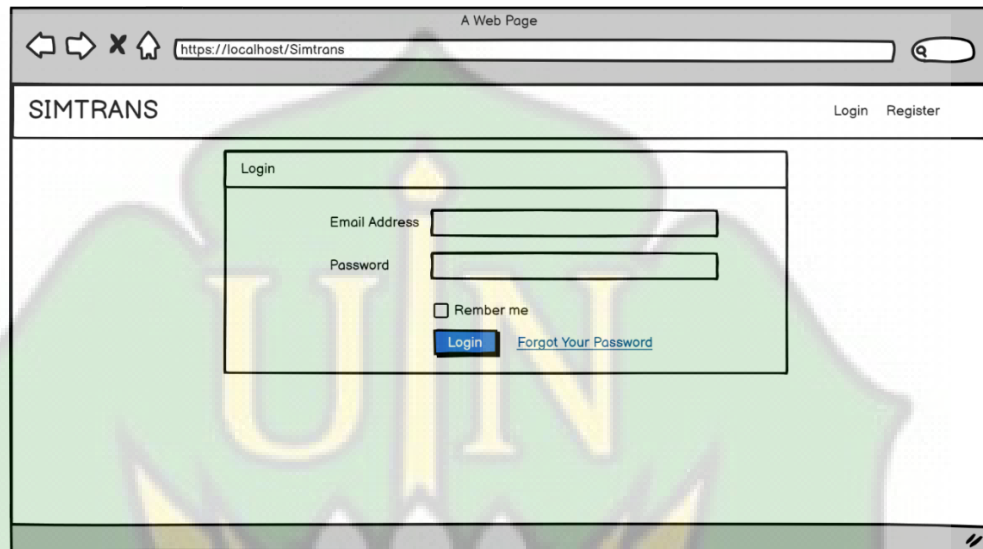


Gambar III.45 Diagram Database

2. Rancangan *User Interface*

UI atau antarmuka pengguna adalah desain yang fokus pada tampilan yang menarik dan pemilihan warna yang tepat serta elemen-elemen lainnya yang membuat situs *web* e-commerce lebih menarik. (Rizki, 2019).

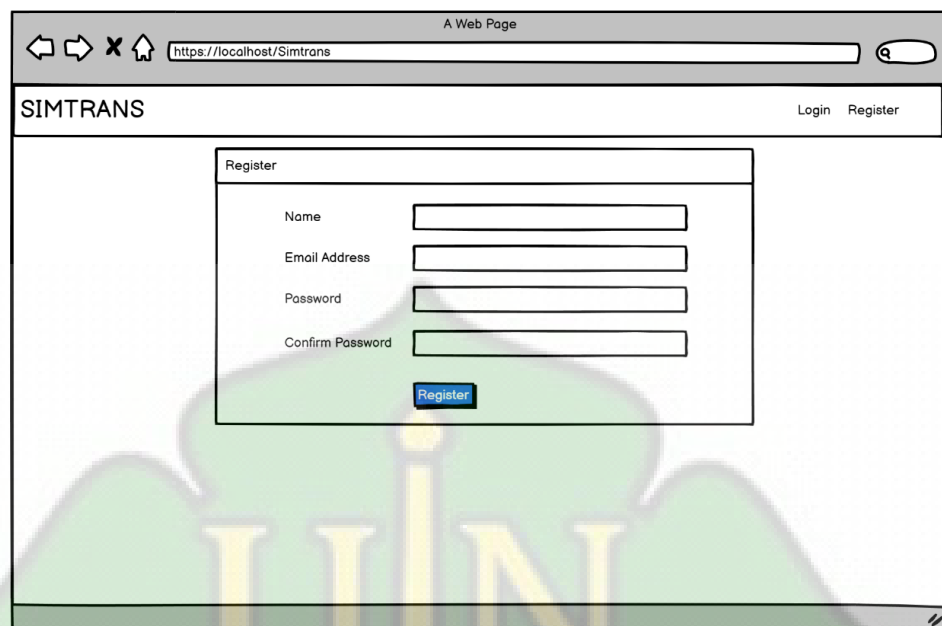
a. Tampilan *Login*



Gambar III.46 Tampilan *Login*

Desain tampilan *login* dirancang dengan menampilkan dua tombol, yaitu tombol *login* dan *forgot your password*, serta dua kolom teks yang dapat diedit, yaitu kolom *email* dan *password*. Perancangan tampilan *login* ini dapat dilihat pada gambar III.46.

b. Tampilan Register



The image shows a web browser window with the address bar displaying 'https://localhost/Simtrans'. The page title is 'SIMTRANS'. In the top right corner, there are links for 'Login' and 'Register'. The main content area features a 'Register' form with the following fields:

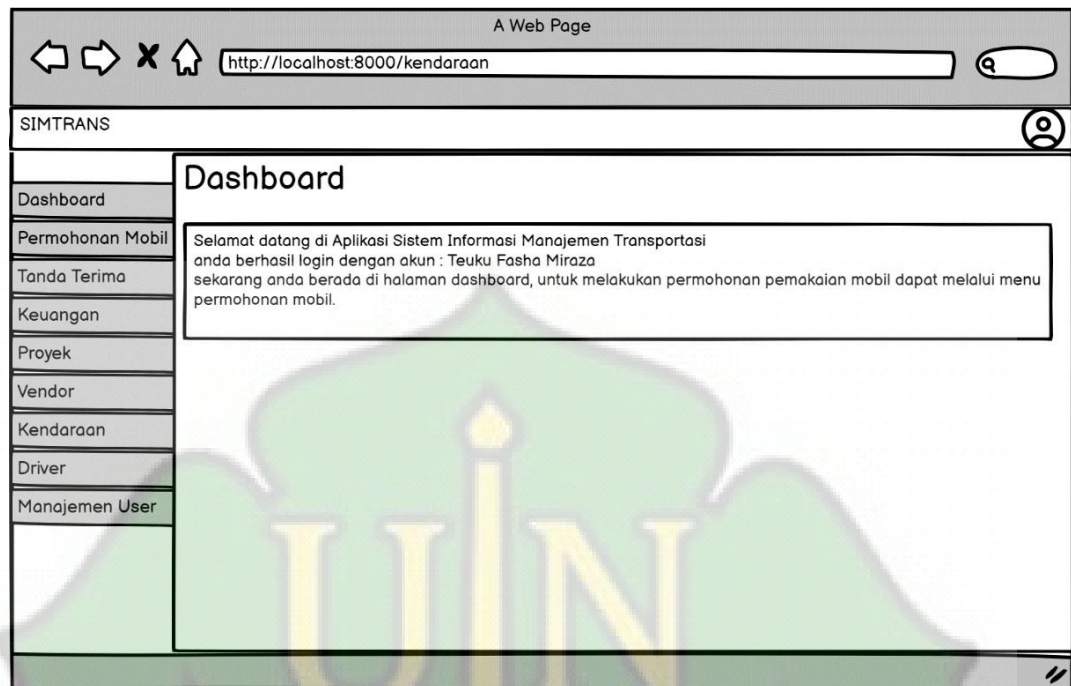
- Name
- Email Address
- Password
- Confirm Password

Below these fields is a blue 'Register' button.

Gambar III.47 Tampilan Register

Desain tampilan untuk registrasi menampilkan satu tombol (register) dan empat kolom teks untuk diisi, yaitu kolom nama, *email*, *password*, dan konfirmasi *password*. Hal ini terlihat pada gambar III.47.

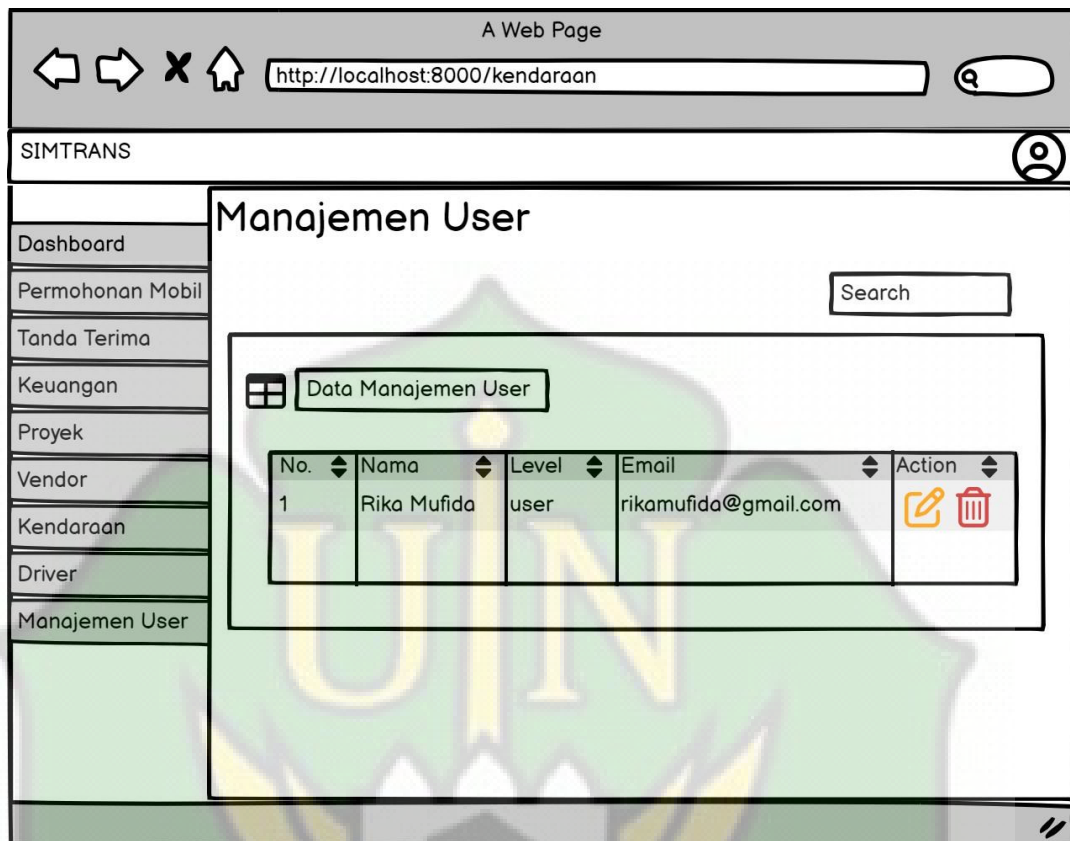
c. Tampilan *Dashboard*



Gambar III.48 Tampilan Dashboard

Tampilan *dashboard* dirancang dengan menampilkan halaman utama yang memberikan gambaran informasi terkini dan penting mengenai aktivitas transportasi yang terjadi pada sistem. Sehingga tampilan dashbord dapat dilihat pada gambar III.48.

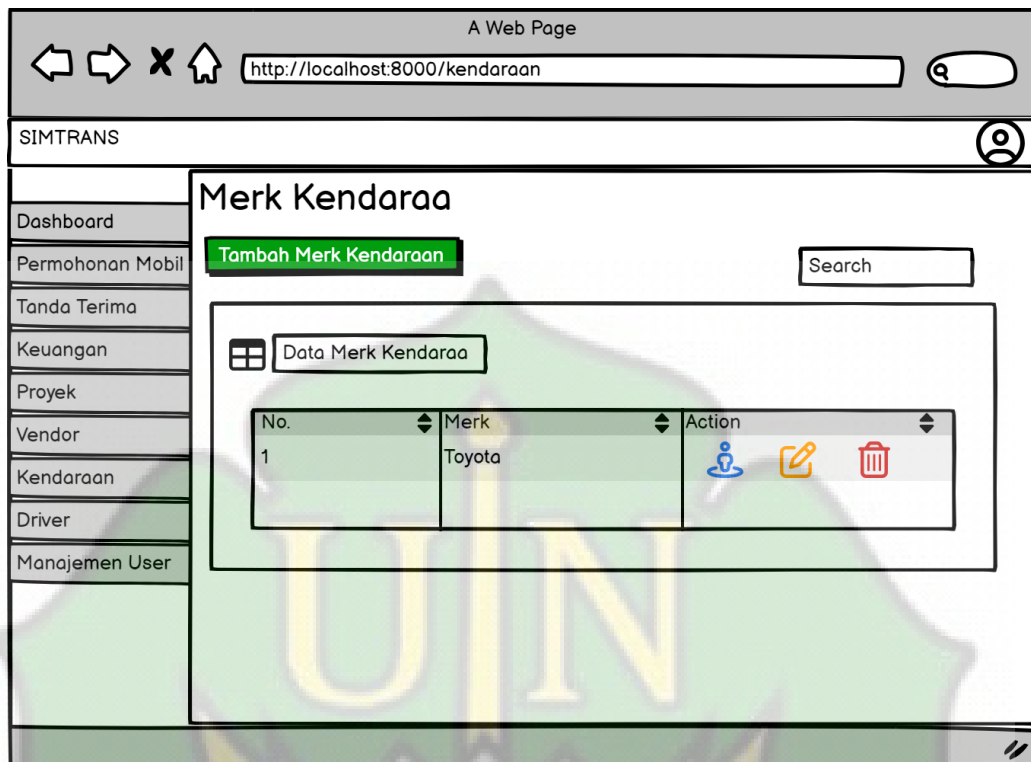
d. Tampilan Manajemen *User*



Gambar III.49 Tampilan Manajemen *User*

Tampilan manajemen *user* dirancang dengan menggunakan 2 *button* (edit dan hapus), pada halaman ini menyimpan data-data *user* yang telah *login* dan hanya admin yang bisa mengubah hak akses pada para *user*. Di tabel data manajemen *user* ada terdapat nama, level, email dan actions. Sehingga tampilan manajemen *user* dapat dilihat pada gambar III.49.

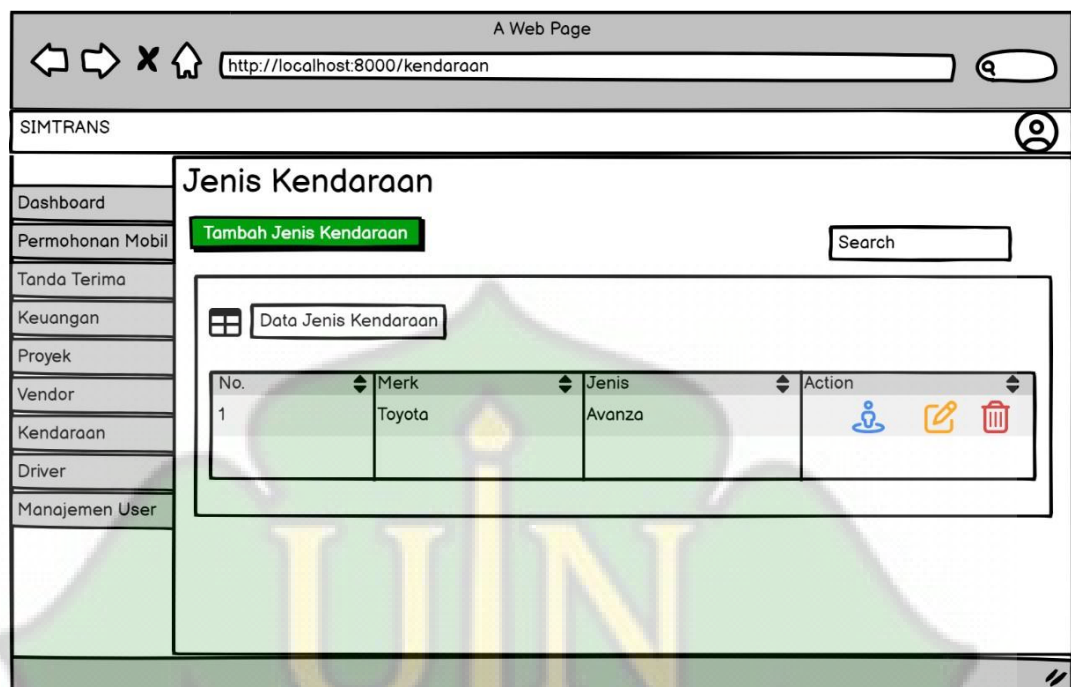
e. Tampilan Merk Kendaraan



Gambar III.50 Tampilan Merk Kendaraan

Perancangan tampilan merk kendaraan terdiri dari 4 *button*, yaitu tambah kendaraan, *view*, edit, dan hapus. Button tambah kendaraan berfungsi untuk memasukkan data merk kendaraan. Pada tabel data merk kendaraan, terdapat kolom merk kendaraan dan *actions*. Rancangan tampilan ini dapat dilihat pada gambar III.50.

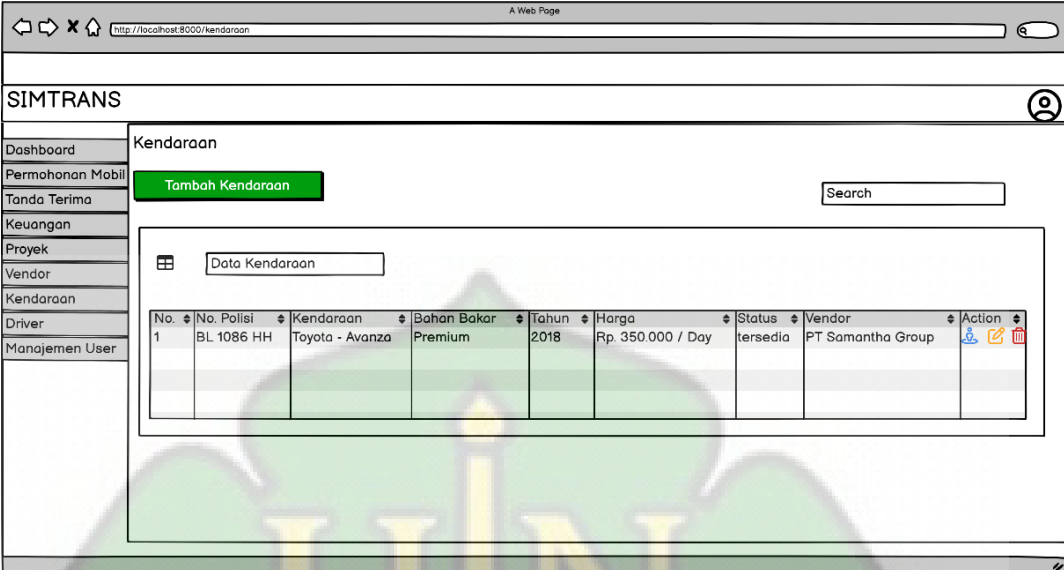
f. Tampilan Jenis Kendaraan



Gambar III.51 Tampilan Jenis Kendaraan

Perancangan tampilan jenis kendaraan terdiri dari 4 *button*, yaitu tambah kendaraan, view, edit, dan hapus. Button tambah jenis kendaraan berfungsi untuk memasukkan data jenis kendaraan. Pada tabel data jenis kendaraan, terdapat kolom merk kendaraan, masa pajak dan *actions*. Rancangan tampilan ini dapat dilihat pada gambar III.51.

g. Tampilan Data Kendaraan

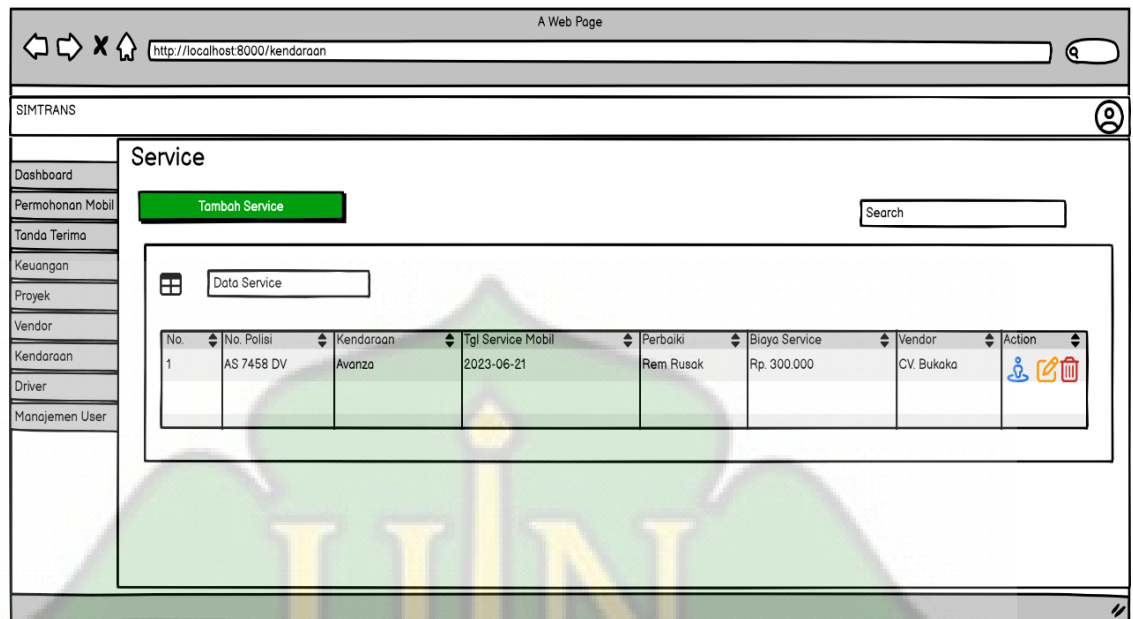


No.	No. Polisi	Kendaraan	Bahan Bakar	Tahun	Harga	Status	Vendor	Action
1	BL 1086 HH	Toyota - Avanza	Premium	2018	Rp. 350.000 / Day	tersedia	PT Samantha Group	  

Gambar III.52 Tampilan Data Kendaraan

Perancangan tampilan kendaraan terdiri dari 4 *button*, yaitu tambah kendaraan, view, edit, dan hapus. Button tambah kendaraan berfungsi untuk memasukkan data merk dan jenis kendaraan. Pada tabel data kendaraan, terdapat kolom nomor polisi, kendaraan, bahan bakar, tahun, harga, status, vendor, dan *actions*. Rancangan tampilan ini dapat dilihat pada gambar III.52.

