

BAB IV. HASIL DAN ANALISIS PENELITIAN

4.1 Analisa Perancangan Sistem

Pada sub bab ini menguraikan secara menyeluruh tahapan analisis dan perancangan sistem sebagai fondasi utama dalam pengembangan aplikasi pergudangan sparepart kendaraan operasional berbasis web di PT OPQ. Tahapan ini bertujuan untuk menggali kebutuhan sistem secara mendalam, merumuskan solusi yang tepat atas permasalahan yang dihadapi perusahaan, serta merancang sistem yang selaras dengan kebutuhan aktual operasional gudang. Analisis sistem yang tepat dan perancangan yang terstruktur menjadi kunci untuk menciptakan sistem yang optimal, efisien, dan mudah diterapkan di lingkungan kerja.

Dalam prosesnya, analisa terhadap sistem sebelum dirancang dilakukan pengumpulan informasi melalui dua metode utama , yakni wawancara dan observasi secara langsung di lapangan. Wawancara dilangsungkan dengan pihak-pihak yang berperan dalam operasional gudang, seperti manajemen, staf gudang, dan teknisi, guna mendapatkan pemahaman mengenai alur kerja yang berlangsung, kendala-kendala yang dihadapi, serta harapan terhadap sistem baru. Di sisi lain, observasi dilakukan dengan mengamati aktivitas secara langsung di area gudang, mulai dari penerimaan barang, penyimpanan, pencatatan manual, hingga proses pengeluaran dan distribusi. Data yang diperoleh dari kedua metode ini kemudian dianalisis untuk mengidentifikasi permasalahan inti serta peluang perbaikan melalui sistem yang akan dikembangkan.

Waterfall merupakan metodologi yang diterapkan pada pengembangan sistem ini, yaitu pendekatan bertahap dan tersistem yang mengharuskan penyelesaian setiap fase sebelum melanjutkan ke tahap berikutnya. Metode ini dianggap tepat karena proyek memiliki lingkup kerja yang jelas, kebutuhan yang stabil, serta jadwal pengembangan yang dapat direncanakan dengan baik. Setiap tahapan mulai dari analisis, desain,

implementasi, pengujian, hingga pemeliharaan dilaksanakan secara bertahap dengan dokumentasi dan pengawasan mutu yang ketat.

Dengan penerapan proses analisis dan desain berdasarkan data riil dari lapangan, sistem pergudangan berbasis web yang dikembangkan diharapkan mampu menjadi solusi efektif atas tantangan yang dihadapi PT OPQ. Target utama pada sistem ini yakni dapat mendukung dalam hal peningkatan efisiensi alur kerja, meminimalkan kesalahan pada saat pencatatan, mempercepat proses pengambilan keputusan, serta mendukung keakuratan dan transparansi data stok sparepart kendaraan operasional..

4.2 Perancangan Diagram Sistem Usulan

4.2.1 *Use Case Diagram*

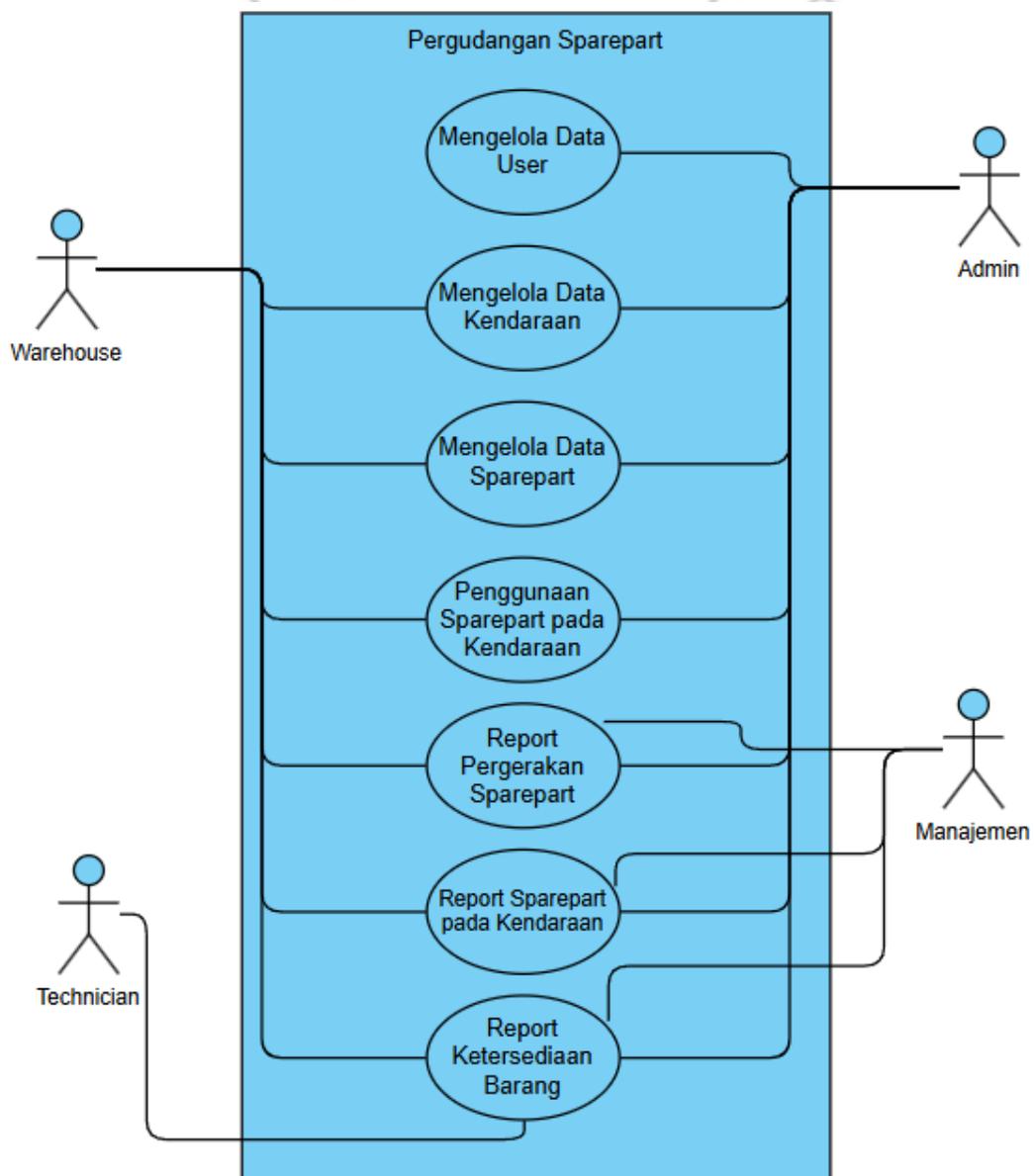
Use Case Diagram berfungsi sebagai representasi fitur-fitur sistem dengan perspektif aktor atau pengguna. Diagram ini menggambarkan keterkaitan antara aktor dengan sejumlah aktivitas atau layanan (*use case*) yang ditawarkan oleh sistem. Diagram ini berfokus untuk mengidentifikasi kebutuhan pengguna dan menjamin bahwa sistem yang dibangun dapat berjalan sesuai dengan fungsi-fungsinya dan memenuhi ekspektasi penggunaannya.

Dalam pengembangan sistem informasi pergudangan *sparepart* kendaraan operasional berbasis web di PT OPQ, *use case diagram* berfungsi sebagai dasar untuk merancang bagaimana sistem akan bekerja dan siapa saja yang akan berinteraksi dengannya. Aktor-aktor yang terlibat pada sistem ini di antaranya adalah *Warehouse*, *Technician*, dan Manajemen aktivitas yang berbeda, seperti mengelola data *sparepart*, melakukan pencatatan barang masuk dan keluar, memantau stok, serta pemantauan laporan bulanan.

Use case diagram mempermudah proses analisis dan perancangan sistem karena menyajikan gambaran visual yang jelas mengenai fungsi-fungsi yang harus tersedia. Dengan diagram ini, pengembang dapat lebih mudah dalam mengidentifikasi kebutuhan pengguna, menetapkan batasan sistem, serta menentukan skenario-skenario penggunaan yang relevan. Selain itu, *use case diagram* juga mengakomodir komunikasi antara

pengembang dengan pemangku kepentingan, agar kesalahan dalam interpretasi kebutuhan dapat diminimalkan sejak tahap awal pengembangan.

Secara umum, *use case diagram* menjadi sarana yang sangat efektif dalam proses analisis sistem karena mampu menggambarkan fungsi-fungsi yang dimiliki oleh sistem serta cara interaksi pengguna dengan sistem tersebut. Dengan demikian, diagram ini membantu menciptakan rancangan sistem yang bersifat terorganisir juga selaras dengan kebutuhan operasional perusahaan.



Gambar 4.1 Use Case Diagram

4.2.2 Use Case Diagram Description

Use Case Diagram Description merupakan uraian mendetail dari tiap-tiap *use case* yang tercantum dalam diagram *use case* yang telah digambarkan. Jika diagram hanya sebatas penggambaran hubungan secara umum antara aktor dan fitur sistem, maka deskripsi ini menjabarkan secara lengkap alur proses dari setiap fungsi tersebut. Penjabaran ini sangat penting untuk menjamin bahwa tiap fitur dalam sistem dirancang dengan alur yang terdefinisi dengan baik, sistematis, dan sesuai dengan kebutuhan dari pengguna. Berikut merupakan Deskripsi *Use Case Diagram* dari aplikasi manajemen pergudangan *sparepart* milik PT OPQ.

a) Mengelola Data User

Tabel 4.1. Use Case Diagram Description Mengelola Data User

Use Case Name	Mengelola Data User	
Actors	Admin	
Trigger	User akan mengakses ke menu <i>master data user</i> untuk mengelola data <i>user</i>	
Preconditions	-User harus terdaftar dan dapat <i>login</i> -User memiliki <i>role</i> Admin	
Postcondition	User berhasil mengakses ke menu <i>master data user</i> dan dapat melakukan <i>CRUD user</i> pada menu tersebut	
Success Scenario	Actor	System
	1. User membuka menu <i>master data user</i>	Menampilkan halaman <i>master data user</i>
	2. User menambah data <i>user</i>	Menambahkan data <i>user</i> ke database dan menampilkannya di list
	3. User mengubah data <i>user</i>	Memperbaharui data <i>user</i> ke database dan menampilkannya di list

	4. <i>User</i> menghapus data <i>user</i>	Menghapus data <i>user</i> di database dan menghilangkannya dari list
Exception	Sistem menampilkan warning data tidak lengkap Sistem menampilkan warning data berhasil disimpan, diedit, atau dihapus	

b) Mengelola Data *Sparepart*

Tabel 4.2. Use Case Diagram Description Mengelola Data *Sparepart*

Use Case Name	Mengelola Data <i>Sparepart</i>	
Actors	<i>Warehouse</i> , Admin	
Trigger	<i>User</i> akan mengakses ke menu <i>master</i> data <i>sparepart</i> , lokasi <i>sparepart</i> , <i>sparepart incoming</i> , dan <i>sparepart outcoming</i> untuk mengelola data <i>sparepart</i>	
Preconditions	- <i>User</i> harus terdaftar dan dapat <i>login</i> - <i>User</i> memiliki role <i>Warehouse</i> dan Admin	
Postcondition	<i>User</i> berhasil mengakses ke menu <i>master</i> data <i>sparepart</i> , <i>sparepart incoming</i> , dan <i>sparepart outcoming</i> dan dapat melakukan CRUD <i>sparepart</i> pada menu tersebut.	
Success Scenario	Actor	System
	1. <i>User</i> membuka menu <i>master</i> data <i>sparepart</i>	Menampilkan halaman <i>master</i> data <i>sparepart</i>
	2. <i>User</i> menambah data <i>sparepart</i>	Menambahkan data <i>sparepart</i> ke database dan menampilkannya di list
	3. <i>User</i> mengubah data <i>sparepart</i>	Memperbaharui data <i>sparepart</i> ke database dan menampilkannya di list
	4. <i>User</i> menghapus data <i>sparepart</i>	Menghapus data <i>sparepart</i> di database dan menghilangkan dari list

	5. <i>User</i> membuka menu <i>master location</i>	Menampilkan halaman <i>master location</i>
	6. <i>User</i> menambah data <i>location</i>	Menambahkan data <i>location</i> ke database dan menampilkannya di list
	7. <i>User</i> mengubah data <i>location</i>	Memperbaharui data <i>location</i> ke database dan menampilkannya di list
	8. <i>User</i> menghapus data <i>location</i>	Menghapus data <i>location</i> di database dan menghilangkan dari list
	9. <i>User</i> membuka menu <i>incoming sparepart</i>	Menampilkan halaman <i>incoming sparepart</i>
	10. <i>User</i> menambah data <i>sparepart</i> masuk	Menambahkan <i>sparepart</i> masuk ke database dan menampilkannya di list
	11. <i>User</i> mengubah data <i>sparepart</i> masuk	Memperbaharui data <i>sparepart</i> masuk ke database dan menampilkannya di list
	12. <i>User</i> menghapus data <i>sparepart</i> masuk	Menghapus data <i>sparepart</i> masuk ke database dan menghilangkan dari list
	13. <i>User</i> membuka menu <i>outcoming sparepart</i>	Menampilkan halaman <i>outcoming sparepart</i>
	14. <i>User</i> menambah data <i>sparepart</i> keluar	Menambahkan <i>sparepart</i> keluar ke database dan menampilkannya di list
	15. <i>User</i> mengubah data <i>sparepart</i> keluar	Memperbaharui data <i>sparepart</i> keluar ke

		database dan menampilkannya di list
	16. <i>User</i> menghapus data <i>sparepart</i> keluar	Menghapus data <i>sparepart</i> keluar ke database dan menghilangkan dari list
Exception	Sistem menampilkan warning data tidak lengkap Sistem menampilkan warning data berhasil disimpan, diedit, atau dihapus	

c) Mengelola Data Kendaraan

Tabel 4.3. Use Case Diagram Description Mengelola Data Kendaraan

Use Case Name	Mengelola Data Kendaraan	
Actors	<i>Warehouse</i> , Admin	
Trigger	<i>User</i> akan mengakses ke menu <i>master</i> data kendaraan untuk mengelola data kendaraan	
Preconditions	- <i>User</i> harus terdaftar dan dapat <i>login</i> - <i>User</i> memiliki <i>role Warehouse</i> dan Admin	
Postcondition	<i>User</i> berhasil mengakses ke menu <i>master</i> data kendaraan dan dapat melakukan CRUD kendaraan pada menu tersebut	
Success Scenario	Actor	System
	1. <i>User</i> membuka menu <i>master</i> data kendaraan	Menampilkan halaman <i>master</i> data kendaraan
	2. <i>User</i> menambah data kendaraan	Menambahkan data kendaraan ke database dan menampilkannya di list
	3. <i>User</i> mengubah data kendaraan	Memperbaharui data kendaraan ke database dan menampilkannya di list
	4. <i>User</i> menghapus data kendaraan	Menghapus data kendaraan di database dan menampilkannya di list

Exception	Sistem menampilkan warning data tidak lengkap Sistem menampilkan warning data berhasil disimpan, diedit, atau dihapus
------------------	--

d) Penggunaan *Sparepart* pada Kendaraan

Tabel 4.4. Use Case Diagram Description Penggunaan *Sparepart* pada Kendaraan

Use Case Name	Penggunaan <i>Sparepart</i> pada Kendaraan	
Actors	<i>Warehouse, Admin</i>	
Trigger	<i>User akan mengakses ke menu <i>outcoming sparepart</i> untuk mendata <i>sparepart</i> yang dipakai pada kendaraan dalam sistem</i>	
Preconditions	- <i>User harus terdaftar dan dapat login</i> - <i>User memiliki role Warehouse dan Admin</i>	
Postcondition	<i>User berhasil mengakses ke menu <i>outcoming sparepart</i> dan dapat menambahkan, mengubah, dan menghapus data <i>sparepart</i> yang dipakai pada kendaraan</i>	
Success Scenario	Actor	System
	1. <i>User membuka menu <i>outcoming sparepart</i></i>	Menampilkan halaman input <i>outcoming sparepart</i>
	2. <i>User menambah data <i>sparepart</i> dipakai pada kendaraan</i>	Menambahkan data <i>sparepart</i> dipakai pada kendaraan ke database dan menampilkannya di list
	3. <i>User mengubah data <i>sparepart</i> dipakai pada kendaraan</i>	Memperbaharui <i>sparepart</i> dipakai pada kendaraan ke database dan menampilkannya di list
	4. <i>User menghapus data <i>sparepart</i> dipakai pada kendaraan</i>	menghapus <i>sparepart</i> dipakai pada kendaraan ke database dan menghilangkannya dari list
Exception	Sistem menampilkan warning data tidak lengkap	

	Sistem menampilkan warning data berhasil disimpan atau diedit
--	---

e) *Report Pergerakan Sparepart*

Tabel 4.5. Use Case Diagram Description Report Pergerakan Sparepart

Use Case Name	Report Pergerakan Sparepart	
Actors	Warehouse, Manajemen, dan Admin	
Trigger	User akan mengakses ke menu <i>report pergerakan sparepart</i>	
Preconditions	-User harus terdaftar dan dapat <i>login</i> -User memiliki role Warehouse / Manajemen / Admin	
Postcondition	User berhasil mengakses ke menu <i>report pergerakan sparepart</i>	
Success Scenario	Actor	System
	1. User membuka menu <i>report pergerakan sparepart</i>	Menampilkan halaman <i>report pergerakan sparepart</i>
	2. User menentukan periode atau jangka waktu yang akan ditampilkan di <i>report</i>	Menampilkan <i>report pergerakan sparepart</i> dengan periode atau jangka waktu yang ditentukan
Exception	Sistem menampilkan warning masukan periode atau jangka waktu yang ingin ditentukan	

f) *Report Sparepart Pada Kendaraan*

Tabel 4.6. Use Case Diagram Description Report Sparepart pada Kendaraan

Use Case Name	Report Sparepart Pada Kendaraan
Actors	Warehouse, Manajemen, dan Admin
Trigger	User akan mengakses ke menu <i>report sparepart pada kendaraan</i>
Preconditions	-User harus terdaftar dan dapat <i>login</i> -User memiliki role Warehouse / Manajemen / Admin

Postcondition	<i>User</i> berhasil mengakses ke menu <i>report sparepart</i> pada kendaraan	
Success Scenario	Actor	System
	1. <i>User</i> membuka menu <i>report sparepart</i> pada kendaraan	Menampilkan halaman <i>report sparepart</i> pada kendaraan
	2. <i>User</i> menentukan kendaraan dan periode atau jangka waktu yang akan ditampilkan di <i>report</i>	Menampilkan <i>report sparepart</i> pada kendaraan dengan kendaraan dan periode atau jangka waktu yang ditentukan
Exception	Sistem menampilkan warning masukan kendaraan yang ingin diperiksa Sistem menampilkan warning masukan periode atau jangka waktu yang ingin ditentukan	

g) *Report* Ketersediaan Barang

Tabel 4.7. Use Case Diagram Description *Report* Ketersediaan Barang

Use Case Name	<i>Report</i> Ketersediaan Barang	
Actors	<i>Warehouse</i> , <i>Technician</i> , Manajemen, dan admin	
Trigger	<i>User</i> akan mengakses ke menu <i>report</i> ketersediaan barang	
Preconditions	- <i>User</i> harus terdaftar dan dapat <i>login</i> - <i>User</i> memiliki role <i>Warehouse</i> / <i>Technician</i> / Manajemen / Admin	
Postcondition	<i>User</i> berhasil mengakses ke menu <i>report</i> ketersediaan barang	
Success Scenario	Actor	System
	1. <i>User</i> membuka menu <i>report</i> ketersediaan barang	Menampilkan halaman <i>report</i> ketersediaan barang
	2. <i>User</i> melakukan pencarian barang yang ingin dilihat	Menampilkan barang yang dicari ke dalam list

Exception	Sistem menampilkan warning barang tidak ditemukan
------------------	---

4.2.3 *Activity Diagram*

Activity Diagram adalah salah satu dari sekian banyak diagram bagian penting dalam pemodelan sistem menggunakan *Unified Modeling Language* (UML) yang memiliki manfaat untuk menggambarkan alur aktivitas atau proses kerja dari sebuah sistem secara visual dan sistematis. Diagram ini menunjukkan bagaimana aktivitas dimulai, dijalankan, bercabang, hingga berakhir. *Activity diagram* memiliki tujuan utama yaitu memberikan pemahaman yang lebih rinci dan jelas mengenai urutan proses yang terjadi dalam sistem, baik itu proses otomatis oleh sistem maupun interaksi pengguna secara manual.

Dalam pengembangan sistem informasi, terutama pada sistem pergudangan seperti di PT OPQ, *activity diagram* digunakan untuk memetakan langkah-langkah operasional secara runtut, mulai dari proses penerimaan barang, pengecekan stok, pengeluaran barang, hingga pelaporan. Diagram ini mampu memperlihatkan interaksi antar bagian, seperti *warehouse*, *technician*, hingga bagian manajemen dalam menjalankan aktivitas sehari-hari yang berkaitan dengan pengelolaan *sparepart* kendaraan operasional.

Activity diagram terdiri dari beberapa komponen penting. Proses dimulai dari *Initial Node*, yang menandai dimulainya aktivitas. Setiap langkah kegiatan atau proses digambarkan dalam bentuk *Activity Node*. Bila dalam proses terdapat kondisi yang harus dipilih, digunakan *Decision Node* untuk menunjukkan percabangan berdasarkan keputusan tertentu. Jika beberapa alur kembali bergabung menjadi satu, digunakan *Merge Node*. Proses kemudian diakhiri dengan *Final Node* yang menunjukkan bahwa seluruh aktivitas telah selesai dijalankan. Selain itu, digunakan *Swimlane* untuk memisahkan aktivitas berdasarkan pelaku atau unit kerja yang terlibat, sehingga alur kerja antar bagian bisa terlihat lebih jelas.

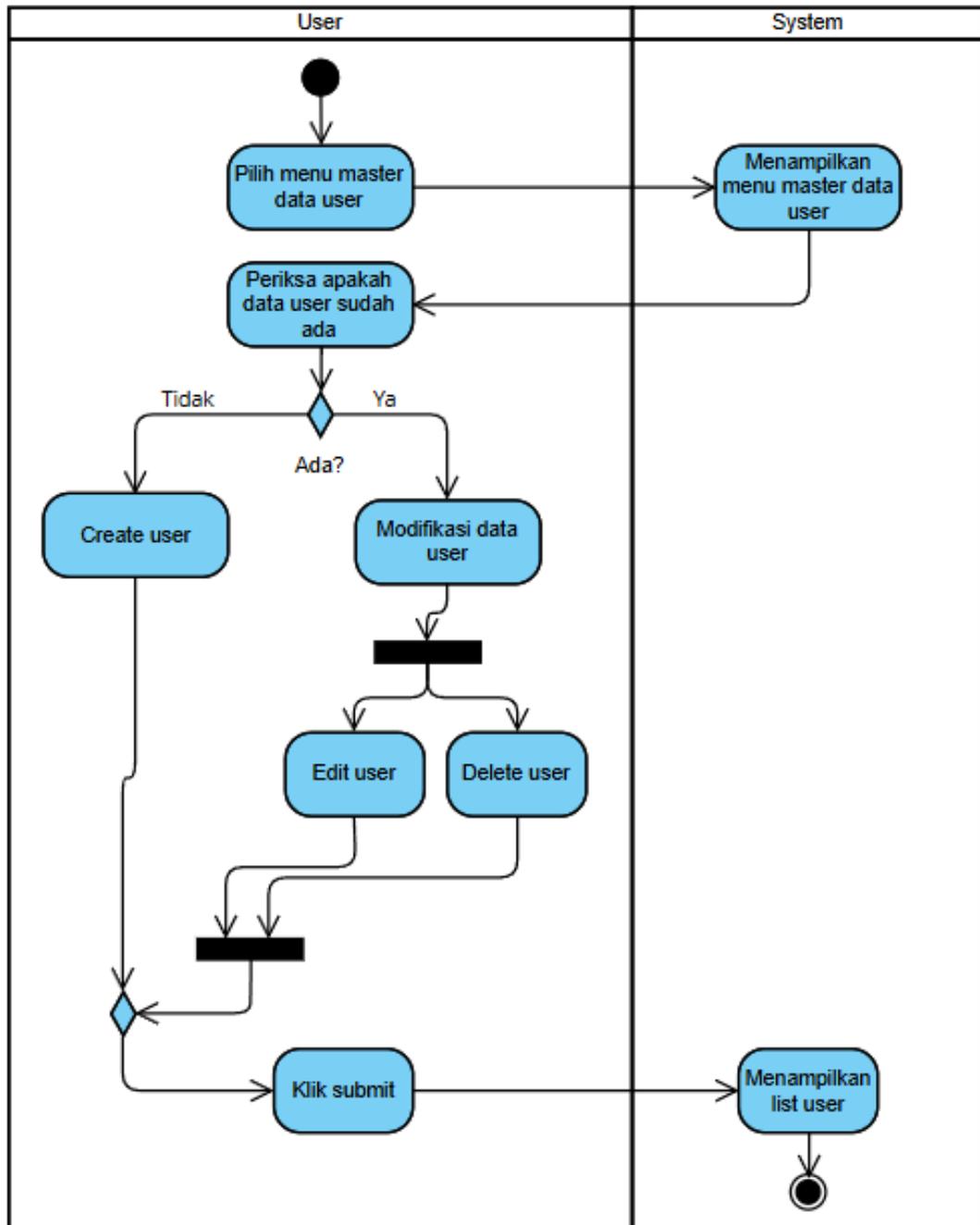
Keunggulan dari *activity diagram* adalah kemampuannya dalam memvisualisasikan proses bisnis secara logis dan menyeluruh. Diagram ini membantu semua pihak yang terlibat dalam proyek mulai dari analisis sistem,

pengembang, hingga pengguna akhir untuk memahami bagaimana sistem akan beroperasi, apa saja yang dibutuhkan dalam proses, dan di mana saja kemungkinan kendala bisa muncul. Dengan begitu, pengembangan sistem dapat dilakukan secara lebih terarah, efisien, dan minim risiko kesalahan.

Dalam sistem yang kompleks, penggunaan *activity diagram* sangat efektif untuk memperjelas komunikasi antar tim dan menjadikan proses perancangan lebih terstruktur. Oleh karena itu, dalam pengembangan sistem per gudangan berbasis web di PT OPQ, *activity diagram* menjadi alat bantu penting untuk merancang dan mengevaluasi setiap alur kerja agar *software* yang dirancang dan dibangun dipastikan dapat memenuhi kebutuhan operasional perusahaan.

1. *Activity Diagram* Mengelola Data User

User membuka halaman *master data user* kemudian sistem menampilkan halaman *master data user*, *user* kemudian bisa memeriksa apakah akun yang dibutuhkan sudah tertera, apabila belum ada *user* dapat membuat akun. Apabila akun sudah ada *user* dapat mengubah atau menghapus akun *user* sesuai dengan kebutuhan. Semua aksi dapat dilakukan dengan menekan tombol *submit* sehingga data yang diinput oleh *user* dapat diproses oleh sistem dan sistem menampilkan kembali list *user* terbaru.

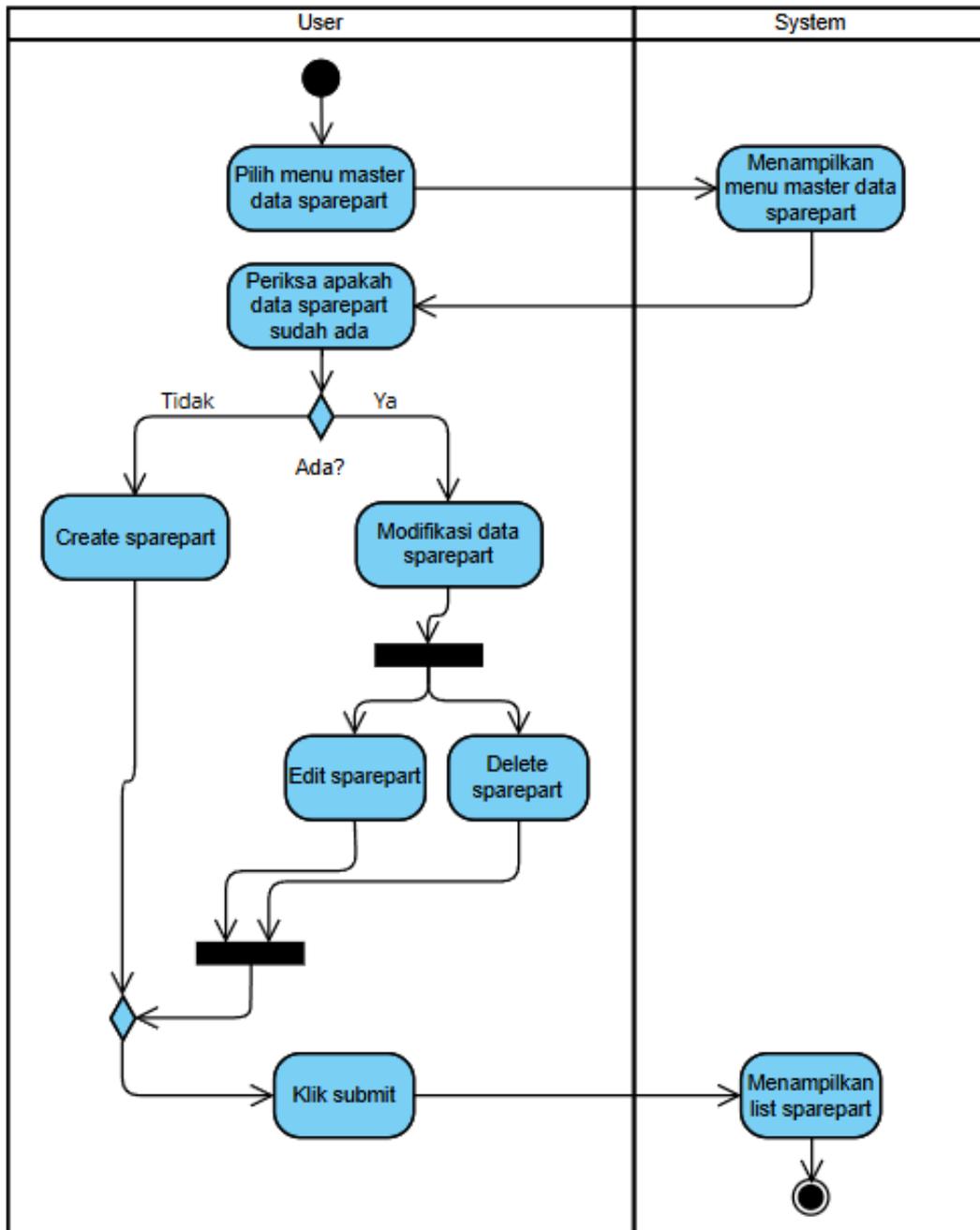


Gambar 4.2 Activity Diagram Mengelola Data User

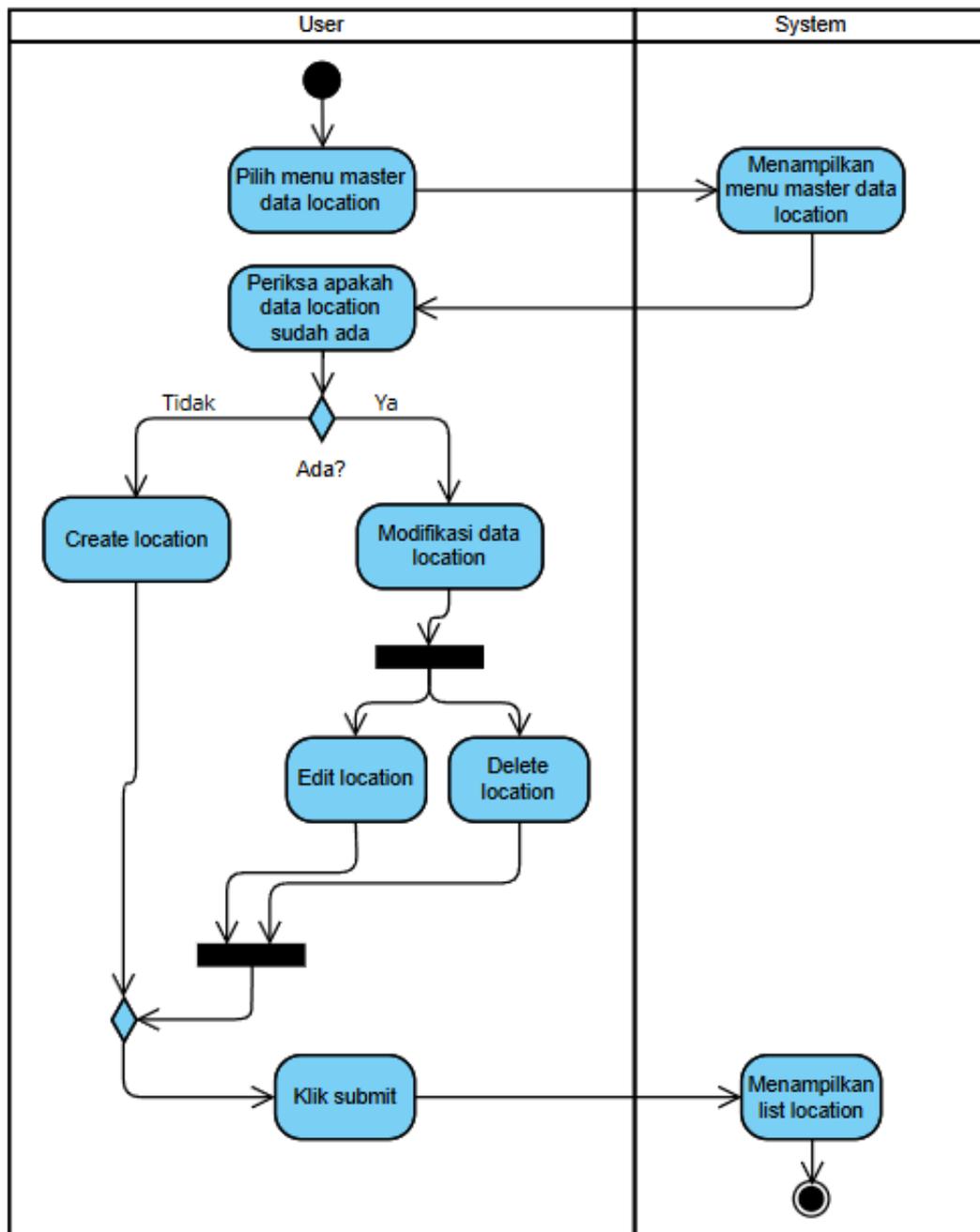
2. Activity Diagram Mengelola Data Sparepart

User membuka halaman *master data sparepart* kemudian sistem menampilkan halaman *master data sparepart*, user kemudian bisa memeriksa apakah *sparepart* yang dibutuhkan sudah ada atau belum jika belum ada user dapat membuat data *sparepart*. Apabila data *sparepart* sudah ada user dapat mengubah atau menghapus data *sparepart* sesuai dengan kebutuhan. Semua aksi dapat dilakukan dengan

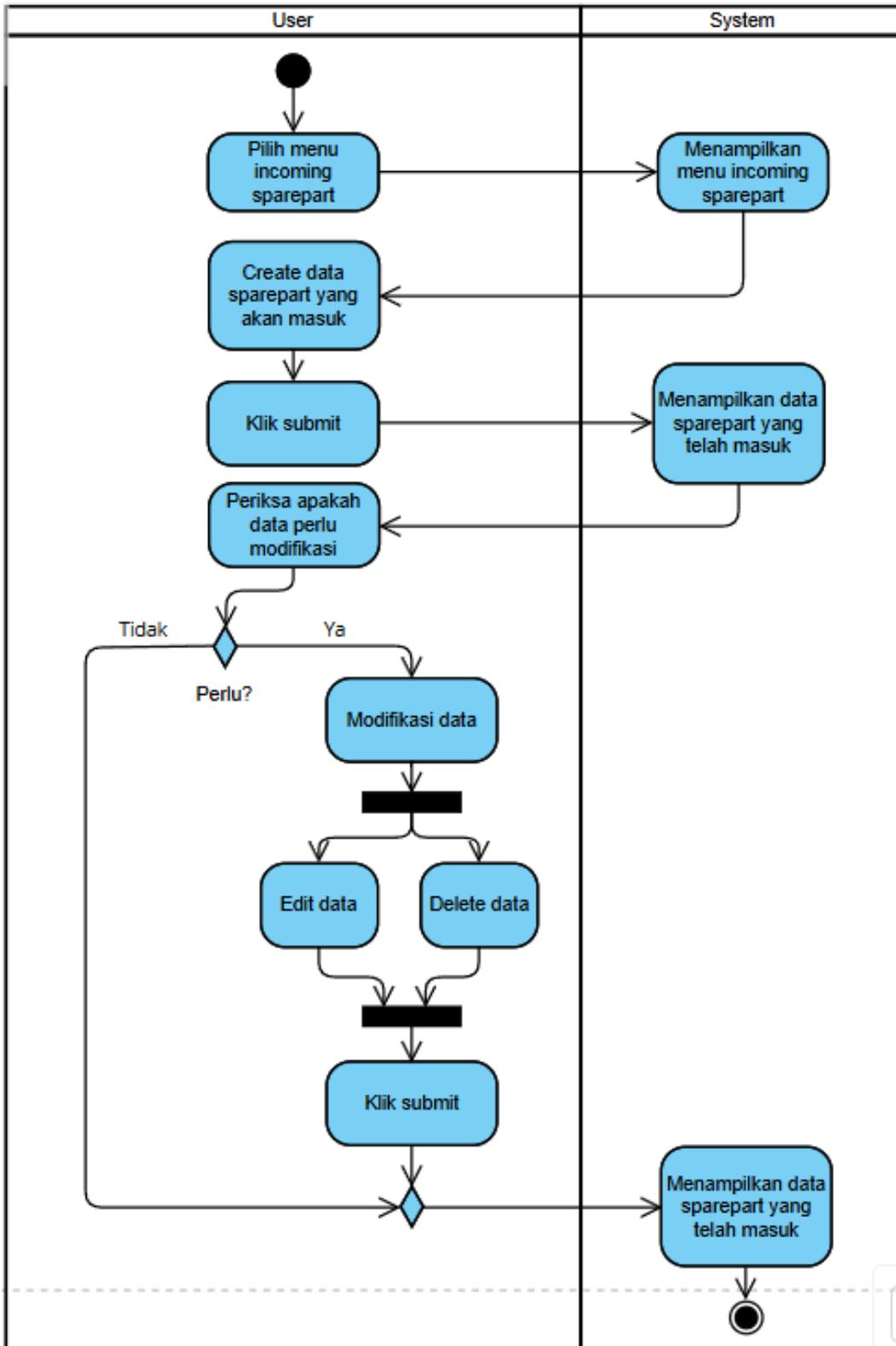
menekan tombol *submit* sehingga data yang diinput oleh *user* dapat diproses oleh sistem dan sistem menampilkan kembali list data *sparepart* terbaru. Selain mengelola data *sparepart* *user* dapat melakukan penambahan, modifikasi, dan penghapusan dari data lokasi, *sparepart* masuk, dan *sparepart* keluar sesuai dengan kebutuhan.



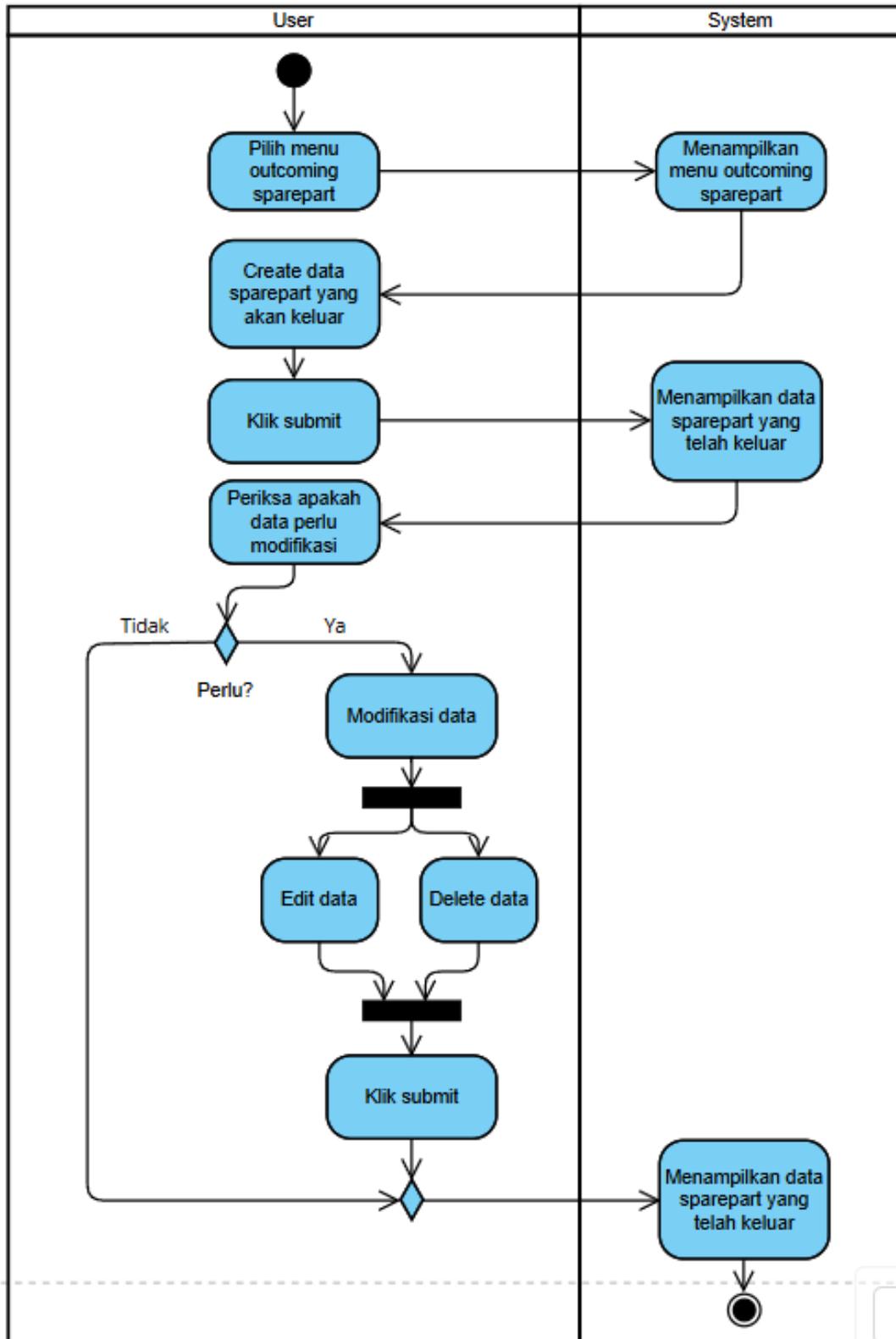
Gambar 4.3 Activity Diagram Mengelola Data Sparepart



Gambar 4.4 Activity Diagram Mengelola Data Location



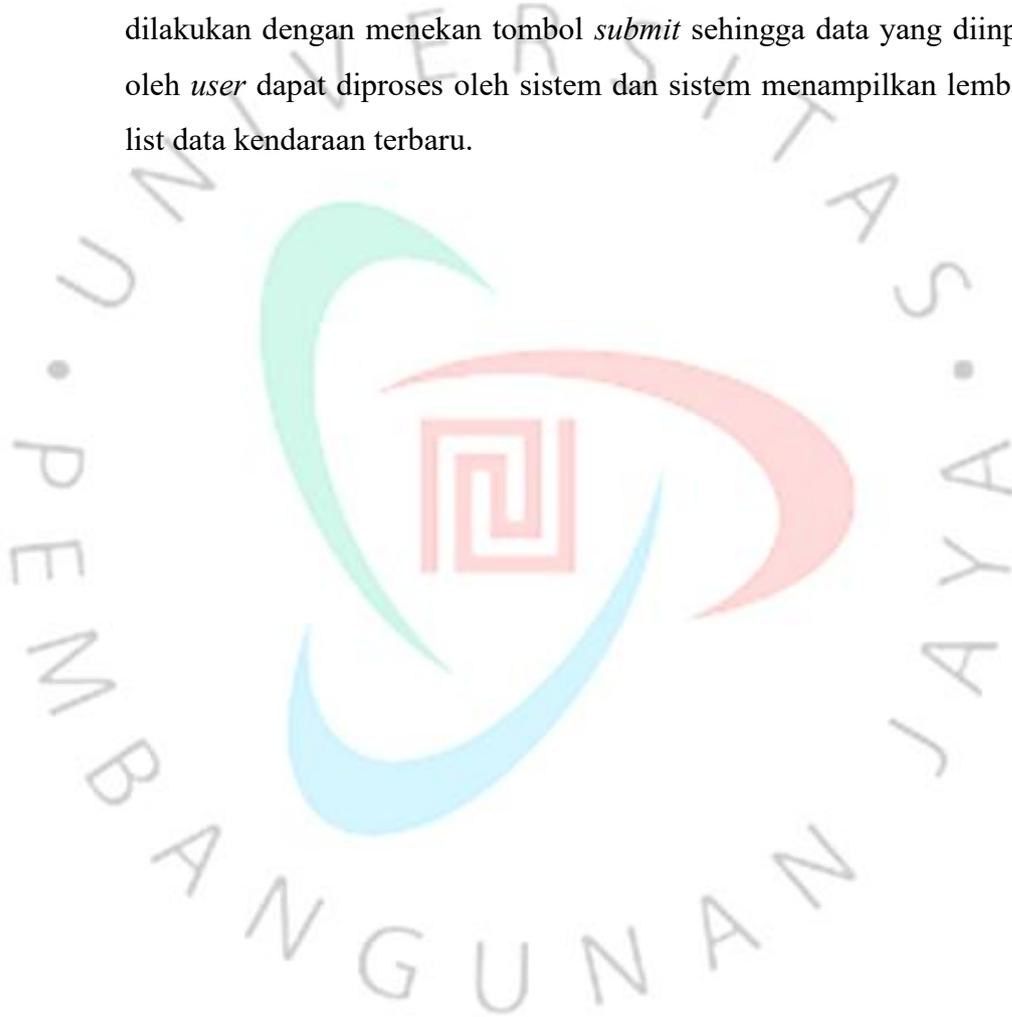
Gambar 4.5 Activity Diagram Mengelola Data Incoming Sparepart

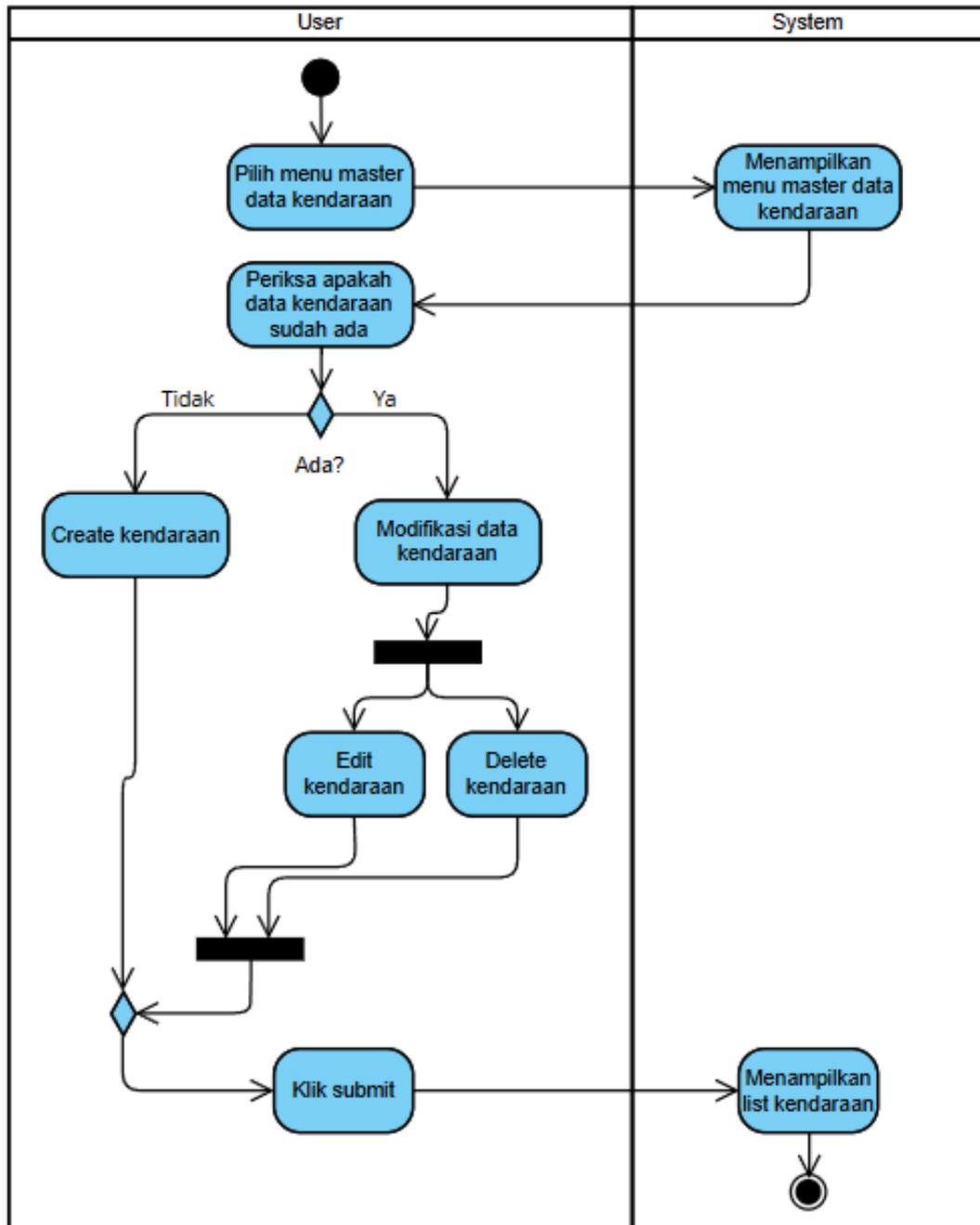


Gambar 4.6 Activity Diagram Mengelola Data Outcoming Sparepart

3. *Activity Diagram* Mengelola Data Kendaraan

User membuka halaman *master* data kendaraan kemudian sistem menampilkan halaman *master* data kendaraan, *user* kemudian bisa memeriksa apakah data kendaraan yang dibutuhkan terdapat pada aplikasi atau tidak, jika belum ada *user* dapat membuat data kendaraan. Apabila data kendaraan sudah ada *user* dapat mengubah atau menghapus data kendaraan sesuai dengan kebutuhan. Semua aksi dapat dilakukan dengan menekan tombol *submit* sehingga data yang diinput oleh *user* dapat diproses oleh sistem dan sistem menampilkan kembali list data kendaraan terbaru.





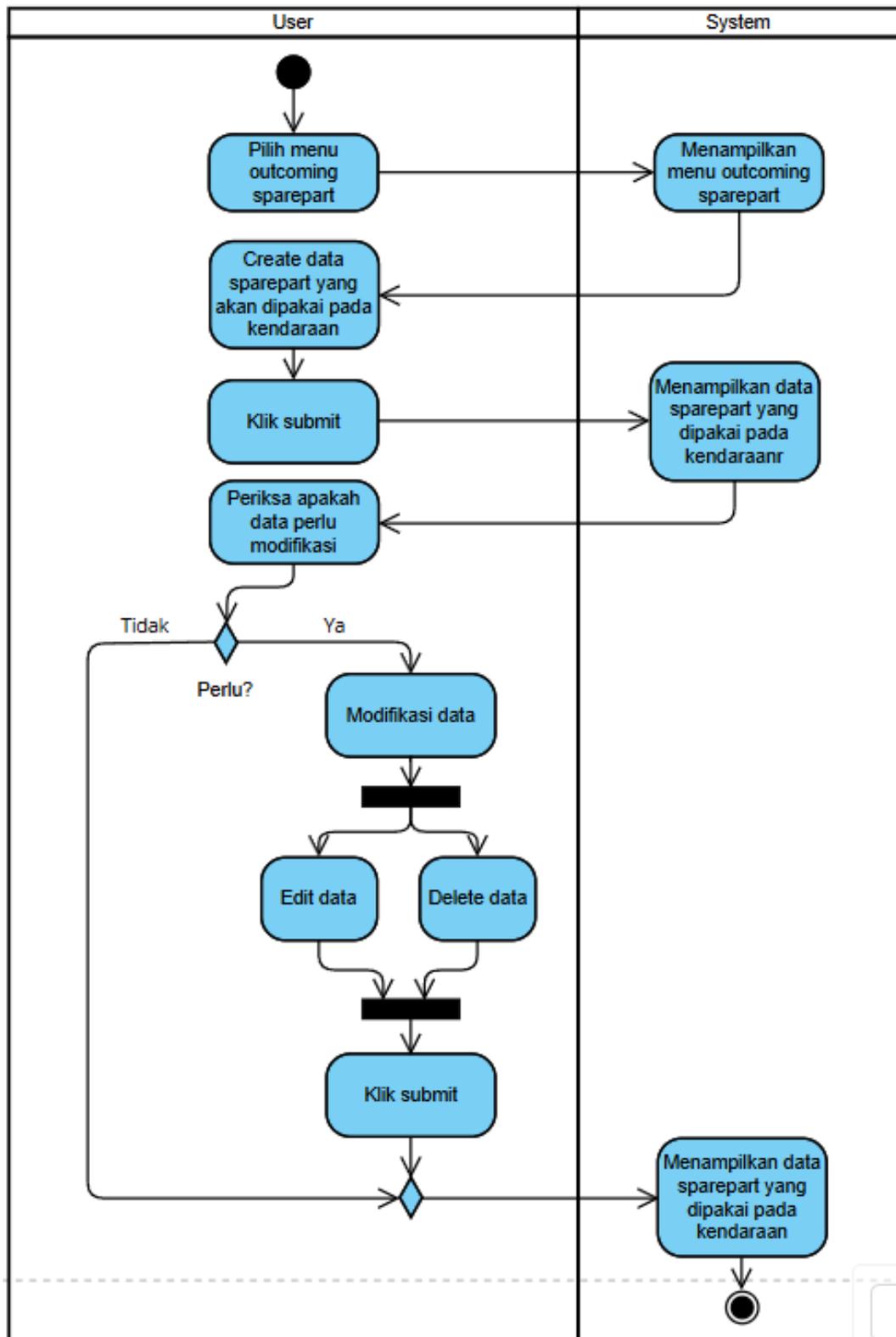
Gambar 4.7 Activity Diagram Mengelola Data Kendaraan

4. Activity Diagram Penggunaan Sparepart pada Kendaraan

User membuka halaman *outcoming sparepart* kemudian sistem menampilkan halaman *outcoming sparepart*, user kemudian bisa memeriksa apakah data *sparepart* pada kendaraan yang dibutuhkan sudah ada atau belum jika belum ada user dapat membuat data *sparepart* pada kendaraan. Apabila data *sparepart* pada kendaraan sudah ada user dapat mengubah atau menghapus data *sparepart* pada kendaraan sesuai

dengan kebutuhan. Semua aksi dapat dilakukan dengan menekan tombol *submit* sehingga data yang diinput oleh *user* dapat diproses oleh sistem dan sistem menampilkan kembali list data *sparepart* pada kendaraan terbaru.



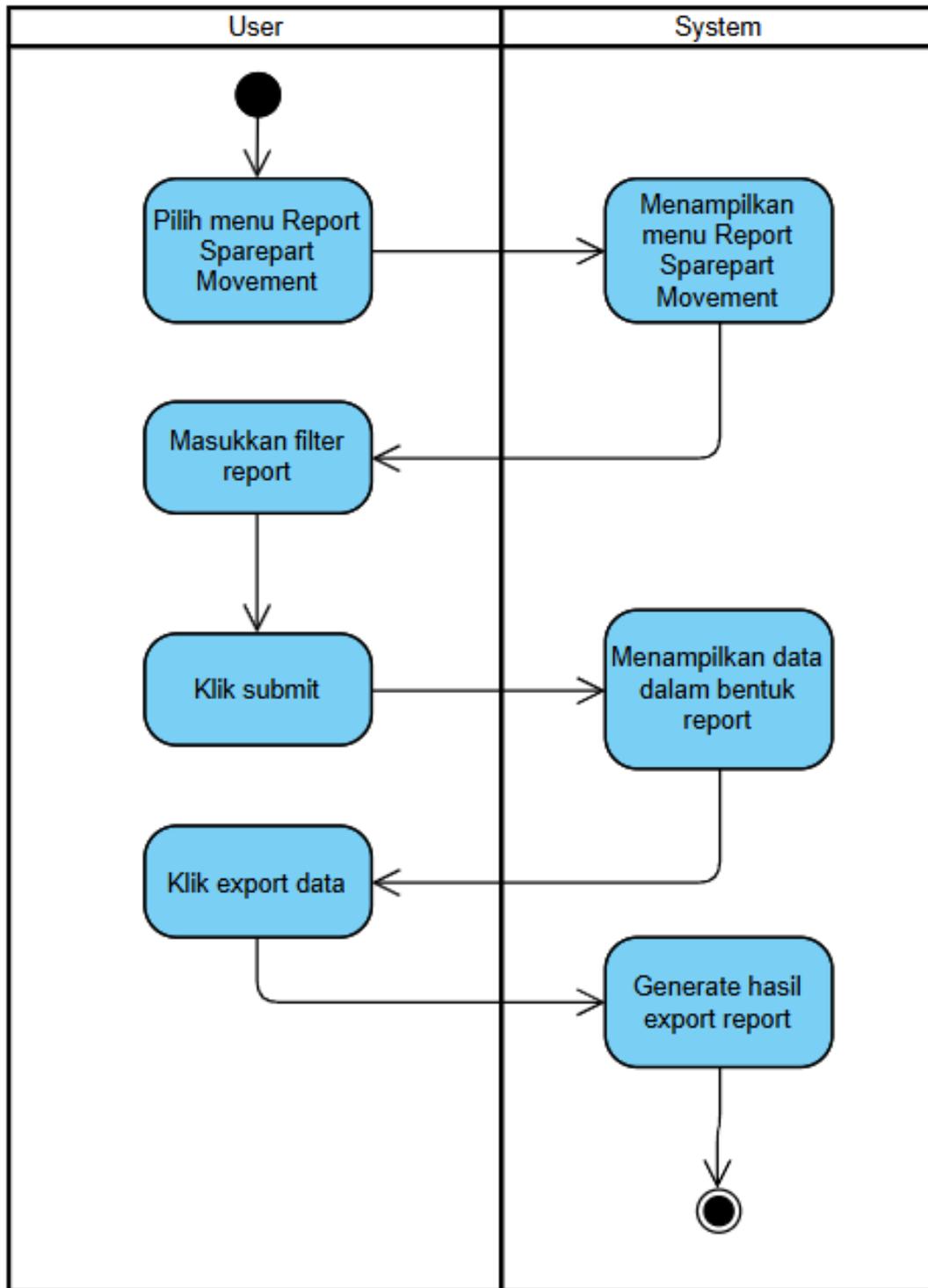


Gambar 4.8 Activity Diagram Mengelola Data Penggunaan Sparepart pada Kendaraan

5. *Activity Diagram Report Pergerakan Sparepart*

User membuka halaman *report sparepart movement*, kemudian sistem akan menampilkan halaman *report sparepart movement*. *User* dapat menginput filter untuk *report* sesuai dengan kebutuhan. Setelah *user submit* filter, sistem akan memproses dan menampilkan hasil dari *report* yang dibutuhkan. *User* juga dapat meng *export* hasil *report* dalam bentuk *excel* yang akan di *export* oleh sistem dan otomatis terunduh ke dalam penyimpanan *user*.