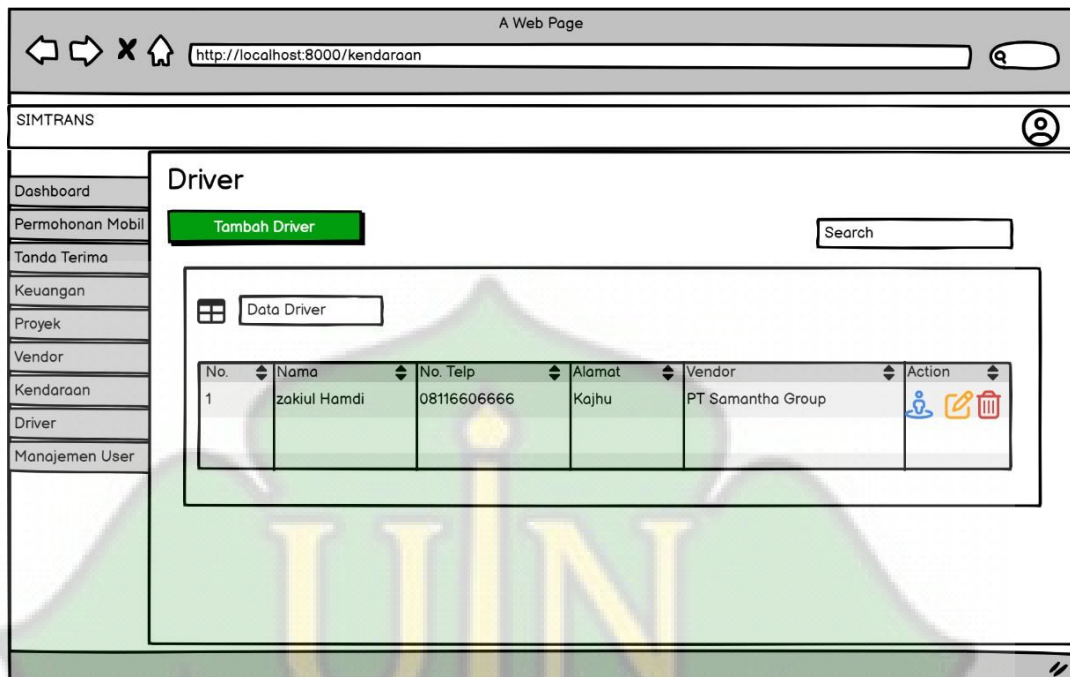
h. Tampilan Service



Gambar III.53 Tampilan Service

Tampilan dirancang dengan 4 *button* (tambah, *view*, edit, dan hapus), pada *button* tambah berfungsi untuk mengisi data pada kendaraan. Di tabel data ada terdapat no. polisi, kendaraan, tanggal, perbaikan, biaya, vendor, dan *actions*. *Service* dapat dilihat pada gambar III.53.

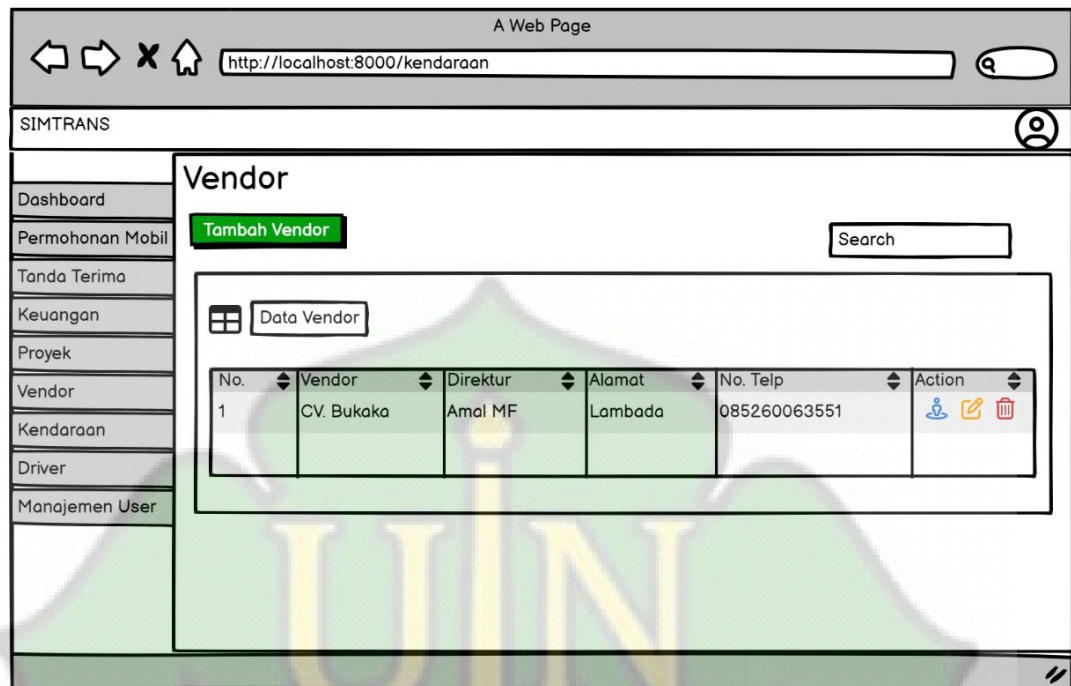
i. Tampilan *Driver*



Gambar III.54 Tampilan *Driver*

Tampilan untuk bagian didesain dengan 4 tombol (tambah *driver*, *view*, edit, dan hapus), di mana tombol tambah *driver* berfungsi untuk mengisi data *driver*. Pada tabel data terdapat kolom untuk nama, nomor telepon, alamat, vendor, dan *action*. Gambaran perancangan antarmuka ini dapat dilihat pada gambar III.54.

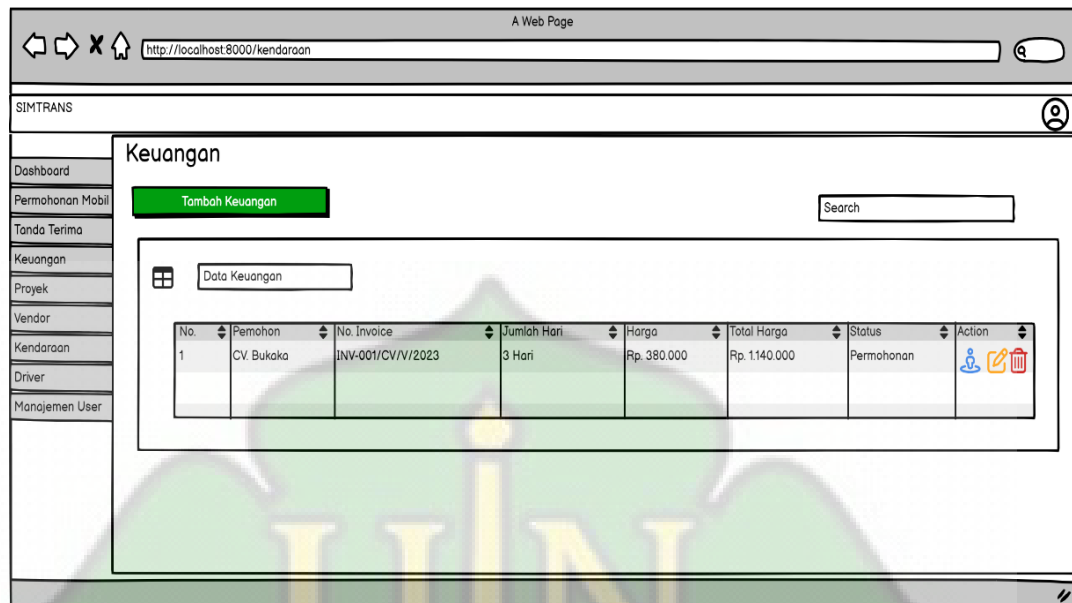
j. Tampilan Vendor



Gambar III.55 Tampilan Vendor

Desain tampilan vendor memiliki 4 tombol fungsi yaitu untuk menambahkan, melihat, mengedit, dan menghapus data vendor. Tombol "tambah vendor" berfungsi untuk mengisi data vendor. Di dalam tabel data vendor, terdapat informasi mengenai nama vendor, direktur, alamat, nomor telepon, dan *actions* yang dapat dilakukan. Rancangan tampilan vendor dapat dilihat pada gambar III.55.

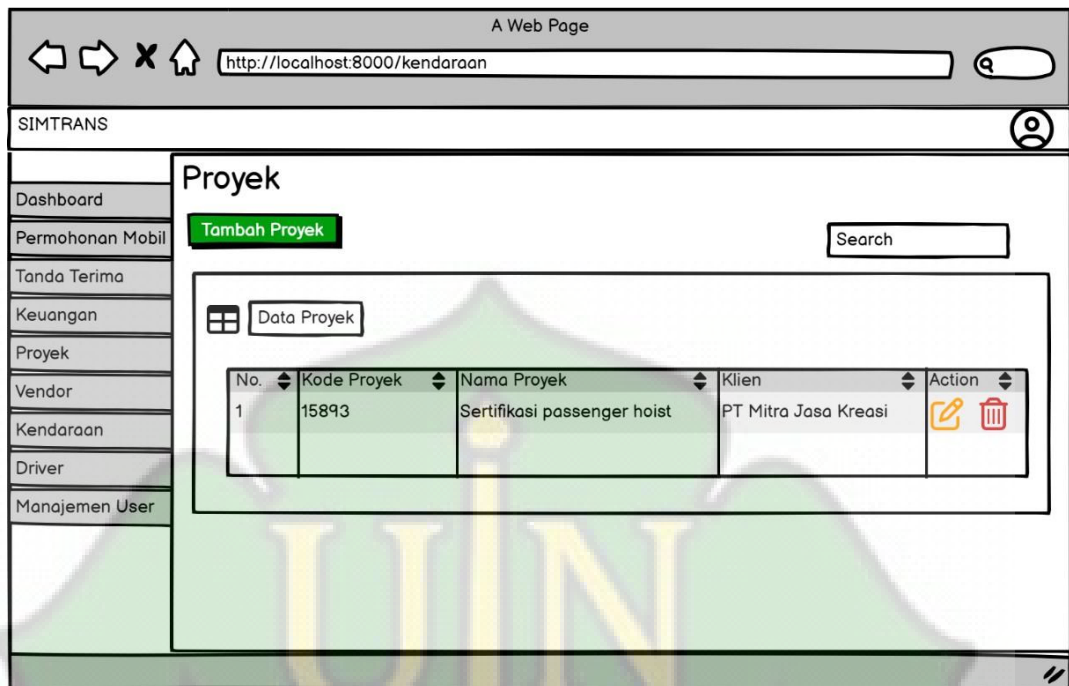
k. Tampilan Keuangan



Gambar III.56 Tampilan Keuangan

Tampilan keuangan dirancang dengan 4 *button* (tambah keuangan, view, edit, dan hapus), pada *button* tambah keuangan berfungsi untuk mengisi data keuangan. Lalu di tabel data keuangan ada terdapat pemohon, no. *invoice*, tanggal *invoice*, jumlah hari, harga, total harga, status dan *actions*. Dapat dilihat pada gambar III.56.

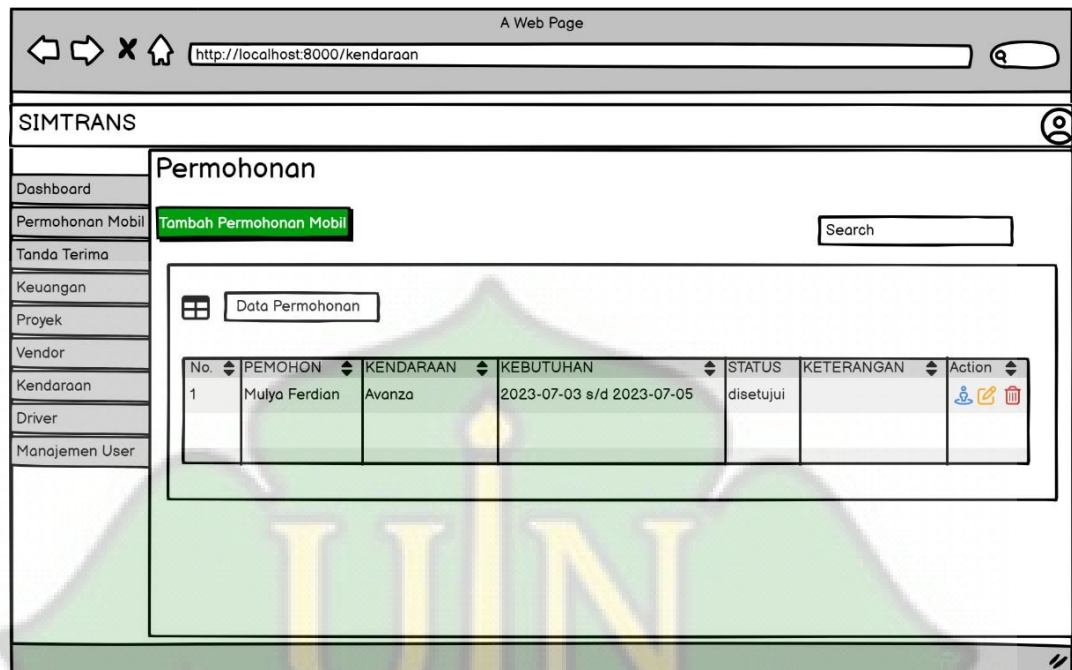
1. Tampilan Proyek



Gambar III.57 Tampilan Proyek

Tampilan proyek dirancang dengan 3 *button* (tambah harga kendaraan, edit, dan hapus), pada *button* tambah proyek berfungsi untuk mengisi data proyek. Lalu di tabel data proyek ada terdapat kode proyek, nama proyek, klien, dan *actions*. Dapat dilihat pada gambar III.57.

m. Tampilan Permohonan Mobil *User*



Gambar III.58 Tampilan Permohonan Mobil *User*

Tampilan permohonan mobil pada *user* dirancang dengan 4 *button* (tambah permohonan mobil, *view*, edit, dan hapus), pada *button* tambah permohonan mobil berfungsi untuk mengisi data permohonan mobil. Lalu di tabel data permohonan mobil ada terdapat pemohon, kendaraan, kebutuhan, status, keterangan, dan *actions*. Dapat dilihat pada gambar III.58.

III.3.3 Implementasi

Tahap di mana sistem yang telah dirancang dan dibuat pada tahap sebelumnya akan dijalankan dan diuji coba. Pada tahap ini, sistem akan diinstal dan dikonfigurasi pada server dan *database* yang telah dipersiapkan sebelumnya.

Setelah sistem terpasang, dilakukan pengujian fungsionalitas dan keamanannya untuk memastikan sistem berjalan dengan baik dan memenuhi kebutuhan yang telah ditentukan pada tahap analisis kebutuhan. Jika terdapat kesalahan atau bug, maka akan diperbaiki dan diuji kembali sebelum sistem siap untuk diimplementasikan secara penuh.

III.3.4 Pengujian Sistem

Proses pengujian dilakukan untuk memeriksa dan mengevaluasi sistem agar sesuai dengan kebutuhan fungsional dan non-fungsional yang telah ditentukan pada tahap analisis kebutuhan. Proses pengujian ini melibatkan rangkaian uji mulai dari uji unit, integrasi, sistem hingga uji penerimaan pengguna.

Uji unit dilakukan untuk mengevaluasi setiap unit program atau komponen sistem, sehingga memastikan kinerja optimal dan menghasilkan output yang diharapkan. Selain itu, uji integrasi dilakukan untuk mengevaluasi kesesuaian antara komponen dalam satu sistem.

III.4 Tempat dan Waktu Penelitian

Penelitian ini dilakukan di PT Surveyor Indonesia di Jl. Jenderal Sudirman, Geuceu Iniem, Kec. Banda Raya, Kota Banda Aceh. Waktu dimulai dari 1 Agustus s/d 24 Oktober 2022.

BAB IV

HASIL DAN PEMBAHASAN

IV.1 Implementasi

Dalam tahap implementasi, sistem informasi manajemen transportasi (SIMTRANS) berbasis *web* menggunakan *framework Laravel* telah berhasil dikembangkan. Sistem ini melibatkan proses pengkodean, pengujian, dan validasi untuk memastikan bahwa sistem berjalan dengan baik dan sesuai dengan kebutuhan PT Surveyor Indonesia. Dalam aplikasi tersebut sudah dikembangkan fitur-fitur menunya yaitu menu keuangan, vendor, tanda terima kendaraan, merk kendaraan, jenis kendaraan, data kendaraan, permohonan mobil, *service*, *driver*, dan proyek yang berhubungan dengan permohonan mobil.

IV.1.2 Hasil metode wawancara

Wawancara dilakukan pada tanggal 12 Juni 2023 pada PT Surveyor Indonesia dengan pegawai SI yaitu ibuk Nelly. Adapun hasil wawancara seperti pada tabel dibawah sebagai berikut:

Tabel IV.1.2.1 Hasil Wawancara

No	Pewawancara	Pegawai SI
1	Bagaimana proses peminjaman mobil yang dilakukan oleh karyawan SI selama ini?	Untuk saat ini kami melakukan proses peminjaman mobil melalui vendor dengan via telepon saja, kami telepon vendor langsung dan mengantarnya ke kantor.
2	Apakah selama ini Surveyor Indonesia (SI) bekerjasama dengan beberapa vendor dalam penyediaan mobil atau hanya mengandalkan ketersediaan mobil kantor saja?	Untuk saat ini kami memakai beberapa vendor untuk menyewa mobil operasional. Karena seandainya saya hubungi vendor A ternyata tidak tersedia mobil yang kami inginkan, maka kami menghubungi vendor B.

		Jadi kami ada beberapa vendor untuk melakukan penyewaan mobil operasional jadi tidak pegang satu vendor saja. Kami mencari info vendor dengan menggunakan via telp.
3	Bagaimana cara SI melakukan pemeliharaan terhadap mobil operasional?	Kalau untuk pemeliharaan mobil operasional tetap di <i>service</i> oleh pihak vendor karena itu tanggung jawab dari sewa-menyewa mobil. Jadi kami di setiap mobil itu kan ada kartu peringatan sekian KM untuk ganti oli sekalian <i>service</i> . Jadi kami pantau dari kartu itu aja, seandainya sudah dibawa waktunya untuk <i>service</i> dan ganti oli, kami menghubungi vendor tersebut jadi nanti di pihak vendor mengutus salah seorang anggotanya untuk ke kantor mengambil mobil itu untuk dibawa ke bengkel yang di tunjuk. Jadi kami terima mobil dalam keadaan normal. Lalu jika memelihara mobil dari pihak SI, maka pihak SI yang akan membawa mobil ke bengkel.
4	Apakah kantor SI sering melakukan kegiatan dinas luar kantor yang mengharuskan penggunaan mobil operasional?	Saat ini kami masih sering menyewa mobil operasional dari vendor. Karena bahwa hanya mobil operasional di kantor SI cuma satu sedangkan kami ada beberapa titik kegiatan di luar daerah. Tidak mungkin karyawan SI mematok satu mobil operasional yang di kantor SI, kami tetap memakai mobil vendor kalau seandainya pihak A mau

		dinas ke Lhokseumawe atau pihak B mau dinas ke Aceh Barat dan sebagainya. Jadi kami tetap menyewa mobil dari pihak vendor untuk melakukan pekerjaan operasional di luar daerah.
5	Bagaimana cara pihak SI membayar vendor setelah menyewa mobil?	Setelah karyawan atau anggota kami pulang dari daerah. Vendor tersebut menyerahkan <i>invoice</i> -nya atau tagihannya ke admin, nanti admin yang membuat tagihannya untuk dibayarkan. Kami bayar melalui via transfer menurut <i>invoice</i> yang masuk ke kami.
6	Apakah selama ini SI mencatat setiap peminjaman mobil operasional?	Tidak ada
7	Menurut anda, apakah aplikasi SIMTRANS ini bisa mengefisiensikan proses peminjaman mobil?	Menurut saya, aplikasinya sudah lengkap dengan semua informasi yang saya butuhkan. Dari kebutuhan mobil yang diminta oleh karyawan dan anggaran dari proyek serta saat pembayaran ke vendor, jadi aplikasinya sudah bagus, lengkap, dan jelas.