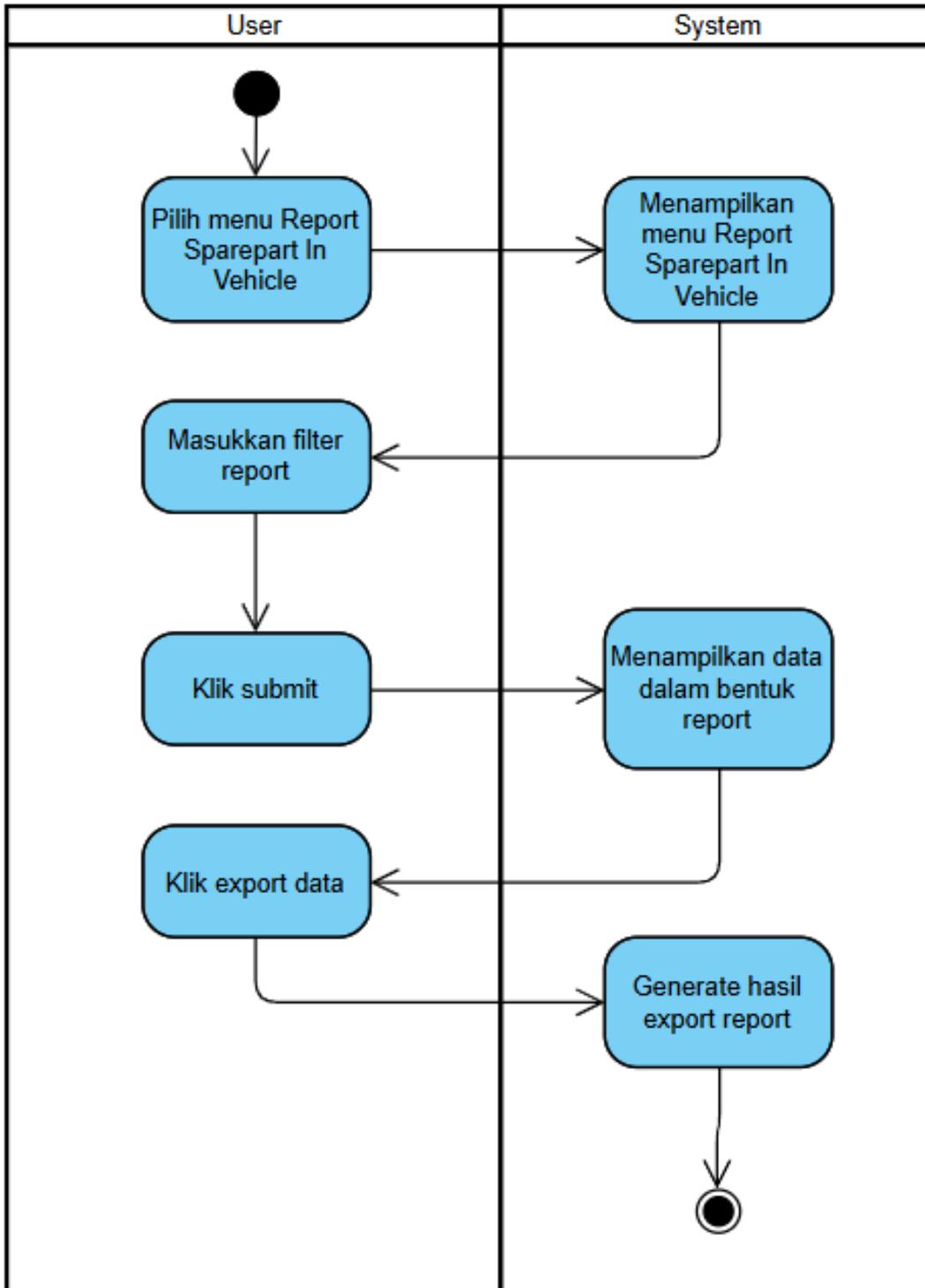


Gambar 4.9 Activity Diagram Report Pergerakan Sparepart

6. Activity Diagram Report Sparepart pada Kendaraan

User membuka halaman *report sparepart in vehicle*, kemudian sistem akan menampilkan halaman *report sparepart in vehicle*. User dapat menginput filter untuk *report* sesuai dengan kebutuhan. Setelah

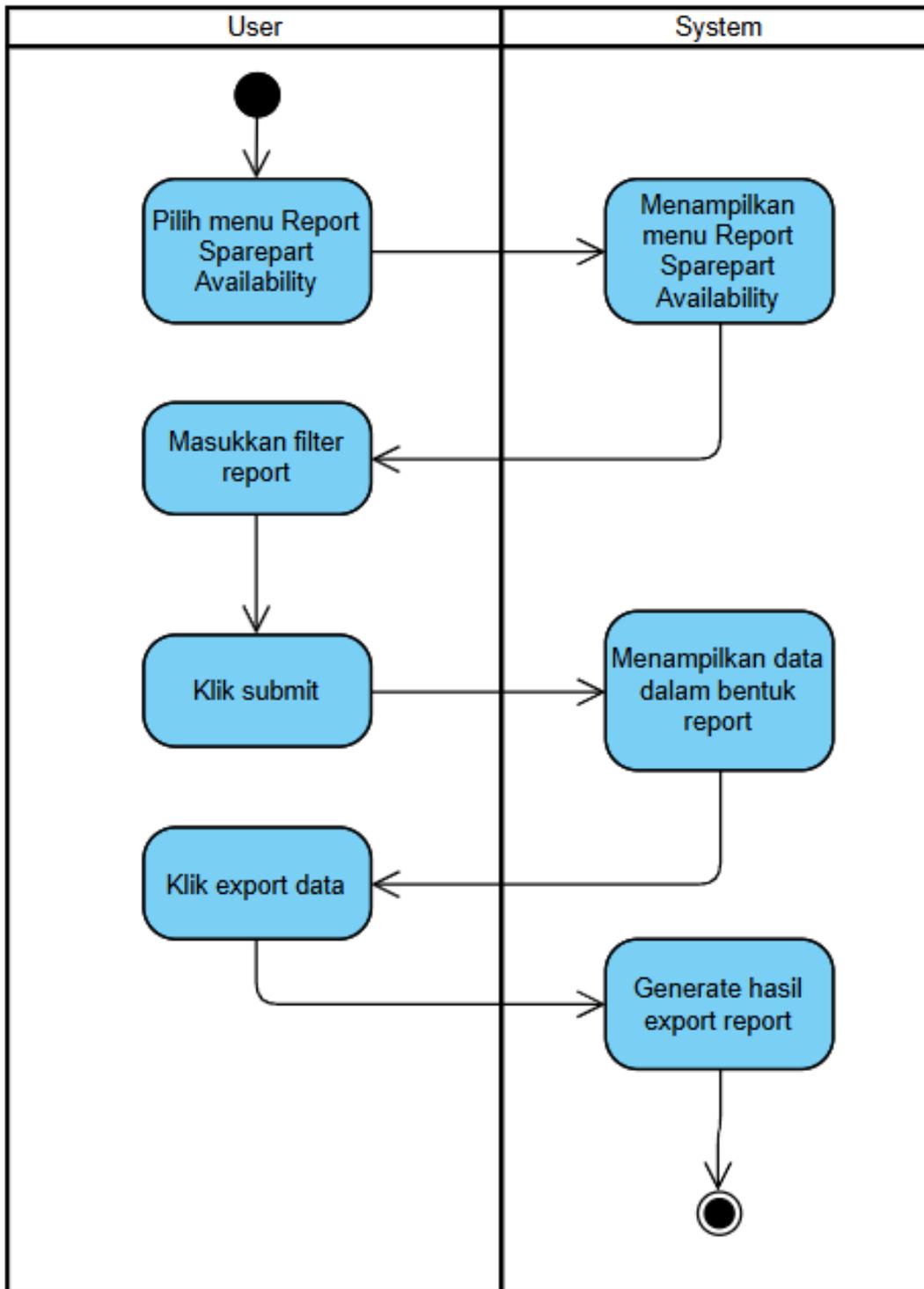
user submit filter, sistem akan memproses dan menampilkan hasil dari *report* yang dibutuhkan. *User* juga dapat meng *export* hasil *report* dalam bentuk *excel* yang akan di *export* oleh sistem dan otomatis terunduh ke dalam penyimpanan *user*.



Gambar 4.10 Activity Diagram Report Sparepart pada Kendaraan

7. *Activity Diagram Report Ketersediaan Barang*

User membuka halaman *report sparepart availability*, kemudian sistem akan menampilkan halaman *report sparepart availability*. *User* dapat menginput filter untuk *report* sesuai dengan kebutuhan. Setelah *user submit* filter, sistem akan memproses dan menampilkan hasil dari *report* yang dibutuhkan. *User* juga dapat meng *export* hasil *report* dalam bentuk *excel* yang akan di *export* oleh sistem dan otomatis terunduh ke dalam penyimpanan *user*.



Gambar 4.11 Activity Diagram Report Ketersediaan Barang

4.2.4 Class Diagram

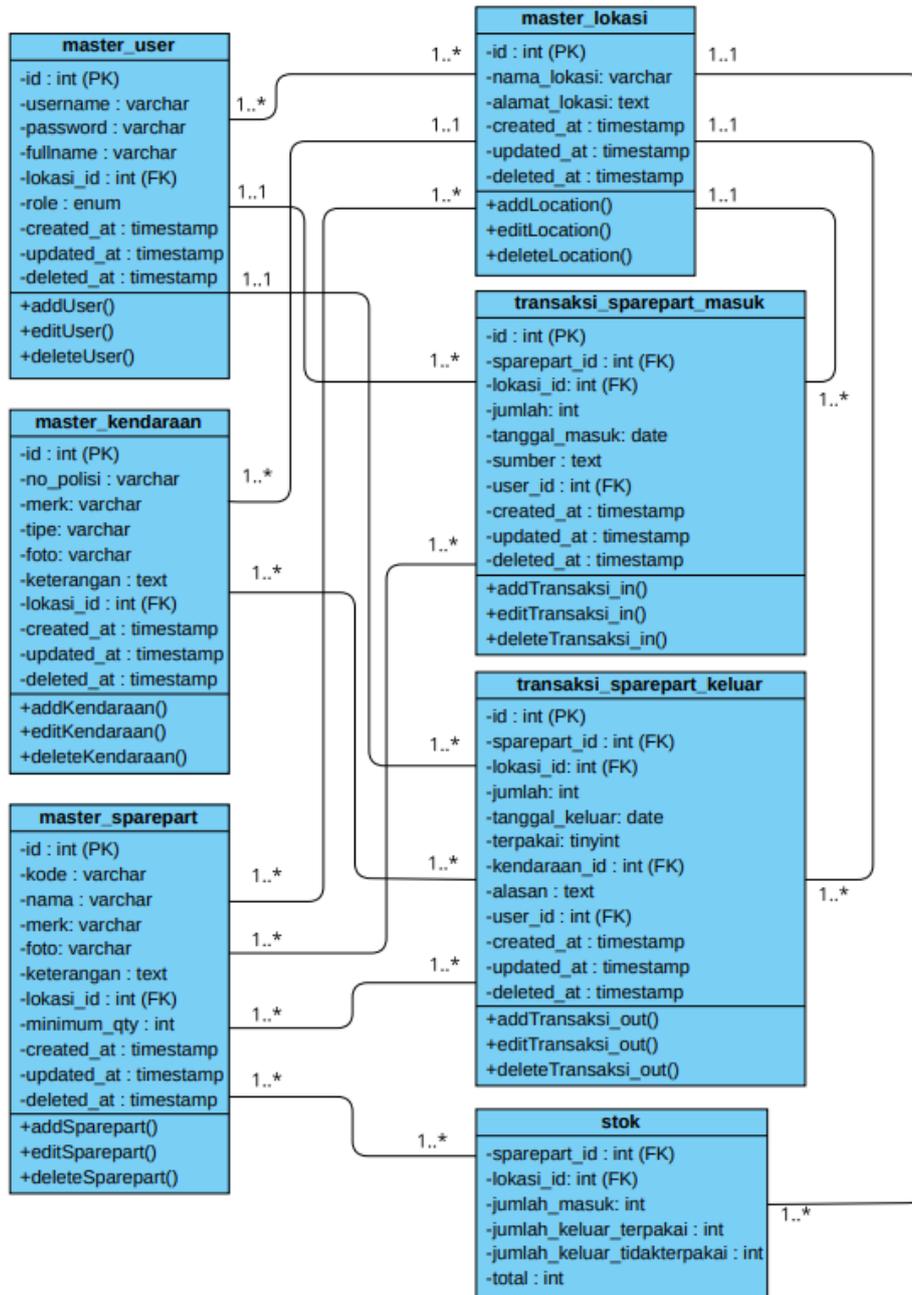
Class Diagram adalah salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk memvisualisasikan struktur statis dari sistem berbasis objek. Diagram ini menampilkan

berbagai *class* penyusun sistem, lengkap dengan berbagai atribut yang terkandung dalam *class* yang ada, fungsi atau metode yang dapat dijalankan, serta relasi antar *class* seperti asosiasi, pewarisan (*inheritance*), dan dependensi. Class diagram memainkan peran penting dalam tahap analisis dan perancangan karena menyajikan gambaran menyeluruh dan sistematis terhadap arsitektur sistem.

Setiap komponen dalam class diagram ditampilkan dalam bentuk persegi panjang yang terbagi menjadi tiga komponen. Komponen atas memuat nama kelas, bagian tengah berisi banyak atribut yang menggambarkan karakteristik atau data yang dimiliki oleh kelas, dan bagian bawah mencantumkan metode atau operasi yang bisa dilakukan oleh kelas tersebut. Relasi antar kelas digambarkan dengan garis penghubung, di mana simbol-simbol khusus digunakan untuk menjelaskan jenis relasi, seperti panah untuk menunjukkan pewarisan, atau garis putus-putus berpana untuk menandakan dependensi.

Class diagram sangat bermanfaat dalam proses pengembangan sistem karena membantu pengembang untuk memahami bagaimana data diorganisir dan bagaimana objek dalam sistem saling berinteraksi. Dengan *class diagram*, pengembang dapat mengetahui struktur data sistem secara rinci dan bagaimana logika program akan bekerja dalam kaitannya dengan objek-objek tersebut. Diagram ini juga memudahkan proses implementasi kode karena sudah terdapat kerangka dasar dari setiap kelas beserta atribut dan fungsinya.

Secara keseluruhan, *class diagram* merupakan alat bantu visual yang sangat penting untuk merancang sistem yang kompleks dan berskala besar. Diagram ini mempermudah kolaborasi antar anggota tim pengembang karena menyajikan informasi struktur sistem dengan jelas dan mudah dipahami. Dalam laporan tugas akhir, *class diagram* digunakan untuk memperkuat penjelasan mengenai desain sistem, serta menunjukkan rancangan struktur data dan relasi antar entitas dalam sistem yang akan dibangun. Berikut ini merupakan *class diagram* dari aplikasi pergudangan *sparepart* PT. OPQ.



Gambar 4.12 Class Diagram

4.2.5 Class Diagram Specification

Class Diagram Specification adalah penjabaran lebih rinci dari elemen-elemen yang terdapat pada sebuah *class diagram*. Spesifikasi ini memberikan detail tentang setiap kelas, mencakup nama kelas, atribut-atribut yang dimilikinya, tipe data dari masing-masing atribut, serta metode atau fungsi (*operations*) yang tersedia di dalam kelas tersebut, termasuk

parameter dan tipe kembalian (*return type*). Selain itu, juga dijelaskan relasi antar kelas seperti asosiasi, agregasi, komposisi, dan pewarisan.

Dalam pengembangan perangkat lunak, *class diagram specification* membantu tim pengembang memahami setiap komponen dalam *class diagram* secara menyeluruh sebelum tahap implementasi dimulai. Misalnya, untuk atribut dalam sebuah kelas, spesifikasi akan menjelaskan apakah atribut tersebut bersifat *private*, *public*, atau *protected*, dan apakah memiliki nilai *default*. Demikian juga untuk metode, spesifikasi akan mencantumkan nama metode, tipe parameter masukan, dan nilai balik (jika ada), serta deskripsi singkat mengenai fungsi dari metode tersebut.

Selain itu, spesifikasi ini juga mencakup deskripsi relasi antar kelas. Misalnya, apakah hubungan tersebut bersifat *one-to-one*, *one-to-many*, atau *many-to-many*. Informasi ini penting dengan tujuan mempermudah proses pembuatan database dan penulisan kode program yang konsisten dengan rancangan yang telah dibuat. Setiap relasi biasanya dijelaskan dengan peran (*role*) masing-masing kelas dalam relasi tersebut dan kardinalitasnya.

Dengan adanya *class diagram specification*, dokumentasi sistem menjadi lebih jelas dan terstruktur. Spesifikasi ini sangat berguna dalam proses validasi dan verifikasi sistem karena memberikan panduan lengkap bagi pengembang, tester, maupun tim dokumentasi dalam memahami bagaimana sistem dirancang dan diharapkan beroperasi. Dalam laporan tugas akhir, bagian ini berfungsi sebagai validasi bahwa desain sistem dibangun selain dibuat secara visual, namun juga dijabarkan secara detail dan teknis untuk mendukung implementasi yang akurat.

1. Table *Master User*

Tabel 4.8. Table *Master Data User*

Nama	Tipe Data	Key	Ket
id	Int (11)	Primary	Id <i>user</i>
<i>username</i>	Varchar(50)		<i>Username user</i>

<i>password</i>	Varchar(255)		<i>Password user</i>
<i>fullname</i>	Varchar(100)		Nama lengkap <i>user</i>
lokasi_id	Int(11)	Foreign	Id lokasi
<i>role</i>	Enum		<i>Role user</i>
created_at	timestamp		Waktu dibuat
updated_at	timestamp		Waktu update
deleted_at	datetime		Waktu <i>delete</i>

2. Table *Master Sparepart*

Tabel 4.9. Table *Master Data Sparepart*

ama	Tipe Data	Key	Ket
id	Int (11)	Primary	Id <i>sparepart</i>
kode	Varchar(50)		Kode <i>sparepart</i>
nama	Varchar(100)		Nama <i>sparepart</i>
merk	Varchar(50)		Merk <i>sparepart</i>
foto	Varchar(255)		Foto <i>sparepart</i>
keterangan	text		Keterangan
lokasi_id	Int (11)	Foreign	Id lokasi
minimum_qty	Int (11)		Minimum jumlah <i>sparepart</i>
created_at	timestamp		Waktu dibuat
updated_at	timestamp		Waktu update
deleted_at	datetime		Waktu <i>delete</i>

3. Table *Master* Kendaraan

Tabel 4.10. Table Master Data Kendaraan

Nama	Tipe Data	Key	Ket
id	Int (11)	Primary	Id kendaraan
no_polisi	Varchar(20)		Nomor polisi kendaraan
Merk	Varchar(50)		Merk kendaraan
Tipe	Varchar(50)		Tipe kendaraan
foto	Varchar(255)		Foto kendaraan
keterangan	text		Keterangan
lokasi_id	Int (11)	Foreign	Id lokasi
created_at	timestamp		Waktu dibuat
updated_at	timestamp		Waktu update
deleted_at	datetime		Waktu <i>delete</i>

4. Table *Master* Lokasi

Tabel 4.11. Table Master Data Lokasi

Nama	Tipe Data	Key	Ket
id	Int (11)	Primary	Id lokasi
nama_lokasi	Varchar(20)		Nama lokasi
alamat_lokasi	text		Alamat lokasi
created_at	timestamp		Waktu dibuat
updated_at	timestamp		Waktu update
deleted_at	datetime		Waktu <i>delete</i>

5. Table Transaksi *Sparepart* Masuk

Tabel 4.12. Table Transaksi *Sparepart* Masuk

Nama	Tipe Data	Key	Ket
id	Int (11)	Primary	Id transaksi masuk
<i>sparepart_id</i>	Int (11)	Foreign	Id <i>sparepart</i>
lokasi_id	Int (11)	Foreign	Id lokasi
jumlah	Int (11)		Jumlah <i>sparepart</i> masuk
tanggal_masuk	<i>date</i>		Tanggal masuk <i>sparepart</i>
sumber	text		Sumber <i>sparepart</i> masuk
<i>user_id</i>	Int (11)	Foreign	Id <i>user</i>
created_at	timestamp		Waktu dibuat
updated_at	timestamp		Waktu update
deleted_at	datetime		Waktu <i>delete</i>

6. Table Transaksi *Sparepart* Keluar

Tabel 4.13. Table Transaksi *Sparepart* Keluar

Nama	Tipe Data	Key	Ket
id	Int (11)	Primary	Id transaksi keluar
<i>sparepart_id</i>	Int (11)	Foreign	Id <i>sparepart</i>
lokasi_id	Int (11)	Foreign	Id lokasi
tanggal_keluar	<i>date</i>		Tanggal keluar <i>sparepart</i>

jumlah	Int (11)		Jumlah <i>sparepart</i> keluar
terpakai	text		Pilihan apakah <i>sparepart</i> terpakai di kendaraan atau tidak
kendaraan_id	Int (11)	Foreign	Id kendaraan
alasan	text		Alasan pemakaian
<i>user_id</i>	Int (11)	Foreign	Id <i>user</i>
created_at	timestamp		Waktu dibuat
updated_at	timestamp		Waktu update

7. Table Stok

Tabel 4.14. Table Stok

Nama	Tipe Data	Key	Ket
<i>sparepart_id</i>	Int (11)	Foreign	Id <i>sparepart</i>
lokasi_id	Int (11)	Foreign	Id lokasi
jumlah_masuk	Int (11)		Jumlah <i>sparepart</i> masuk
jumlah_keluar_terpakai	Int (11)		Jumlah <i>sparepart</i> keluar terpakai
jumlah_keluar_tidakterpakai	Int (11)		Jumlah <i>sparepart</i>

			keluar tidak terpakai
total	Int (11)		Total qty <i>sparepart</i>

4.2.6 Sequence Diagram

Diagram bertujuan agar memvisualisasikan urutan interaksi tiap objek yang menyusun suatu *software* berdasarkan alur *timeline* adalah *Sequence Diagram*. Diagram ini menekankan komunikasi antar objek maupun aktor yang terlibat, serta memperlihatkan aliran pesan yang dikirim dan diterima dalam suatu proses secara berurutan. *Sequence* diagram umumnya digunakan pada tahap analisis dan desain sistem, terutama dalam proses pengembangan sistem *software* yang berbasis objek. Fokus utama diagram ini adalah memperlihatkan bagaimana objek-objek saling berinteraksi selama eksekusi sebuah skenario atau fungsi sistem. Setiap pesan yang dipertukarkan digambarkan secara kronologis, dari atas ke bawah, untuk memberikan pemahaman yang jelas mengenai jalannya komunikasi dan respon antar objek dalam suatu skenario sistem tertentu.

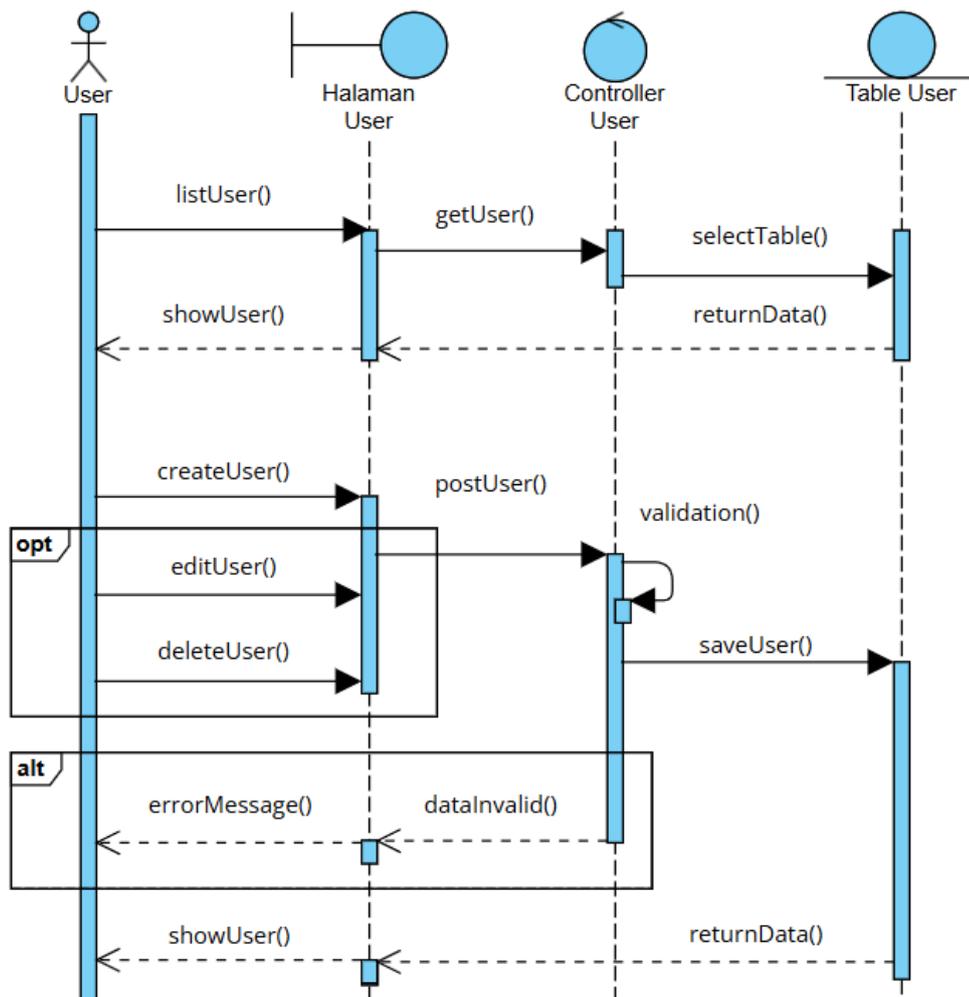
Dalam *sequence diagram*, elemen-elemen utama yang ditampilkan terdiri dari aktor atau objek, lifeline, pesan (message), dan activation bar. Aktor atau objek digambarkan dalam bentuk kotak di bagian atas diagram, yang kemudian dihubungkan dengan garis vertikal ke bawah yang disebut lifeline. Garis lifeline merepresentasikan keberadaan objek dalam rentang waktu tertentu. Sementara itu, pesan antar objek digambarkan dalam bentuk panah horizontal dari satu objek ke objek lain, menunjukkan bahwa suatu aksi atau metode dipanggil dalam urutan tertentu. Activation bar (balok aktivitas) menunjukkan saat suatu objek sedang memproses sebuah aksi atau metode.

Sequence diagram sangat berguna dalam menjelaskan bagaimana sistem bekerja secara dinamis, karena menggambarkan alur komunikasi antara pengguna dan sistem atau antar komponen sistem itu sendiri secara detail. Dengan adanya *sequence diagram*, pengembang sistem dapat

memahami lebih jelas bagaimana setiap komponen saling terhubung dan bekerja sama untuk menyelesaikan suatu proses. Selain itu, diagram ini juga membantu tim pengembang untuk mengidentifikasi kebutuhan sistem, potensi kesalahan logika, serta mempermudah proses implementasi kode karena alur proses sudah digambarkan dengan jelas sebelumnya.

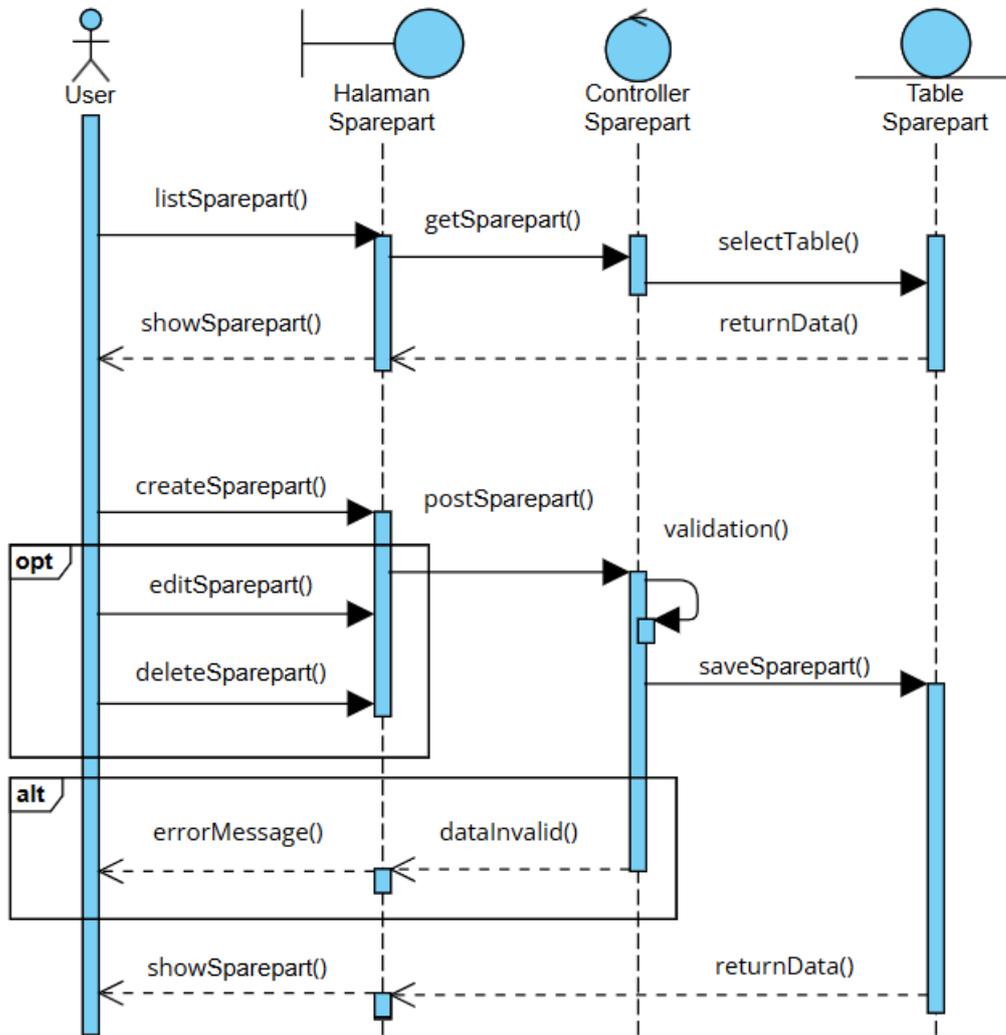
Secara umum, *sequence diagram* diterapkan untuk membantu dalam penggambaran perancangan sistem yang kompleks secara efektif, karena diagram ini menyajikan alur komunikasi secara runut dan sistematis. Dalam laporan tugas akhir, penyajian *sequence diagram* sangat penting untuk mendukung penjelasan proses-proses dalam sistem yang sedang dirancang, serta memberikan gambaran konkret tentang interaksi antar entitas yang terlibat dalam sistem tersebut.