Berdasarkan wawancara di atas yang dilakukan pada tanggal 12 Juni 2023 pada PT Surveyor Indonesia, dapat disimpulkan sebagai berikut:

1. Proses peminjaman mobil yang dilakukan oleh karyawan SI selama ini dilakukan proses penyewaan mobil melalui vendor dengan menggunakan telepon.

2. Selama ini PT Surveyor Indonesia (SI) bekerjasama dengan beberapa vendor dengan menggunakan telp. untuk mencari informasi vendor. Pihak SI memakai beberapa vendor untuk menyewa mobil operasional ketika pihak SI menghubungi vendor tidak hanya di satu tempat, tapi bisa menghubungi vendor lain jika diperlukan.
3. Cara pihak SI melakukan pemeliharaan terhadap mobil operasional yaitu dengan membawa mobil operasional ke bengkel. Ketika pihak SI menyewa mobil operasional dari vendor, maka pihak vendorlah yang akan *service* ke bengkel karena itu tanggung jawab dari pihak vendor.
4. PT SI sering melakukan kegiatan dinas luar kantor dengan menggunakan menyewa mobil operasional dari vendor. Karena pihak SI hanya memiliki satu mobil operasional. Dengan begitu pihak SI tetap menyewa mobil dari pihak vendor untuk melakukan pekerjaan operasional di luar daerah.
5. Proses pihak SI membayar vendor setelah menyewa mobil dengan cara menyerahkan invoice atau tagihan ke admin SI. Yang mana pihak SI akan membayar melalui transfer menurut invoice.
6. Selama ini pihak SI tidak ada catatan peminjaman mobil operasional.
7. Aplikasi SIMTRANS sudah dilengkapi dengan semua informasi yang dibutuhkan. Dari kebutuhan mobil yang dibutuhkan karyawan, anggaran dari proyek, hingga pembayaran ke vendor.

IV.1.1 Fitur Authentication

Adapun hasil dari laravel untuk fitur *Authentication* yaitu *route* pada *web.php* yang dapat dibawah ini.

1. Route Laravel

Coding dibawah ini merupakan konfigurasi rute (*route*) dalam *framework laravel* yang digunakan untuk mengatur navigasi dan akses pada aplikasi *web* seperti pada gambar IV.1. Berikut adalah penjelasan lebih detail mengenai masing-masing baris kode:

12. **Auth::routes()**: Mendaftarkan rute-rute otentikasi default yang disediakan oleh *Laravel*, seperti login, register, reset *password*, dan lain-lain. Rute-rute tersebut akan terhubung dengan *controller* bawaan *Laravel* yang mengatur logika otentikasi.
13. **Route::get('/', [App\Http\Controllers\HomeController::class, 'index'])->name('home')**: Menentukan rute untuk halaman utama aplikasi. Ketika pengguna mengakses '/' (root), *HomeController* akan dipanggil dengan aksi 'index'. Nama rute 'home' digunakan untuk mengidentifikasi rute ini.
14. **Route::get('/home', [App\Http\Controllers\HomeController::class, 'index'])->name('home')**: Menentukan rute untuk halaman 'home' aplikasi. Ketika pengguna mengakses '/home', *HomeController* akan dipanggil dengan aksi 'index'. Nama rute 'home' digunakan untuk mengidentifikasi rute ini.
15. **Route::get('/logout', function() {...})**: Menentukan rute untuk proses logout pengguna. Ketika pengguna mengakses '/logout', fungsi closure akan dieksekusi. Pada fungsi closure, pemanggilan *Auth::Logout()* digunakan untuk melakukan logout pengguna. Setelah logout, pengguna akan diarahkan kembali ke halaman utama '/home'.

```
Auth::routes();
Route::get('/', [App\Http\Controllers\HomeController::class, 'index'])->name('home');
Route::get('/home', [App\Http\Controllers\HomeController::class, 'index'])->name('home');

Route::get('/logout', function() {
    \Auth::Logout();
    return redirect('/home');
});
```

Gambar IV.1 Create Account laravel

a. Login

Pada halaman ini pengguna bisa melakukan login untuk melakukan peminjaman mobil pada aplikasi. Halaman login bertanggung jawab untuk memverifikasi identitas pengguna yang mencoba sistem. Untuk melakukan login *user* harus mengisi *email* dan *password* pada form tersedia. Seperti pada gambar IV.2.

Gambar IV.2 Hasil Halaman *Login*

b. Register

Halaman register berfungsi untuk diperlukan agar setiap pengguna/*user* dapat mendaftarkan diri ke aplikasi. Pada halaman register ini ada 4 data yaitu *username*, *email*, *password*, dan konfirmasi *password* yang mana harus diisi oleh pengguna untuk mengumpulkan informasi pengguna, validasi data masukan, pembuatan akun pengguna, dan keamanan dan perlindungan data. Seperti pada gambar IV.3.

Gambar IV.3 Hasil Halaman Register

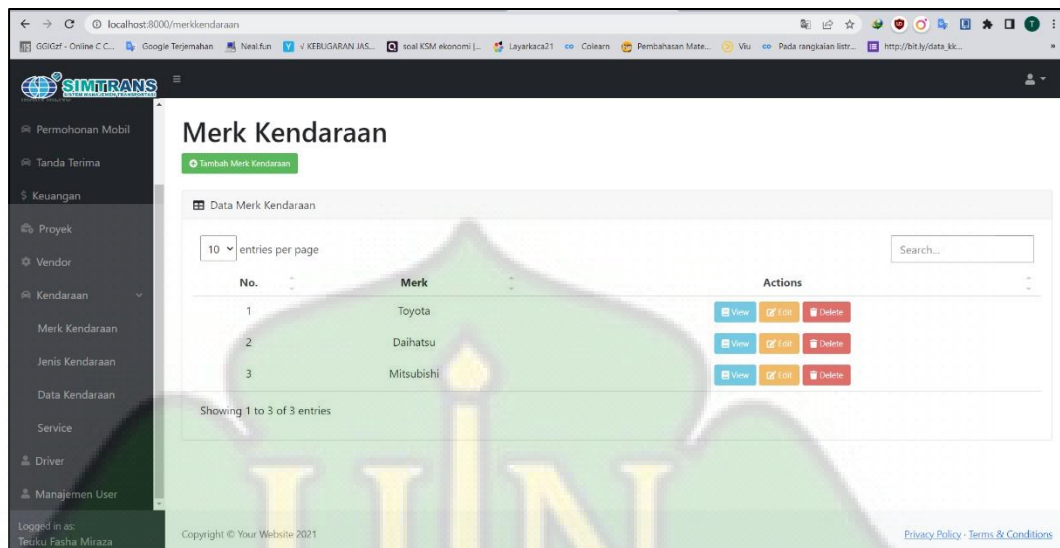
2. Fitur Menu Kendaraan

Fitur kendaraan di admin terbagi 3 yaitu, merk kendaraan, jenis kendaraan, dan data kendaraan. Sedangkan fitur kendaraan di vendor hanya satu yaitu, data kendaraan.

a. Merk Kendaraan

Fitur merk kendaraan ini berfungsi untuk menambah merk kendaraan yang dibutuhkan dari admin. Fitur ini memungkinkan admin untuk mencatat dan

mengelola data merk kendaraan yang tersedia dalam sistem. Seperti pada gambar IV.4.



Gambar IV.4 Hasil Menu Merk Kendaraan

Adapun coding yang menghubungkan dengan fitur merk kendaraan dapat dilihat pada gambar IV.5. Model merk kendaraan.php yang mempresentasikan entitas merk kendaraan dalam tabel 'merkkendaraan' di *database*. Menggunakan *trait HasFactory* untuk mendukung pembuatan *instance* model menggunakan *factory*. Menentukan nama tabel, *primary key*, dan kolom yang dapat diisi. *Route* merk kendaraan menggunakan *resourceful routing* untuk menghubungkan rute *'/merkkendaraan'* dengan *controller MerkKendaraanController*.

MerkKendaraanController.php di *function create()* digunakan untuk menampilkan halaman pembuatan merk kendaraan baru. Mendapatkan pengguna yang saat ini terautentikasi. Mengirimkan data pengguna ke halaman pembuatan merk kendaraan. *Create.blade.php* merk kendaraan menampilkan form untuk membuat merk kendaraan baru. Form ini memiliki input *field* untuk merk kendaraan. Nilai dari input *field* merk kendaraan diisi oleh pengguna.

1. Model merk kendaraan

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Merkkendaraan extends Model
{
    use HasFactory;
    protected $table = 'merkkendaraan';
    protected $primaryKey = 'id';
    protected $fillable = ['merk'];
}
```

2. Route merk kendaraan

```
Route::resource("/merkkendaraan", MerkkendaraanController::class);
```

3. MerkkendaraanController.php

```
public function create()
{
    $user = Auth::user();
    return view('merkkendaraan.create', compact('user'));
    // return view('service.create');
}
```

4. Create.blade.php merk kendaraan

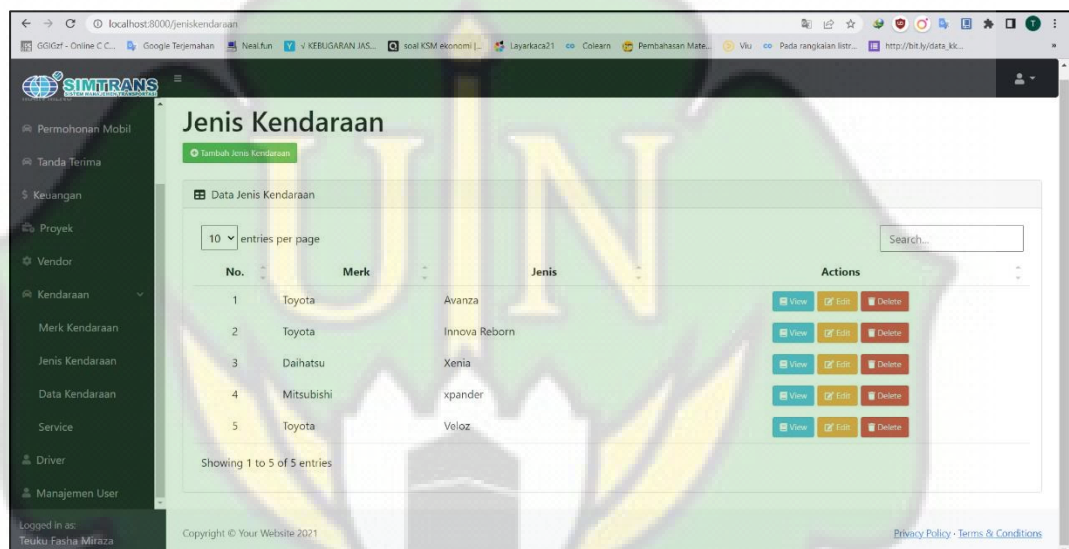
```
<table align="center">
    <tr>
        <td>Merk</td>
        <td>:</td>
        <td><input type="text" name="merk" id="merk"
            class="form-control"></td>
    </tr>
</table>
```

Gambar IV.5 Algoritma dari fitur merk kendaraan

b. Jenis Kendaraan

Fitur jenis kendaraan ini berguna untuk mengelompokkan kendaraan berdasarkan jenisnya. Admin dapat membuat jenis kendaraan seperti avanza, Xenia, Br-V atau jenis kendaraan lainnya yang sesuai dengan kebutuhan perusahaan.

Hal ini membantu dalam mengorganisir dan mengelompokkan kendaraan berdasarkan kategori tertentu. Dengan adanya fitur ini, admin dapat dengan mudah mengidentifikasi dan memilih jenis kendaraan saat melakukan berbagai kegiatan dalam sistem. Seperti pada gambar 4.6.



Gambar IV.6 Hasil Menu Jenis Kendaraan

Adapun coding yang menghubungkan dengan fitur merk kendaraan dapat dilihat pada gambar IV.7. Model jenis kendaraan yang merepresentasikan entitas jenis kendaraan dalam tabel '**jeniskendaraan**' di *database* . Menggunakan *trait HasFactory* untuk mendukung pembuatan *instance* model menggunakan *factory*. Menentukan nama tabel, *primary key*, dan kolom yang dapat diisi. *Route* jenis kendaraan menggunakan *resourceful* routing untuk menghubungkan rute '**/jeniskendaraan**' dengan *controller jeniskendaraanController*.

JenisKendaraanController.php di *function index()* digunakan untuk menampilkan halaman daftar jenis kendaraan. Mengecek otorisasi pengguna dan hanya mengizinkan akses untuk pengguna dengan level '**admin**'. Mengambil

semua data jenis kendaraan dan merk kendaraan. Mengirimkan data jenis kendaraan dan merk kendaraan ke halaman daftar jenis kendaraan.

Index.blade.php jenis kendaraan menampilkan tabel dengan daftar jenis kendaraan. Setiap baris tabel menampilkan nomor iterasi, merk kendaraan, dan jenis kendaraan. Baris tabel juga memiliki kolom untuk aksi yang dapat dilakukan pada jenis kendaraan.

1. Model jenis kendaraan

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class jeniskendaraan extends Model
{
    use HasFactory;
    protected $table = 'jeniskendaraan';
    protected $primaryKey = 'id';
    protected $fillable = ['id_merk', 'jenis'];
}
```

2. Route jeniskendaraan

```
Route::resource("/jeniskendaraan", jeniskendaraanController::class);
```

3. JeniskendaraanController.php

```
public function index()
{
    $this->authorize('admin');
    $jeniskendaraan = jeniskendaraan::all();
    $merkkendaraan = merkkendaraan::pluck('merk','id');
    return view('jeniskendaraan.index',
    compact('jeniskendaraan', 'merkkendaraan'));
}
```

4. Index.blade.php jenis kendaraan

```
<tbody>
    @foreach($jeniskendaraan as $item)
        <tr>
            <td><center>{{ $loop->iteration }}</center></td>
            <td>{{ $merkkendaraan[$item->id_merk] }}</td>
            <td>{{ $item->jenis }}</td>
            <td><center>
```

Gambar IV.7 Algoritma dari fitur jenis kendaraan

c. Kendaraan

Fitur ini digunakan untuk menyimpan informasi penting tentang setiap kendaraan dalam sistem. Informasi yang dapat disimpan meliputi nomor polisi, merk kendaraan, jenis kendaraan, tahun produksi, bahan bakar yang digunakan, status kendaraan, dan vendor yang terkait. Data kendaraan yang terperinci dan teratur membantu dalam melacak dan mengelola inventaris kendaraan dengan lebih efisien.

Fitur ini memungkinkan admin untuk mengelola data kendaraan secara terpusat. Admin dapat menambahkan, mengedit, dan menghapus data kendaraan yang ada dalam sistem. Hal ini membantu dalam pemeliharaan dan pembaruan informasi terkait kendaraan perusahaan. Seperti pada gambar IV.8.

No.	No. Polisi	Kendaraan	Bahan Bakar	Tahun	Harga	Status	Vendor	Actions
1	BL 1086 HH	Toyota - Avanza		2018	Rp. 0 / Day	Tersedia		
2	BL 1123 CK	Toyota - Avanza		2018	Rp. 0 / Day	Tersedia		
3	BL 1171 PB	Toyota - Avanza	Premium	2022	Rp. 350.000 / Day	Tersedia	PT Samantha Group	
4	B 2598 SYP	Toyota - Innova Reborn	Premium	2018	Rp. 550.000 / Day	Tersedia	PT Samantha Group	
5	BL 1559	Daihatsu - Xenia	Premium	2022	Rp. 350.000 / Day	Tersedia		

Gambar IV.8 Hasil Menu Kendaraan

Adapun coding yang menghubungkan dengan fitur kendaraan dapat dilihat pada gambar IV.9. Model kendaraan digunakan untuk berintraksi dengan entitas kendaraan dalam basis data. Model ini menggunakan *trait HasFactory* yang menyediakan metode pembuatan *instance* model dengan mudah. Properti “*\$table*” diatur dengan nilai ‘*kendaraan*’ yang menunjukkan tabel yang terkait dengan model ini dalam basis data. Properti “*\$primaryKey*” diatur dengan nilai ‘*id*’ yang menunjukkan kolom kunci utama pada tabel kendaraan. Properti “*\$fillable*” diatur dengan nilai [‘*platmobil*’, ‘*stnk*’, ‘*pajak*’, ‘*id_vendor*’, ‘*id_jenis*’, ‘*bbm*’, ‘*harga*’] yang menunjukkan atribut-atribut pada model ini yang dapat diisi secara massal.

Route kendaraan menggunakan metode *Route::resource()* untuk menghasilkan rute CRUD (*create, read, update, delete*) untuk model **Kendaraan**. Rute ini terhubung ke “*kendaraansController*” yang akan mengelola permintaan terkait kendaraan.

Lalu “*kendaraansController.php*” di *function create()* digunakan untuk menampilkan halaman pembuatan kendaraan. Mengambil pengguna yang sedang login menggunakan *Auth::user* dan menyimpannya dalam variabel *\$user*. Melakukan perulangan *foreach* untuk mengambil semua merk kendaraan dari model merkkendaraan dan menyimpannya dalam array *\$merk* dengan id merk sebagai kunci dan nama merk sebagai nilai. Mengembalikan tampilan

'**kendaraan.create**' dengan variabel “**\$user**” dan “**\$merk**” yang dikompak untuk digunakan dalam tampilan tersebut.

Sedangkan *create.blade.php* kendaraan adalah tampilan yang menampilkan formulir untuk membuat kendaraan. Tabel memiliki beberapa baris untuk memasukkan data kendaraan. Pada baris pertama, menggunakan “**Form::select**” untuk membuat dropdown berisi pilihan merk kendaraan. Nilai dari dropdown ini diberikan oleh variabel “**\$merk**” yang berisi daftar merk kendaraan. Pada baris kedua, juga menggunakan “**Form::select()**” untuk membuat dropdown berisi pilihan jenis kendaraan. Namun, pada saat ini dropdown jenis kendaraan tidak memiliki nilai apapun (array kosong) karena nilai tersebut belum ditentukan.

1. Model kendaraan

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Kendaraan extends Model
{
    use HasFactory;
    protected $table = 'kendaraan';
    protected $primaryKey = 'id';
    protected $fillable = ['platmobil', 'stnk', 'pajak',
        'id_vendor', 'id_jenis', 'bbm', 'harga'];
}
```

2. Route kendaraan

```
Route::resource("/kendaraan", kendaraanController::class);
```

3. KendaraanController.php

```
public function create()
{
    $user = Auth::user();
    $merk = array('' => '');
    foreach(merkkendaraan::all() as $row)
    {
        $merk[$row->id] = $row->merk;
    }
    return view('kendaraan.create', compact('user', 'merk'));
    // return view('service.create');
```

4. Create.blade.php kendaraan

```
{!! csrf_field() !!}
<table align="center">
<tr>
<td>Merk</td>
<td></td>
<td>{!! Form::select('id_merk', $merk, null,
    array('id' => 'merk', 'style'=>'width: 190px')) !!}</td>
</tr>
<tr>
<td>Jenis</td>
<td></td>
<td>{!! Form::select('id_jenis', array(), null,
    array('id' => 'jeniskendaraan', 'style'=>'width: 190px')) !!}</td>
</tr>
```

Gambar IV.9 Algoritma dari fitur kendaraan

3. Fitur Menu Service

Fitur ini dapat mencatat semua kegiatan *service* kendaraan yang dilakukan, termasuk tanggal *service*, jenis perbaikan, dan informasi terkait lainnya. Hal ini membantu dalam melacak riwayat perawatan kendaraan dan memastikan bahwa semua kegiatan perawatan tercatat dengan baik. Seperti pada gambar IV.10.

No.	No. Polisi	Kendaraan	Tgl Service Mobil	Perbaiki	Biaya Service	Vendor	Actions
1	BL 1086 HH	Avanza	10 Oktober 2022	ganti oli	Rp. 500,000		
2	BL 1123 CK	Avanza	10 Oktober 2022	service AC, ganti Oli, ganti Busi	Rp. 1,200,000		
3	BL 1559	Xenia	10 Oktober 2022	ganti oli	Rp. 500,000		
4	BL 1171 PB	Avanza	10 Oktober 2022	service AC, ganti Oli, ganti Busi	Rp. 500,000	PT Samantha Group	
5	AS 7458 DV	Avanza	20 September 2023	Rem Rusak	Rp. 300,000	CV. Bukaka	
6	PP 03621 XN	Xenia	2023-06-28	Ganti Oli	Rp. 200,000	CV. Bukaka	

Gambar IV.10 Hasil Menu Service Kendaraan

Adapun coding untuk pembuatan fitur *service* seperti pada gambar IV.11. Model *service* yang akan digunakan untuk berintraksi dengan entitas layanan dalam basis data. Model ini menggunakan *trait HasFactory* yang menyediakan metode pembuatan *instance* model dengan mudah. Properti “*\$table*” diatur dengan nilai ‘*service*’ yang menunjukkan tabel yang terkait dengan model ini dalam basis data. Properti “*\$primaryKey*” diatur dengan nilai ‘*id*’ yang menunjukkan kolom kunci utama pada tabel layanan. Properti “*\$fillable*” diatur dengan nilai [‘*tgl_service_mobil*’, ‘*perbaiki*’, ‘*biaya_service*’, ‘*id_kendaraan*’, ‘*id_vendor*’] yang menunjukkan atribut-atribut pada model ini yang dapat diisi secara massal.

Route service yang menggunakan metode “*Route::resource()*” untuk menghasilkan rute CRUD (*create, read, update, delete*) untuk model *Service*. Rute ini terhubung ke “*servicesController*” yang akan mengelola permintaan terkait layanan.

Kemudian *serviceController.php* memiliki fungsi yang digunakan untuk menampilkan halaman detail layanan. Menerima parameter “*\$id*” yang merupakan ID layanan yang akan ditampilkan. Menggunakan model *Service* untuk mencari layanan dengan ID yang sesuai menggunakan metode “*find(\$id)*”. Mengembalikan tampilan ‘*service.show*’ dengan variabel “*\$service*” yang berisi data layanan yang ditemukan.

File *service show.blade.php* adalah tampilan yang menampilkan detail layanan. Tabel ini memiliki beberapa baris yang menampilkan atribut-atribut layanan, seperti tanggal layanan mobil dan perbaikan yang dilakukan. Nilai dari atribut-atribut tersebut diambil dari variabel “*\$service*” yang diteruskan dari kontroler. Masing-masing atribut ditampilkan dalam elemen input dengan nilai yang sesuai menggunakan sintaks “*{{ \$service->tgl_service_mobil }}*” dan “*{{ \$service->perbaiki }}*”.

1. Model service

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Service extends Model
{
    use HasFactory;
    protected $table = 'service';
    protected $primaryKey = 'id';
    protected $fillable = ['tgl_service_mobil', 'perbaiki',
        'biaya_service', 'id_kendaraan', 'id_vendor'];
}
```

2. Route service

```
Route::resource("/service", servicesController::class);
```

3. ServiceController.php

```
public function show($id)
{
    $service = Service::find($id);
    return view('service.show')->with('service', $service);
}
```

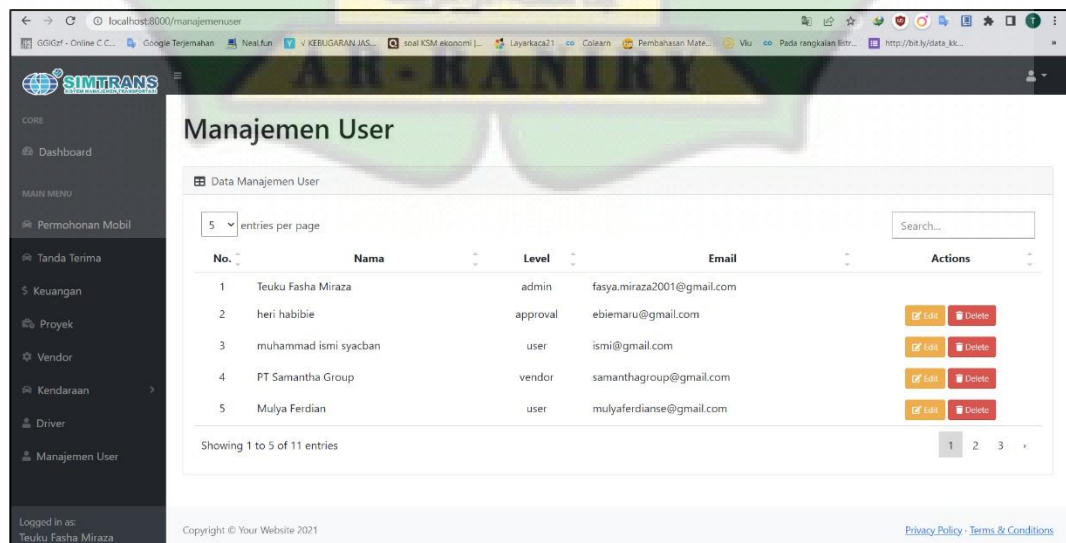
4. Show.blade.php service

```
<table align="center">
<tr>
<td>Tgl Service Mobil</td>
<td></td>
<td><input type="text" name="tgl_service_mobil" id="tgl_service_mobil"
class="form-control" value="{{ $service->tgl_service_mobil }}"></td>
</tr>
<tr>
<td>Perbaiki</td>
<td></td>
<td><input type="text" name="perbaiki" id="perbaiki"
class="form-control" value="{{ $service->perbaiki }}"></td>
</tr>
```

Gambar IV.11 Algoritma dari fitur *service*

4. Fitur Menu Manajemen User

Tampilan manajemen *user* berfungsi untuk mengelola pengguna atau anggota yang memiliki akses ke sistem. Pada halaman ini admin bisa menyimpan data-data *user* yang sudah login dan bisa mengatur level *user* pada nama *user*. Seperti pada gambar IV.12.



Gambar IV.12 Hasil Menu Manajemen *User*

Adapun coding untuk membuat fitur manajemen *user* seperti pada gambar IV.13. Model manajemen *user* yang akan digunakan untuk berinteraksi dengan entitas pengguna dalam basis data. Model ini menggunakan *trait* “**HasFactory**” yang menyediakan metode pembuatan *instance* model dengan mudah. Properti “**\$table**” diatur dengan nilai ‘*users*’ yang menunjukkan tabel yang terkait dengan model ini dalam basis data. Properti “**\$primaryKey**” diatur dengan nilai ‘*id*’ yang menunjukkan kolom kunci utama pada tabel pengguna. Properti “**\$fillable**” diatur dengan nilai [‘*nama*’, ‘*level*’, ‘*email*’, ‘*password*’, ‘*homebase*’] yang menunjukkan atribut-atribut pada model ini yang dapat diisi secara massal.

Untuk *route* manajemen *user* menggunakan metode “**Route::resource()**” untuk menghasilkan rute CRUD (*create*, *read*, *update*, *delete*) untuk model **Manajemenuser**. Rute ini terhubung ke “**manajemenusersController**” yang akan mengelola permintaan terkait pengguna.

Kemudian di file “**manajemenusersController.php**” pada *function* edit(\$id) digunakan untuk menampilkan halaman pengeditan pengguna. Menerima parameter “**\$id**” yang merupakan ID pengguna yang akan diedit.

Menggunakan model **Manajemenuser** untuk mencari pengguna dengan ID yang sesuai menggunakan metode “**find(\$id)**”. Membuat array “**\$vendor**” yang berisi daftar vendor dengan format [‘*id*’ => ‘*vendor*’] menggunakan model vendor. Mengembalikan tampilan ‘**manajemenuser.edit**’ dengan variabel \$manajemenuser yang berisi data pengguna yang ditemukan dan “**\$vendor**” yang berisi daftar vendor.

Yang terakhir ada manajemen *user* edit.blade.php yang digunakan untuk mengedit pengguna. Terdapat dua elemen input yang akan diedit: Level dan Vendor. Level ditampilkan dalam elemen select dengan beberapa opsi yang dapat dipilih.

Vendor ditampilkan dalam elemen select dengan opsi yang berasal dari array “**\$vendor**”. Nilai yang dipilih untuk Level dan Vendor ditentukan oleh data yang diteruskan dari kontroler, yaitu “**\$manajemenuser->level**” dan “**\$manajemenuser->homebase**”.

1. Model manajemen user

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Manajemenuser extends Model
{
    use HasFactory;
    protected $table = 'users';
    protected $primaryKey = 'id';
    protected $fillable = ['nama', 'level', 'email',
        'password', 'homebase'];
}
```

2. Route manajemen user

```
Route::resource("/manajemenuser", manajemenusersController::class);
```

3. ManajemenusersController.php

```
*/
public function edit($id)
{
    $manajemenuser = Manajemenuser::find($id);
    $vendor = array('' => '');
    foreach($vendor::all() as $row)
        $vendor[$row->id] = $row->vendor;
    return view('manajemenuser.edit', compact('manajemenuser', 'vendor'));
}
```

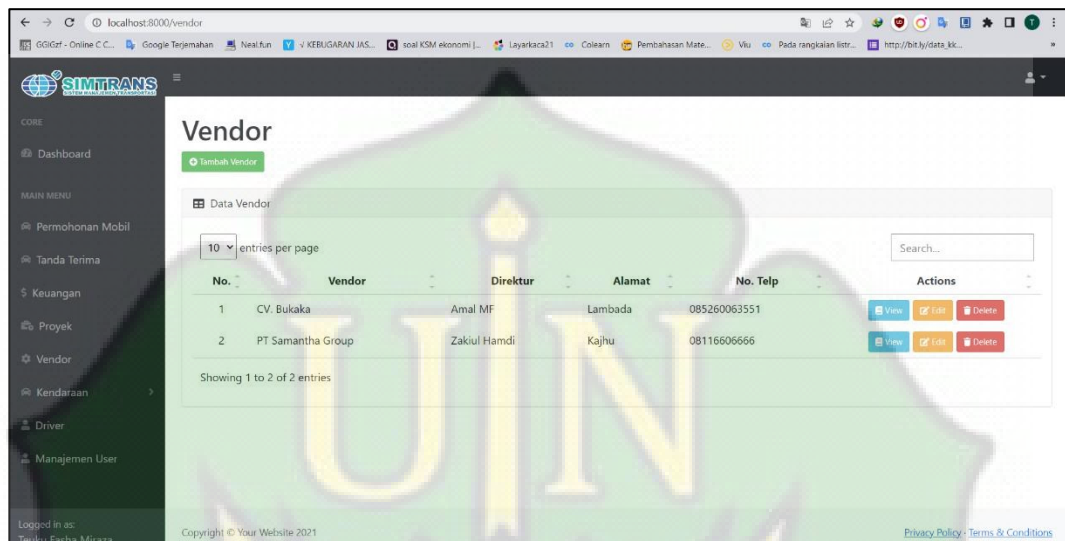
4. Show.blade.php manajemen user

```
<table align="center">
<tr>
<td>Nama</td>
<td>:</td>
<td><input type="text" name="nama" id="nama"
class="form-control" value="{{ $manajemenuser->nama }}"></td>
</tr>
<tr>
<td>Level</td>
<td>:</td>
<td><input type="text" name="level" id="level"
class="form-control" value="{{ $manajemenuser->level }}"></td>
</tr>
```

Gambar IV.13 Algoritma dari fitur manajemen user

5. Fitur Menu Vendor

Tampilan vendor ini berfungsi untuk mengelola informasi dan interaksi dengan vendor atau pemasok yang terkait dengan kebutuhan transportasi dalam sistem. Pada halaman ini admin bisa mengisi data-data vendor atau pemasok untuk menyewa kendaraan. Seperti pada gambar IV.14.



Gambar IV.14 Hasil Menu Vendor

Adapaun coding untuk membuat fitur vendor seperti pada gambar IV.15. Model vendor digunakan untuk berinteraksi dengan entitas vendor dalam basis data. Model ini menggunakan *trait* “**HasFactory**” yang menyediakan metode pembuatan *instance* model dengan mudah. Properti “**\$table**” diatur dengan nilai 'vendor' yang menunjukkan tabel yang terkait dengan model ini dalam basis data. Properti “**\$primaryKey**” diatur dengan nilai 'id' yang menunjukkan kolom kunci utama pada tabel vendor. Properti “**\$fillable**” diatur dengan nilai ['alamat', 'no_telp', 'vendor', 'direktur'] yang menunjukkan atribut-atribut pada model ini yang dapat diisi secara massal.

Route vendor menggunakan metode “**Route::resource()**” untuk menghasilkan rute CRUD (*create, read, update, delete*) untuk model Vendor. Rute ini terhubung ke “**VendorsController**” yang akan mengelola permintaan terkait vendor.

Lalu “**VendorsController.php**” di *function* index untuk menampilkan halaman daftar vendor. Hanya pengguna dengan peran 'admin' yang diotorisasi

untuk mengakses halaman ini. Menggunakan model Vendor untuk mengambil semua data vendor dengan metode **all()**. Mengembalikan tampilan '**vendor.index**' dengan variabel \$vendor yang berisi data vendor.

Vendor index.blade.php untuk menampilkan daftar vendor. Menggunakan loop **foreach** untuk mengulang setiap item vendor dalam array "**\$vendor**". Masing-masing atribut vendor, seperti vendor *name*, direktur, alamat, dan nomor telepon, ditampilkan dalam kolom terpisah di setiap baris.

1. Model vendor

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Vendor extends Model
{
    use HasFactory;
    protected $table = 'vendor';
    protected $primaryKey = 'id';
    protected $fillable = ['alamat', 'no_telp', 'vendor', 'direktur'];
}
```

2. Route vendor

```
Route::resource("/vendor", VendorsController::class);
```

3. VendorController.php

```
|  
|  
| public function index()  
| {  
|     $this->authorize('admin');  
|     $vendor = vendor::all();  
|     return view ('vendor.index')->with('vendor', $vendor);  
| }
```

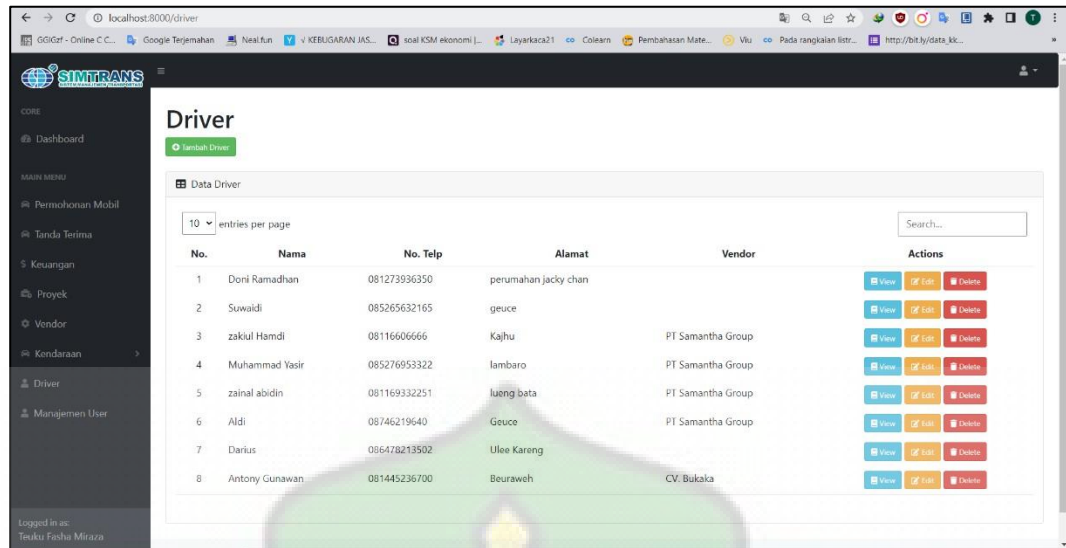
4. Index.blade.php vendor

```
<tbody>  
    @foreach($vendor as $item)  
        <tr>  
            <td><center>{{ $loop->iteration }}</center></td>  
            <td>{{ $item->vendor }}</td>  
            <td>{{ $item->direktur }}</td>  
            <td>{{ $item->alamat }}</td>  
            <td>{{ $item->no_telp }}</td>  
            <td><center>
```

Gambar IV.15 Algoritma dari fitur vendor

6. Fitur Menu Driver

Tampilan *driver* ini berguna untuk mengelola data dan informasi terkait dengan para pengemudi (*driver*) yang terlibat dalam layanan transportasi. Pada halaman ini admin bisa mengisi nama *driver*, no. telp, alamat, dan identifikasi vendor di dalam sistem. Seperti pada gambar IV.16.



Gambar IV.16 Hasil Menu Driver

Adapun coding untuk membuat fitur menu *driver* seperti pada gambar IV.17. Model *driver* yang merepresentasikan entitas “*driver*” dalam tabel di *database* . Menggunakan *trait HasFactory* untuk mendukung pembuatan *instance* model menggunakan *factory*. Menentukan nama tabel, *primary key*, dan kolom yang dapat diisi. *Route driver* menggunakan *resourceful* routing untuk menghubungkan rute *'/driver'* dengan *controller DriversController*.

DriverController.php pada *function update()* menerima objek *request* dan objek *driver* sebagai argumen. Menggunakan metode “*where*” untuk mencari entitas *Driver* berdasarkan ID yang diberikan. Menggunakan metode “*update*” untuk memperbarui kolom 'nama', 'no_telp', dan 'alamat' pada entitas *Driver* yang ditemukan. Mengembalikan ke halaman sebelumnya dengan pesan flash yang menyatakan bahwa *driver* telah diperbarui.

Lalu *create.blade.php driver* yang berguna untuk menampilkan form untuk membuat *driver* baru. Form ini memiliki input *field* untuk mengisi nama dan nomor telepon *driver*. Input *field* untuk nama memiliki atribut ***name="nama"*** dan ID "nama". Input *field* untuk nomor telepon memiliki atribut ***name="no_telp"*** dan ID "no_telp".