1. *Sequence Diagram* Mengelola Data User

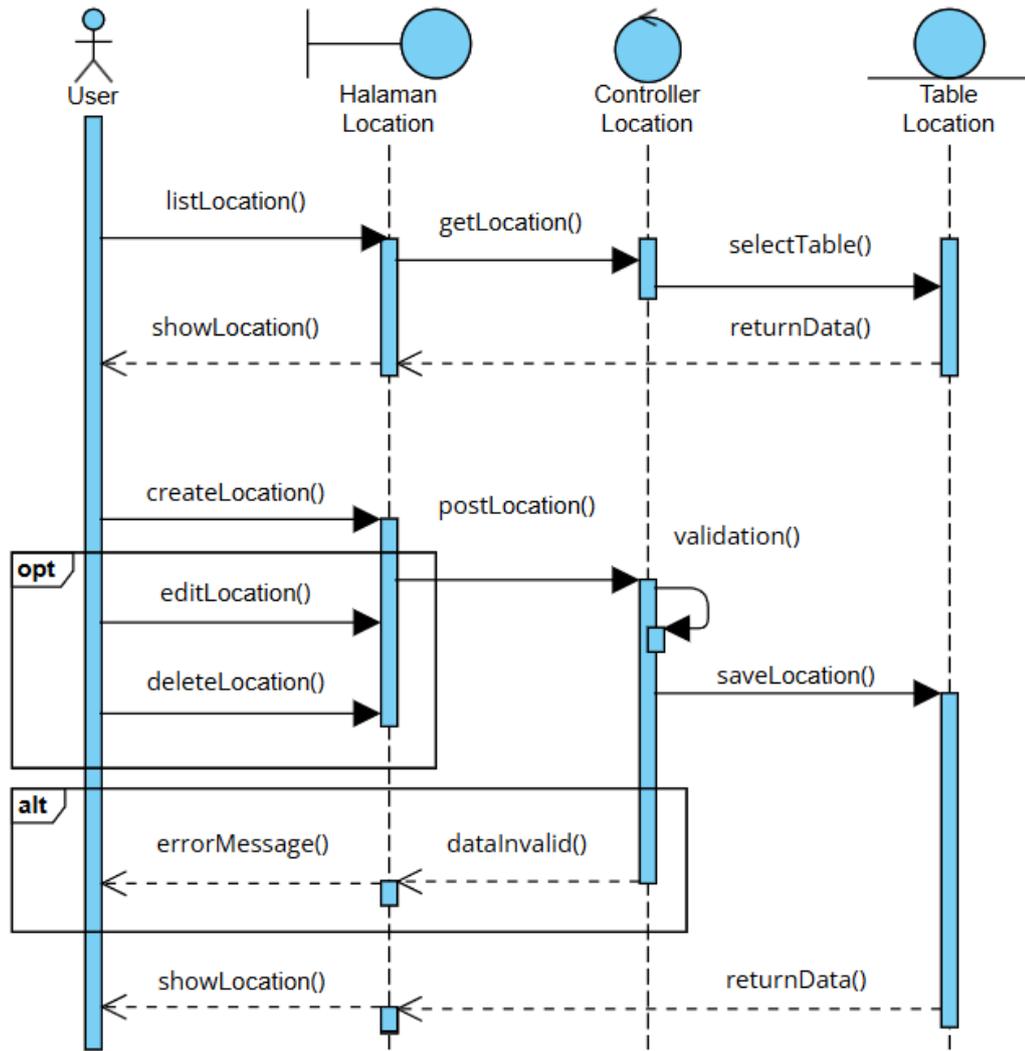


Gambar 4.13 *Sequence Diagram* Kelola Data User

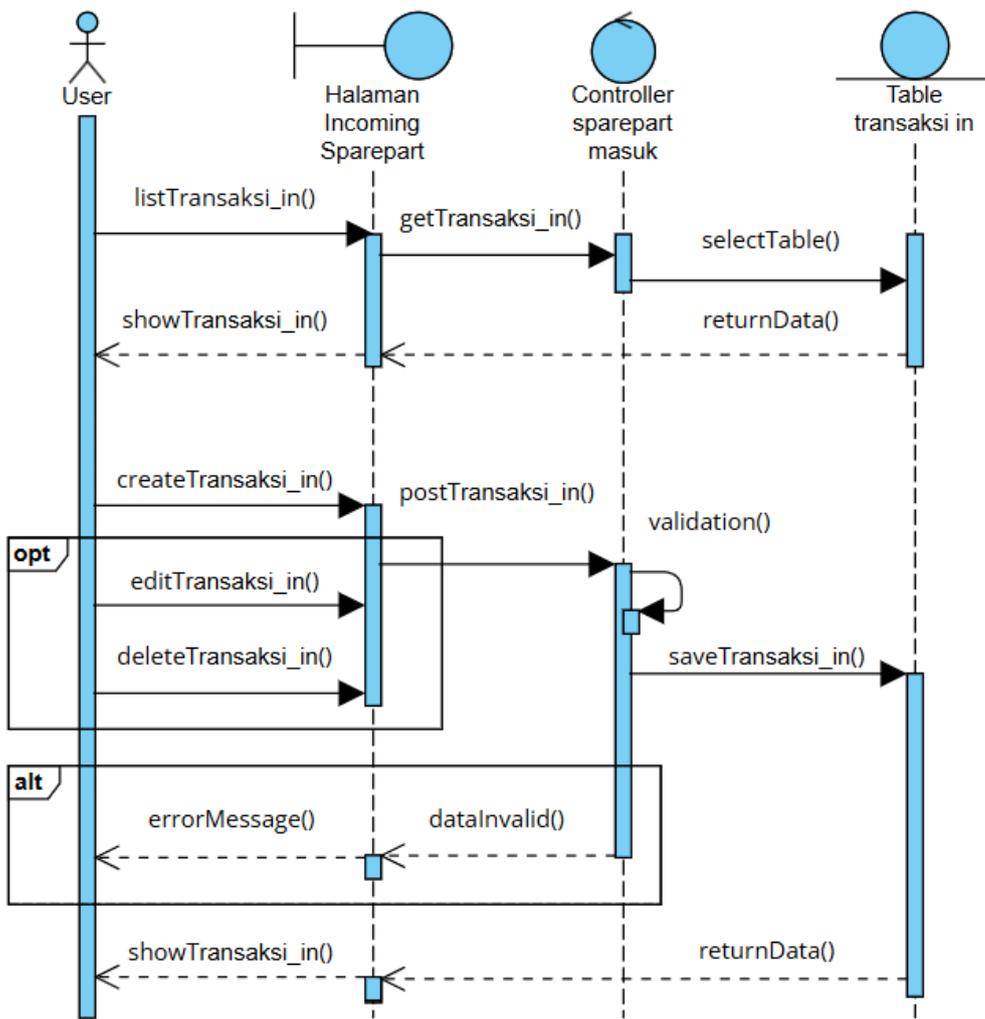
2. Sequence Diagram Mengelola Data Sparepart



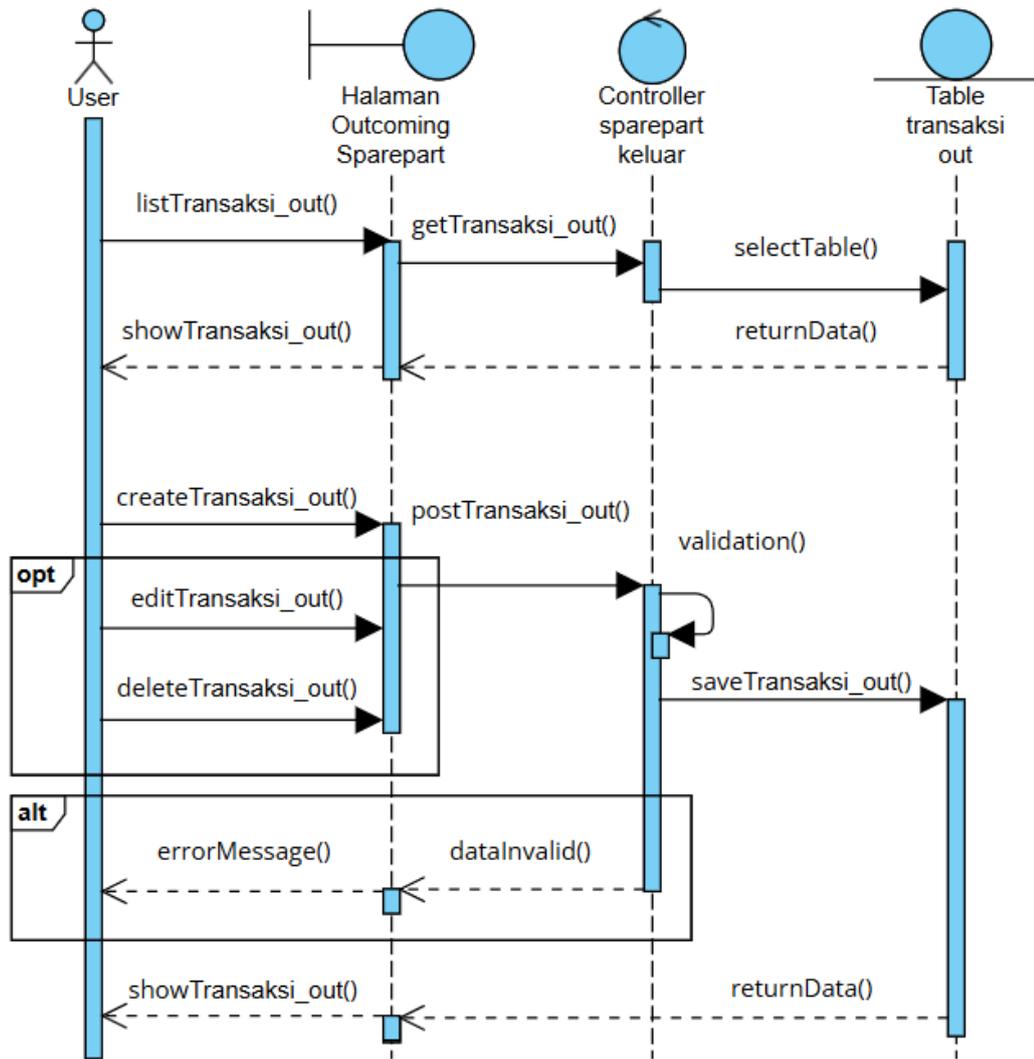
Gambar 4.14 Sequence Diagram Kelola Data Sparepart



Gambar 4.15 Sequence Diagram Kelola Data Location

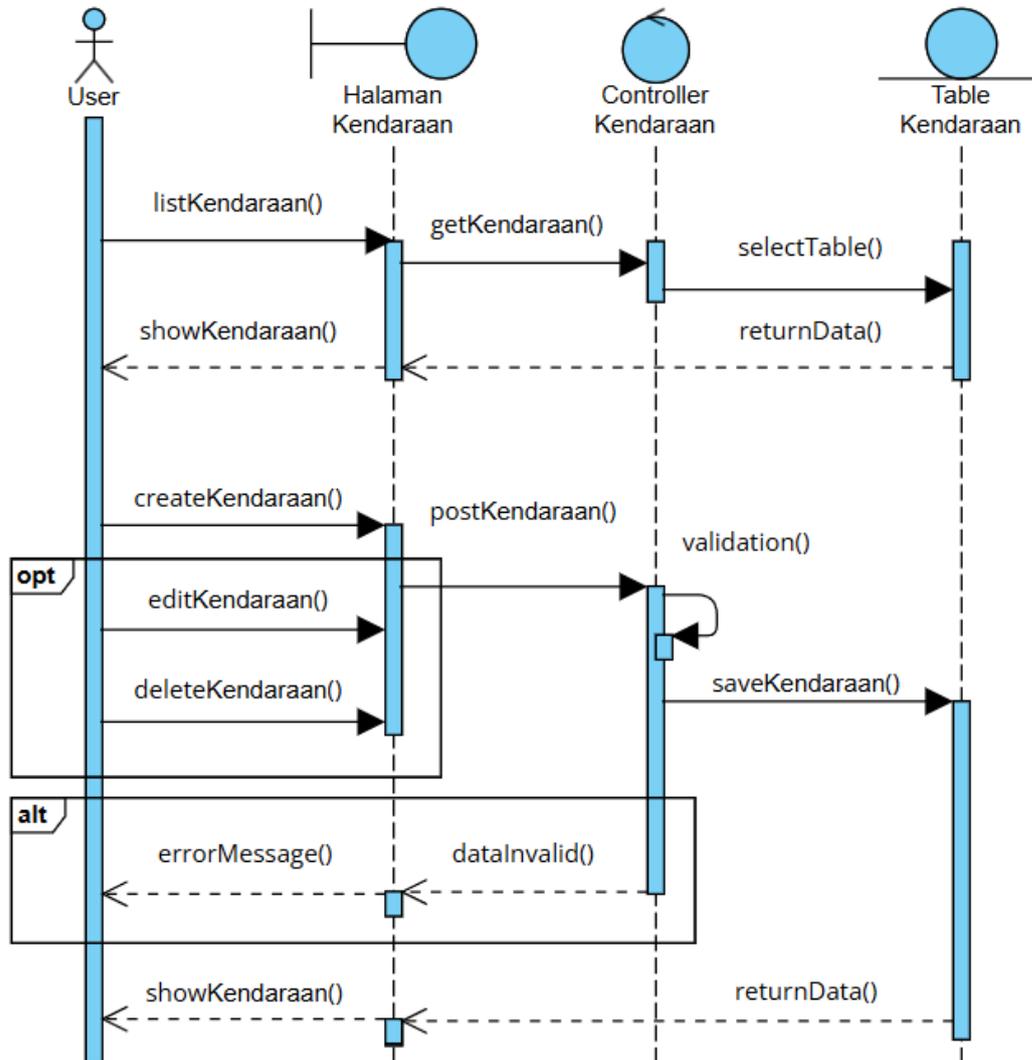


Gambar 4.16 Sequence Diagram Kelola Data Sparepart Masuk



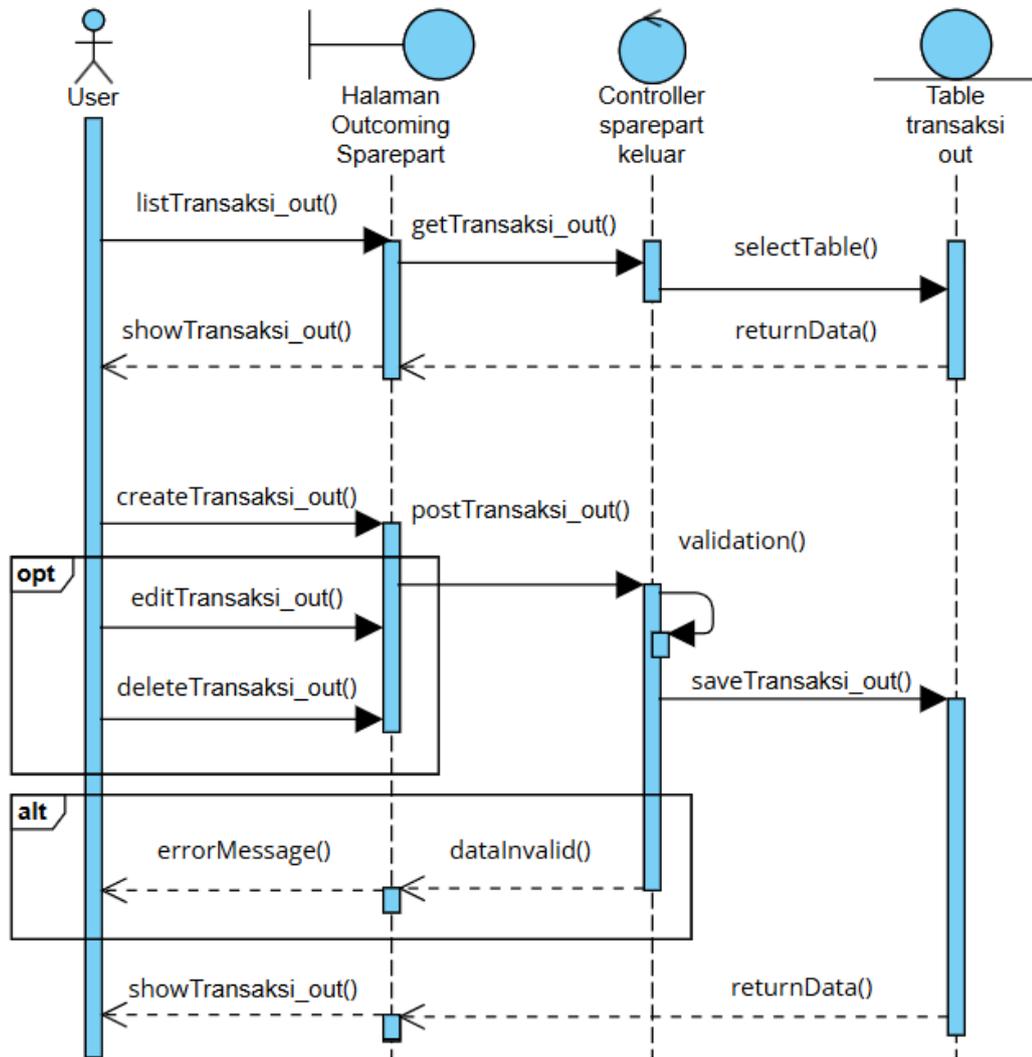
Gambar 4.17 Sequence Diagram Kelola Data Sparepart Keluar

3. Sequence Diagram Mengelola Data Kendaraan



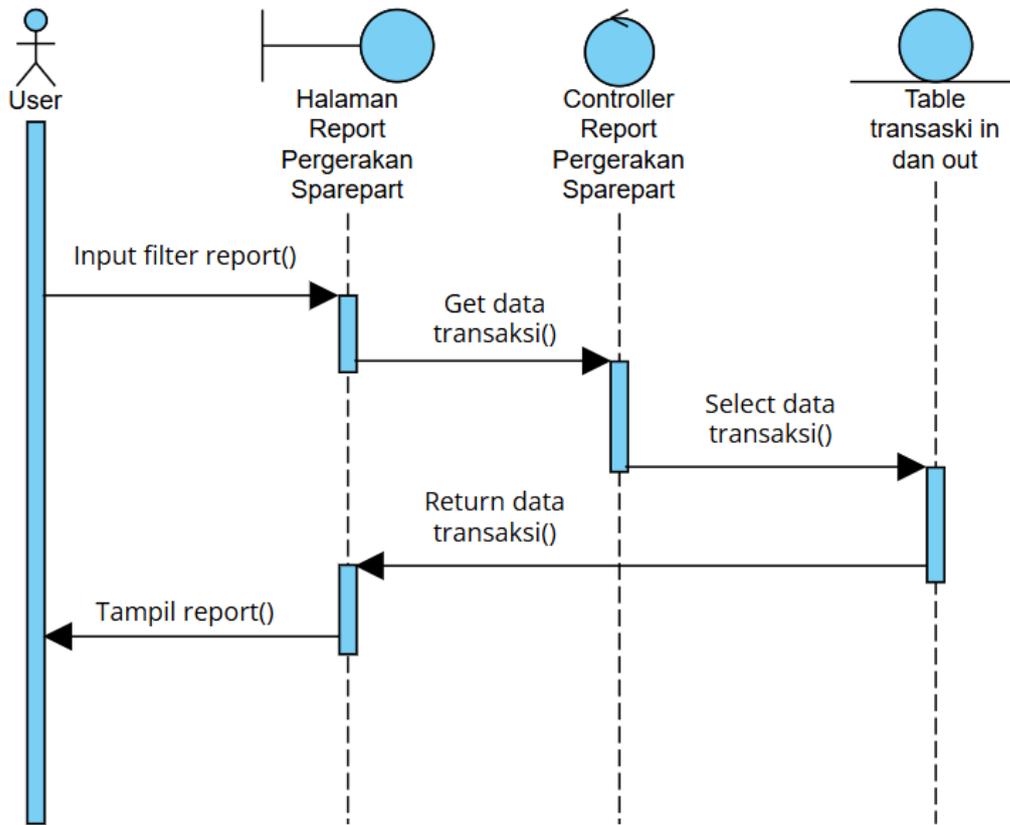
Gambar 4.18 Sequence Diagram Kelola Data Kendaraan

4. Sequence Diagram Penggunaan Sparepart pada Kendaraan

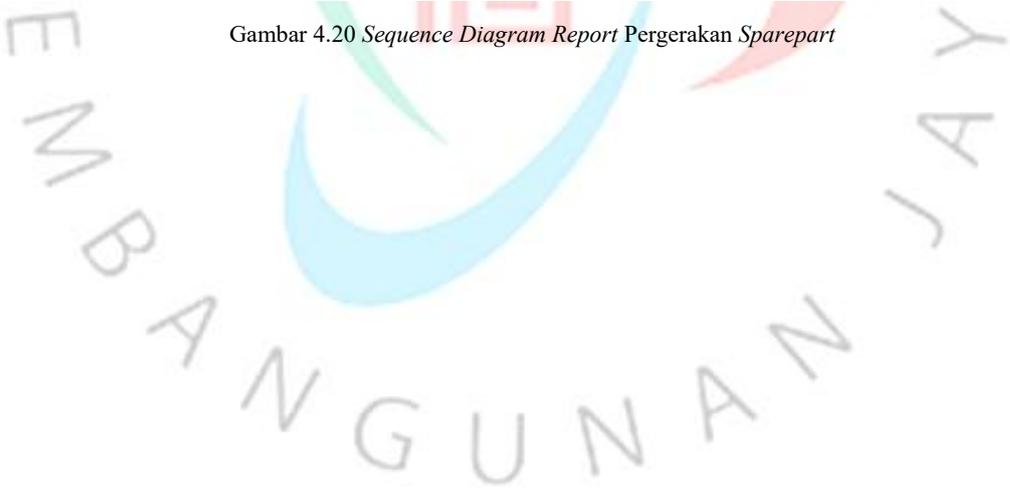


Gambar 4.19 Sequence Diagram Kelola Data Penggunaan Sparepart pada Kendaraan

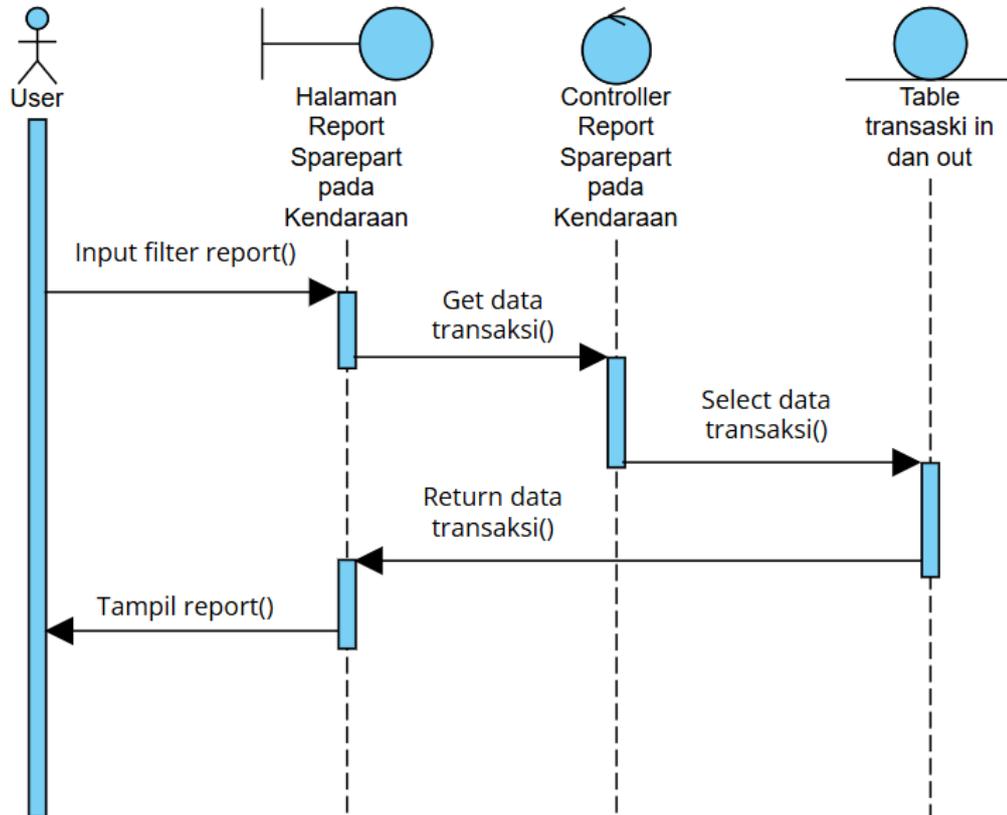
5. Sequence Diagram Report Pergerakan Sparepart



Gambar 4.20 Sequence Diagram Report Pergerakan Sparepart

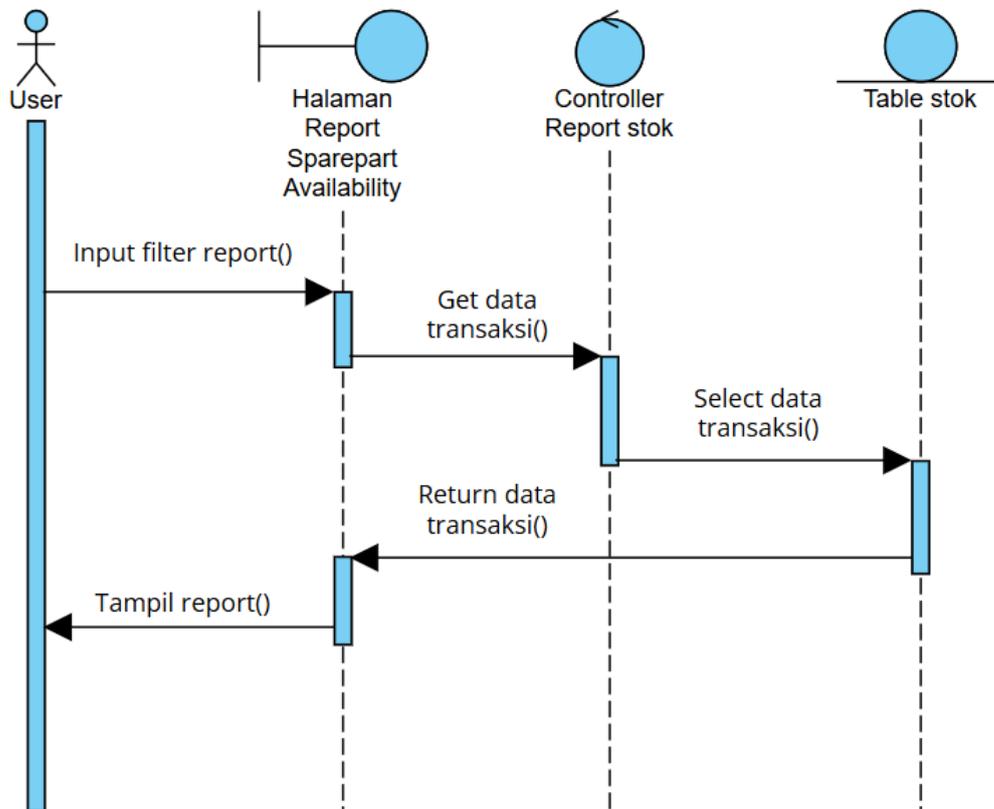


6. *Sequence Diagram Report Sparepart pada Kendaraan*



Gambar 4.21 *Sequence Diagram Report Sparepart pada Kendaraan*

7. Sequence Diagram Report Ketersediaan Barang



Gambar 4.22 Sequence Diagram Report Ketersediaan Barang

4.3 Perancangan Antar Muka Pengguna

Perancangan antarmuka pengguna atau *user interface* (UI) merupakan bagian dalam pengembangan sistem informasi yang penting karena menjadi titik interaksi langsung antara pengguna dan sistem. Tujuan dari perancangan ini yaitu untuk menggambarkan tampilan yang mudah dipahami, intuitif, dan nyaman digunakan oleh pengguna dalam menjalankan fungsionalitas yang tersedia dalam sistem. Dalam laporan tugas akhir, bagian ini menjelaskan rancangan tampilan dari halaman-halaman utama sistem sesuai kebutuhan dan alur penggunaan yang telah dianalisis sebelumnya.

Perancangan UI dimulai dengan menentukan struktur dan navigasi sistem berdasarkan fungsionalitas utama, seperti halaman *login*, *dashboard*, pengelolaan data, pencarian, input, dan laporan. Setiap tampilan dirancang berdasarkan prinsip *usability* yang baik, seperti konsistensi desain,

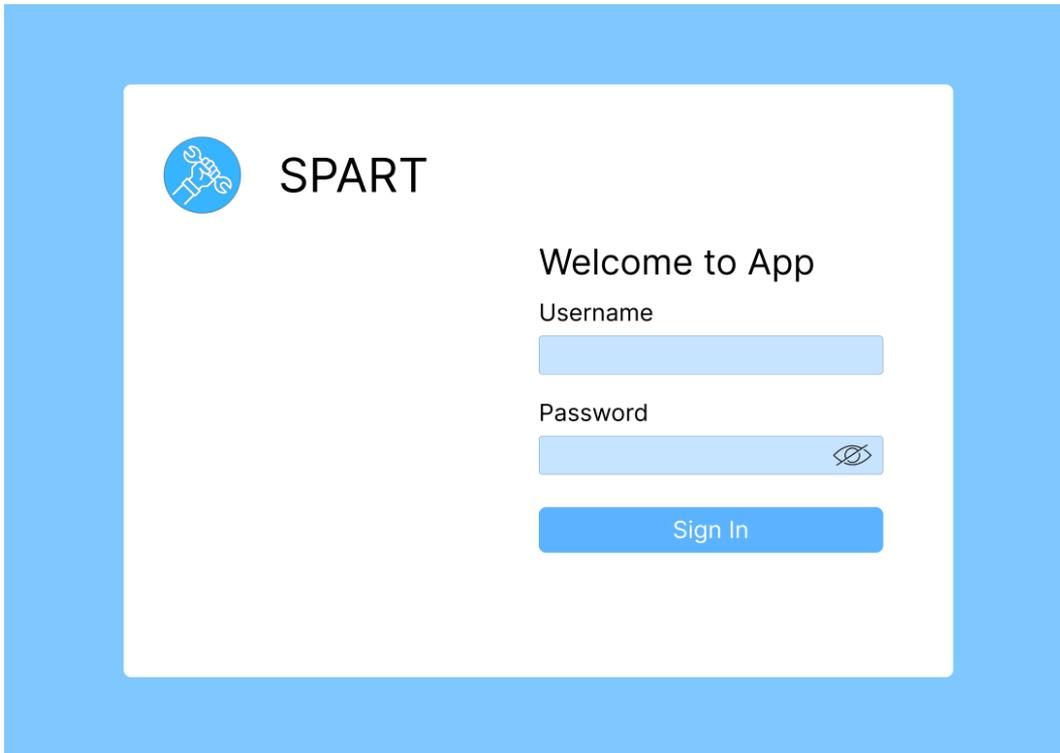
kemudahan navigasi, serta penggunaan ikon dan warna yang informatif. Hal ini bertujuan agar pengguna tidak mengalami kebingungan saat menggunakan sistem.

Rancangan antarmuka biasanya disajikan dalam bentuk mockup atau wireframe, yang merupakan gambaran visual dari tata letak elemen-elemen seperti tombol, form, tabel, dan menu. Selain itu, diberikan pula penjelasan mengenai fungsi dari masing-masing elemen, serta bagaimana interaksi pengguna terhadap elemen-elemen tersebut. Misalnya, tombol “Simpan” berfungsi untuk menyimpan data yang diinput, atau form “Cari Data” digunakan untuk melakukan pencarian berdasarkan kata kunci tertentu.

Dalam sistem berbasis web, desain UI juga mempertimbangkan aspek responsif, yaitu kemampuan antarmuka untuk menyesuaikan diri dengan berbagai ukuran layar perangkat, baik *desktop* maupun perangkat *mobile*. Hal memiliki tujuan yaitu untuk meningkatkan aksesibilitas dan kenyamanan pengguna dari berbagai *platform*.