1. Model driver

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Driver extends Model
{
    use HasFactory;
    protected $table = 'driver';
    protected $primaryKey = 'id';
    protected $fillable = ['nama', 'no_telp', 'alamat', 'id_vendor'];
}
```

2. Route driver

```
Route::resource("/driver", DriversController::class);
```

3. DriverController.php

```
*/
public function update(Request $request, Driver $driver)
{
    Driver::where('id', $driver->id)->update([
        'nama' => $request->nama,
        'no_telp' => $request->no_telp,
        'alamat' => $request->alamat,
    ]);
    return back()->with('flash_message', 'driver Update!');
}
```

4. Create.blade.php driver

```
<table align="center">
<tr>
<td>Nama</td>
<td>:</td>
<td><input type="text" name="nama" id="nama"
class="form-control"></td>
</tr>
<tr>
<td>No. Telp </td>
<td>:</td>
<td><input type="text" name="no_telp" id="no_telp"
class="form-control"></td>
</tr>
<tr>
```

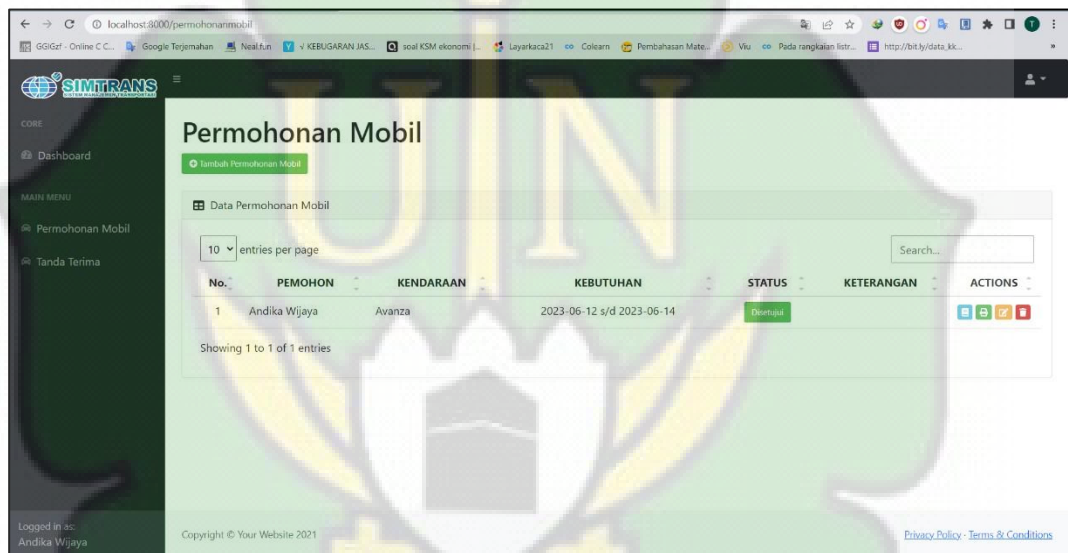
Gambar IV.17 Algoritma dari fitur *driver*

7. Fitur Menu Permohonan Mobil

Adapun peran yang dilakukan *user*, admin, *approval*, dan vendor pada permohonan mobil, yaitu sebagai berikut:

a. User

Di permohonan mobil ini *user* memiliki tindakan untuk mengisi data permohonan mobil yaitu, mengajukan permohonan mobil, memilih jenis kendaraan, memilih tanggal dan durasi untuk pergi dan pulang, melihat status permohonan, mengedit atau menghapus permohonan dan lain-lain. Seperti pada gambar IV.18 dan IV.19.



Gambar IV.18 Hasil Menu Permohonan Mobil (*user*)

Tambah Data Permohonan Mobil

Nama : Andika Wijaya

Merk :

Jenis :

Jumlah Orang :

Proyek :

Tujuan :

Barang :

Tgl Berangkat : 06/14/2023

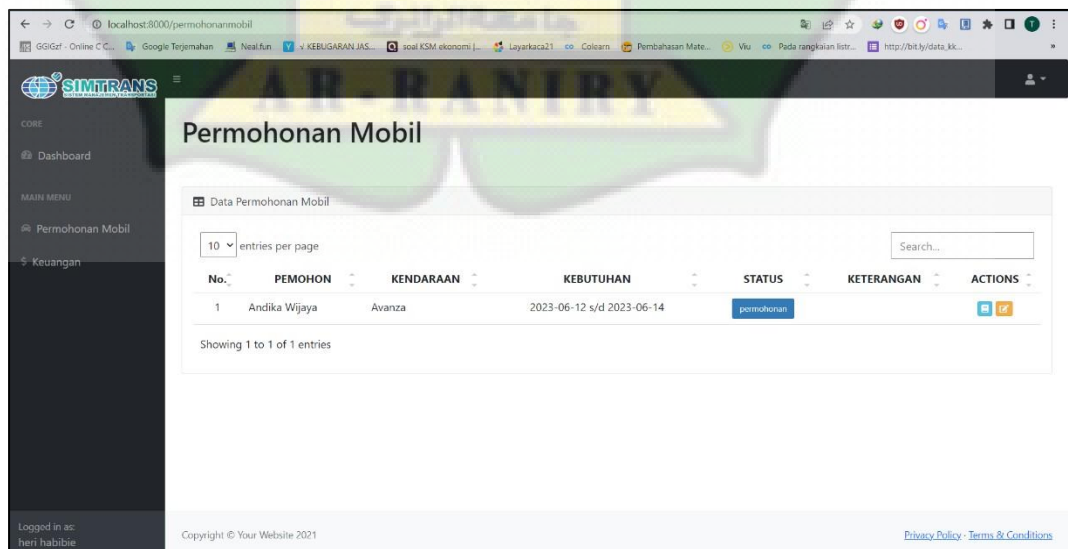
Tgl Pulang : 05/23/2023

Gambar IV.19 Hasil Tambah Data Permohonan Mobil (*user*)

b. Approval

Di permohonan mobil ini *approval* memiliki peran yang penting dalam proses persetujuan dan peninjauan permohonan dari *user*. Tugas-tugas yang dilakukan *approval* adalah, sebagai berikut:

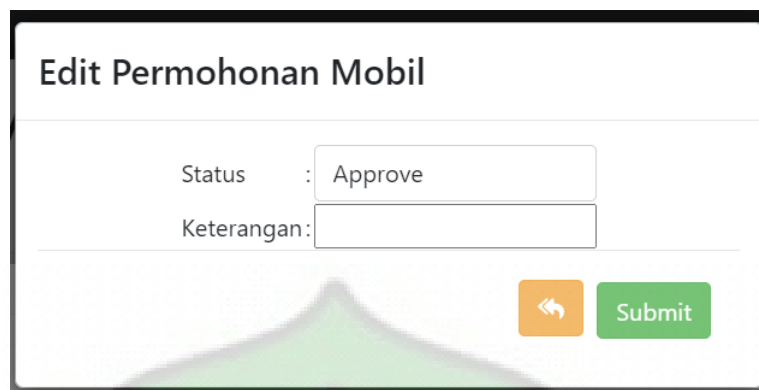
- **Evaluasi persyaratan:** *approval* mengevaluasi permohonan berdasarkan persyaratan yang telah ditetapkan. Mereka akan memeriksa apakah permohonan memenuhi syarat-syarat tertentu, seperti ketersediaan mobil pada tanggal yang diminta, durasi sewa yang diizinkan, dan persyaratan kelayakan lainnya. Hal ini dilakukan untuk memastikan bahwa permohonan sesuai dengan kebijakan perusahaan atau organisasi terkait.
- **Keputusan persetujuan atau penolakan:** setelah meninjau permohonan, *approval* akan memberikan keputusan persetujuan atau penolakan. Jika permohonan memenuhi persyaratan yang ditetapkan, mereka akan memberikan persetujuan dan melanjutkan dengan langkah-langkah selanjutnya, seperti penugasan pengemudi atau pengaturan jadwal pengambilan dan pengembalian mobil. Namun, jika permohonan tidak memenuhi persyaratan atau ada ketidaksesuaian, mereka dapat menolak permohonan dengan memberikan alasan yang jelas kepada pengguna. Seperti pada gambar IV.20 dan IV.21.



No.	PEMOHON	KENDARAAN	KEBUTUHAN	STATUS	KETERANGAN	ACTIONS
1	Andika Wijaya	Avanza	2023-06-12 s/d 2023-06-14	permohonan		

Showing 1 to 1 of 1 entries

Gambar IV.20 Hasil Menu Permohonan Mobil (*approval*)



Edit Permohonan Mobil

Status :

Keterangan:

Gambar IV.21 Hasil Edit Permohonan Mobil (*approval*)

c. Admin

Di permohonan mobil admin memiliki tanggung jawab yang penting dalam mengelola dan mengawasi proses permohonan. Apa saja yang dilakukan oleh admin adalah sebagai berikut:

- Admin bisa memilih vendor apa saja yang ada di sistem, ketika admin memilih internal perusahaan (SI) maka secara otomatis akan muncul no. polisi dan *driver* dari SI sendiri yang sesuai dengan jenis kendaraan yang diminta oleh pemohon.
- Ketika admin memilih vendor lain, maka secara otomatis akan muncul no. polisi dan *driver* dari vendor tersebut yang sesuai dengan jenis kendaraan yang diminta oleh pemohon.
- Setelah admin memilih data kendaraan yang sesuai dari pemohon, akan submit data kendaraan ke *user*.
- Jika data kendaraan yang berasal dari SI maka akan submit di edit permohonan dengan status “disetujui”. Sedangkan data kendaraan yang berasal dari vendor maka akan submit di edit permohonan dengan status “permintaan vendor”. Jadi vendor bisa melihat data permohonan *user* atas permintaan admin. Seperti pada gambar IV.22 dan IV.23.

No.	PEMOHON	KENDARAAN	KEBUTUHAN	STATUS	KETERANGAN	ACTIONS
1	muhammad imi syaban	Avanza	2023-06-05 s/d 2023-06-06	Ditunggu		[Icons]
2	Mulya Ferdian	Avanza	2023-06-05 s/d 2023-06-06	Ditunggu		[Icons]
3	muhammad imi syaban	Innova Reborn	2023-06-05 s/d 2023-06-06	Ditunggu		[Icons]
4	Rika Mulda	Avanza	2023-06-06 s/d 2023-06-06	Ditunggu		[Icons]
5	Putri Tawarnate	Avanza	2023-06-06 s/d 2023-06-06	Ditunggu		[Icons]
6	Nelly	Avanza	2023-06-06 s/d 2023-06-08	Ditunggu		[Icons]
7	Susi Nurjanah	Innova Reborn	2023-06-27 s/d 2023-06-29	Ditunggu		[Icons]

Gambar IV.22 Hasil Menu Permohonan Mobil (admin)

Edit Permohonan Mobil

Vendor :

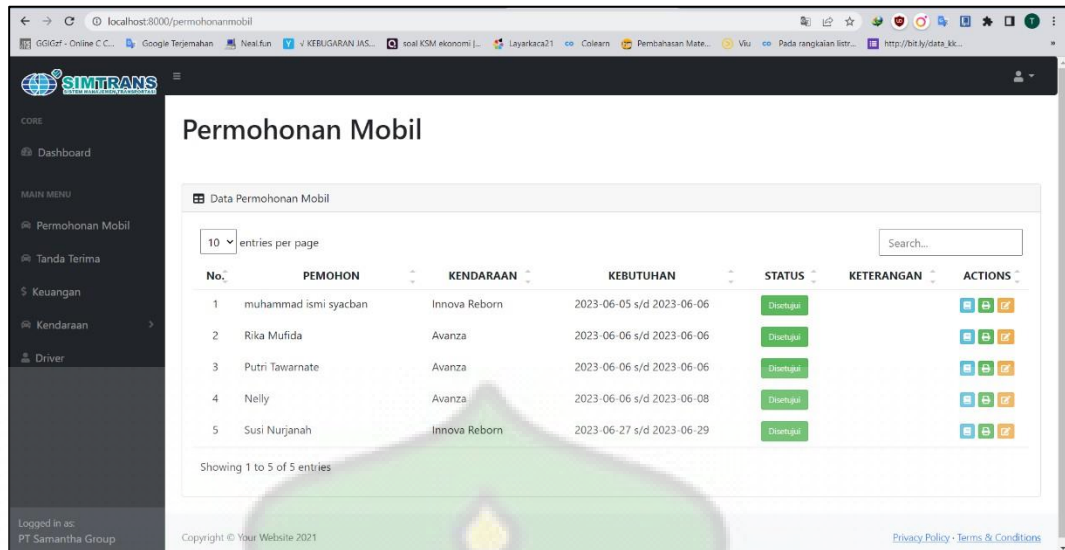
No. Polisi:

Driver :

Gambar IV.23 Hasil Edit Permohonan Mobil (admin)

d. Vendor

Di permohonan mobil vendor memiliki tanggung jawab yang penting dalam mengelola dan mengawasi proses permohonan yang hampir sama seperti admin. Yang dilakukan oleh vendor ketika vendor melihat data permohonan mobil yang diterima oleh *user*, vendor bisa memilih kendaraan dan *driver* di edit permohonan dan submit, akan berubah status dari “permintaan vendor” menjadi “disetujui”. Seperti pada gambar IV.24 dan IV.25.



Gambar IV.24 Hasil Menu Permohonan Mobil (vendor)

Gambar IV.25 Hasil Edit Permohonan Mobil (vendor)

Adapun coding untuk membuat fitur permohonan mobil, seperti pada gambar IV.26. Model permohonan mobil yang mempresentasikan entitas permohonan mobil dalam tabel 'permohonanmobil' di *database*. Menggunakan *trait HasFactory* untuk mendukung pembuatan *instance* model menggunakan *factory*. Menentukan nama tabel, *primary key*, dan kolom yang dapat diisi. Route permohonan mobil menggunakan *resourceful* routing untuk menghubungkan rute '/permohonanmobil' dengan *controller* *PermohonansmobilController*.

PermohonansmobilController.php di *function create()* digunakan untuk menampilkan halaman pembuatan permohonan mobil. Mengambil data jenis kendaraan dari tabel 'jeniskendaraan' dan merk kendaraan dari tabel

'merkkendaraan'. Mengambil data proyek dari tabel 'proyek'. Menampilkan halaman *create* permohonan mobil dengan data jenis, merk, dan proyek.

Sedangkan *create.blade.php* permohonan mobil menampilkan form untuk membuat permohonan mobil baru. Form ini memiliki input *field* untuk mengisi nama pemohon, merk kendaraan, jenis kendaraan, dan lainnya. Input *field* untuk nama pemohon diisi secara otomatis dengan nama pengguna yang sedang terautentikasi. Input *field* untuk merk kendaraan menggunakan *dropdown select* yang diisi dengan data merk kendaraan. Input *field* untuk jenis kendaraan menggunakan *dropdown select* yang awalnya kosong, tetapi akan terisi saat merk kendaraan dipilih.

1. Model permohonan mobil

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Permohonanmobil extends Model
{
    use HasFactory;
    protected $table = 'permohonanmobil';
    protected $primaryKey = 'id';
    protected $fillable = ['nama_pemohon', 'id_jenis', 'jumlah_orang',
        'id_proyek', 'tujuan', 'tgl_berangkat', 'tgl_pulang', 'status',
        'no_polisi', 'driver', 'no_telp_driver', 'keterangan', 'barang'];
}
```

2. Route permohonan mobil

```
Route::resource("/permohonanmobil", PermohonansmobilController::class);
```

3. PermohonanmobilController.php

```
public function create()
{
    $jenis = jeniskendaraan::pluck('jenis','id');
    $merk = array('' => '');
    foreach(merkkendaraan::all() as $row)
        $merk[$row->id] = $row->merk;
    $proyek = array('' => '');
    foreach(proyek::all() as $row)
        $proyek[$row->id] = $row->kode_proyek.' _ '.$row->nama_proyek;
    return view('permohonanmobil.create', compact('jenis', 'merk', 'proyek'));
}
```

4. Create.blade.php permohonan mobil

```
<table align="center">
<tr>
<td>Nama </td>
<td><input type="text" name="nama_pemohon" id="nama_pemohon"
class="form-control" value="{{ Auth::user()->name }}" readonly></td>
</tr>
<tr>
<td>Merk</td>
<td><select id="merk" name="merk" class="form-control">
<option value="">Pilih Merk
<option value="{{ $merk[$row->id] }}">{{ $row->merk }}
</select></td>
</tr>
<tr>
<td>Jenis</td>
<td><select id="jenis" name="jenis" class="form-control">
<option value="">Pilih Jenis
<option value="{{ $jenis[$row->id] }}">{{ $row->jenis }}
</select></td>
</tr>
</table>
```

Gambar IV.26 Algoritma dari fitur permohonan mobil

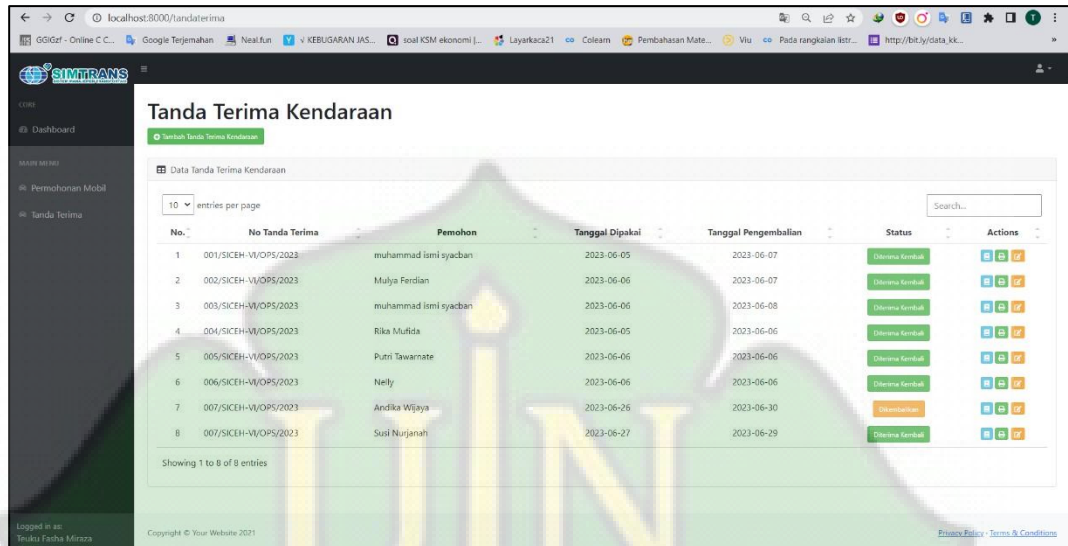
8. Fitur Menu Tanda Terima

Fitur ini menampilkan informasi tentang tanda terima kendaraan, termasuk pemohon (nama atau identitas pemohon), kendaraan yang dipinjam, kebutuhan atau tujuan penggunaan kendaraan, status tanda terima, keterangan atau catatan terkait, dan tindakan yang tersedia. Adapun fungsi fitur ini bagi setiap pengguna masing-masing, yaitu:

a. Admin

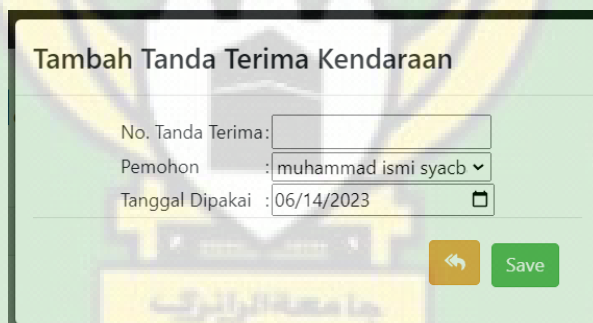
Setelah admin mengisi data kendaraan dari permohonan *user*, admin akan menambahkan tanda terima kendaraan untuk *user*. Yang mana admin bisa mengisi data tanda terima kendaraan dengan data, seperti no. Tanda terima, nama pemohon, dan tanggal dipakai dengan status “diserahkan” yang akan dikirim ke *user*. Tapi

kalo admin yang sudah memilih data kendaraan dari vendor maka vendor tersebut yang akan mengisinya, karna harus meminta persetujuan dari vendor. Seperti pada gambar IV.27 dan IV.28.