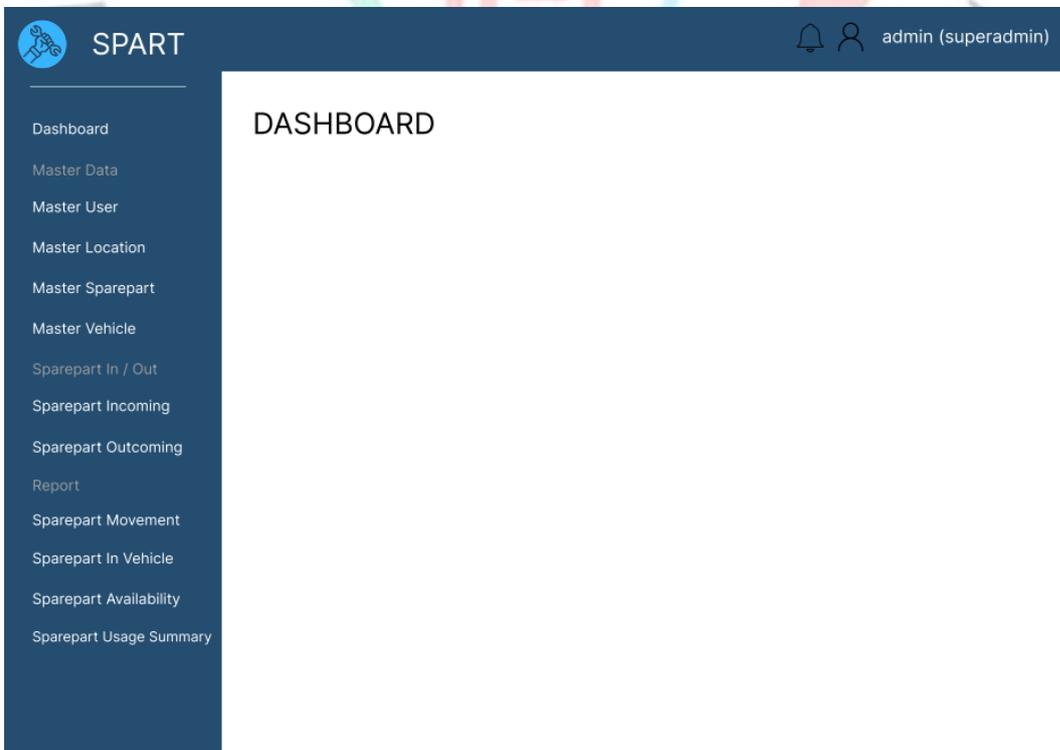
Bagian perancangan antarmuka pengguna ini menjadi dasar dalam proses implementasi sistem, karena semua komponen visual yang dirancang akan diwujudkan dalam bentuk kode program oleh tim pengembang. Oleh karena itu, rancangan UI yang dibuat dalam laporan harus jelas, lengkap, dan sesuai dengan kebutuhan pengguna yang diidentifikasi terlebih dahulu sebelumnya melalui proses analisis sistem.

1. Tampilan *Login*



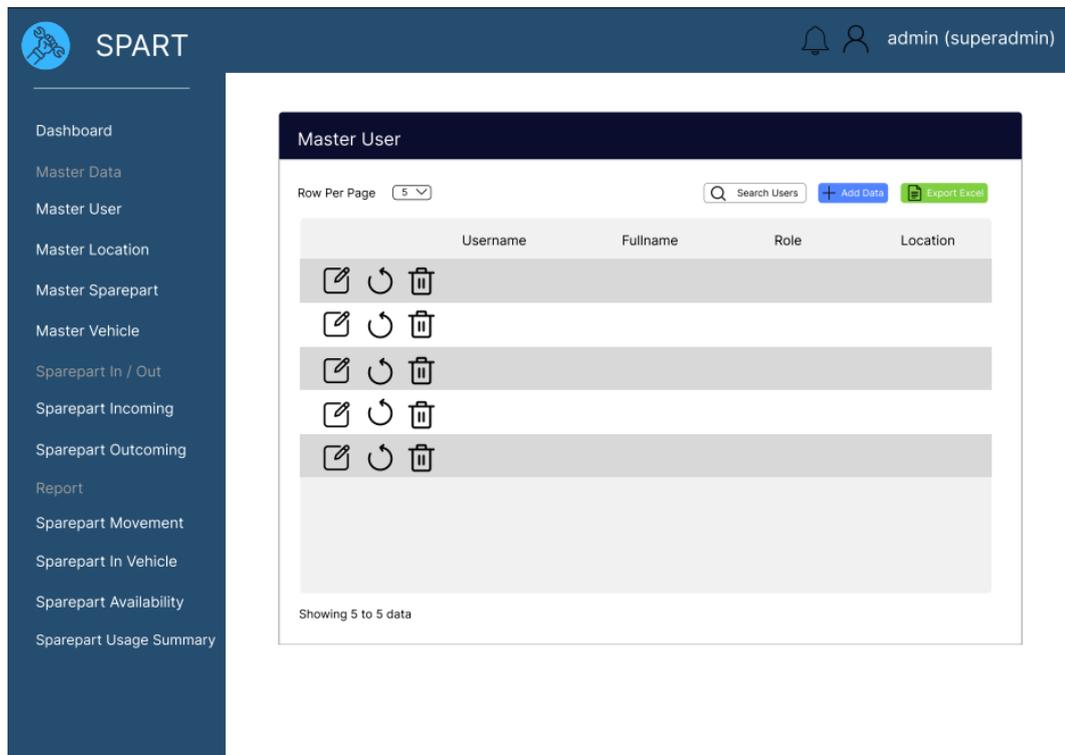
Gambar 4.23 Mockup Halaman *Login*

2. Tampilan *Dashboard*



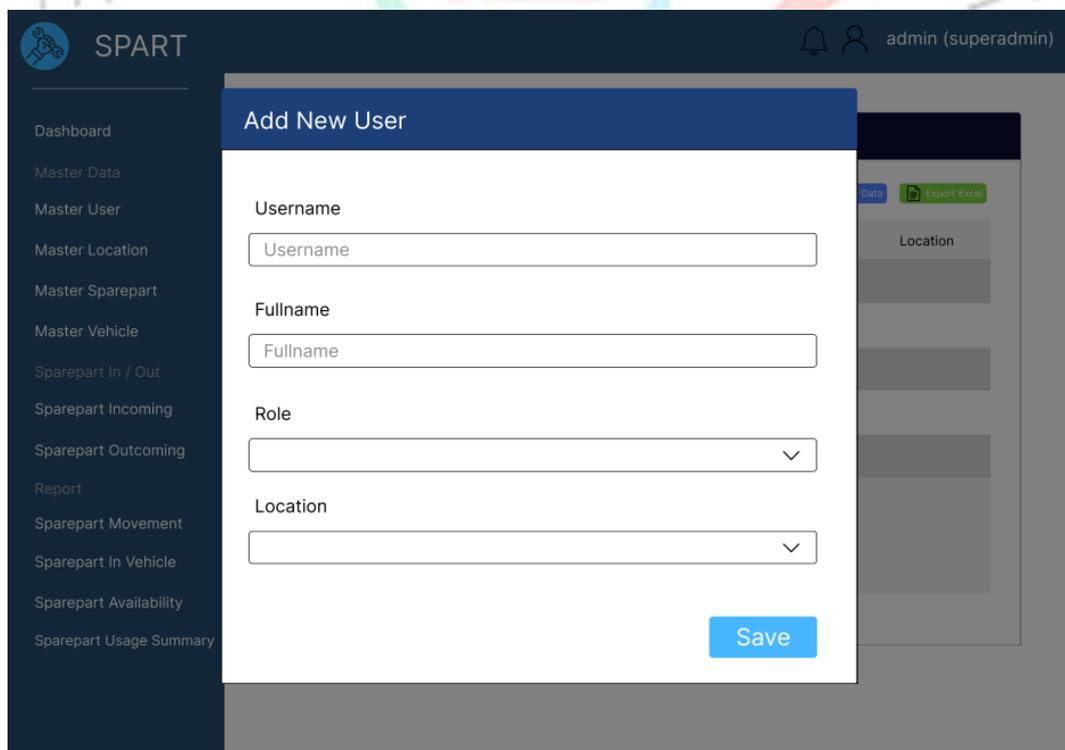
Gambar 4.24 Mockup Halaman *Dashboard*

3. Tampilan *Master Data User*



Gambar 4.25 Mockup Halaman *Master Data User*

4. Tampilan *Create Data User*



Gambar 4.26 Mockup Halaman *Add User*

5. Tampilan *Edit Data User*

The screenshot displays the SPART application interface. On the left is a dark sidebar menu with items like Dashboard, Master Data, Master User, Master Location, Master Sparepart, Master Vehicle, Sparepart In / Out, Sparepart Incoming, Sparepart Outcoming, Report, Sparepart Movement, Sparepart In Vehicle, Sparepart Availability, and Sparepart Usage Summary. The top right shows a user profile for 'admin (superadmin)'. A modal window titled 'Edit User Data' is open, containing the following fields: 'Username' (text input), 'Fullname' (text input), 'Role' (dropdown menu), and 'Location' (dropdown menu). A blue 'Save' button is positioned at the bottom right of the modal.

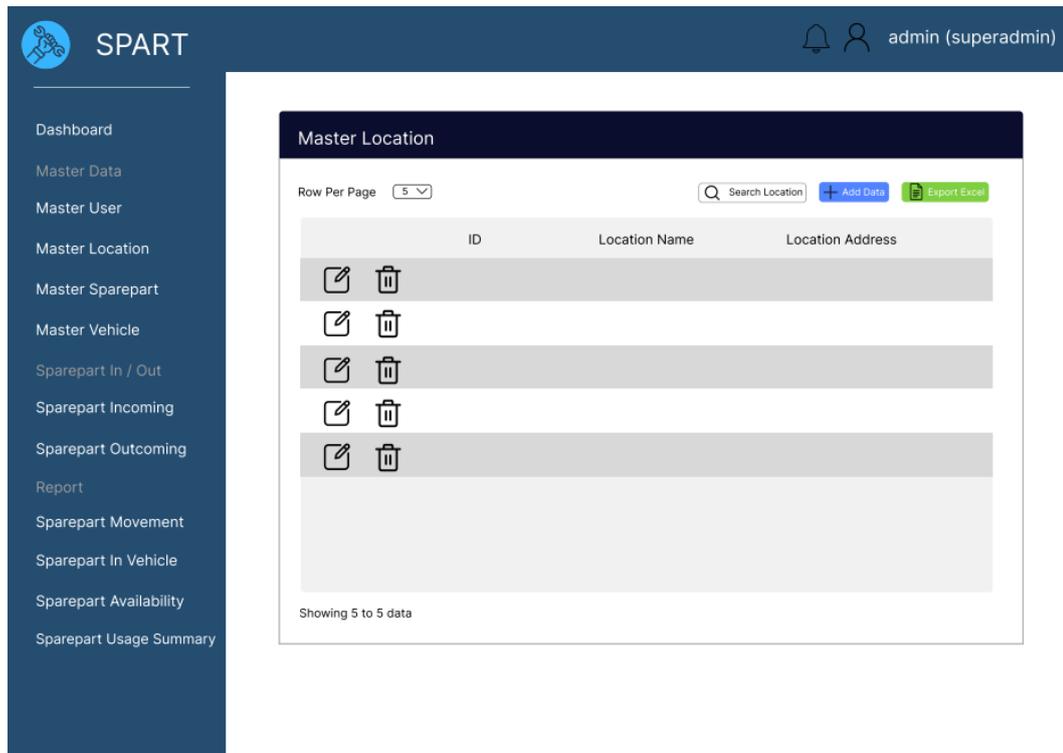
Gambar 4.27 Mockup Halaman *Edit User*

6. Tampilan *Delete Data User*

The screenshot shows the SPART application with the same sidebar and user profile as the previous image. A confirmation dialog box is centered on the screen. The dialog has the title 'Delete User Account' and the text 'Delete User Account Warehouse'. At the bottom of the dialog are two buttons: a blue 'Yes, Delete' button and a red 'Cancel' button. The background shows a blurred view of a data table with icons for edit, refresh, and delete, and a status 'Showing 5 to 5 data'.

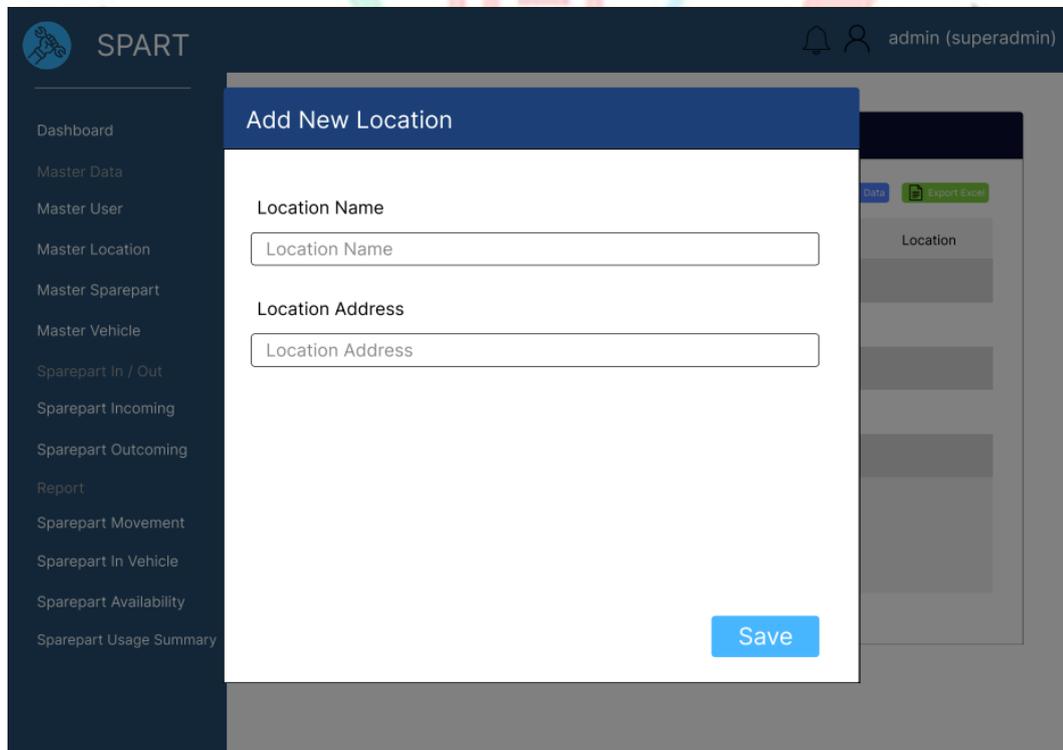
Gambar 4.28 Mockup Halaman *Delete User*

7. Tampilan *Master Data Location*



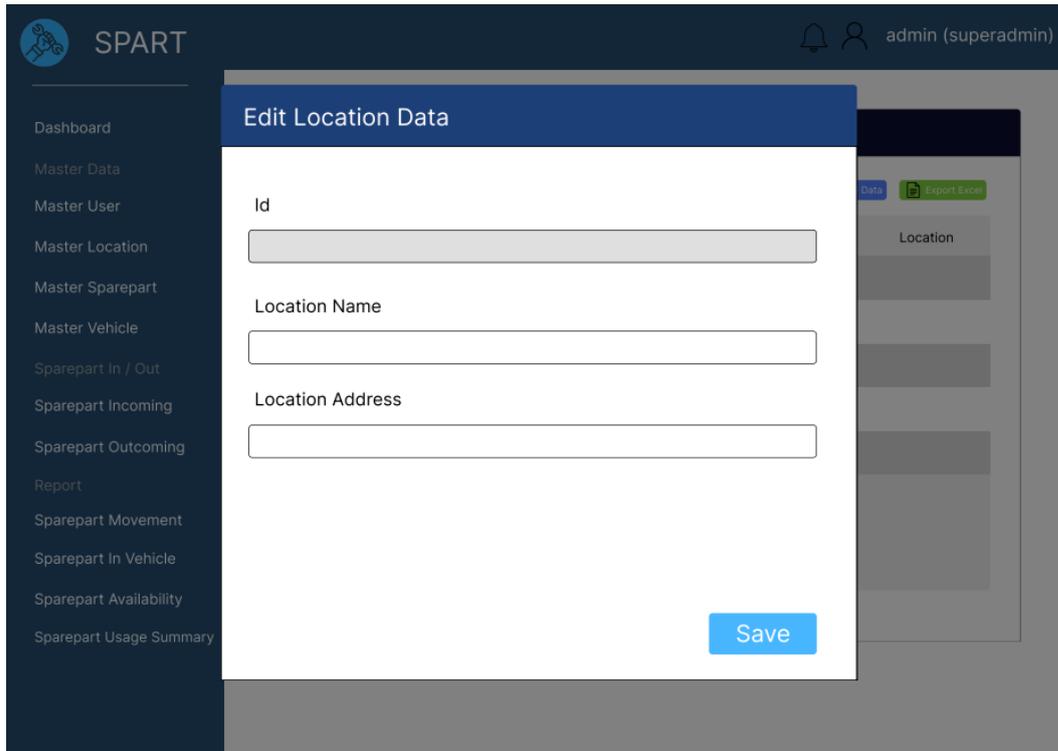
Gambar 4.29 Mockup Halaman *Master Data Location*

8. Tampilan *Create Data Location*



Gambar 4.30 Mockup Halaman *Create Location*

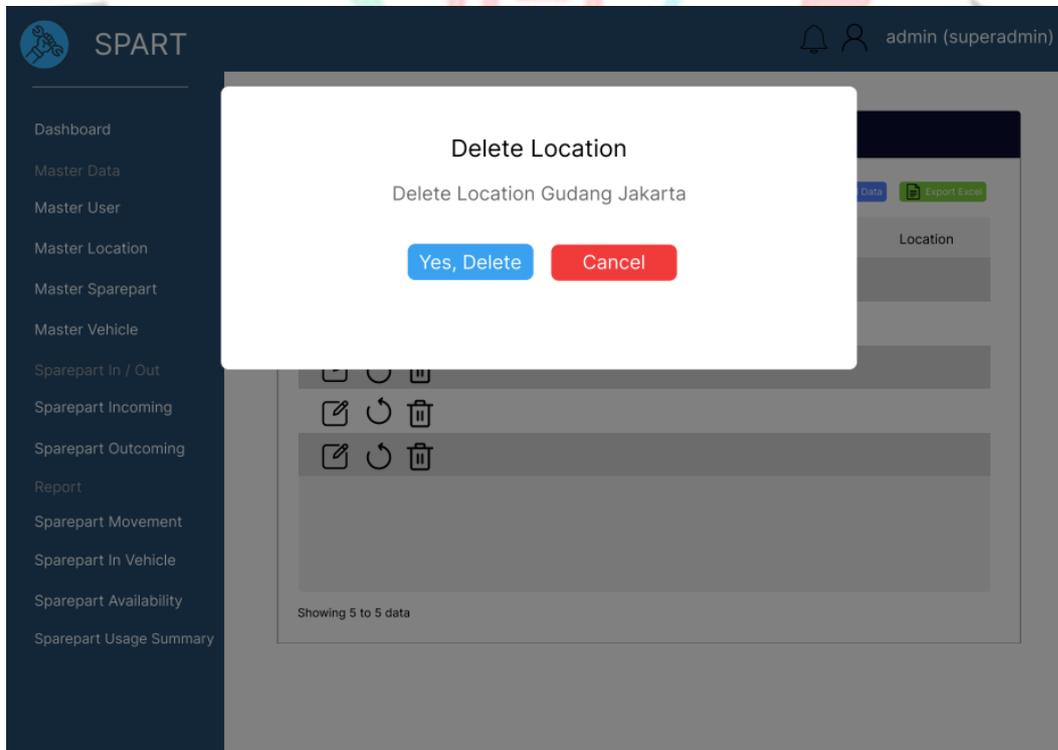
9. Tampilan *Edit Data Location*



The screenshot shows the SPART application interface. On the left is a dark sidebar with a menu including Dashboard, Master Data, Master User, Master Location, Master Sparepart, Master Vehicle, Sparepart In / Out, Sparepart Incoming, Sparepart Outcoming, Report, Sparepart Movement, Sparepart In Vehicle, Sparepart Availability, and Sparepart Usage Summary. The top right shows a user profile for 'admin (superadmin)'. A modal window titled 'Edit Location Data' is open, containing three text input fields labeled 'Id', 'Location Name', and 'Location Address', and a blue 'Save' button at the bottom right.

Gambar 4.31 Mockup Halaman *Edit Location*

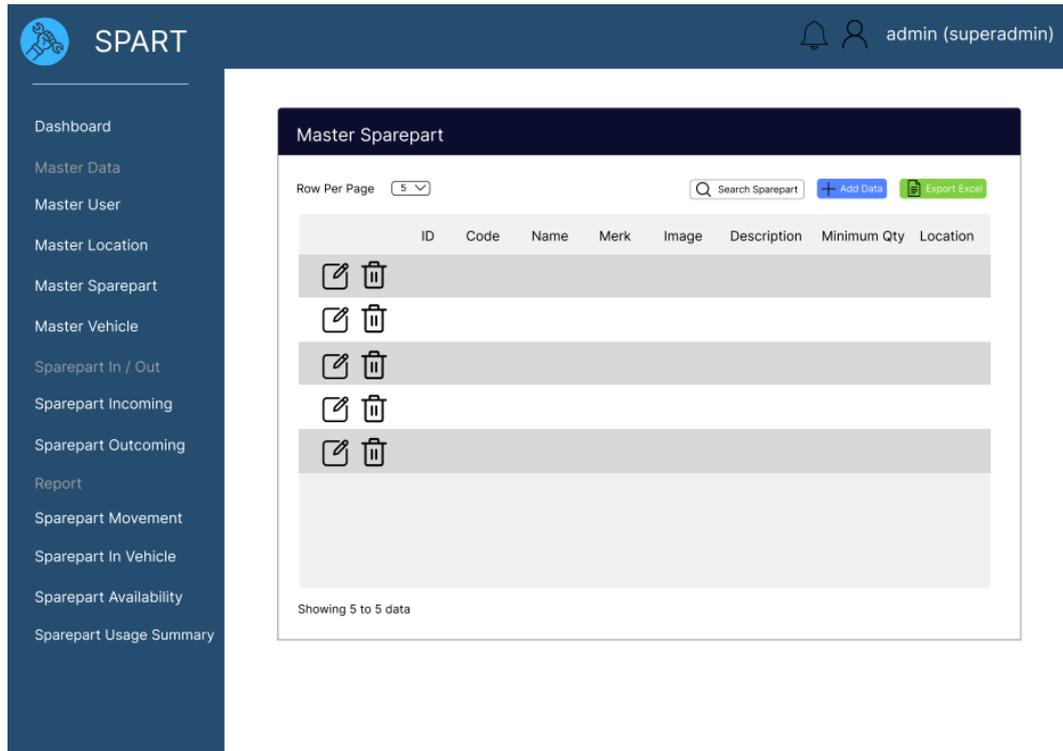
10. Tampilan Detele Data *Location*



The screenshot shows the SPART application interface with a confirmation dialog box. The dialog is titled 'Delete Location' and contains the text 'Delete Location Gudang Jakarta'. At the bottom of the dialog are two buttons: a blue 'Yes, Delete' button and a red 'Cancel' button. The background shows a table with location data and icons for edit, refresh, and delete.

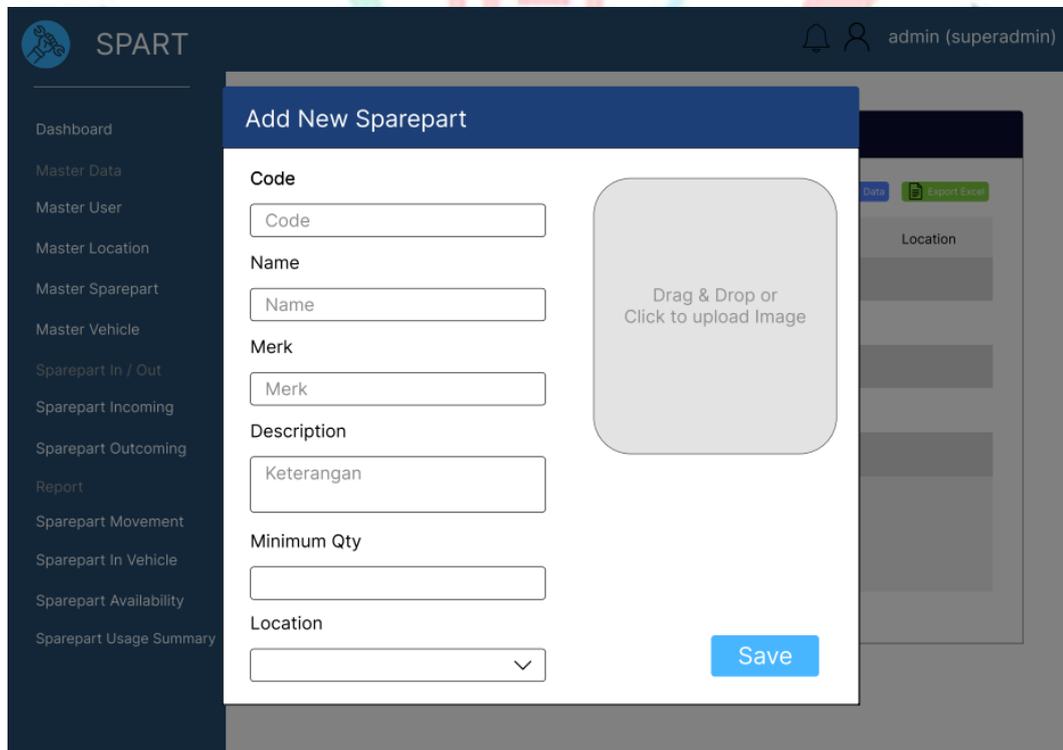
Gambar 4.32 Mockup Halaman *Delete Location*

11. Tampilan *Master Data Sparepart*



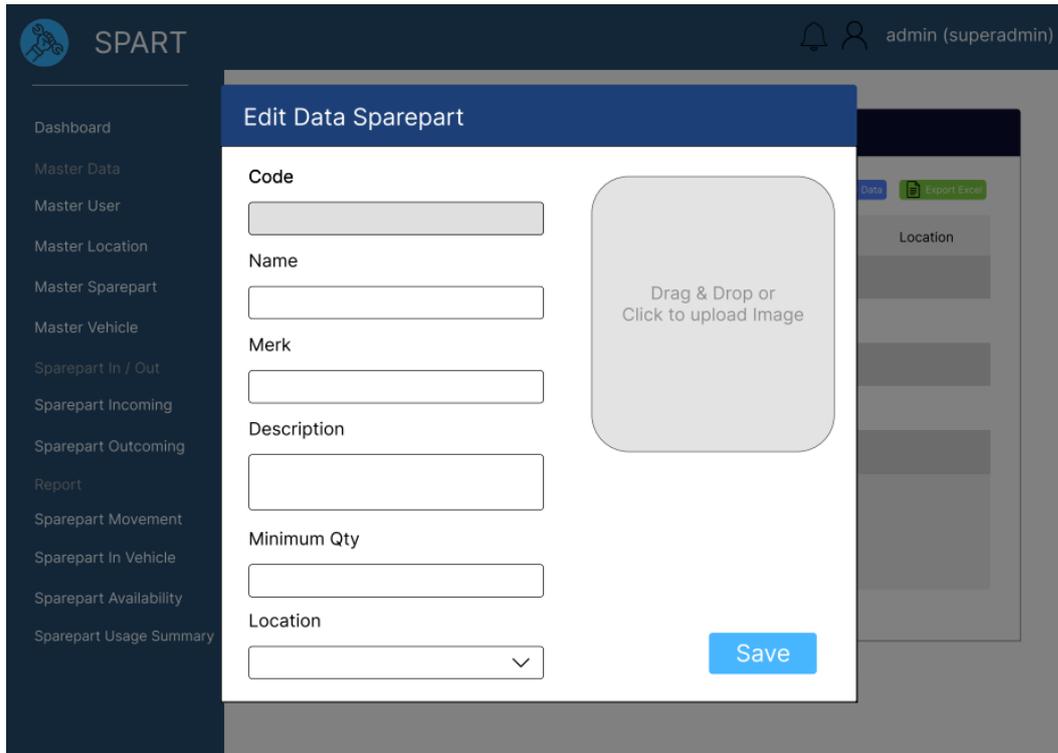
Gambar 4.33 Mockup Halaman *Master Data Sparepart*

12. Tampilan *Create Data Sparepart*



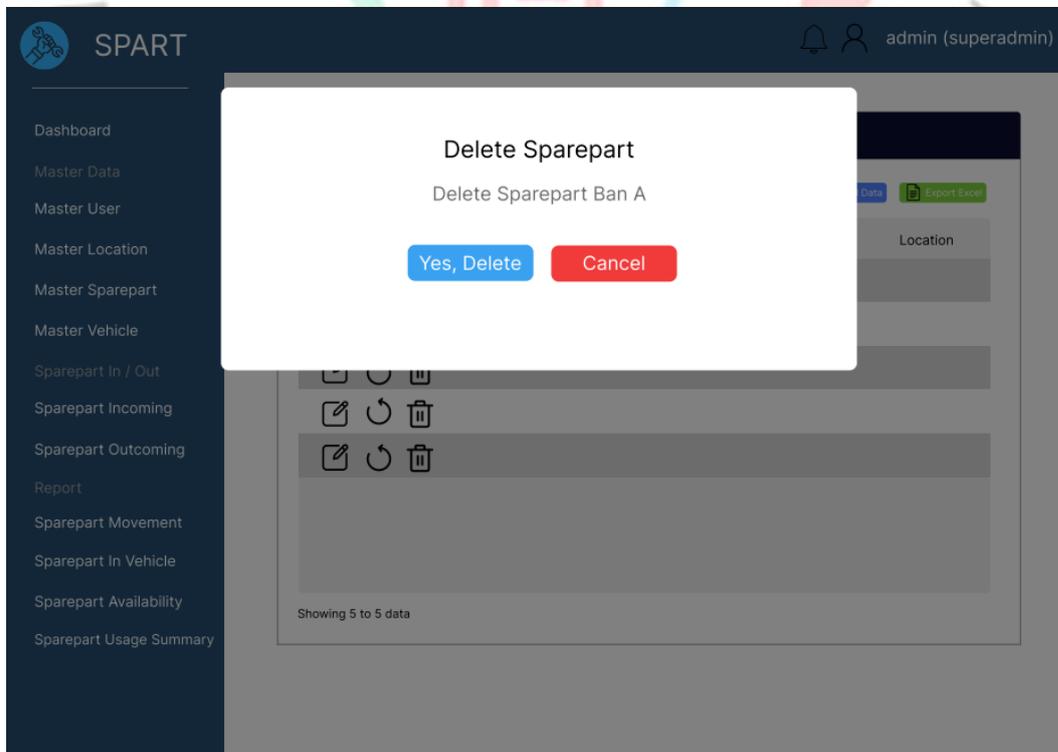
Gambar 4.34 Mockup Halaman *Create Sparepart*

13. Tampilan *Edit Data Sparepart*



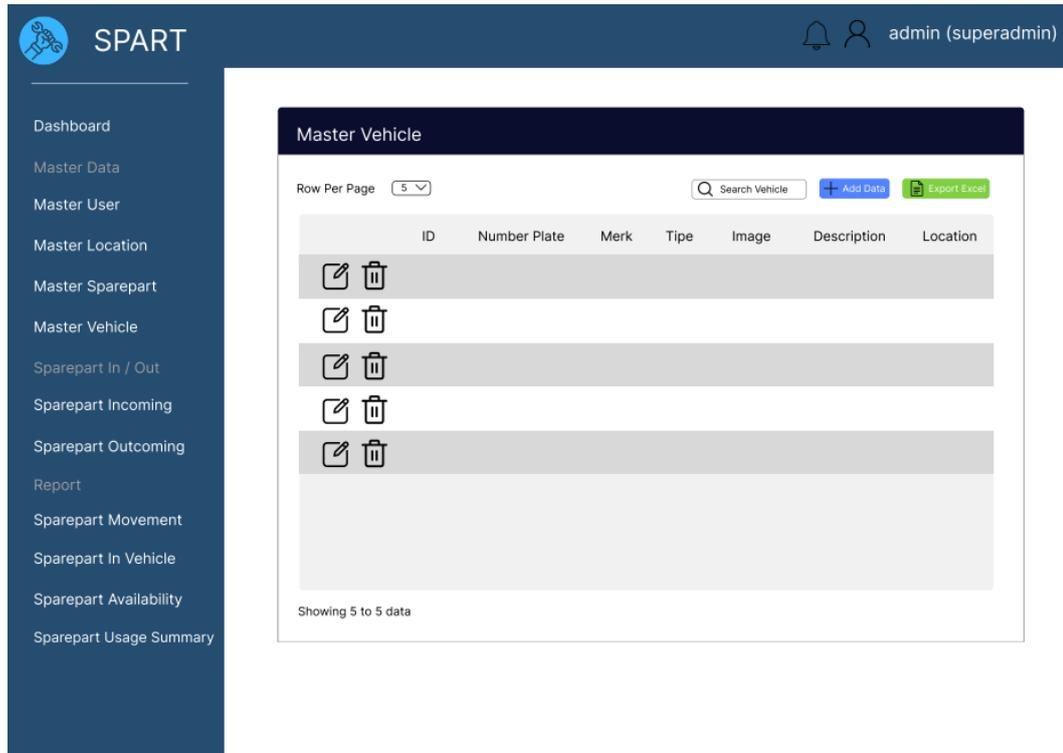
Gambar 4.35 Mockup Halaman *Edit Sparepart*

14. Tampilan *Delete Data Sparepart*



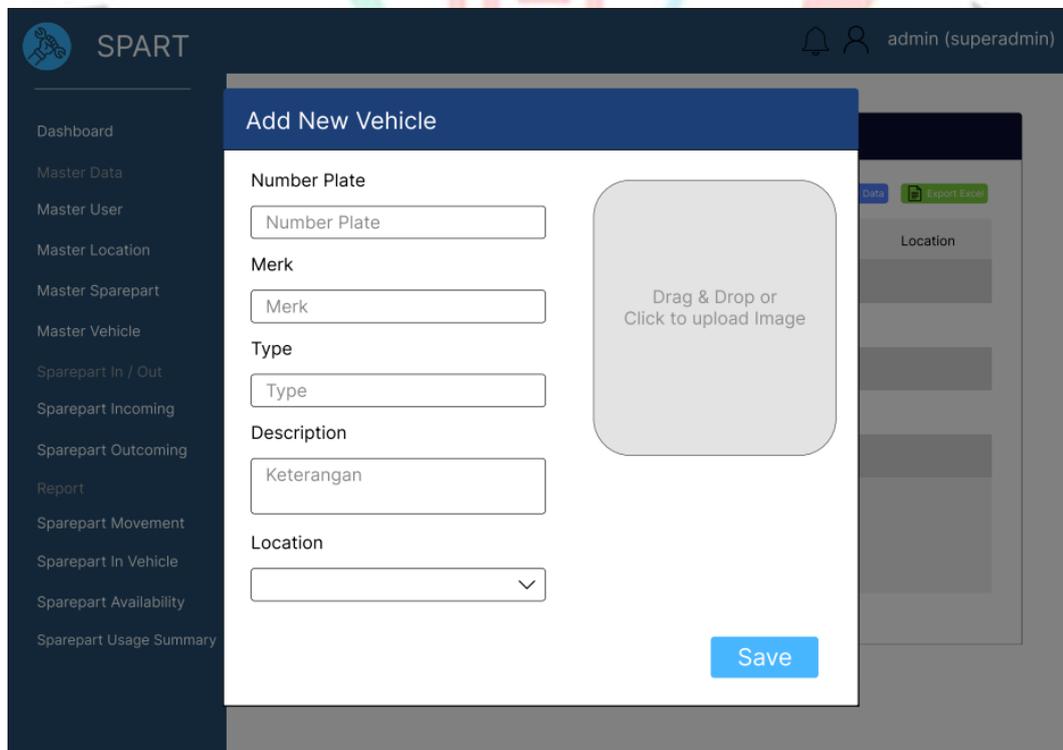
Gambar 4.36 Mockup Halaman *Delete Sparepart*

15. Tampilan *Master Data Vehicle*



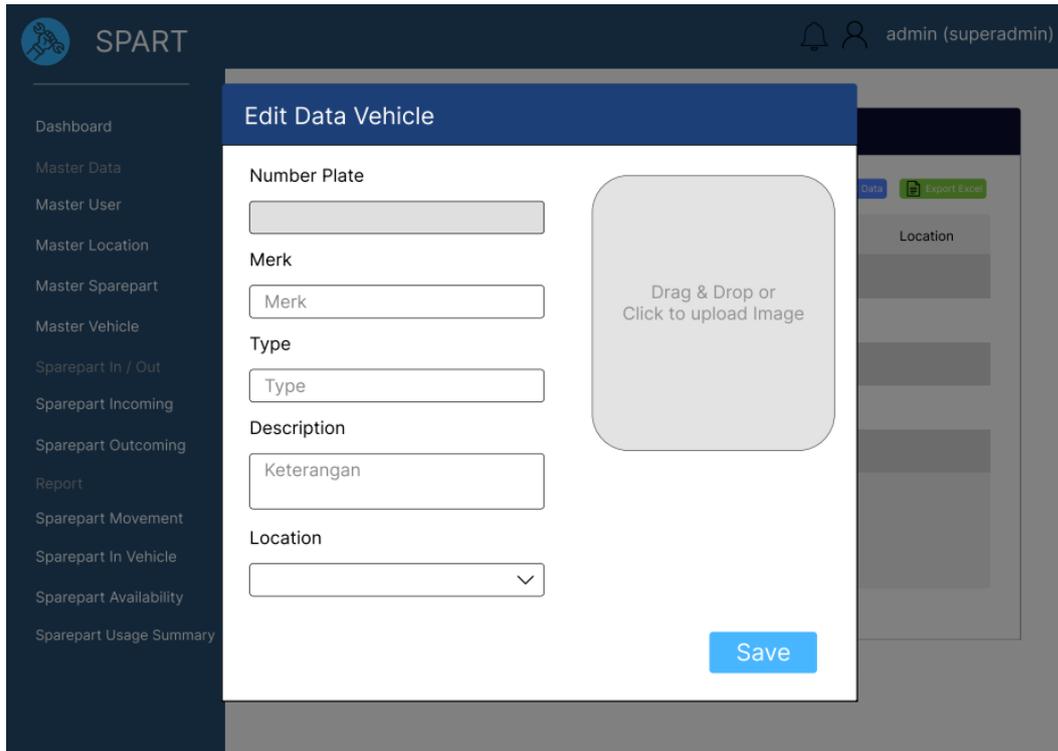
Gambar 4.37 Mockup Halaman *Master Data Vehicle*

16. Tampilan *Create Data Vehicle*



Gambar 4.38 Mockup Halaman *Create Vehicle*

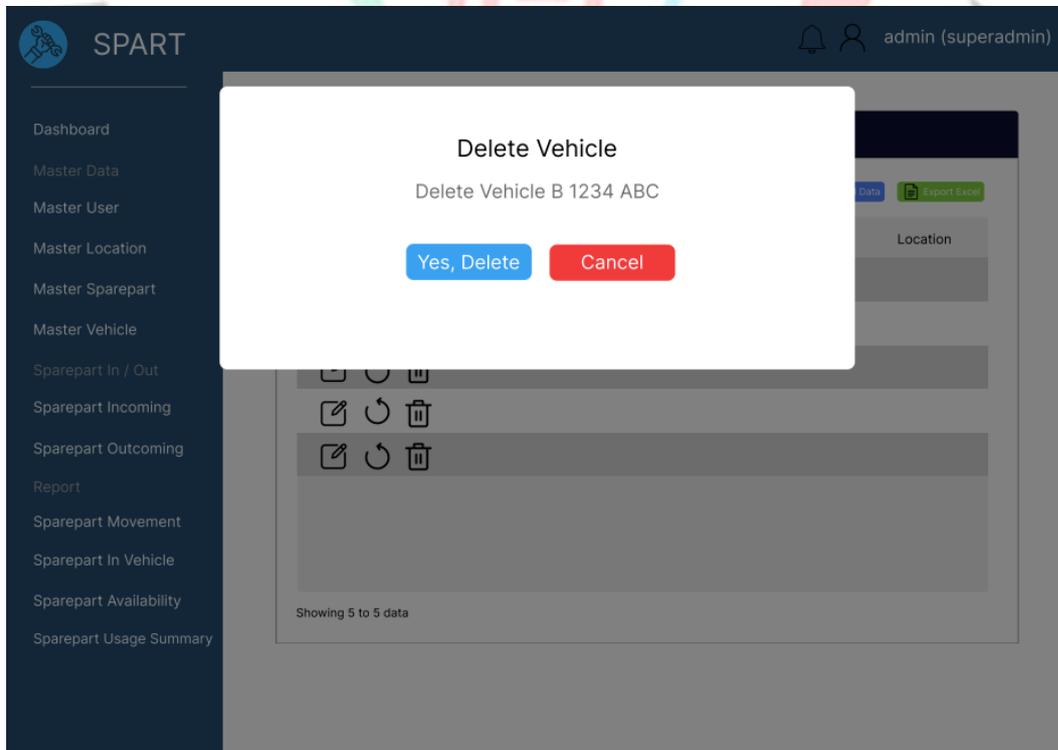
17. Tampilan *Edit Data Vehicle*



The screenshot shows the 'Edit Data Vehicle' form in the SPART system. The form is titled 'Edit Data Vehicle' and is located in the center of the screen. It contains several input fields: 'Number Plate' (empty), 'Merk' (containing 'Merk'), 'Type' (containing 'Type'), 'Description' (containing 'Keterangan'), and 'Location' (a dropdown menu). To the right of the form is a large grey box with the text 'Drag & Drop or Click to upload Image'. At the bottom right of the form is a blue 'Save' button. The background shows a sidebar with navigation options and a user profile 'admin (superadmin)' in the top right corner.

Gambar 4.39 Mockup Halaman *Edit Vehicle*

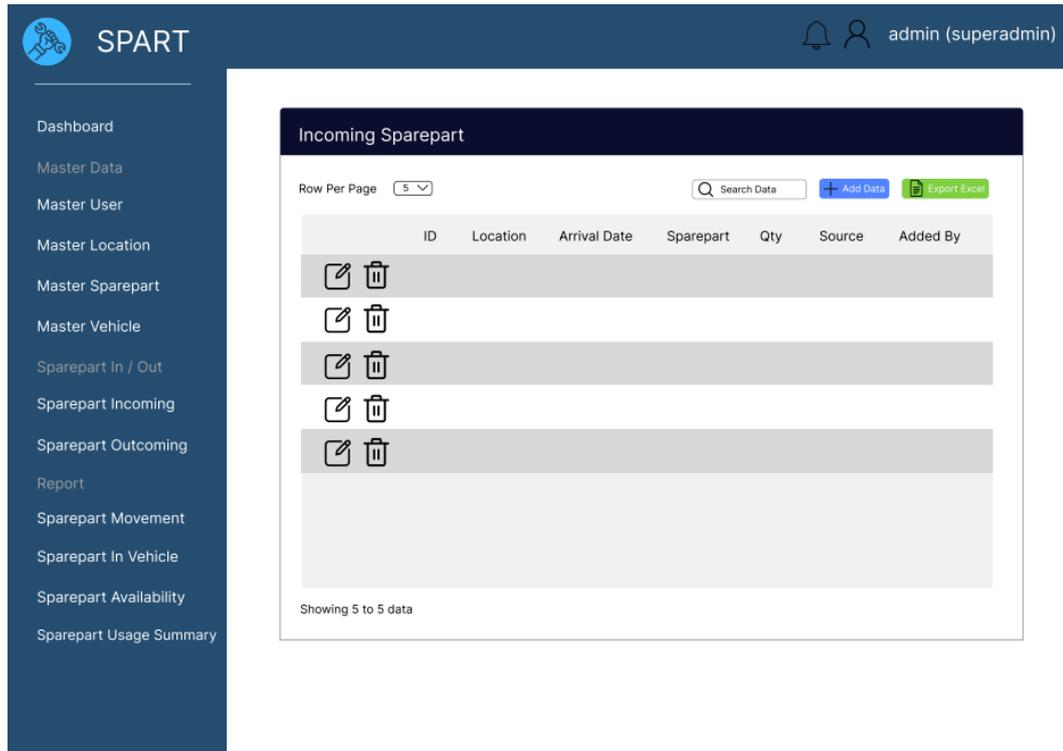
18. Tampilan *Delete Data Vehicle*



The screenshot shows a confirmation dialog box titled 'Delete Vehicle' in the SPART system. The dialog box is centered on the screen and contains the text 'Delete Vehicle B 1234 ABC'. Below the text are two buttons: a blue 'Yes, Delete' button and a red 'Cancel' button. The background shows the same sidebar and user profile as in the previous screenshot, but the main content area is dimmed.

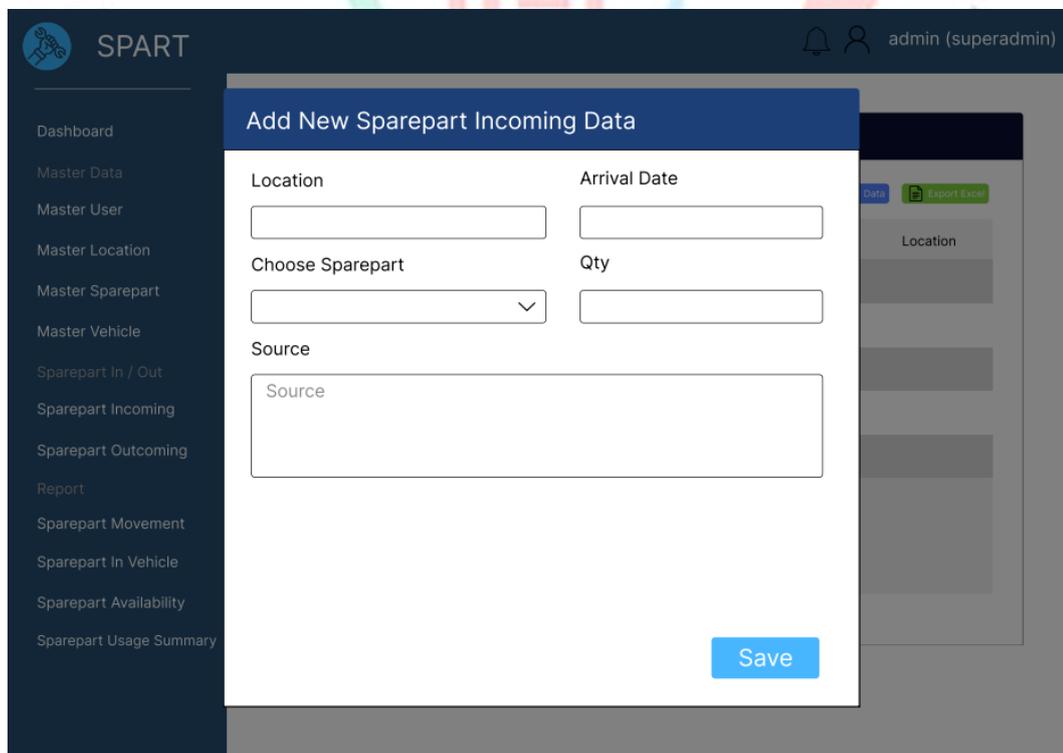
Gambar 4.40 Mockup Halaman *Delete Vehicle*

19. Tampilan *Sparepart Incoming*



Gambar 4.41 Mockup Halaman *Incoming Sparepart*

20. Tampilan *Create Data Sparepart Incoming*



Gambar 4.42 Mockup Halaman *Create Incoming Sparepart*

21. Tampilan *Edit Data Sparepart Incoming*

The screenshot displays the SPART application interface. On the left is a dark sidebar menu with options: Dashboard, Master Data, Master User, Master Location, Master Sparepart, Master Vehicle, Sparepart In / Out, Sparepart Incoming, Sparepart Outcoming, Report, Sparepart Movement, Sparepart In Vehicle, Sparepart Availability, and Sparepart Usage Summary. The top right shows a user profile for 'admin (superadmin)'. A modal window titled 'Edit Sparepart Incoming Data' is open, containing the following fields: ID (text input), Location (text input), Arrival Date (text input), Choose Sparepart (dropdown menu), Qty (text input), and Source (text area). A blue 'Save' button is located at the bottom right of the modal.

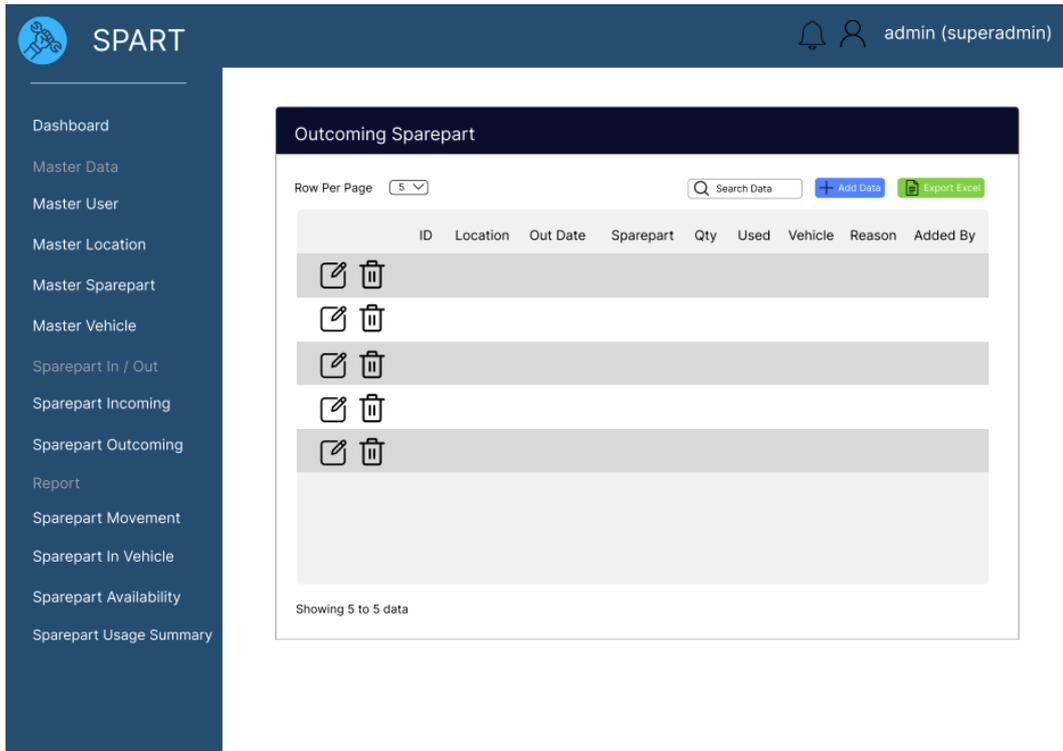
Gambar 4.43 Mockup Halaman *Edit Incoming Sparepart*

22. Tampilan *Delete Data Sparepart Incoming*

The screenshot displays the SPART application interface. On the left is a dark sidebar menu with options: Dashboard, Master Data, Master User, Master Location, Master Sparepart, Master Vehicle, Sparepart In / Out, Sparepart Incoming, Sparepart Outcoming, Report, Sparepart Movement, Sparepart In Vehicle, Sparepart Availability, and Sparepart Usage Summary. The top right shows a user profile for 'admin (superadmin)'. A modal window titled 'Delete Sparepart Incoming Data' is open, displaying the text 'Delete Sparepart Incoming Data Id 1'. Below the text are two buttons: a blue 'Yes, Delete' button and a red 'Cancel' button. The background shows a table with icons for edit, refresh, and delete, and a 'Showing 5 to 5 data' indicator.

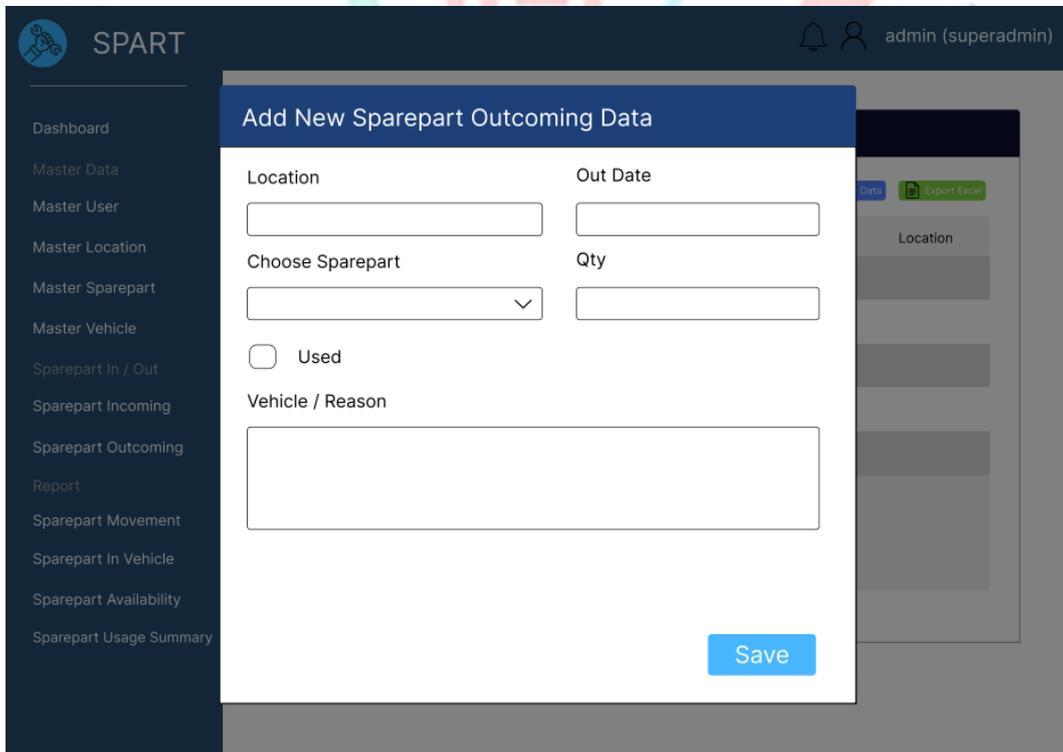
Gambar 4.44 Mockup Halaman *Delete Incoming Sparepart*

23. Tampilan *Sparepart Outcoming*



Gambar 4.45 Mockup Halaman *Outcoming Sparepart*

24. Tampilan *Create Data Sparepart Outcoming*



Gambar 4.46 Mockup Halaman *Create Outcoming Sparepart*

25. Tampilan *Edit Data Sparepart Outcoming*

The screenshot displays the SPART application interface. On the left is a dark sidebar menu with items: Dashboard, Master Data, Master User, Master Location, Master Sparepart, Master Vehicle, Sparepart In / Out, Sparepart Incoming, Sparepart Outcoming, Report, Sparepart Movement, Sparepart In Vehicle, Sparepart Availability, and Sparepart Usage Summary. The top right shows a user profile for 'admin (superadmin)'. A modal window titled 'Edit Sparepart Outcoming Data' is open, containing the following fields:

- ID:
- Location:
- Out Date:
- Choose Sparepart:
- Qty:
- Used
- Vehicle / Reason:

A blue 'Save' button is located at the bottom right of the modal.

Gambar 4.47 Mockup Halaman *Edit Outcoming Sparepart*

26. Tampilan *Delete Data Sparepart Outcoming*

The screenshot shows the SPART application with a confirmation dialog box titled 'Delete Sparepart Outcoming Data'. The dialog contains the text 'Delete Sparepart Outcoming Data Id 1' and two buttons: 'Yes, Delete' (blue) and 'Cancel' (red). In the background, a table of sparepart outcoming data is visible, showing columns for ID, Location, and Out Date. The table has two rows of data, each with edit, refresh, and delete icons. At the bottom of the table, it says 'Showing 5 to 5 data'.

Gambar 4.48 Mockup Halaman *Delete Outcoming Sparepart*

27. Tampilan *Report Sparepart Movement*

The screenshot shows the 'Report Sparepart Movement' interface. At the top left is the SPART logo. The top right shows a user profile for 'admin (superadmin)'. A sidebar menu on the left lists various system components. The main content area features a header 'Report Sparepart Movement' and a filter section with dropdowns for 'Choose Location', 'Choose Sparepart', and input fields for 'Start Date' and 'End Date'. A 'Load Report' button is present. Below the filters, there is a 'Row Per Page' dropdown set to 5, a search bar, and an 'Export Excel' button. The table below has the following structure:

Location	Sparepart Name	Action	Note	Qty	Date

Showing 5 to 5 data

Gambar 4.49 Mockup Halaman *Report Sparepart Movement*

28. Tampilan *Report Sparepart In Vehicle*

The screenshot shows the 'Report Sparepart In Vehicle' interface. At the top left is the SPART logo. The top right shows a user profile for 'admin (superadmin)'. A sidebar menu on the left lists various system components. The main content area features a header 'Report Sparepart In Vehicle' and a filter section with dropdowns for 'Choose Location', 'Choose Sparepart', and 'Choose Vehicle'. A 'Load Report' button is present. Below the filters, there is a 'Row Per Page' dropdown set to 5, a search bar, and an 'Export Excel' button. The table below has the following structure:

Location	Sparepart Name	Number Plate	Qty

Showing 5 to 5 data

Gambar 4.50 Mockup Halaman *Report Sparepart In Vehicle*

29. Tampilan *Report Availability*

The screenshot displays the 'Report Sparepart Availability' interface. On the left is a dark blue sidebar with a menu containing: Dashboard, Master Data, Master User, Master Location, Master Sparepart, Master Vehicle, Sparepart In / Out, Sparepart Incoming, Sparepart Outcoming, Report, Sparepart Movement, Sparepart In Vehicle, Sparepart Availability, and Sparepart Usage Summary. The top header features the SPART logo, the text 'SPART', and a user profile icon with the name 'admin (superadmin)'. The main content area is titled 'Report Sparepart Availability' and contains a form with 'Choose Location' and 'Choose Sparepart' dropdown menus, a 'Load Report' button, a 'Row Per Page' dropdown set to '5', a 'Search Data' input field, and an 'Export Excel' button. Below the form is a table with the following columns: Location, Sparepart Name, Minimum Qty, In, Used, Out, and Available. The table body is currently empty, and a status message at the bottom indicates 'Showing 5 to 5 data'.

Gambar 4.51 Mockup Halaman *Report Availability*

30. Tampilan *Report Sparepart Usage Summary*

The screenshot displays the 'Report Sparepart Usage Summary' interface. It features the same sidebar and header as the previous page. The main content area is titled 'Report Sparepart Usage Summary' and includes a form with 'Choose Location', 'Start Date', and 'End Date' input fields, a 'Load Report' button, a 'Row Per Page' dropdown set to '5', a 'Search Data' input field, and an 'Export Excel' button. Below the form is a table with columns: Location, Sparepart Name, and Qty. The table body is empty, and a status message at the bottom indicates 'Showing 5 to 5 data'.

Gambar 4.52 Mockup Halaman *Report Sparepart Usage Summary*

4.4 Perancangan Implementasi

4.4.1 Tahapan Pengembangan

Tahap implementasi merupakan bagian penting dari proses pengembangan sistem, di mana seluruh hasil rancangan dan spesifikasi teknis yang telah dirancang pada tahapan-tahapan sebelumnya diwujudkan menjadi bentuk program aplikasi yang nyata juga dapat digunakan. Dalam konteks pengembangan Aplikasi *Inventory Sparepart*, tahap implementasi ini mencakup proses pembuatan kode program, pengembangan basis data, serta penyusunan antarmuka pengguna (*user interface*) sesuai dengan perancangan yang telah disetujui.

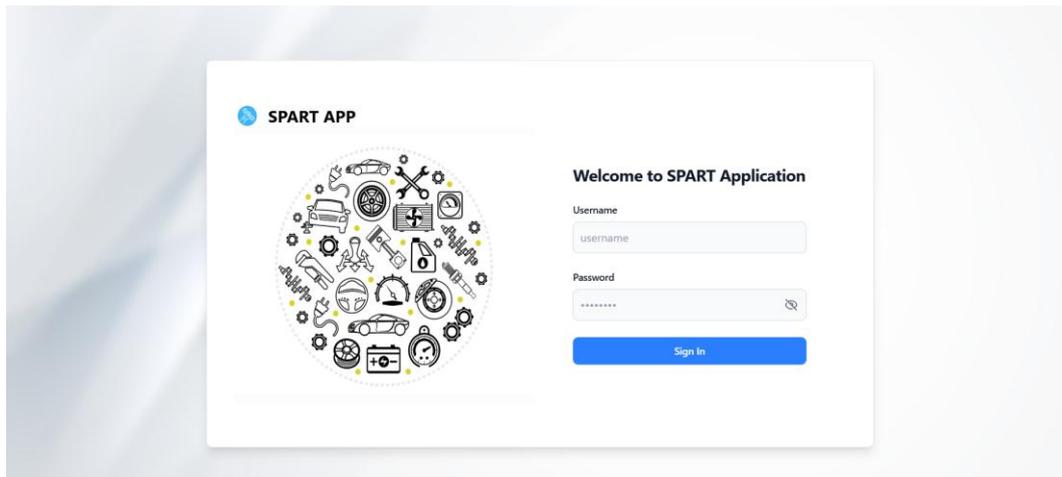
Implementasi dilakukan dengan mengacu pada metode *Waterfall* yang bersifat sistematis dan bertahap. Artinya, implementasi baru dimulai setelah proses analisis dan perancangan selesai dengan baik. Tujuan utama dari metodologi tersebut untuk memastikan bahwa proses pembangunan sistem diterapkan secara terstruktur dan memenuhi kebutuhan fungsional yang telah dirumuskan sebelumnya.