No.	No Tanda Terima	Pemohon	Tanggal Dipakai	Tanggal Pengembalian	Status	Actions
1	001/SICEH-W/OPS/2023	muhammad ismi syacban	2023-06-05	2023-06-07	Diterima Kembali	[Edit] [Delete] [Add]
2	002/SICEH-W/OPS/2023	Mulya Ferdian	2023-06-06	2023-06-07	Diterima Kembali	[Edit] [Delete] [Add]
3	003/SICEH-W/OPS/2023	muhammad ismi syacban	2023-06-06	2023-06-08	Diterima Kembali	[Edit] [Delete] [Add]
4	004/SICEH-W/OPS/2023	Rika Mutiada	2023-06-05	2023-06-06	Diterima Kembali	[Edit] [Delete] [Add]
5	005/SICEH-W/OPS/2023	Putri Tawarnate	2023-06-06	2023-06-06	Diterima Kembali	[Edit] [Delete] [Add]
6	006/SICEH-W/OPS/2023	Nelly	2023-06-06	2023-06-06	Diterima Kembali	[Edit] [Delete] [Add]
7	007/SICEH-W/OPS/2023	Andika Wijaya	2023-06-26	2023-06-30	Dikembalikan	[Edit] [Delete] [Add]
8	007/SICEH-W/OPS/2023	Susi Nurjanah	2023-06-27	2023-06-29	Diterima Kembali	[Edit] [Delete] [Add]

Gambar IV.27 Hasil Menu Tanda Terima Kendaraan (admin)



Tambah Tanda Terima Kendaraan

No. Tanda Terima:

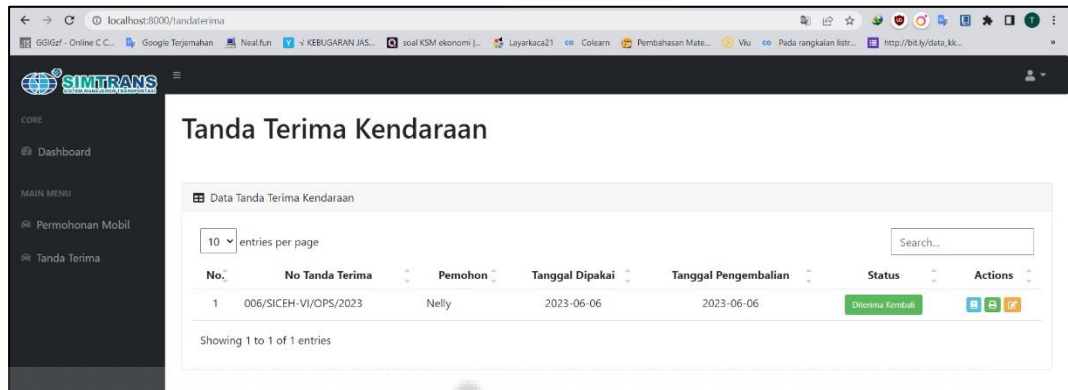
Pemohon : muhammad ismi syacb

Tanggal Dipakai : 06/14/2023

Gambar IV.28 Hasil Tambah Tanda Terima Kendaraan (admin)

b. User

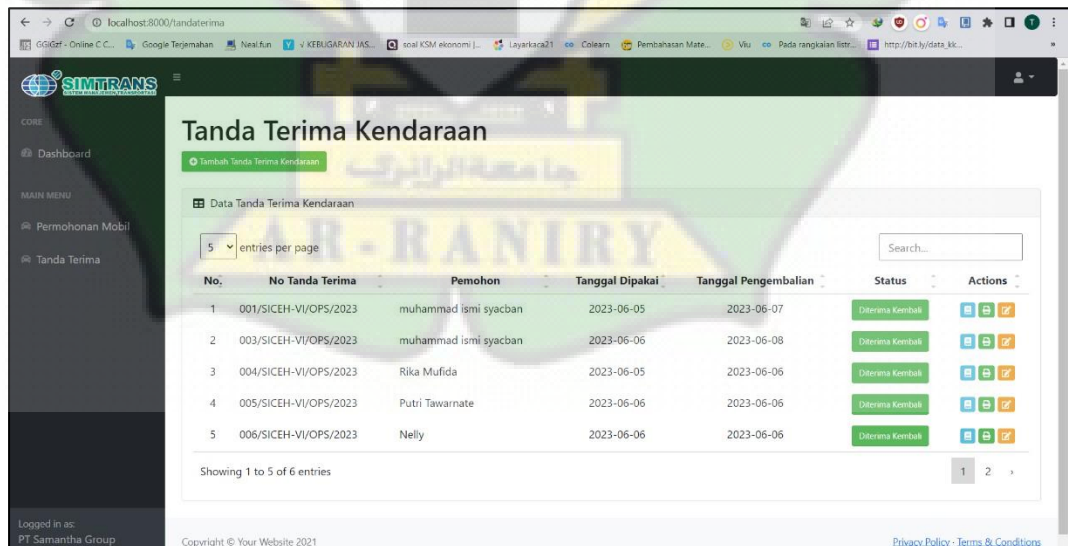
Setelah mendapat data tanda terima kendaraan dari admin, *user* akan mengisi tanggal pengembalian dan status. *User* akan memilih “dikembalikan” di status agar data yang diterima oleh *user* akan dikembalikan kepada admin. Setelah admin mendapatkan kembali data dari *user*, admin akan memilih “diterima kembali” di status pada edit di actions. Seperti pada gambar IV.29.



Gambar IV.29 Hasil Menu Tanda Terima Kendaraan (*user*)

c. Vendor

Setelah vendor mengisi data kendaraan dari permohonan *user*, vendor akan menambahkan tanda terima kendaraan untuk *user*. Yang mana vendor bisa mengisi data tanda terima kendaraan dengan data, seperti no. Tanda terima, nama pemohon, dan tanggal dipakai dengan status “diserahkan” yang akan dikirim ke *user*. Tapi kalo vendor yang sudah memilih data kendaraan dari vendor maka vendor tersebut yang akan mengisinya, karna harus meminta persetujuan dari vendor. Seperti pada gambar IV.30.



Gambar IV.30 Hasil Menu Tanda Terima Kendaraan (*vendor*)

Adapun coding untuk pembuatan fitur menu tanda terima kendaraan seperti pada gambar IV.31. Model tanda terima yang merepresentasikan entitas tanda terima kendaraan dalam tabel '**tandaterima**' di *database* . Menggunakan *trait HasFactory* untuk mendukung pembuatan *instance* model menggunakan *factory*. Menentukan nama tabel, *primary key*, dan kolom yang dapat diisi. *Route* tanda terima menggunakan *resourceful* routing untuk menghubungkan rute '/tandaterima' dengan *controller* *tandaterimaController*.

Kemudian "**tandaterimaController.php**" di *function* *edit()* memiliki fungsi untuk menampilkan halaman pengeditan tanda terima. Mengambil data tanda terima dengan ID yang sesuai dari tabel '**tandaterima**'. Menampilkan halaman edit tanda terima dengan data tanda terima yang akan diedit.

Yang terakhir *edit.blade.php* tanda terima menampilkan form untuk mengedit tanda terima. Form ini memiliki input *field* untuk mengedit status dan tanggal selesai. Input *field* untuk status menggunakan *dropdown select* dengan opsi "**Diterima**" dan "**Dikembalikan**". Input *field* untuk tanggal selesai menggunakan *date picker* dengan nilai default tanggal saat ini.

1. Model tanda terima

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class tandaterima extends Model
{
    use HasFactory;
    protected $table = 'tandaterima';
    protected $primaryKey = 'id';
    protected $fillable = ['no_tt', 'id_permohonanmobil',
        'tgl_mulai', 'tgl_selesai', 'status'];
}
```

2. Route tanda terima

```
Route::resource("/tandaterima", tandaterimaController::class);
```

3. TandaterimaController.php

```
public function edit($id)
{
    $standaterima = tandaterima::find($id);
    return view('tandaterima.edit', [
        'tandaterima' => $standaterima
    ]);

    // $service = Kendaraan::find($id);
    // return view('service.edit')->with('service', $service);
}
```

4. Edit.blade.php tanda terima

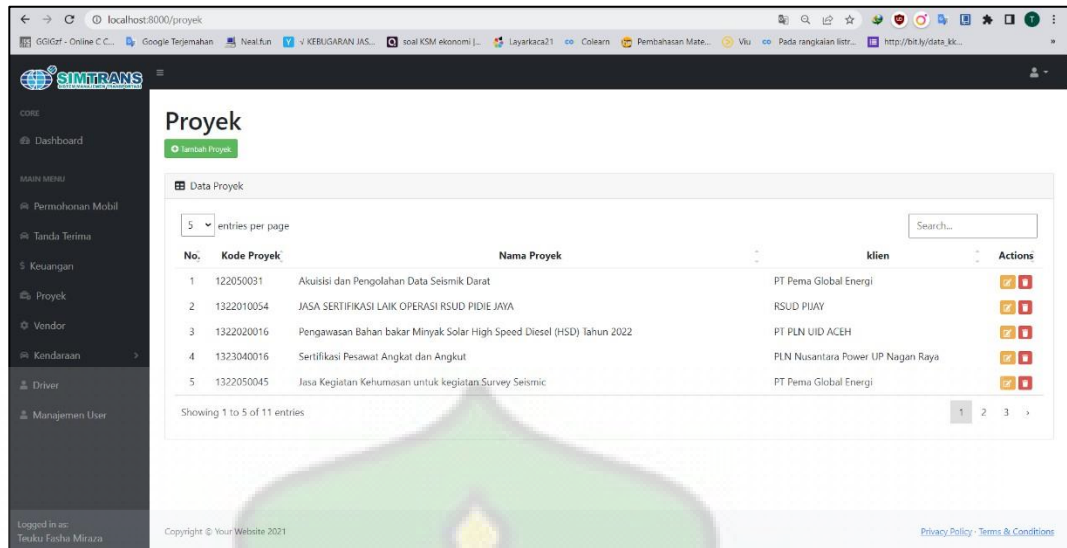
```
<table align="center">
  @if(Auth::user()->level == "user")
    <tr>
      <td>Status</td>
      <td>
        <div>
          {!! Form::select('status', array("Diterima"=>"Diterima",
            "Dikembalikan"=>"Dikembalikan"),null,array('id' => 'jeniskendaraan',
            'style'=>'width: 188px')) !!}
        </div>
      </td>
    </tr>
    <tr>
      <td>Tanggal Dikembalikan</td>
      <td>
        <div class="input-append date" id="dp3" data-date="2023-05-23"
          data-date-format="yyyy-mm-dd">
          <input class="span2" size="16" type="date"
            value="{{ \Carbon\Carbon::now()->format('Y-m-d') }}"
            name="tgl_selesai" id="tgl_selesai" style="width: 190px">
          <span class="add-on"><i class="icon-th"></i></span>
        </div>
      </td>
    </tr>
  </tr>
</table>
```

Gambar IV.31 Algoritma dari fitur tanda terima kendaraan

9. Fitur Menu Proyek

Fitur ini berfungsi untuk membuat dan mengelola proyek-proyek yang terkait dengan transportasi. Admin dapat mengisi informasi seperti kode proyek, nama proyek, dan klien yang terkait dengan proyek tersebut. Admin juga bisa melakukan pembaruan atau penghapusan proyek yang sudah ada jika diperlukan.

Fitur ini juga menghubungkan proyek dengan permohonan mobil. Admin dapat menetapkan proyek yang relevan dengan permohonan mobil tertentu, sehingga memudahkan dalam mengorganisasi dan melacak pemakaian kendaraan berdasarkan proyek yang sedang berjalan. Seperti pada gambar IV.32.



Gambar IV.32 Hasil Menu Proyek

Adapun coding untuk pembuatan fitur menu proyek seperti pada gambar IV.33. Model proyek yang mempresentasikan entitas proyek dalam tabel 'proyek' di *database*. Menggunakan *trait HasFactory* untuk mendukung pembuatan *instance* model menggunakan *factory*. Menentukan nama tabel, *primary key*, dan kolom yang dapat diisi. *Route* proyek menggunakan *resourceful* routing untuk menghubungkan rute '/proyek' dengan *controller* *proyekController*.

ProyekController.php di *function store()* digunakan untuk menyimpan data proyek yang dikirim melalui *request*. Mengambil semua data dari *request*. Membuat entitas proyek baru dengan menggunakan metode *create* dan menyimpan data yang diambil dari *request*. Kembali ke halaman sebelumnya dengan pesan flash yang menyatakan bahwa proyek telah ditambahkan.

Edit.blade.php proyek menampilkan form untuk mengedit data proyek. Form ini memiliki input *field* untuk mengedit kode proyek, nama proyek, dan klien. Nilai dari input *field* diinisialisasi dengan nilai saat ini dari atribut *kode_proyek*, *nama_proyek*, dan *klien*.

1. Model proyek

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Proyek extends Model
{
    use HasFactory;
    protected $table = 'proyek';
    protected $primaryKey = 'id';
    protected $fillable = ['kode_proyek', 'nama_proyek', 'klien'];
}
```

2. Route proyek

```
Route::resource("/proyek", proyekController::class);
```

3. ProyekController.php

```
public function store(Request $request)
{
    $input = $request->all();
    proyek::create($input);
    return back()->with('flash_message', 'Proyek Addedd!');
}
```

4. Edit.blade.php proyek

```
<table align="center">
<tr>
    <td>Kode Proyek</td>
    <td></td>
    <td><input type="text" name="kode_proyek" id="kode_proyek"
        class="form-control" value="{{ $proyek->kode_proyek }}"></td>
</tr>
<tr>
    <td>Nama Proyek</td>
    <td></td>
    <td><input type="text" name="nama_proyek" id="nama_proyek"
        class="form-control" value="{{ $proyek->nama_proyek }}"></td>
</tr>
<tr>
    <td>Klien</td>
    <td></td>
    <td><input type="text" name="klien" id="klien"
        class="form-control" value="{{ $proyek->klien }}"></td>
</tr>
</table>
```

Gambar IV.33 Algoritma dari fitur proyek

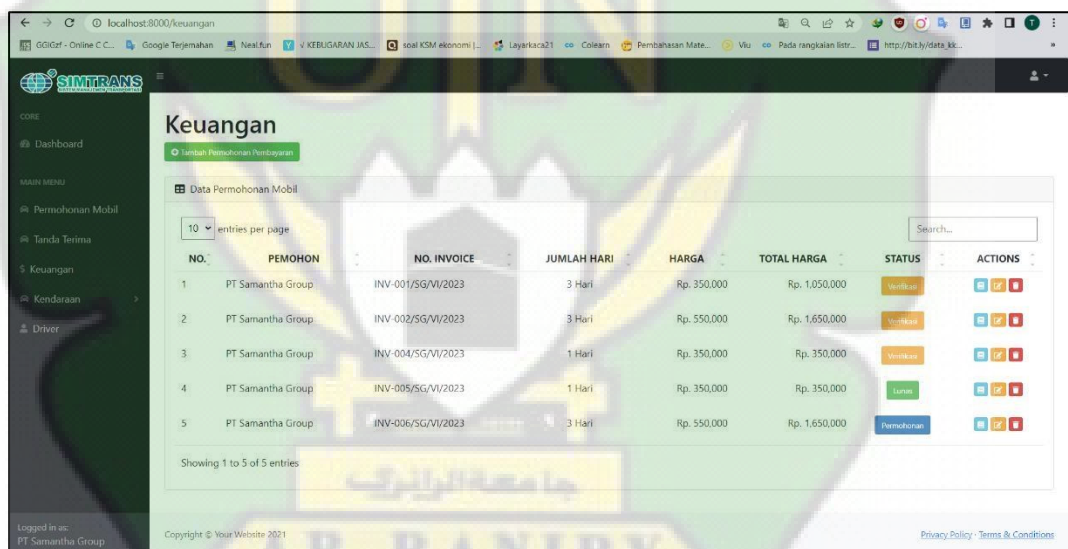
10. Fitur Menu Keuangan

Fitur keuangan memiliki fungsi masing-masing di setiap pengguna, yaitu sebagai berikut:

a. Vendor

Fitur keuangan ini memungkinkan vendor untuk melihat informasi tentang pemohon, yaitu pengguna atau entitas yang melakukan permohonan sewa kendaraan. Informasi yang ditampilkan mungkin mencakup nama pemohon atau nomor identifikasi pemohon.

Fitur ini juga bisa menampilkan informasi *invoice*, menampilkan informasi jumlah hari, menampilkan informasi harga, menampilkan informasi total harga, menampilkan informasi status. Setelah itu data tersebut otomatis terkirim ke admin dan *approval*. Seperti pada gambar IV.34.



The screenshot shows a web application interface for a vendor. The main menu on the left includes 'Dashboard', 'Permohonan Mobil', 'Tanda Terima', 'Keuangan', 'Kendaraan', and 'Driver'. The 'Keuangan' (Finance) menu is selected, displaying a table titled 'Data Permohonan Mobil'. The table has columns for NO., PEMOHON, NO. INVOICE, JUMLAH HARI, HARGA, TOTAL HARGA, STATUS, and ACTIONS. There are 5 entries in the table, all from 'PT Samantha Group'. The status of the requests is 'Verifikasi' (Verification) for the first four and 'Lulus' (Approved) for the fifth. The bottom of the page shows 'Showing 1 to 5 of 5 entries' and a copyright notice for 2021.

NO.	PEMOHON	NO. INVOICE	JUMLAH HARI	HARGA	TOTAL HARGA	STATUS	ACTIONS
1	PT Samantha Group	INV-001/SG/VV/2023	3 Hari	Rp. 350,000	Rp. 1,050,000	Verifikasi	Edit Hapus
2	PT Samantha Group	INV-002/SG/VV/2023	3 Hari	Rp. 550,000	Rp. 1,650,000	Verifikasi	Edit Hapus
3	PT Samantha Group	INV-004/SG/VV/2023	1 Hari	Rp. 350,000	Rp. 350,000	Verifikasi	Edit Hapus
4	PT Samantha Group	INV-005/SG/VV/2023	1 Hari	Rp. 350,000	Rp. 350,000	Lulus	Edit Hapus
5	PT Samantha Group	INV-006/SG/VV/2023	3 Hari	Rp. 550,000	Rp. 1,650,000	Permohonan	Edit Hapus

Gambar IV.34 Hasil Menu Keuangan (vendor)

b. Approval

Fitur keuangan di bagian *approval* untuk melihat informasi tentang pemohon yang terkait dengan transaksi keuangan. Setelah itu *approval* menyetujui / menolak data pembayaran sewa kendaraan keuangan dari vendor. Seperti pada gambar IV.35.

Keuangan

Data Permohonan Mobil

10 entries per page

NO.	PEMOHON	NO. INVOICE	JUMLAH HARI	HARGA	TOTAL HARGA	STATUS	ACTIONS
1	PT Samantha Group	INV-001/SG/VI/2023	3 Hari	Rp. 350,000	Rp. 1,050,000	Verifikasi	[Icons]
2	PT Samantha Group	INV-002/SG/VI/2023	3 Hari	Rp. 550,000	Rp. 1,650,000	Verifikasi	[Icons]
3	PT Samantha Group	INV-004/SG/VI/2023	1 Hari	Rp. 350,000	Rp. 350,000	Verifikasi	[Icons]

Showing 1 to 3 of 3 entries

Gambar IV.35 Hasil Menu Keuangan (*approval*)

c. Admin

Fitur keuangan di bagian admin untuk melihat informasi tentang pemohon yang terkait dengan transaksi keuangan. Admin hanya menyetujui atau menolak. Jika admin menyetujui maka data keuangan akan dikirim ke *approval* dan setelah mendapatkan kembali data keuangan yang sudah disetujui *approval*, maka statusnya akan menjadi “kasir” yang artinya sedang melakukan proses pembayaran keuangan dari vendor. Admin akan mengubah status menjadi “lunas”. Seperti pada gambar IV.36.

Keuangan

Data Permohonan Mobil

10 entries per page

NO.	PEMOHON	NO. INVOICE	JUMLAH HARI	HARGA	TOTAL HARGA	STATUS	ACTIONS
1	PT Samantha Group	INV-002/SG/VI/2023	3 Hari	Rp. 550,000	Rp. 1,650,000	Verifikasi	[Icons]
2	PT Samantha Group	INV-001/SG/VI/2023	3 Hari	Rp. 350,000	Rp. 1,050,000	Verifikasi	[Icons]
3	PT Samantha Group	INV-006/SG/VI/2023	3 Hari	Rp. 550,000	Rp. 1,650,000	Permohonan	[Icons]
4	PT Samantha Group	INV-004/SG/VI/2023	1 Hari	Rp. 350,000	Rp. 350,000	Verifikasi	[Icons]
5	PT Samantha Group	INV-005/SG/VI/2023	1 Hari	Rp. 350,000	Rp. 350,000	Lunas	[Icons]

Showing 1 to 5 of 5 entries

Gambar IV.36 Hasil Menu Keuangan (vendor)

Adapun coding untuk pembuatan fitur menu keuangan seperti pada gambar IV.37. Model keuangan yang mempresentasikan entitas keuangan dalam tabel 'keuangan' di *database* . Menggunakan *trait HasFactory* untuk mendukung pembuatan *instance* model menggunakan *factory*. Menentukan nama tabel, *primary key*, dan kolom yang dapat diisi. *Route* keuangan menggunakan *resourceful* routing untuk menghubungkan rute *'/keuangan'* dengan *controller keuanganController*.

keuanganController.php di *function create()* digunakan untuk menampilkan halaman pembuatan keuangan baru. Mendapatkan pengguna yang saat ini terautentikasi. Mengambil data tanda terima yang memenuhi kondisi tertentu dan mengaturnya dalam array dengan menggunakan *pluck*. Mengirimkan data tanda terima ke halaman pembuatan keuangan.

Create.blade.php keuangan menampilkan form untuk membuat keuangan baru. Form ini memiliki input *field* untuk nomor *invoice* dan pilihan tanda terima. Nilai dari input *field* nomor *invoice* diisikan oleh pengguna. Pilihan tanda terima diisi menggunakan *select box* yang memanfaatkan data tanda terima yang diteruskan dari *controller*.