Tahap implementasi pada aplikasi *inventory sparepart* ini menjadi kunci utama dalam mewujudkan rancangan sistem menjadi sebuah produk *software* yang siap digunakan. Dengan mengikuti tahapan yang sistematis dalam model *Waterfall*, proses implementasi berjalan sesuai urutan logis dan menghasilkan aplikasi yang stabil, mudah digunakan, dan sesuai kebutuhan pengguna. Implementasi ini memastikan bahwa sistem dapat dijalankan di lingkungan nyata dan siap dilanjutkan ke tahap pengujian formal. Hasil dari pengembangan sistem ini sebagai berikut:

1. Halaman *Login*

Halaman *login* pada aplikasi SPART merupakan pintu masuk utama yang berfungsi untuk melakukan proses otentikasi pengguna sebelum mengakses sistem. Desain antarmuka halaman ini dibuat sederhana dan *user-friendly* dengan menampilkan dua kolom utama, yaitu ilustrasi bertema otomotif di sisi kiri dan form *login* di sisi kanan. Form *login* terdiri dari input *username*, input *password* dengan fitur *show/hide*, dan tombol *Sign In* untuk masuk ke dalam aplikasi.

Halaman ini berfungsi memverifikasi kredensial pengguna melalui data yang tersimpan dalam basis data. Jika data cocok, pengguna diarahkan ke halaman *dashboard* sesuai hak aksesnya. Desain visual yang digunakan juga mendukung kesan profesional dan mencerminkan tema aplikasi *inventory sparepart*, sehingga memberikan pengalaman pertama yang baik bagi pengguna.

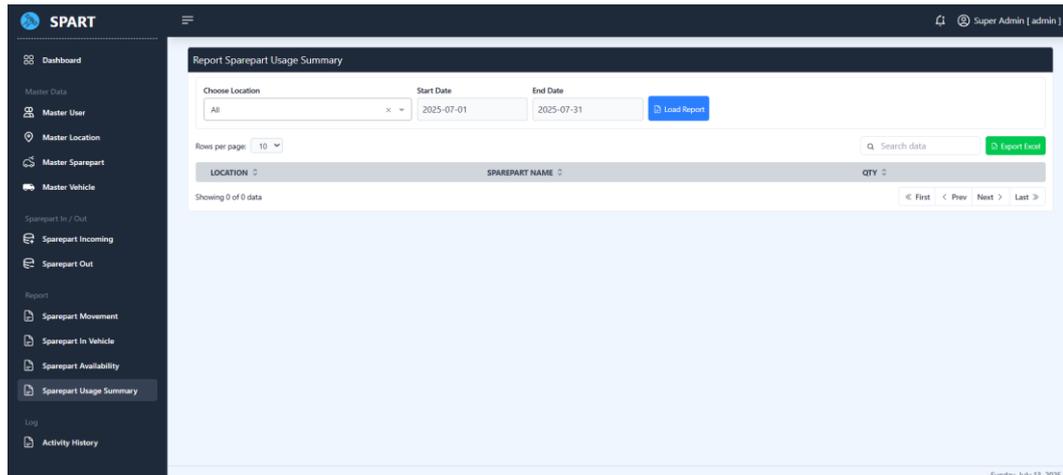


Gambar 4.53 Halaman *Login*

2. Halaman *Dashboard*

Halaman *dashboard* pada aplikasi SPART berfungsi sebagai beranda utama. Tampilan *dashboard* menyediakan sambutan kepada pengguna sesuai peran yang dimiliki, dalam hal ini sebagai Super Admin. Informasi pada halaman ini menjelaskan bahwa pengguna telah berhasil masuk dan dapat mulai menggunakan fitur-fitur utama seperti monitoring stok, pencatatan *sparepart* masuk dan keluar, serta pengecekan ketersediaan barang berdasarkan lokasi.

Selain itu, halaman ini juga menampilkan menu navigasi di sisi kiri layar yang terstruktur dalam beberapa kategori seperti *Master Data*, *Sparepart In/Out*, *Report*, dan *Log*. Navigasi ini memudahkan pengguna dalam mengakses fitur-fitur yang tersedia tanpa harus kembali ke halaman utama. Desain antarmuka dibuat responsif dan mudah dipahami, sehingga mendukung efisiensi kerja pengguna dalam mengelola data *inventory* secara real time.

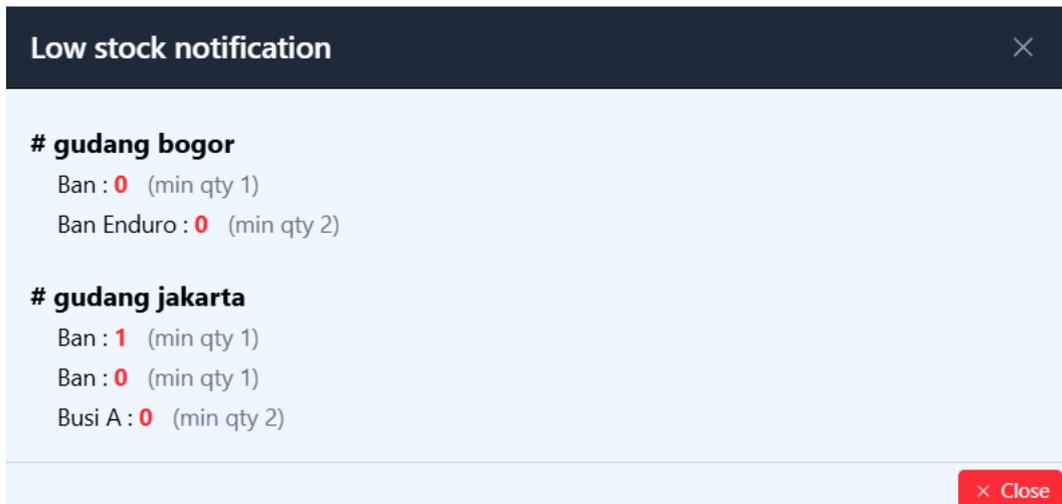


Gambar 4.54 Halaman *Dashboard*

3. *Low Stock Notification*

Halaman notifikasi “*Low Stock Notification*” pada aplikasi SPART berfungsi untuk memberikan informasi secara langsung kepada pengguna terkait stok barang yang berada di bawah jumlah minimum yang telah ditentukan di masing-masing gudang. Notifikasi ini muncul secara otomatis saat pengguna *login* ke dalam aplikasi atau ketika ikon lonceng di bagian kanan atas navbar ditekan. Dengan tampilan yang ringkas dan jelas, pengguna dapat segera mengetahui barang mana yang perlu segera dilakukan pengadaan ulang.

Pada gambar ditampilkan dua lokasi gudang, yaitu gudang bogor dan gudang jakarta, lengkap dengan daftar item dan jumlah stok terkini yang ditandai berwarna merah apabila berada di bawah batas minimum (min qty). Misalnya, “Ban Enduro” di gudang bogor memiliki stok 0 dari minimum 2, dan “Busi A” di gudang jakarta juga menunjukkan kondisi serupa. Fitur ini sangat membantu dalam pengambilan keputusan cepat agar operasional tidak terganggu akibat kekurangan stok barang penting.

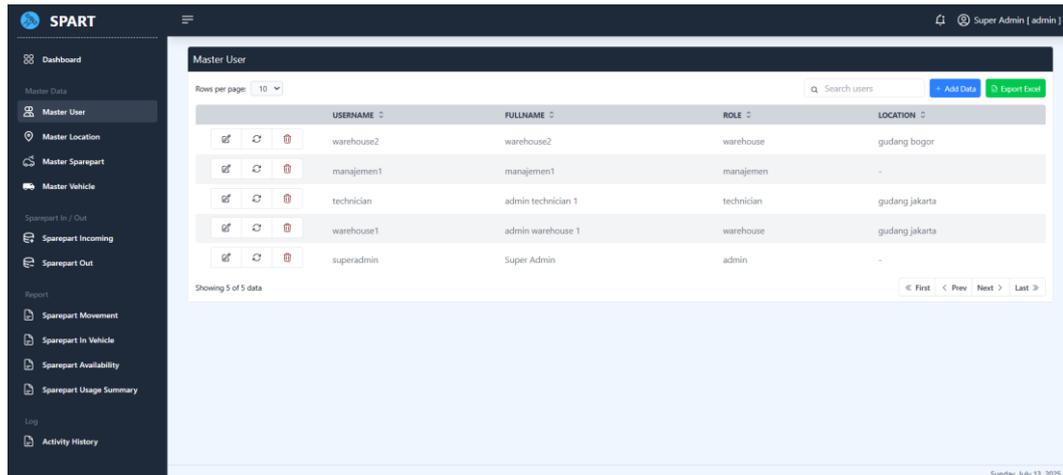


Gambar 4.55 Halaman *Low Stock Notification*

4. Halaman *Master Data User*

Halaman “*Master User*” pada aplikasi SPART digunakan untuk mengelola data pengguna yang memiliki akses ke dalam sistem. Di sini, admin dapat melihat daftar seluruh pengguna lengkap dengan informasi seperti *username*, nama lengkap (*fullname*), peran atau hak akses (*role*), serta lokasi gudang yang terkait. Halaman ini juga menyediakan fitur pencarian pengguna, serta tombol untuk menambahkan data baru (*Add Data*) dan ekspor data ke file *Excel* (*Export Excel*), yang sangat membantu untuk keperluan dokumentasi atau analisis.

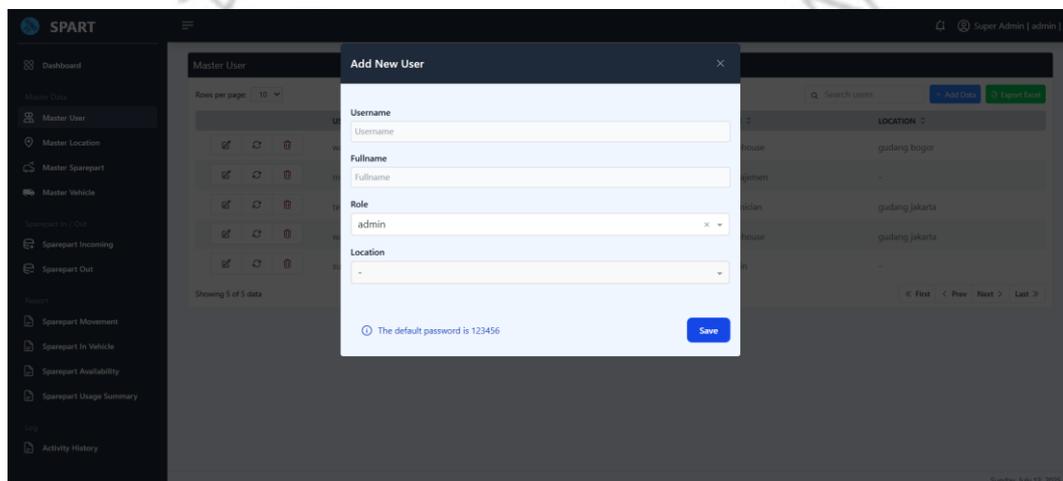
Setiap entri pengguna dilengkapi dengan tombol aksi yang memungkinkan admin melakukan *edit* data, reset *password*, atau menghapus akun pengguna dengan mudah. Peran yang tercantum seperti *admin*, *warehouse*, *technician*, dan manajemen menandakan hak akses yang berbeda sesuai dengan fungsi jabatan masing-masing. Fitur ini penting untuk memastikan bahwa setiap pengguna hanya memiliki akses terhadap fitur yang relevan dengan tugasnya di dalam organisasi.



Gambar 4.56 Halaman Master Data User

5. Halaman Create User

Tampilan pada gambar adalah *pop-up* “Add New User” dalam modul *Master User* yang digunakan untuk menambahkan pengguna baru ke dalam sistem aplikasi SPART. Formulir ini berisi empat kolom isian penting yaitu *Username*, *Fullname*, *Role*, dan *Location*. Admin dapat menentukan peran pengguna seperti *admin*, *warehouse*, *technician*, atau *manajemen*, serta mengaitkannya dengan lokasi gudang tertentu jika diperlukan. Di bagian bawah form, terdapat informasi bahwa *password default* untuk *user* baru adalah 123456, yang dapat diubah oleh pengguna. Setelah semua data diisi, admin cukup menekan tombol “Save” untuk menyimpan informasi pengguna ke sistem. Fitur ini memudahkan pengelolaan akses *user* dan menjamin setiap pengguna mendapat peran serta hak akses yang sesuai dalam aplikasi.

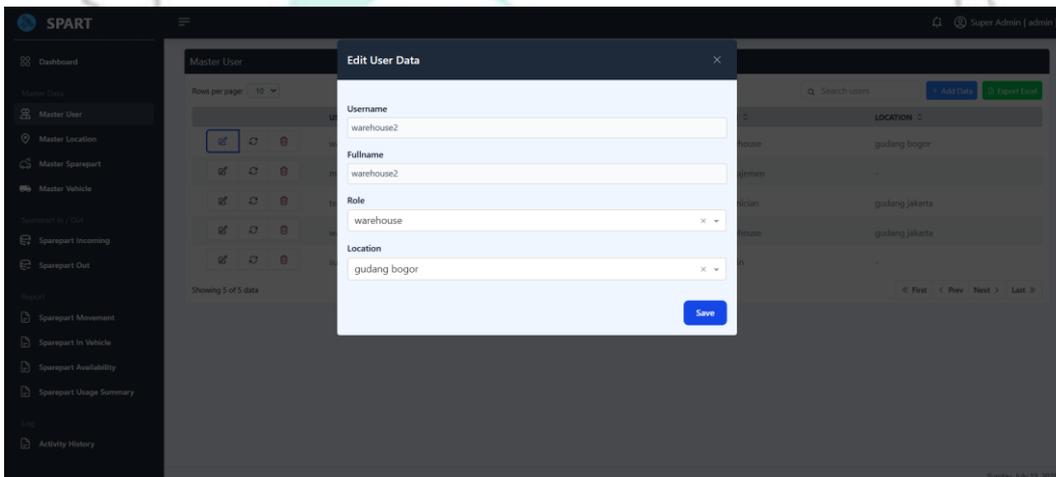


Gambar 4.57 Halaman Create User

6. Halaman *Edit User*

Gambar yang ditampilkan menunjukkan tampilan *pop-up* “*Edit User Data*” dalam menu *Master User* pada aplikasi SPART. Formulir ini digunakan untuk memperbarui data pengguna yang telah ada di dalam sistem. Terdapat empat *field* yang dapat diedit, yaitu *Username*, *Fullname*, *Role*, dan *Location*.

Setelah perubahan dilakukan, pengguna cukup mengklik tombol “*Save*” berwarna biru di bagian bawah untuk menyimpan pembaruan data tersebut. Fitur ini sangat penting untuk menjaga akurasi dan update informasi pengguna, serta memudahkan admin dalam mengelola akses sesuai dengan perubahan struktur organisasi atau penempatan lokasi kerja.



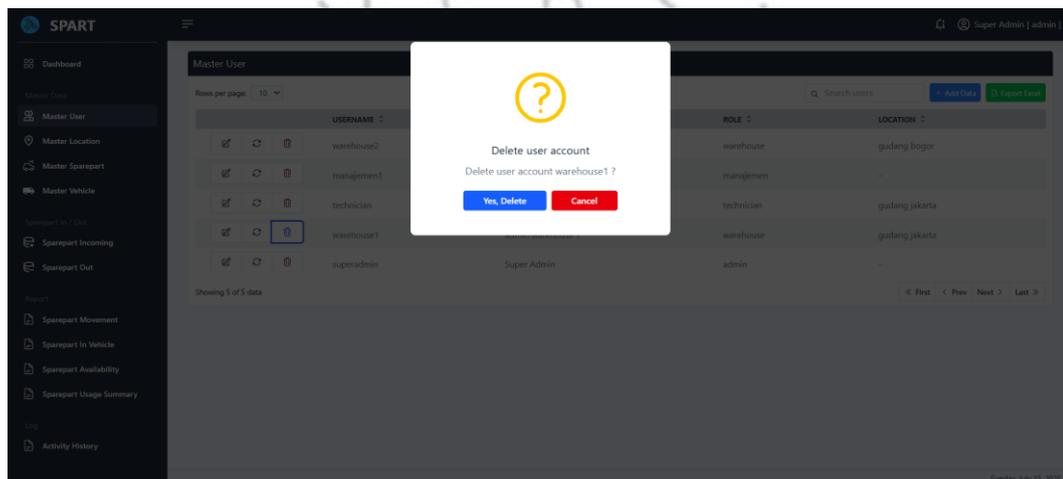
Gambar 4.58 Halaman *Edit User*

7. Halaman *Delete User*

Gambar yang ditampilkan menunjukkan tampilan *pop-up* konfirmasi “*Delete User Account*” pada menu *Master User* dalam aplikasi SPART. Dialog ini muncul ketika *user* menekan tombol ikon tempat sampah (hapus) pada salah satu baris data pengguna. Dalam kasus ini, sistem menanyakan apakah benar ingin menghapus akun pengguna dengan *username* warehouse2. *Pop-up* ini menyediakan dua opsi tindakan:

1. Tombol biru “*Yes, Delete*” untuk mengonfirmasi penghapusan akun, yang akan secara permanen menghapus data *user* dari sistem.
2. Tombol merah “*Cancel*” untuk membatalkan proses dan menutup jendela konfirmasi.

Fitur ini penting untuk mencegah penghapusan data secara tidak sengaja dan memberikan kontrol penuh kepada administrator dalam pengelolaan data *user*.



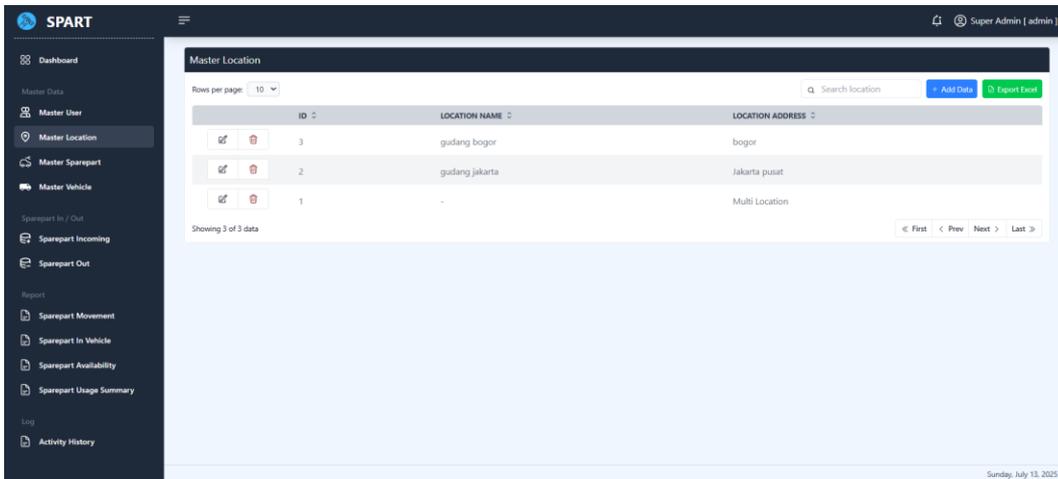
Gambar 4.59 Halaman *Delete User*

8. Halaman *Master Data Location*

Halaman *Master Location* pada aplikasi SPART digunakan untuk mengelola data lokasi gudang penyimpanan *sparepart*. Tampilan utamanya berupa tabel yang menampilkan daftar lokasi dengan tiga kolom utama: ID, nama lokasi, dan alamat lokasi. Setiap entri dilengkapi dengan tombol *edit* (ikon pensil) dan *hapus* (ikon tempat sampah) untuk memudahkan pengelolaan data. Fitur pencarian tersedia untuk memfilter lokasi berdasarkan kata kunci, dan pengguna juga bisa memilih jumlah data yang ditampilkan per halaman.

Selain itu, halaman ini menyediakan tombol "+ *Add Data*" untuk menambahkan lokasi baru dan tombol "*Export Excel*" untuk mengunduh data lokasi dalam format *Excel*. Salah satu data memiliki nama lokasi kosong dengan alamat "*Multi Location*" yang digunakan untuk pengguna dengan akses ke lebih dari satu gudang. Fungsionalitas ini

sangat penting untuk mendukung pengelolaan *sparepart* yang tersebar di berbagai lokasi secara efisien dan terstruktur.



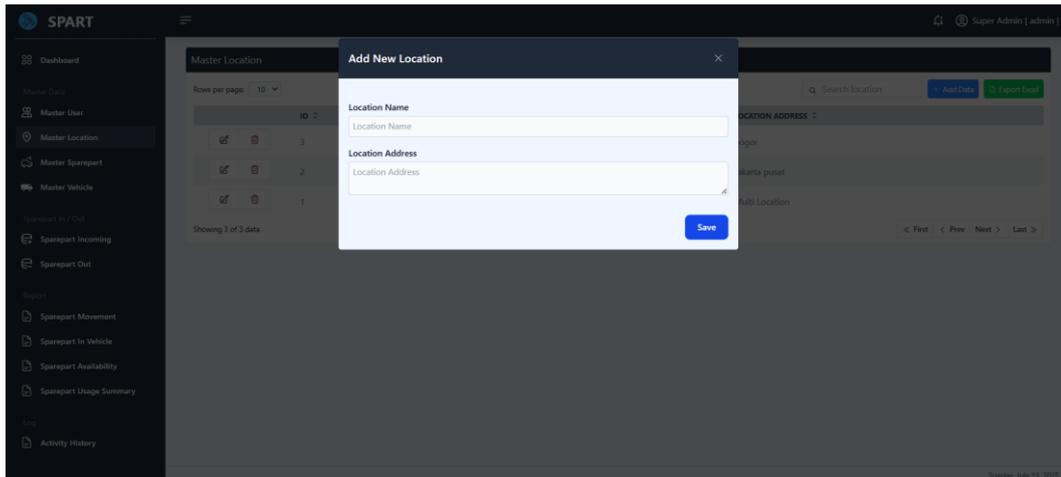
ID	LOCATION NAME	LOCATION ADDRESS
3	gudang bogor	bogor
2	gudang jakarta	Jakarta pusat
1	-	Multi Location

Gambar 4.60 Halaman *Master Data Location*

9. Halaman *Create Location*

Gambar ini menampilkan tampilan form tambah lokasi baru (*Add New Location*) pada modul *Master Location* aplikasi SPART. Form ini muncul sebagai *pop-up* ketika pengguna mengklik tombol "+ Add Data" dan digunakan untuk memasukkan informasi lokasi gudang baru. Terdapat dua *field* input utama yang harus diisi: *Location Name* (nama lokasi) dan *Location Address* (alamat lokasi). Setelah data diisi, pengguna dapat menyimpannya dengan menekan tombol *Save* berwarna biru di bagian bawah.

Tampilan ini dirancang sederhana dan intuitif untuk memastikan pengguna dapat menambahkan data dengan cepat tanpa kebingungan. Tombol "X" di pojok kanan atas memungkinkan pengguna menutup form jika ingin membatalkan penambahan lokasi. Fitur ini sangat berguna untuk memperbarui database lokasi secara langsung melalui antarmuka tanpa perlu melakukan input manual melalui backend atau file eksternal.

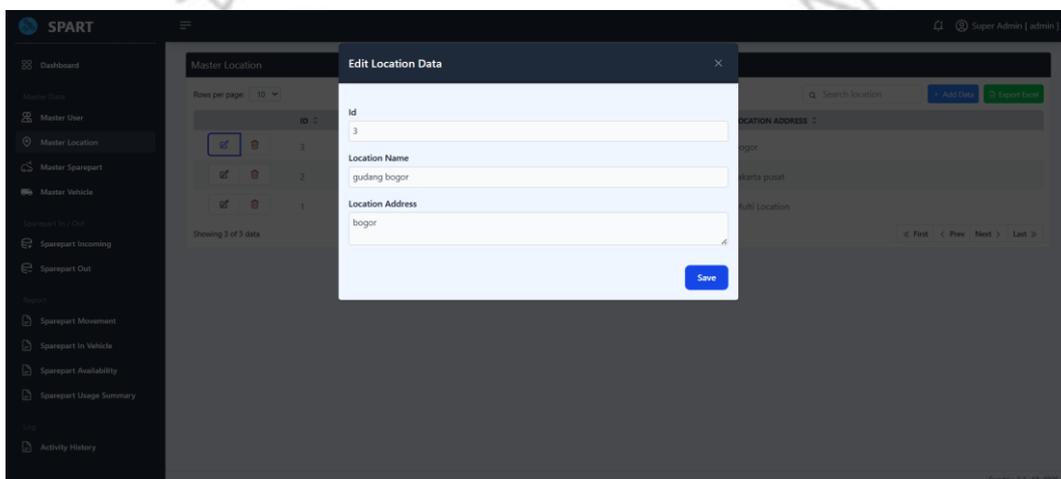


Gambar 4.61 Halaman *Create Location*

10. Halaman *Edit Location*

Gambar ini menampilkan tampilan form *edit* lokasi (*Edit Location Data*) dalam aplikasi SPART. Form ini ditampilkan dalam bentuk *pop-up* ketika pengguna mengklik ikon pensil pada baris data lokasi tertentu. Form ini berisi tiga *field*: Id (yang bersifat *non-editable*), *Location Name*, dan *Location Address*, yang memungkinkan pengguna untuk memperbarui nama dan alamat lokasi yang sudah ada di sistem.

Tampilan ini dirancang agar pengguna dapat melakukan pembaruan data dengan mudah tanpa meninggalkan halaman utama. Setelah melakukan perubahan yang diperlukan, pengguna cukup menekan tombol *Save* berwarna biru untuk menyimpan pembaruan tersebut. Fitur ini penting untuk menjaga akurasi dan konsistensi data lokasi dalam manajemen *inventory sparepart*.



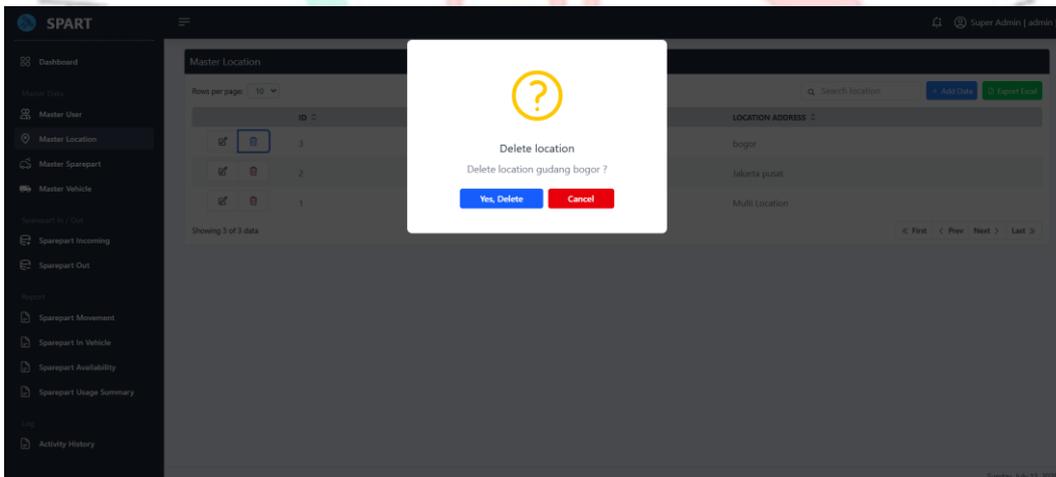
Gambar 4.62 Halaman *Edit Location*

11. Halaman *Delete Location*

Gambar yang ditampilkan menunjukkan tampilan *pop-up* konfirmasi "*Delete Location*" pada menu *Master Location* dalam aplikasi SPART. Dialog ini muncul ketika *user* menekan tombol ikon tempat sampah (hapus) pada salah satu baris data *location*. Dalam kasus ini, sistem menanyakan apakah benar ingin menghapus *location* dengan nama "gudang bogor". *Pop-up* ini menyediakan dua opsi tindakan:

1. Tombol biru "*Yes, Delete*" untuk mengonfirmasi penghapusan *location*, yang akan secara permanen menghapus data *location* dari sistem.
2. Tombol merah "*Cancel*" untuk membatalkan proses dan menutup jendela konfirmasi.

Fitur ini penting untuk mencegah penghapusan data secara tidak sengaja dan memberikan kontrol penuh kepada administrator dalam pengelolaan data *location*.



Gambar 4.63 Halaman *Delete Location*

12. Halaman *Master Data Sparepart*

Halaman "*Master Sparepart*" pada aplikasi SPART digunakan untuk mengelola data *sparepart*. Di sini, *user* dapat melihat daftar seluruh *sparepart* lengkap dengan informasi seperti *Code* (kode *sparepart*), *Name*, Merk, Image, *Description*, Minimum Qty, dan lokasi gudang yang terkait. Halaman ini juga menyediakan fitur pencarian *sparepart*, serta tombol untuk menambahkan data baru (*Add Data*) dan

ekspor data ke file *Excel* (*Export Excel*), yang sangat membantu untuk keperluan dokumentasi atau analisis.

Setiap entri *user* dilengkapi dengan tombol aksi yang memungkinkan admin melakukan *edit* data atau menghapus data *sparepart* dengan mudah. Fungsionalitas ini sangat penting untuk mendukung pengelolaan *sparepart* yang tersebar di berbagai lokasi secara efisien dan terstruktur.