1. Model keuangan

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class keuangan extends Model
{
    use HasFactory;
    protected $table = 'keuangan';
    protected $primaryKey = 'id';
    protected $fillable = ['no_invoice', 'id_vendor',
        'id_tt', 'jml_hari', 'harga', 'total_harga', 'status'];
}
```

2. Route keuangan

```
Route::resource("/keuangan", keuanganController::class);
```

3. KeuanganController.php

```
public function create()
{
    $user = Auth::user();
    $standaterima = [ NULL => 'Pilih Tanda Terima' ] +
    tandaterima::where('id_vendor', $user->homebase)
    ->where('status', 'Diterima Kembali')
    ->where('invoice', NULL)->pluck('no_tt', 'id')->all();

    return view('keuangan.create', compact('tandaterima'));
}
```

4. Create.blade.php keuangan

```
<table align="center">
<tr>
<td>No. Invoice </td>
<td></td>
<td><input type="text" name="no_invoice" id="no_invoice"
class="form-control"></td>
</tr>
<tr>
<td>Tanda Terima</td>
<td></td>
<td>{!! Form::select('id_tt', $standaterima, null,
array('id' => 'tandaterima', 'style'=>'width: 188px')) !!}</td>
</tr>
```

Gambar IV.37 Algoritma dari fitur keuangan

BAB V

KESIMPULAN DAN SARAN

V.1 Kesimpulan

Setelah dilakukan penelitian dan evaluasi, dapat disimpulkan bahwa:

1. Aplikasi SIMTRANS ini sudah berhasil dibuat menggunakan *framework Laravel* pada PT Surveyor Indonesia, yang menunjukkan kemajuan dalam efisiensi pengelolaan data transportasi di perusahaan tersebut. Dengan penerapan aplikasi ini, PT Surveyor Indonesia dapat memperoleh manfaat dalam pengelolaan data transportasi, termasuk peningkatan efisiensi dan keterpaduan informasi terkait transportasi perusahaannya.
2. Aplikasi sistem informasi manajemen transportasi berbasis web menggunakan *framework Laravel* telah membawa manfaat dalam meningkatkan efisiensi pengelolaan data transportasi di PT Surveyor Indonesia.
3. Fitur-fitur seperti vendor, keuangan, tanda terima kendaraan memberikan nilai tambah dalam pengelolaan transportasi, pengendalian keuangan, dan pemantauan kendaraan. Hal ini membantu PT Surveyor Indonesia dalam meningkatkan efektivitas operasional dan pengambilan keputusan.

V.2 Saran

Setelah melakukan penelitian ini, terdapat beberapa saran untuk penelitian selanjutnya yang dapat dilakukan agar dapat meningkatkan manfaat dan kegunaan aplikasi sistem informasi manajemen transportasi berbasis *web* bagi penggunaanya. Adapun saran yang dapat diberikan untuk penelitian selanjutnya adalah:

1. Dalam pengembangan sistem ini, perlu diperhatikan aspek keamanan data. Melalui implementasi mekanisme otentikasi dan otorisasi yang kuat, serta enkripsi data sensitif, PT Surveyor Indonesia dapat meningkatkan keamanan sistem agar data transportasi tetap terlindungi.

2. Penting untuk memberikan pelatihan kepada pengguna sistem agar mereka dapat mengoptimalkan penggunaan sistem ini. Selain itu, perlu dilakukan pemeliharaan rutin, termasuk pembaruan keamanan dan pemecahan masalah, untuk memastikan sistem tetap berjalan dengan baik.



DAFTAR PUSTAKA

- Januhari, Ni Nyoman Utami, I. Ketut Dedy Suryawan, and I. Made Adi Purwantara. "MANAJEMEN E-SCHEDULING TRANSPORTASI DENGAN METODE IS RESEARCH." *Jurnal Teknologi Informasi dan Komputer* 5.2 (2019).
- Malang, P. K., T. Informatika, and P. K. Malang. "Implementasi *Framework Laravel* Dalam Sistem." vol 2: 35-42.
- Hanif, Afril Caesar Muhammad, and Magdalena A. Ineke Pakereng. "Pengembangan Aplikasi Sistem Informasi Operasional Bus Berbasis *Website* Menggunakan *Framework Laravel*." *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)* 8.3 (2021): 1027-1039.
- Sugianto, Sugianto, and Muhammad Arief Kurniawan. "Tingkat Ketertarikan Masyarakat terhadap Transportasi Online, Angkutan Pribadi dan Angkutan Umum Berdasarkan Persepsi." *Jurnal Teknologi Transportasi dan Logistik* 1.2 (2020): 51-58.
- Hidayah, Abdurrahman, and Ahmad Yani. "MEMBANGUN WEBSITE SMA PGRI GUNUNG RAYA RANAU MENGGUNAKAN PHP DAN MYSQL." *JTIM: Jurnal Teknik Informatika Mahakarya* 2.2 (2019): 41-52.
- Rustandi, Riyani Aprilia, Wahyu Hidayat, and Ely Rosely. "Perangkat Lunak Manajemen *User* Untuk Oracle 11g Dengan Java." *eProceedings of Applied Science* 1.3 (2015).
- Wibowati, JanuarIda. "Pengaruh Kualitas Pelayanan Terhadap Kepuasan Pelanggan Pada Pt Muarakati Baru Satu Palembang." *Jurnal Manajemen* 8.2 (2020): 15-31.
- (2023). Retrieved 27 February 2023, from <http://lib.unnes.ac.id/38448/1/4611413042.pdf>
- Hamzan, Muhammad Abdulloh, I. Made Ari Nrratha, and I. Ketut Wiryajati. "Rancang Bangun Sistem Pemantauan Daya Listrik Berbasis Android Menggunakan Teknologi React Native." *DIELEKTRIKA* 9.1 (2022): 42-50.
- Setiyani, Lila, and Evelyn Tjandra. "Analisis kebutuhan fungsional aplikasi penanganan keluhan mahasiswa studi kasus: stmik rosma karawang." *Jurnal Inovasi Pendidikan dan Teknologi Informasi (JIPTI)* 2.1 (2021): 8-17.
- Perbedaan Kebutuhan Fungsional dan Non Fungsional, Lengkap Contoh dan Penjelasan. (2023). Retrieved 5 March 2023, from <https://www.materidosen.com/2017/03/perbedaan-kebutuhan-fungsional-dan-non.html>.

Yusri, A., Hayati, U., & Amalia, D. R. (2022). SISTEM INFORMASI LALU LINTAS HARIAN RATA-RATA BERBASIS WEB MENGGUNAAN FRAMEWORK LARAVEL. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 6(2), 446-453.



LAMPIRAN

1. Merk kendaraan

a. Perintah memasukkan data merk kendaraan

```
<form action="{{ url('merkkendaraan') }}" method="post">
    {!! csrf_field() !!}
    <table align="center">
        <tr class="hidden">
            <td>Id</td>
            <td>:</td>
            <td><input type="text" name="id" id="id"
class="form-control" value="{{
\App\Models\Merkkendaraan::orderBy('id', 'desc')->first()->id+1 }}"
readonly></td>
        </tr>
        <tr>
            <td>Merk</td>
            <td>:</td>
            <td><input type="text" name="merk" id="merk"
class="form-control"></td>
        </tr>
    </table>
```

b. Perintah melakukan perbaikan data merk kendaraan

```
<form action="/merkkendaraan/{{ $merkkendaraan->id }}"
method="post">
    @method('put')
    {!! csrf_field() !!}
    <table align="center">
        <tr>
            <td>Merk</td>
            <td>:</td>
            <td><input type="text" name="merk" id="merk"
class="form-control" value="{{ $merkkendaraan->merk }}"></td>
        </tr>
    </table>
```

c. Perintah melakukan hapus data merk kendaraan

```
<form method="POST" action="{{ url('/merkkendaraan' . '/' . $item-
>id) }}" accept-charset="UTF-8" style="display:inline">
```

```

{{ method_field('DELETE') }}
{{ csrf_field() }}
<button type="submit" class="btn
btn-danger btn-sm" title="Delete Kendaraan" onclick="return
confirm('Confirm delete?')"><i class="fa fa-trash" aria-
hidden="true"></i> Delete</button>
</form>

```

d. Perintah menampilkan data merk kendaraan

```

<form action="/merkkendaraan/{{ $merkkendaraan->id }}"
method="post">
  @method('get')
  {!! csrf_field() !!}
  <table align="center">
    <tr>
      <td>Merk</td>
      <td>:</td>
      <td><input type="text" name="merk" id="merk"
class="form-control" value="{{ $merkkendaraan->merk }}"></td>
    </tr>
  </table>

```

2. Jenis Kendaraan

a. Perintah memasukkan data jenis kendaraan

```

<form action="{{ url('jeniskendaraan') }}" method="post">
  {!! csrf_field() !!}
  <table align="center">
    <tr class="hidden">
      <td>Id</td>
      <td>:</td>
      <td><input type="text" name="id" id="id"
class="form-control" value="{{
\App\Models\jeniskendaraan::orderBy('id', 'desc')->first()->id+1 }}"
readonly></td>
    </tr>
    <tr>
      <td>Merk</td>
      <td>:</td>
      <td>{!! Form::select('id_merk', $merkkendaraan,
array('id' => 'unit', 'style'=>'width: 190px')) !!}</td>
    </tr>
    <tr>
      <td>Jenis</td>
      <td>:</td>

```

```

        <td><input type="text" name="jenis" id="jenis"
class="form-control"></td>
    </tr>
</table>
<div class="modal-footer">
    <a href="{{ url('jeniskendaraan')}}" title="Back"
class="btn btn-warning"><span class="fa fa-reply-all"></span></a>
    <input type="submit" value="Save" class="btn btn-
success"></br>
</div>
</form>

```

b. Perintah melakukan perbaikan data jenis kendaraan

```

<form action="/jeniskendaraan/{{ $jeniskendaraan->id }}"
method="post">
    @method('put')
    {!! csrf_field() !!}
    <table align="center">
        <tr>
            <td>Merk</td>
            <td>:</td>
            <td>{!! Form::select('id_merk', $merkkendaraan,
$jeniskendaraan->id_merk, array('id' => 'unit', 'style'=>'width:
190px')) !!}</td>
        </tr>
        <tr>
            <td>Jenis</td>
            <td>:</td>
            <td><input type="text" name="jenis" id="jenis"
class="form-control" value="{{ $jeniskendaraan->jenis }}"></td>
        </tr>
    </table>

```

c. Perintah melakukan hapus data jenis kendaraan

```

<form method="POST" action="{{ url('/jeniskendaraan' . '/' . $item-
>id) }}" accept-charset="UTF-8" style="display:inline">
    {{ method_field('DELETE') }}
    {{ csrf_field() }}
    <button type="submit" class="btn
btn-danger btn-sm" title="Delete Kendaraan" onclick="return
confirm('Confirm delete?')"><i class="fa fa-trash" aria-
hidden="true"></i> Delete</button>
</form>

```

d. Perintah menampilkan data jenis kendaraan

```
<table align="center">
  <tr>
    <td>Merk</td>
    <td>:</td>
    <td><input type="text" name="merk" id="merk"
class="form-control" value="{{ $merkkendaraan[$jeniskendaraan-
>id_merk] }}"></td>
  </tr>
  <tr>
    <td>Jenis</td>
    <td>:</td>
    <td><input type="text" name="merk" id="merk"
class="form-control" value="{{ $jeniskendaraan->jenis }}"></td>
  </tr>
</table>
```

3. Kendaraan

a. Perintah memasukkan data kendaraan

```
<form action="{{ url('kendaraan') }}" method="post">
  {!! csrf_field() !!}
  <table align="center">
    <tr class="hidden">
      <td>Id</td>
      <td>:</td>
      <td><input type="text" name="id" id="id"
class="form-control" value="{{ \App\Models\Kendaraan::orderBy('id',
'desc')->first()->id+1 }}" readonly></td>
    </tr>
    <tr>
      <td>Merk</td>
      <td>:</td>
      <td>{!! Form::select('id_merk', $merk, null,
array('id' => 'merk', 'style'=>'width: 190px')) !!}</td>
    </tr>
    <tr>
      <td>Jenis</td>
      <td>:</td>
      <td>{!! Form::select('id_jenis', array(), null,
array('id' => 'jeniskendaraan', 'style'=>'width: 190px')) !!}</td>
    </tr>
    <tr>
      <td>No. Polisi</td>
      <td>:</td>
```

```

        <td><input type="text" name="platmobil"
id="platmobil" class="form-control"></td>
    </tr>
    <tr>
        <td>Tahun</td>
        <td><input type="text" name="tahun" id="tahun"
class="form-control"></td>
    </tr>
    <tr>
        <td>Masa Pajak </td>
        <td><input type="date" name="pajak" id="pajak"
class="form-control" style='width: 190px'></td>
    </tr>
    <tr>
        <td>Bahan Bakar</td>
        <td>{!! Form::select('bbm', [
            '' => '',
            'Pertamax Racing' => 'Pertamax Racing',
            'Pertamax Turbo' => 'Pertamax Turbo',
            'Pertamax' => 'Pertamax',
            'Pertalite' => 'Pertalite',
            'Premium' => 'Premium',
            'Pertamina Dex ' => 'Pertamina Dex ',
            'Dexlite ' => 'Dexlite ',
            'Solar ' => 'Solar ',], null, ['style' =>
'width: 190px']) !!}</td>
    </tr>
    <tr>
        <td>Harga</td>
        <td><input type="text" name="harga" id="harga"
class="form-control"></td>
    </tr>
    <tr class="hidden">
        <td>vendor </td>
        <td><input type="text" name="id_vendor"
id="id_vendor" value="{{ $user->homebase }}" class="form-
control"></td>
    </tr>
</table>

```

b. Perintah melakukan perbaikan data kendaraan

```
<script>
$(document).ready(function() {
    $('#merk').change(function(){
        $.get('/merk/' + this.value, function(jeniskendaraans){
            var $jeniskendaraan = $('#jeniskendaraan');

            $jeniskendaraan.find('option').remove().end();

            $.each(jeniskendaraans, function(index,
jeniskendaraan){
                $jeniskendaraan.append('<option value="' +
jeniskendaraan.id + '">' + jeniskendaraan.jenis + '</option>');
            });
        });

        $(".merk_id option[value='0']").attr("disabled","disabled");
    });
</script>
```

c. Perintah melakukan hapus data kendaraan

```
<form method="POST" action="{ { url('/kendaraan' . '/' . $item->id)
}} " accept-charset="UTF-8" style="display:inline">
    { { method_field('DELETE') } }
    { { csrf_field() } }
    <button type="submit" class="btn
btn-danger btn-xs" title="Delete Kendaraan" onclick="return
confirm('Confirm delete?') "><i class="fa fa-trash" aria-
hidden="true"></i></button>
</form>
```

d. Perintah menampilkan data kendaraan

```
<table align="center">
    <tr>
        <td>No. Polisi</td>
        <td>:</td>
        <td><input type="text" name="platmobil"
id="platmobil" class="form-comtrol" value="{ { $kendaraan->platmobil
}} "></td>
    </tr>
    <tr>
        <td>Merk kendaraan</td>
```

```

        <td>:</td>
        <td><input type="text" name="id_merk"
id="id_merk" class="form-control" value="{{ $kendaraan->id_merk
}}"></td>
    </tr>
    <tr>
        <td>Masa Pajak</td>
        <td>:</td>
        <td><input type="text" name="pajak" id="pajak"
class="form-control" value="{{ $kendaraan->pajak }}"></td>
    </tr>
</table>

```

4. Service

a. Perintah memasukkan data service

```

<table align="center">
    <tr>
        <td>Id</td>
        <td>:</td>
        <td><input type="text" name="id" id="id"
class="form-control" value="{{ \App\Models\Service::orderBy('id',
'desc')->first()->id+1 }}" readonly></td>
    </tr>
    <tr>
        <td>Kendaraan</td>
        <td>:</td>
        <td>{!! Form::select('id_kendaraan', $kendaraan,
null, array('id' => 'unit', 'style'=>'width: 190px')) !!}</td>
    </tr>
    <tr>
        <td>Tgl Service Mobil</td>
        <td>:</td>
        <td>{!! Form::date('tgl_service_mobil',
null,['placeholder'=>'dd mm yy','style'=>'width: 190px']) !!}</td>
    </tr>
    <tr>
        <td>Perbaiki </td>
        <td>:</td>
        <td>{!! Form::text('perbaiki',
null,['placeholder'=>'Input keterangan service','style'=>'width:
190px']) !!}</td>
    </tr>
    <tr>
        <td>Biaya Service </td>
        <td>:</td>

```

```

        <td>{!! Form::text('biaya_service',
null,['placeholder'=>'Input biaya service','style'=>'width: 190px'])
!!}</td>
    </tr>
    <tr class="hidden">
        <td>vendor </td>
        <td>:</td>
        <td><input type="text" name="id_vendor"
id="id_vendor" value="{{ $user->homebase }}" class="form-
comtrol"></td>
    </tr>
</table>

```

b. Perintah melakukan perbaikan data service

```

<table align="center">
    <tr>
        <td>Kendaraan</td>
        <td>:</td>
        <td>{!! Form::select('id_kendaraan', $kendaraan,
$service->id_kendaraan, array('id' => 'unit', 'style'=>'width:
190px')) !!}</td>
    </tr>
    <tr>
        <td>Tgl Service Mobil</td>
        <td>:</td>
        <td><input type="date" name="tgl_service_mobil"
id="tgl_service_mobil" class="form-comtrol" value="{{ $service-
>tgl_service_mobil }}" style="width: 190px"></td>
    </tr>
    <tr>
        <td>Perbaiki </td>
        <td>:</td>
        <td><input type="text" name="perbaiki"
id="perbaiki" class="form-comtrol" value="{{ $service->perbaiki
}}"></td>
    </tr>
    <tr>
        <td>Biaya Service</td>
        <td>:</td>
        <td><input type="text" name="biaya_service"
id="biaya_service" class="form-comtrol" value="{{ $service-
>biaya_service }}"></td>
    </tr>
</table>

```

c. Perintah melakukan hapus data service

```

<form method="POST" action="{{ url('/service' . '/' . $item->id) }}"
accept-charset="UTF-8" style="display:inline">

```

```

        {{ method_field('DELETE') }}
        {{ csrf_field() }}
        <button type="submit" class="btn
btn-danger btn-sm" title="Delete service" onclick="return
confirm('Confirm delete?')"><i class="fa fa-trash" aria-
hidden="true"></i></button>
    </form>

```

d. Perintah menampilkan data service

```

<table align="center">
<tr>
<td>Tgl Service Mobil</td>
<td>:</td>
<td><input type="text" name="tgl_service_mobil"
id="tgl_service_mobil" class="form-control" value="{{ $service-
>tgl_service_mobil }}"></td>
</tr>
<tr>
<td>Perbaiki</td>
<td>:</td>
<td><input type="text" name="perbaiki"
id="perbaiki" class="form-control" value="{{ $service->perbaiki
}}"></td>
</tr>
<tr>
<td>Biaya Service</td>
<td>:</td>
<td><input type="text" name="biaya_service"
id="biaya_service" class="form-control" value="{{ $service-
>biaya_service }}"></td>
</tr>
</table>

```

5. Permohonan Mobil

a. Perintah memasukkan data permohonan mobil

```

<form action="{{ url('permohonanmobil') }}" method="post">
    {!! csrf_field() !!}
    <table align="center">
        <tr class="hidden">
            <td>Id</td>
            <td>:</td>
            <td>
                @if(\App\Models\Permohonanmobil::orderBy('id
', 'desc')->count() == 0)

```

```

        <input type="text" name="id" id="id"
class="form-control" value="1" readonly></td>
        @else
        <input type="text" name="id" id="id"
class="form-control" value="{{
\App\Models\Permohonanmobil::orderBy('id', 'desc')->first()->id+1
}}" readonly></td>
        @endif
    </tr>
    <tr>
        <td>Nama </td>
        <td></td>
        <td><input type="text" name="nama_pemohon"
id="nama_pemohon" class="form-control" value="{{ Auth::user()->name
}}" readonly></td>
    </tr>
    <tr>
        <td>Merk</td>
        <td></td>
        <td>{!! Form::select('id_merk', $merk, null,
array('id' => 'merk', 'style'=>'width: 188px')) !!}</td>
    </tr>
    <tr>
        <td>Jenis</td>
        <td></td>
        <td>{!! Form::select('id_jenis', array(), null,
array('id' => 'jeniskendaraan', 'style'=>'width: 188px')) !!}</td>
    </tr>

```

b. Perintah melakukan perbaikan data permohonan mobil

```

<script>
$(document).ready(function() {
    $('#merk').change(function(){
        $.get('/merk/' + this.value, function(jeniskendaraans){
            var $jeniskendaraan = $('#jeniskendaraan');

            $jeniskendaraan.find('option').remove().end();

            $jeniskendaraan.append('<option value="" + "" + ">'
+ 'Pilih Jenis Kendaraan' + '</option>');

            $.each(jeniskendaraans, function(index,
jeniskendaraan){
                $jeniskendaraan.append('<option value="" +
jeniskendaraan.id + ">' + jeniskendaraan.jenis + '</option>');
            });

```



```

</tr>
<tr>
  <td><b>No. Telp Driver</b></td>
  <td>{{ $nohp[$permohonanmobil-
>no_telp_driver] }}</td>
</tr>
@else
  <tr>
    <td><b>No. Polisi</b></td>
    <td></td>
  </tr>
  <tr>
    <td><b>Driver</b></td>
    <td></td>
  </tr>
  <tr>
    <td><b>No. Telp Driver</b></td>
    <td></td>
  </tr>
@endif

```

6. Proyek

a. Perintah memasukkan data proyek

```

<div class="modal-body">
  <form action="{{ url('proyek') }}" method="post">
    {!! csrf_field() !!}
    <table align="center">
      <tr>
        <td>Id</td>
        <td></td>
        <td><input type="text" name="id" id="id"
class="form-control" value="{{ \App\Models\Proyek::orderBy('id',
'desc')->first()->id+1 }}" readonly></td>
      </tr>
      <tr>
        <td>Kode Proyek</td>
        <td></td>
        <td><input type="text" name="kode_proyek"
id="kode_proyek" class="form-control"></td>
      </tr>
      <tr>
        <td>Nama Proyek</td>
        <td></td>
        <td><input type="text" name="nama_proyek"
id="nama_proyek" class="form-control"></td>
      </tr>
    </table>
  </form>
</div>

```

```

        <tr>
            <td>Klien</td>
            <td>:</td>
            <td><input type="text" name="klien" id="klien"
class="form-control"></td>
        </tr>
    </table>

```

b. Perintah melakukan perbaikan data proyek

```

<form action="/proyek/{{ $proyek->id }}" method="post">
    @method('put')
    {!! csrf_field() !!}
    <table align="center">
        <tr>
            <td>Kode Proyek</td>
            <td>:</td>
            <td><input type="text" name="kode_proyek"
id="kode_proyek" class="form-control" value="{{ $proyek->kode_proyek
}}"></td>
        </tr>
        <tr>
            <td>Nama Proyek</td>
            <td>:</td>
            <td><input type="text" name="nama_proyek"
id="nama_proyek" class="form-control" value="{{ $proyek->nama_proyek
}}"></td>
        </tr>
        <tr>
            <td>Klien</td>
            <td>:</td>
            <td><input type="text" name="klien" id="klien"
class="form-control" value="{{ $proyek->klien }}"></td>
        </tr>
    </table>

```

c. Perintah melakukan hapus data proyek

```

<form method="POST" action="{{ url('/proyek' . '/' . $item->id) }}"
accept-charset="UTF-8" style="display:inline">
    {{ method_field('DELETE') }}
    {{ csrf_field() }}
    <button type="submit" class="btn
btn-danger btn-xs" title="Delete Vendor" onclick="return
confirm('Confirm delete?')"><i class="fa fa-trash" aria-
hidden="true"></i></button>
</form>

```

d. Perintah menampilkan data proyek

```
<table align="center">
  <tr>
    <td>Kode Proyek</td>
    <td>:</td>
    <td><input type="text" name="kode_proyek"
id="kode_proyek" class="form-control"></td>
  </tr>
  <tr>
    <td>Nama Proyek</td>
    <td>:</td>
    <td><input type="text" name="nama_proyek"
id="nama_proyek" class="form-control"></td>
  </tr>
  <tr>
    <td>Klien</td>
    <td>:</td>
    <td><input type="text" name="klien" id="klien"
class="form-control"></td>
  </tr>
</table>
```

7. Keuangan

a. Perintah memasukkan data keuangan

```
<div class="modal-body">
  <form action="{ url('keuangan') }" method="post">
    {!! csrf_field() !!}
    <table align="center">
      <tr class="hidden">
        <td>Id</td>
        <td>:</td>
        <td>
          @if(\App\Models\keuangan::orderBy('id',
'desc')->count() == 0)
            <input type="text" name="id" id="id"
class="form-control" value="1" readonly></td>
          @else
            <input type="text" name="id" id="id"
class="form-control" value="{ \App\Models\keuangan::orderBy('id',
'desc')->first()->id+1 }" readonly></td>
          @endif
        </td>
      </tr>
      <tr>
        <td>No. Invoice </td>
        <td>:</td>
        <td><input type="text" name="no_invoice"
id="no_invoice" class="form-control"></td>
      </tr>
    </table>
  </form>
</div>
```

```

</tr>
<tr>
    <td>Tanda Terima</td>
    <td>:</td>
    <td>{!! Form::select('id_tt', $tandaterima,
null, array('id' => 'tandaterima', 'style'=>'width:
188px')) !!}</td>
</tr>
<tr class="hidden">
    <td>Tanda Terima</td>
    <td>:</td>
    <td>{!! Form::select('tgl_mulai', array(), null,
array('id' => 'tglmulai', 'style'=>'width: 188px')) !!}</td>
</tr>
<tr class="hidden">
    <td>Tanda Terima</td>
    <td>:</td>
    <td>{!! Form::select('tgl_selesai', array(),
null, array('id' => 'tglselesai', 'style'=>'width:
188px')) !!}</td>
</tr>
</table>

```

b. Perintah melakukan perbaikan data keuangan

```

<script>
$(document).ready(function() {
    $('#tandaterima').change(function(){
        $.get('/tanggalsewa/' + this.value,
function(tandaterimas){
            var $tandaterima = $('#tglmulai');

            $tandaterima.find('option').remove().end
            ();

            $.each(tandaterimas, function(index,
tandaterima){
                $tandaterima.append('<option
value="" + tandaterima.tgl_mulai + ">" + tandaterima.tgl_mulai +
'</option>');

            });
        });
    });

    $(".merk_id
option[value='0']").attr("disabled","disabled");
});
</script>

```

c. Perintah melakukan hapus data keuangan

```
<form method="POST" action="{ url('/keuangan' . '/' . $item->id) }"
accept-charset="UTF-8" style="display:inline">
    {{ method_field('DELETE') }}
    {{ csrf_field() }}
    <button type="submit"
class="btn btn-danger btn-xs" title="Delete Permohonan Mobil"
onclick="return confirm('Confirm delete?')"><i class="fa fa-trash"
aria-hidden="true"></i></button>
</form>
```

d. Perintah menampilkan data keuangan

```
<div class="modal-body">
    <form action="/keuangan/{{ $keuangan->id }}" method="post">
        @method('get')
        {!! csrf_field() !!}
        <table align="center">
            <tr>
                <td width="250px">Pemohon</td>
                <td>:</td>
                <td width="450px"><b>{{ $vendor[$keuangan-
id_vendor] }}</b></td>
            </tr>
            <tr>
                <td width="250px">No. Invoice</td>
                <td>:</td>
                <td width="450px">{{ $keuangan->no_invoice
}}</td>
            </tr>
            <tr>
                <td width="250px">Tanggal Invoice</td>
                <td>:</td>
                <td width="450px">{{ $keuangan->created_at
}}</td>
            </tr>
            <tr>
                <td width="250px">No. Tanda Terima
Kendaraan</td>
                <td>:</td>
                <td width="450px">{{ $standaterima[$keuangan-
id_tt] }}</td>
            </tr>
            <tr>
                <td width="250px">Kebutuhan</td>
                <td>:</td>
                <td width="450px">Sewa Kendaraan untuk Kegiatan
{{ $namaprojek[$projek[$kebutuhan[$keuangan->id_tt]]] }}</td>
            </tr>
        </table>
    </form>
```

```

</tr>
<tr>
    <td width="250px">Tanggal Kebutuhan</td>
    <td>:</td>
    <td width="450px">{{ $tgl_mulai[$keuangan-
>id_tt] }} s/d {{ $tgl_selesai[$keuangan->id_tt] }}</td>
</tr>

```

8. Vendor

a. Perintah memasukkan data vendor

```

<div class="modal-body">
    <form action="{{ url('vendor') }}" method="post">
        {!! csrf_field() !!}
        <table align="center">
            <tr class="hidden">
                <td>Id</td>
                <td>:</td>
                <td><input type="text" name="id" id="id"
class="form-control" value="{{ \App\Models\Vendor::orderBy('id',
'desc')->first()->id+1 }}" readonly></td>
            </tr>
            <tr>
                <td>Vendor</td>
                <td>:</td>
                <td><input type="text" name="vendor" id="vendor"
class="form-control"></td>
            </tr>
            <tr>
                <td>Direktur</td>
                <td>:</td>
                <td><input type="text" name="direktur"
id="direktur" class="form-control"></td>
            </tr>
            <tr>
                <td>Alamat</td>
                <td>:</td>
                <td><input type="text" name="alamat" id="alamat"
class="form-control"></td>
            </tr>
            <tr>
                <td>No. Telp</td>
                <td>:</td>
                <td><input type="text" name="no_telp"
id="no_telp" class="form-control"></td>
            </tr>
        </table>

```

b. Perintah melakukan perbaikan data vendor

```
<div class="modal-body">
  <form action="/vendor/{{ $vendor->id }}" method="post">
    @method('put')
    {!! csrf_field() !!}
    <table align="center">
      <tr>
        <td>Alamat</td>
        <td>:</td>
        <td><input type="text" name="alamat" id="alamat"
class="form-control" value="{{ $vendor->alamat }}"></td>
      </tr>
      <tr>
        <td>No. Telp</td>
        <td>:</td>
        <td><input type="text" name="no_telp"
id="no_telp" class="form-control" value="{{ $vendor->no_telp
}}"></td>
      </tr>
    </table>
```

c. Perintah melakukan hapus data vendor

```
<form method="POST" action="{{ url('/vendor' . '/' . $item->id) }}"
accept-charset="UTF-8" style="display:inline">
  {{ method_field('DELETE') }}
  {{ csrf_field() }}
  <button type="submit" class="btn
btn-danger btn-sm" title="Delete Vendor" onclick="return
confirm('Confirm delete?')"><i class="fa fa-trash" aria-
hidden="true"></i> Delete</button>
</form>
```

d. Perintah menampilkan data vendor

```
<div class="modal-body">
  <form action="/vendor/{{ $vendor->id }}" method="post">
    @method('get')
    {!! csrf_field() !!}
    <table align="center">
      <tr>
        <td>Alamat</td>
        <td>:</td>
        <td><input type="text" name="alamat" id="alamat"
class="form-control" value="{{ $vendor->alamat }}"></td>
      </tr>
      <tr>
```

```

        <td>No. Telp</td>
        <td>:</td>
        <td><input type="text" name="no_telp"
id="no_telp" class="form-control" value="{{ $vendor->no_telp
}}"></td>
    </tr>
</table>

```

9. Tanda Terima Kendaraan

a. Perintah memasukkan data tanda terima kendaraan

```

<table align="center">
<tr>
<td>Id</td>
<td>:</td>
<td>
@if(\App\Models\tandaterima::orderBy('id',
'desc')->count() == 0)
<input type="text" name="id" id="id"
class="form-control" value="1" readonly></td>
@else
<input type="text" name="id" id="id"
class="form-control" value="{{
\App\Models\tandaterima::orderBy('id', 'desc')->first()->id+1 }}"
readonly></td>
@endif
</tr>
<tr>
<td>No. Tanda Terima</td>
<td>:</td>
<td><input type="text" name="no_tandaterima"
id="no_tandaterima" class="form-control"></td>
</tr>
<tr>
<td>Pemohon</td>
<td>:</td>
<td>{!! Form::select('id_permohonanmobil',
$pemohon, null, array('id' => 'pemohon', 'style'=>'width:
188px')) !!}</td>
</tr>
<tr>
<td>Tanggal Dipakai</td>
<td>:</td>
<td><!--<input type="text" name="tgl_mulai"
id="tgl_mulai" class="form-control">-->
<div class="input-append date" id="dp3"
data-date="12-02-2012" data-date-format="yyyy-mm-dd">

```

```

                <input class="span2" size="16"
type="date" value="{{ \Carbon\Carbon::now()->format('Y-m-d') }}"
name="tgl_mulai" id="tgl_mulai" style="width: 190px">
                <span class="add-on"><i class="icon-
th"></i></span>
            </div>
        </td>
    </tr>
</table>

```

b. Perintah melakukan perbaikan data tanda terima kendaraan

```

<div class="modal-body">
    <form action="/tandaterima/{{ $tandaterima->id }}"
method="post">
        @method('put')
        {!! csrf_field() !!}
        <table align="center">
            @if(Auth::user()->level == "user")
                <tr>
                    <td>Status</td>
                    <td>:</td>
                    <td>{!! Form::select('status',
array("Diterima"=>"Diterima","Dikembalikan"=>"Dikembalikan"),null,array('id' => 'jeniskendaraan', 'style'=>'width: 188px')) !!}</td>
                </tr>
                <tr>
                    <td>Tanggal Dikembalikan</td>
                    <td>:</td>
                    <td>
                        <div class="input-append date" id="dp3"
data-date="2023-05-23" data-date-format="yyyy-mm-dd">
                            <input class="span2" size="16"
type="date" value="{{ \Carbon\Carbon::now()->format('Y-m-d') }}"
name="tgl_selesai" id="tgl_selesai" style="width: 190px">
                            <span class="add-on"><i class="icon-
th"></i></span>
                        </div>
                    </td>
                </tr>
            @else
                <tr>
                    <td>Penerimaan</td>
                    <td>:</td>
                    <td>{!! Form::select('status',
array("Diserahkan"=>"Diserahkan","Diterima Kembali"=>"Diterima
Kembali"),null,array('id' => 'jeniskendaraan', 'style'=>'width:
188px')) !!}</td>
                </tr>
            @endif
        </table>
    </form>
</div>

```

```
@endif
</table>
```

c. Perintah melakukan hapus data tanda terima kendaraan

```
<form method="POST" action="{ url('/tandaterima' . '/' . $item->id)
}}" accept-charset="UTF-8" style="display:inline">
    {{ method_field('DELETE') }}
    {{ csrf_field() }}
    <button type="submit"
class="btn btn-danger btn-xs" title="Delete Kendaraan"
onclick="return confirm('Confirm delete?')"><i class="fa fa-trash"
aria-hidden="true"></i></button>
</form>
```

d. Perintah menampilkan data tanda terima kendaraan

```
<table align="center">
<tr>
    <td colspan="2"><b>DATA PEMOHON</b></td>
</tr>
<tr>
    <td width="250px">
        <table>
            <tr>
                <td width="150px"><b>Nama </b></td>
                <td width="10px">:</td>
                <td width="150px">{{
$pemohon[$tandaterima->id_permohonanmobil] }}</td>
            </tr>
            <tr>
                <td><b>mobil </b></td>
                <td>:</td>
                <td>{{
$jeniskendaraan[$jenis[$tandaterima->id_permohonanmobil]] }}</td>
            </tr>
            <tr>
                <td><b>Jumlah Orang</b></td>
                <td>:</td>
                <td>{{ $jml_orang[$tandaterima-
>id_permohonanmobil] }}</td>
            </tr>
        </table>
    </td>
    <td>
        <table>
            <tr>
                <td width="150px"><b>Tujuan</b></td>
```

```
 </td>  <b>Tanggal Berangkat</b></td> {{ $tgl_brgkat[$standaterima- >id_permohonanmobil] }}</td> </tr> <tr>  <b>Tanggal Pulang</b></td> {{ $tgl_plg[$standaterima- >id_permohonanmobil] }}</td> </tr> </table> </td> </tr> | | |
```

10. Driver

a. Perintah memasukkan data driver

```

<form action="{{ url('driver') }}" method="post">
{!! csrf_field() !!}
<table align="center">
<tr class="hidden">
<td>Id</td>
<td:</td>
<td><input type="text" name="id" id="id"
class="form-control" value="{{ \App\Models\Driver::orderBy('id',
'desc')->first()->id+1 }}" readonly></td>
</tr>
<tr>
<td>Nama</td>
<td:</td>
<td><input type="text" name="nama" id="nama"
class="form-control"></td>
</tr>
<tr>
<td>No. Telp </td>
<td:</td>
<td><input type="text" name="no_telp"
id="no_telp" class="form-control"></td>
</tr>
<tr>
<td>Alamat</td>

```

```

        <td>:</td>
        <td><input type="text" name="alamat" id="alamat"
class="form-control"></td>
    </tr>
    <tr class="hidden">
        <td>vendor </td>
        <td>:</td>
        <td><input type="text" name="id_vendor"
id="id_vendor" value="{{ $user->homebase }}" class="form-
control"></td>
    </tr>
</table>
</div>
</form>

```

b. Perintah melakukan perbaikan data driver

```

<div class="modal-body">
    <form action="/driver/{{ $driver->id }}" method="post">
        @method('put')
        {!! csrf_field() !!}
        <table align="center">
            <tr>
                <td>Nama</td>
                <td>:</td>
                <td><input type="text" name="nama" id="nama"
class="form-control" value="{{ $driver->nama }}"></td>
            </tr>
            <tr>
                <td>No. Telp </td>
                <td>:</td>
                <td><input type="text" name="no_telp"
id="no_telp" class="form-control" value="{{ $driver->no_telp
}}"></td>
            </tr>
            <tr>
                <td>Alamat</td>
                <td>:</td>
                <td><input type="text" name="alamat" id="alamat"
class="form-control" value="{{ $driver->alamat }}"></td>
            </tr>
        </table>
    </form>

```

c. Perintah melakukan hapus data driver

```

<form method="POST" action="{{ url('/driver' . '/' . $item->id) }}"
accept-charset="UTF-8" style="display:inline">
    {{ method_field('DELETE') }}

```

```

{{ csrf_field() }}
<button type="submit" class="btn
btn-danger btn-sm" title="Delete Driver" onclick="return
confirm('Confirm delete?')"><i class="fa fa-trash" aria-
hidden="true"></i> Delete</button>
</form>

```

d. Perintah menampilkan data driver

```

<table align="center">
<tr>
<td>Nama</td>
<td>:</td>
<td><input type="text" name="nama" id="nama"
class="form-control" value="{{ $driver->nama }}"></td>
</tr>
<tr>
<td>No. Telp</td>
<td>:</td>
<td><input type="text" name="no_telp"
id="no_telp" class="form-control" value="{{ $driver->no_telp
}}"></td>
</tr>
<tr>
<td>Alamat</td>
<td>:</td>
<td><input type="text" name="alamat" id="alamat"
class="form-control" value="{{ $driver->alamat }}"></td>
</tr>
</table>

```

11. Manajemen User

a. Perintah memasukkan data manajemen user

```

<form action="{{ url('manajemenuser') }}" method="post">
{!! csrf_field() !!}
<table align="center">
<tr>
<td>Nama</td>
<td>:</td>
<td><input type="text" name="nama" id="nama"
class="form-control"></td>
</tr>
<tr>
<td>Level </td>
<td>:</td>
<td><input type="text" name="level" id="level"
class="form-control"></td>

```

```

</tr>
<tr>
    <td>Email</td>
    <td>:</td>
    <td><input type="text" name="email" id="email"
class="form-control"></td>
</tr>
</table>

```

b. Perintah melakukan perbaikan data manajemen user

```

<table align="center">
<tr>
    <td>Level</td>
    <td>:</td>
    <td>
        {!! Form::select('level',
array("admin"=>"Admin","approval"=>"Approval","user"=>"User","vendor"
=>"Vendor"),null,['style'=>'width: 190px']) !!}
    </td>
</tr>
<tr>
<td>Vendor</td>
    <td>:</td>
    <td>{!! Form::select('homebase', $vendor,
$manajemenuser->homebase, array('id' => 'unit', 'style'=>'width:
190px')) !!}</td>
</tr>
</table>

```

c. Perintah melakukan hapus data manajemen user

```

<form method="POST" action="{{ url('/manajemenuser' . '/' . $item-
>id) }}" accept-charset="UTF-8" style="display:inline">
    {{ method_field('DELETE') }}
    {{ csrf_field() }}
    <button type="submit" class="btn
btn-danger btn-sm" title="Delete Manajemen User" onclick="return
confirm('Confirm delete?')"><i class="fa fa-trash" aria-
hidden="true"></i> Delete</button>
</form>

```

d. Perintah menampilkan data manajemen user

```

<table align="center">
<tr>
    <td>Nama</td>

```

```

        <td>:</td>
        <td><input type="text" name="nama" id="nama"
class="form-control" value="{ $manajemenuser->nama }"></td>
    </tr>
    <tr>
        <td>Level</td>
        <td>:</td>
        <td><input type="text" name="level" id="level"
class="form-control" value="{ $manajemenuser->level }"></td>
    </tr>
    <tr>
        <td>Email</td>
        <td>:</td>
        <td><input type="text" name="email" id="email"
class="form-control" value="{ $manajemenuser->email }"></td>
    </tr>
</table>

```