ID	CODE	NAME	MERK	IMAGE	DESCRIPTION	MINIMUM QTY	LOCATION
4	A4	Busi A	A		Busi A	2	gudang jakarta
3	A3	Ban Enduro	Enduro		Ban Enduro	2	gudang bogor
2	A2	Ban Michelin	Michelin		Ban Michelin	1	gudang jakarta
1	A1	Ban Aspira	Aspira		Ban Aspira	1	gudang bogor

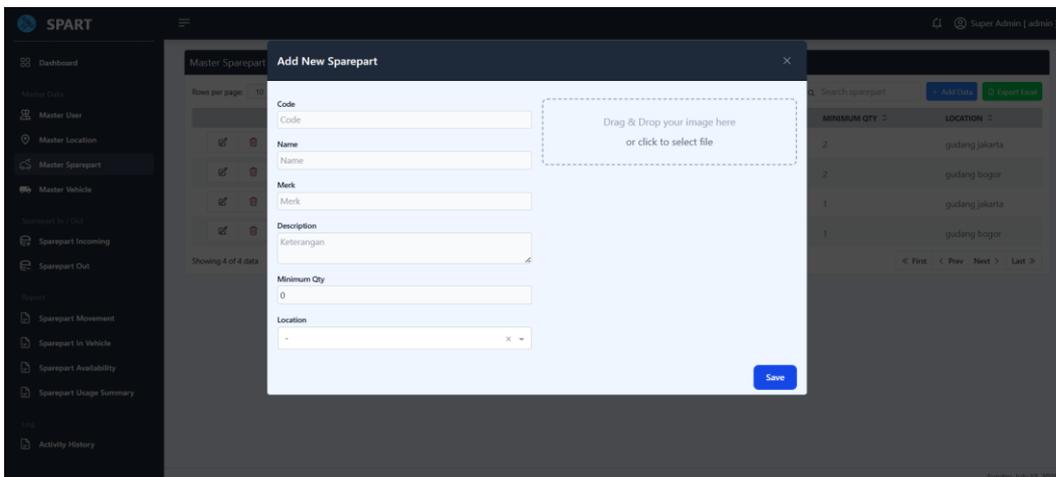
Gambar 4.64 Halaman *Master Data Sparepart*

13. Halaman *Create Sparepart*

Gambar ini menampilkan tampilan form tambah *sparepart* baru (*Add New Sparepart*) pada modul *Master Sparepart* aplikasi SPART. Form ini muncul sebagai *pop-up* ketika pengguna mengklik tombol "+ *Add Data*" dan digunakan untuk memasukkan informasi *sparepart* baru. Terdapat *field* input utama yang harus diisi: *Code* (kode *sparepart*), *Name* (nama *sparepart*), *Merk* (merk *sparepart*), *Description* (deskripsi *sparepart*), *Minimum Qty* (jumlah minimum *sparepart*), *Location* (lokasi *sparepart*), dan upload *Image*. Setelah data diisi, pengguna dapat menyimpannya dengan menekan tombol *Save* berwarna biru di bagian bawah.

Tampilan ini dirancang sederhana dan intuitif untuk memastikan pengguna dapat menambahkan data dengan cepat tanpa kebingungan. Tombol "X" di pojok kanan atas memungkinkan pengguna menutup form jika ingin membatalkan penambahan *sparepart*. Fitur ini sangat berguna untuk memperbarui database *sparepart* secara langsung

melalui antarmuka tanpa perlu melakukan input manual melalui backend atau file eksternal.

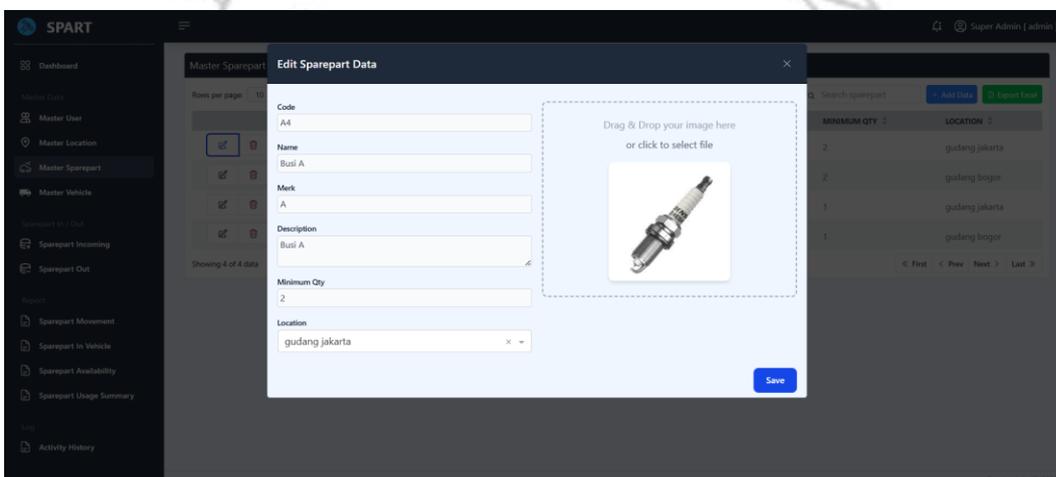


Gambar 4.65 Halaman *Create Sparepart*

14. Halaman *Edit Sparepart*

Gambar yang ditampilkan menunjukkan tampilan *pop-up* “*Edit Sparepart Data*” dalam menu *Master Sparepart* pada aplikasi SPART. Formulir ini digunakan untuk memperbarui data *sparepart* yang telah ada di dalam sistem. Terdapat lima *field* yang dapat diedit, yaitu *Name*, *Merk*, *Description*, *Minimum Qty*, *Location*, dan *Image*.

Setelah perubahan dilakukan, pengguna cukup mengklik tombol “*Save*” berwarna biru di bagian bawah untuk menyimpan pembaruan data tersebut. Fitur ini sangat penting untuk menjaga akurasi dan update informasi *sparepart*, serta memudahkan admin dalam mengelola akses sesuai dengan lokasi *sparepart* tersebut berada.



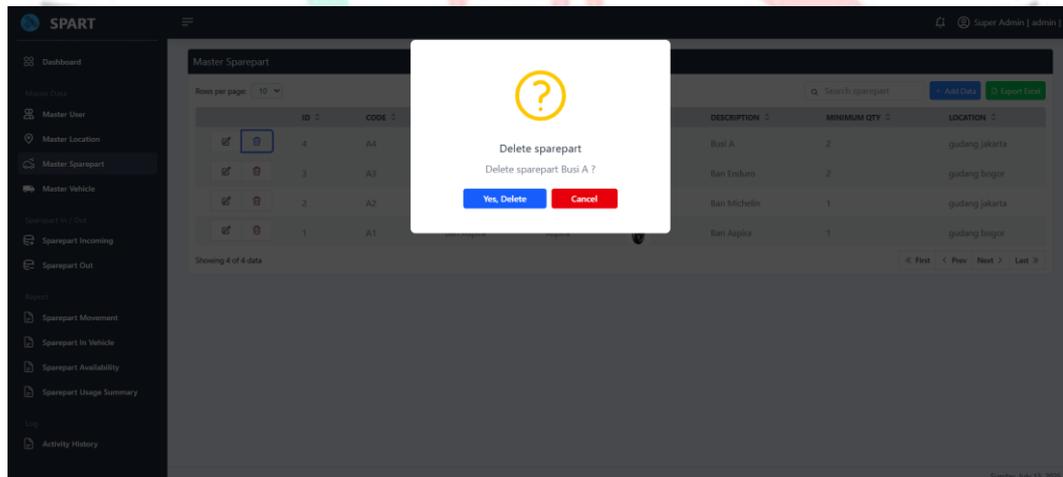
Gambar 4.66 Halaman *Edit Sparepart*

15. Halaman *Delete Sparepart*

Gambar yang ditampilkan menunjukkan tampilan *pop-up* konfirmasi "*Delete Sparepart*" pada menu *Master Sparepart* dalam aplikasi SPART. Dialog ini muncul ketika *user* menekan tombol ikon tempat sampah (hapus) pada salah satu baris data *sparepart*. Dalam kasus ini, sistem menanyakan apakah benar ingin menghapus *sparepart* dengan nama "Busi A". *Pop-up* ini menyediakan dua opsi tindakan:

1. Tombol biru "*Yes, Delete*" untuk mengonfirmasi penghapusan *sparepart*, yang akan secara permanen menghapus data *sparepart* dari sistem.
2. Tombol merah "*Cancel*" untuk membatalkan proses dan menutup jendela konfirmasi.

Fitur ini penting untuk mencegah penghapusan data secara tidak sengaja dan memberikan kontrol penuh kepada administrator dalam pengelolaan data *sparepart*.



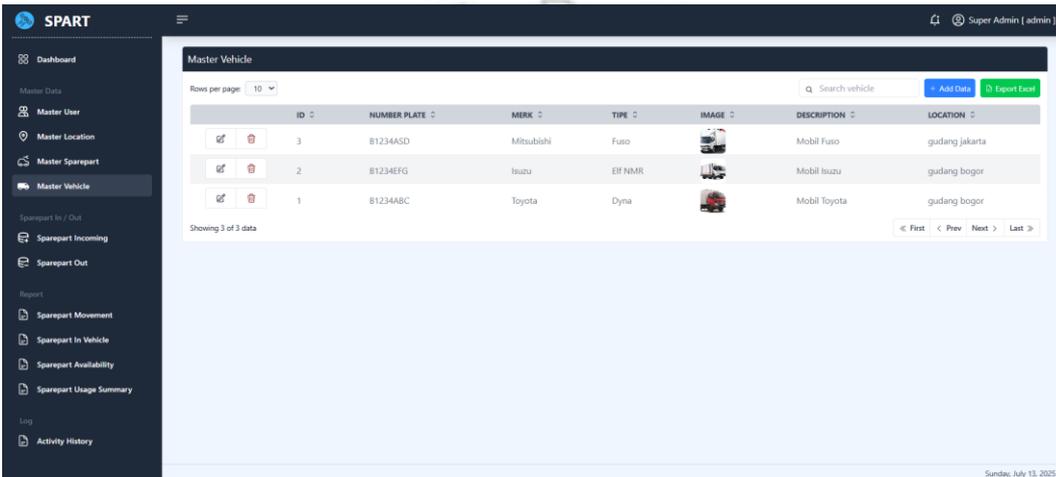
Gambar 4.67 Halaman *Delete Sparepart*

16. Halaman *Master Data Vehicle*

Halaman *Master Vehicle* pada aplikasi SPART digunakan untuk mengelola data kendaraan perusahaan. Tampilan utamanya berupa tabel yang menampilkan daftar kendaraan yang berisikan kolom informasi kendaraan meliputi: ID, *Number Plat* (nomor plat kendaraan), Merk, *Type*, Image, *Description* dan Lokasi. Setiap entri dilengkapi dengan tombol *edit* (ikon pensil) dan hapus (ikon tempat sampah) untuk memudahkan pengelolaan data. Fitur pencarian tersedia untuk

memfilter kendaraan berdasarkan kata kunci, dan pengguna juga bisa memilih jumlah data yang ditampilkan per halaman.

Selain itu, halaman ini menyediakan tombol "+ Add Data" untuk menambahkan kendaraan baru dan tombol "Export Excel" untuk mengunduh data kendaraan dalam format *Excel*. Fungsionalitas ini sangat penting untuk mendukung pengelolaan kendaraan yang tersebar di berbagai lokasi secara efisien dan terstruktur.



The screenshot shows the 'Master Vehicle' page in the SPART application. It features a sidebar menu on the left with options like Dashboard, Master User, Master Location, Master Sparepart, Master Vehicle, Sparepart In / Out, Sparepart Incoming, Sparepart Out, Report, Sparepart Movement, Sparepart In Vehicle, Sparepart Availability, Sparepart Usage Summary, Log, and Activity History. The main content area displays a table with the following data:

ID	NUMBER PLATE	MERK	TIPE	IMAGE	DESCRIPTION	LOCATION
3	B1234ASD	Mitsubishi	Fuso		Mobil Fuso	gudang jakarta
2	B1234EFG	Isuzu	Elf NMR		Mobil Isuzu	gudang bogor
1	B1234ABC	Toyota	Dyna		Mobil Toyota	gudang bogor

At the top of the table, there is a search bar labeled 'Search vehicle', a '+ Add Data' button, and an 'Export Excel' button. The table also includes a 'Rows per page' dropdown set to 10 and pagination controls at the bottom right showing 'Showing 3 of 3 data' and navigation buttons for 'First', 'Prev', 'Next', and 'Last'.

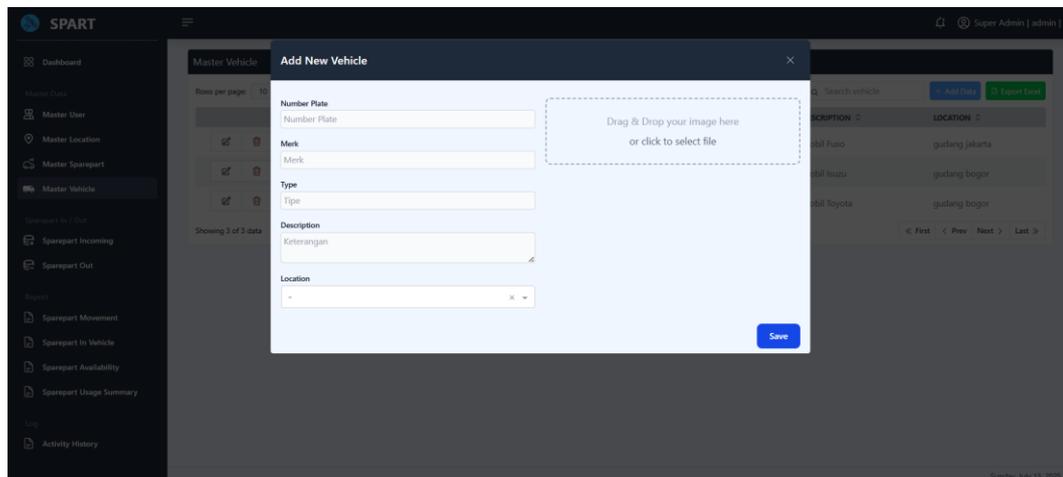
Gambar 4.68 Halaman *Master Data Vehicle*

17. Halaman *Create Vehicle*

Gambar ini menampilkan tampilan form tambah kendaraan baru (*Add New Vehicle*) pada modul *Master Vehicle* aplikasi SPART. Form ini muncul sebagai *pop-up* ketika pengguna mengklik tombol "+ Add Data" dan digunakan untuk memasukkan informasi kendaraan baru. Terdapat *field* input utama yang harus diisi: *Number Plate* (nomor plat kendaraan), *Merk*, *Type*, *Description* dan *Location*. Setelah data diisi, pengguna dapat menyimpannya dengan menekan tombol *Save* berwarna biru di bagian bawah.

Tampilan ini dirancang sederhana dan intuitif untuk memastikan pengguna dapat menambahkan data dengan cepat tanpa kebingungan. Tombol "X" di pojok kanan atas memungkinkan pengguna menutup form jika ingin membatalkan penambahan kendaraan. Fitur ini sangat berguna untuk memperbarui database kendaraan secara langsung

melalui antarmuka tanpa perlu melakukan input manual melalui backend atau file eksternal.

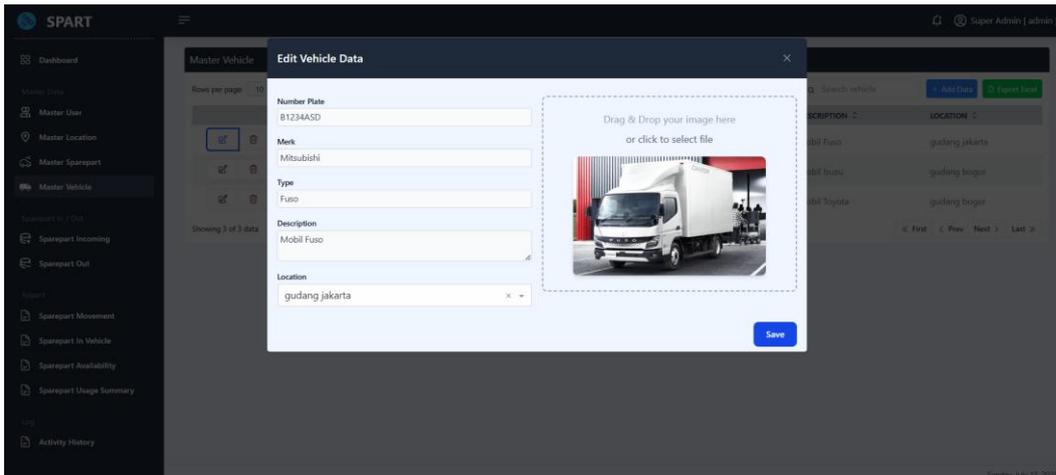


Gambar 4.69 Halaman *Create Vehicle*

18. Halaman *Edit Vehicle*

Gambar ini menampilkan tampilan form *edit* kendaraan (*Edit Vehicle Data*) dalam aplikasi SPART. Form ini ditampilkan dalam bentuk *pop-up* ketika pengguna mengklik ikon pensil pada baris data kendaraan tertentu. Form ini berisi empat *field* yang bisa diubah: *Merk*, *Type*, *Description*, *Location*, dan *Image*, yang memungkinkan *user* untuk memperbarui data kendaraan sudah ada di sistem.

Tampilan ini dirancang agar pengguna dapat melakukan pembaruan data dengan mudah tanpa meninggalkan halaman utama. Setelah melakukan perubahan yang diperlukan, pengguna cukup menekan tombol *Save* berwarna biru untuk menyimpan pembaruan tersebut. Fitur ini penting untuk menjaga akurasi dan konsistensi data kendaraan dalam aplikasi manajemen *inventory sparepart*.



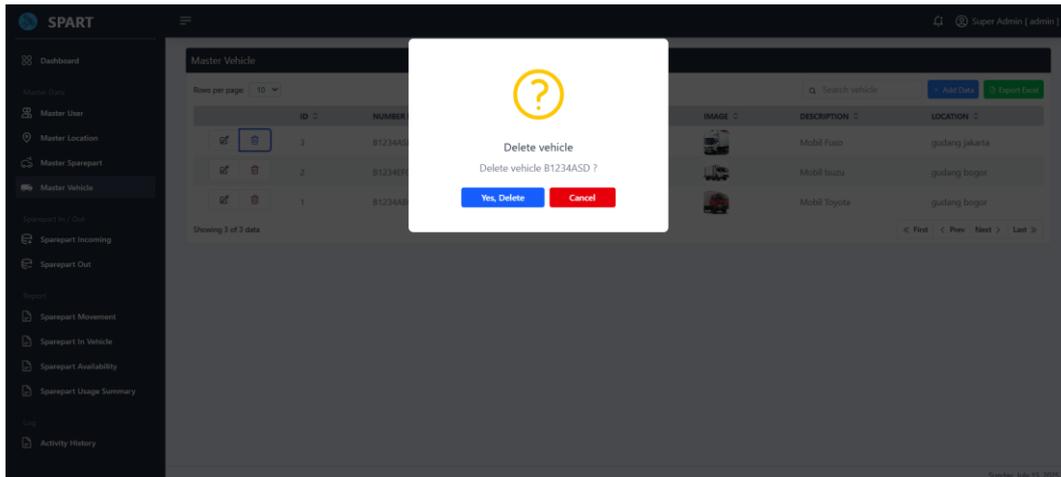
Gambar 4.70 Halaman *Edit Vehicle*

19. Halaman *Delete Vehicle*

Gambar yang ditampilkan menunjukkan tampilan *pop-up* konfirmasi "*Delete Vehicle*" pada menu *Master Vehicle* dalam aplikasi SPART. Dialog ini muncul ketika *user* menekan tombol ikon tempat sampah (hapus) pada salah satu baris data *Vehicle*. Dalam kasus ini, sistem menanyakan apakah benar ingin menghapus *Vehicle* dengan nama "B1234ASD". *Pop-up* ini menyediakan dua opsi tindakan:

1. Tombol biru "*Yes, Delete*" untuk mengonfirmasi penghapusan *location*, yang akan secara permanen menghapus data *Vehicle* dari sistem.
2. Tombol merah "*Cancel*" untuk membatalkan proses dan menutup jendela konfirmasi.

Fitur ini penting untuk mencegah penghapusan data secara tidak sengaja dan memberikan kontrol penuh kepada administrator dalam pengelolaan data *Vehicle*.

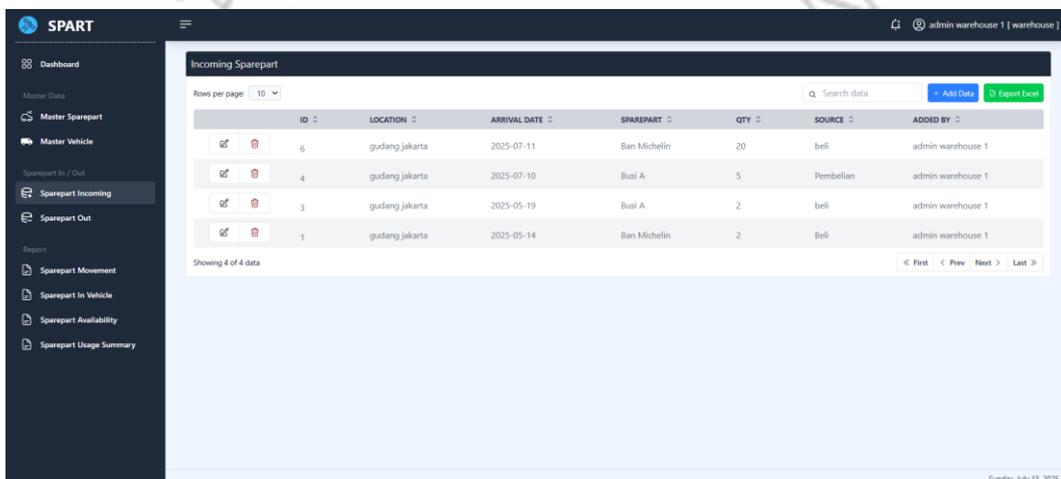


Gambar 4.71 Halaman *Delete Vehicle*

20. Halaman *Sparepart Incoming*

Gambar ini menampilkan tampilan halaman *Incoming Sparepart* dalam aplikasi SPART, yang berfungsi untuk mencatat dan memantau *sparepart* yang masuk ke gudang. Di bagian atas terdapat fitur pencarian data, tombol *Add Data* untuk menambahkan *sparepart* masuk baru, serta tombol *Export Excel* untuk mengunduh laporan dalam format *Excel*.

Tabel yang ditampilkan berisi informasi detail terkait *sparepart* masuk, seperti ID, nama lokasi, tanggal kedatangan (*Arrival Date*), nama *sparepart*, jumlah (*Qty*), sumber *sparepart* (*Source*), serta pengguna yang menambahkan data (*Added By*). Fitur ini memungkinkan pengguna *warehouse* untuk merekam dan menelusuri setiap transaksi masuk *sparepart* secara akurat dan efisien, sehingga stok dapat dikelola dengan baik.

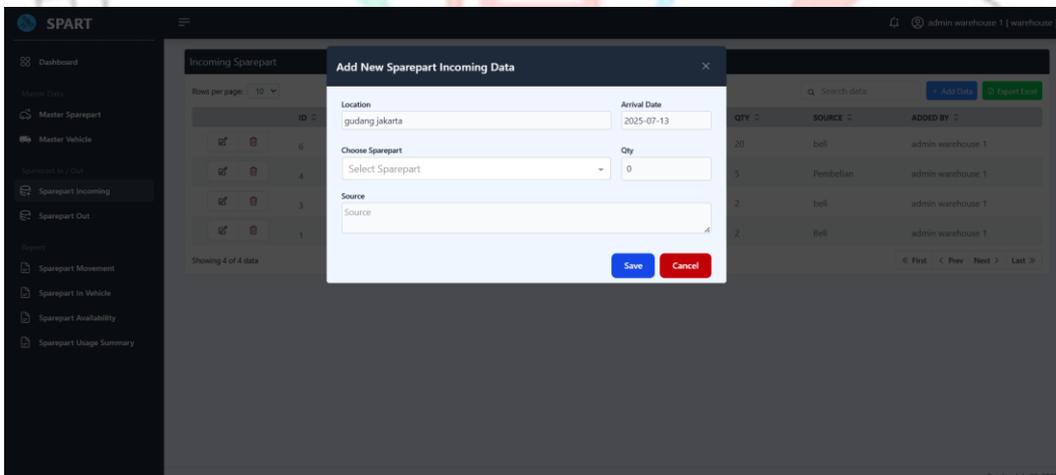


Gambar 4.72 Halaman *Sparepart Incoming*

21. Halaman *Create Sparepart Incoming*

Gambar ini menampilkan tampilan form tambah data *sparepart* masuk (*Add New Sparepart Incoming Data*) pada menu *Incoming Sparepart* aplikasi SPART. Form ini muncul sebagai *pop-up* ketika pengguna mengklik tombol "+ Add Data" dan digunakan untuk memasukkan informasi *sparepart* yang masuk ke gudang. Terdapat *field* input utama yang harus diisi: *Location*, *Arrival Date*, *Choose Sparepart* (nama *sparepart*), *Qty*, dan *Source*. Setelah data diisi, pengguna dapat menyimpannya dengan menekan tombol *Save* berwarna biru di bagian bawah.

Tampilan ini dirancang sederhana dan intuitif untuk memastikan pengguna dapat menambahkan data dengan cepat tanpa kebingungan. Tombol "X" di pojok kanan atas memungkinkan pengguna menutup form jika ingin membatalkan penambahan *sparepart* masuk. Fitur ini sangat berguna untuk memperbarui database transaksi secara langsung melalui antarmuka tanpa perlu melakukan input manual melalui backend atau file eksternal.



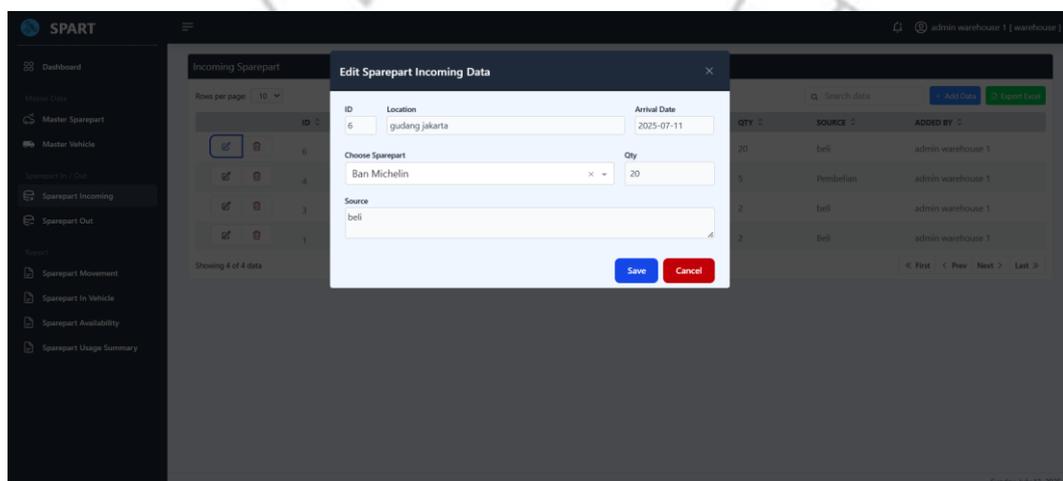
Gambar 4.73 Halaman *Create Sparepart Incoming*

22. Halaman *Edit Sparepart Incoming*

Gambar ini menampilkan tampilan form *edit sparepart* masuk (*Edit Sparepart Incoming Data*) dalam aplikasi SPART. Form ini ditampilkan dalam bentuk *pop-up* ketika pengguna mengklik ikon pensil pada baris data *sparepart* masuk tertentu. Form ini berisi *field* yang bisa diubah:

Arrival Date, *Choose Sparepart* (nama *sparepart*), *Qty*, dan *Source*, yang memungkinkan *user* untuk memperbarui data *sparepart* masuk yang sudah ada di sistem.

Tampilan ini dirancang agar pengguna dapat melakukan pembaruan data dengan mudah tanpa meninggalkan halaman utama. Setelah melakukan perubahan yang diperlukan, pengguna cukup menekan tombol *Save* berwarna biru untuk menyimpan pembaruan tersebut. Fitur ini penting untuk menjaga akurasi dan konsistensi data *sparepart* masuk dalam aplikasi manajemen *inventory sparepart*.



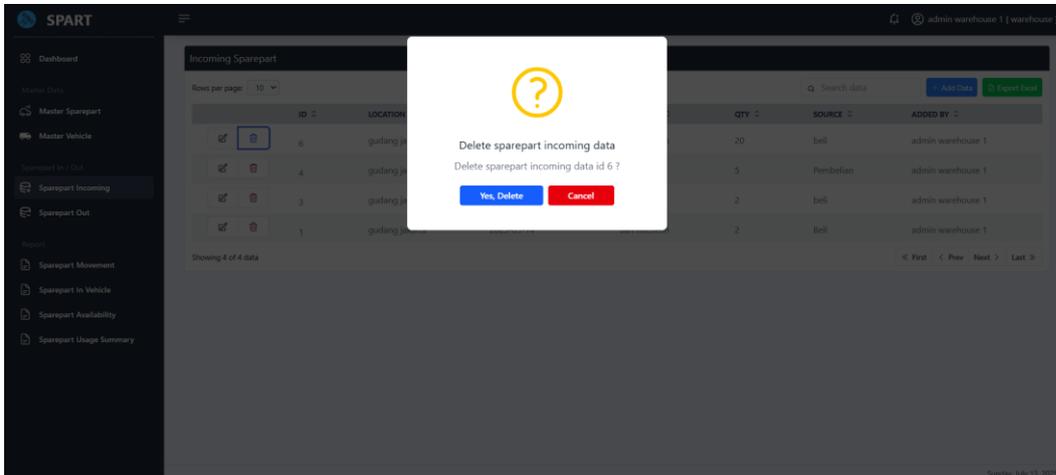
Gambar 4.74 Halaman *Edit Sparepart Incoming*

23. Halaman *Delete Sparepart Incoming*

Gambar yang ditampilkan menunjukkan tampilan *pop-up* konfirmasi "*Delete Sparepart In*" pada menu *Sparepart Incoming* dalam aplikasi SPART. Dialog ini muncul ketika *user* menekan tombol ikon tempat sampah (hapus) pada salah satu baris data *Sparepart In*. Dalam kasus ini, sistem menanyakan apakah benar ingin menghapus *Sparepart In* dengan id "3". *Pop-up* ini menyediakan dua opsi tindakan:

1. Tombol biru "*Yes, Delete*" untuk mengonfirmasi penghapusan *Sparepart In*, yang akan secara permanen menghapus data *Sparepart In* dari sistem.
2. Tombol merah "*Cancel*" untuk membatalkan proses dan menutup jendela konfirmasi.

Fitur ini penting untuk mencegah penghapusan data secara tidak sengaja dan memberikan kontrol penuh kepada administrator dalam pengelolaan data *Sparepart In*.



Gambar 4.75 Halaman *Delete Sparepart Incoming*

24. Halaman *Sparepart Outcoming*

Gambar ini menampilkan halaman *Sparepart Out* pada aplikasi SPART, yang digunakan untuk mencatat pengeluaran *sparepart* dari gudang. Di bagian atas, terdapat tombol *Add Data* untuk menambahkan transaksi baru dan *Export Excel* untuk mengunduh data dalam format spreadsheet. Selain itu, tersedia juga fitur pencarian data dan pengaturan jumlah baris per halaman.

Tabel yang ditampilkan menyajikan informasi lengkap seperti ID transaksi, lokasi, tanggal keluar (*Out Date*), nama *sparepart*, jumlah (*Qty*), status penggunaan (*Used*), nomor kendaraan (*Vehicle*) yang menggunakan *sparepart* tersebut, alasan pengeluaran (*Reason*), serta pengguna yang menambahkan data (*Added By*). Halaman ini memungkinkan pengguna *warehouse* untuk mengelola data *sparepart* keluar secara tertib dan terdokumentasi dengan baik.

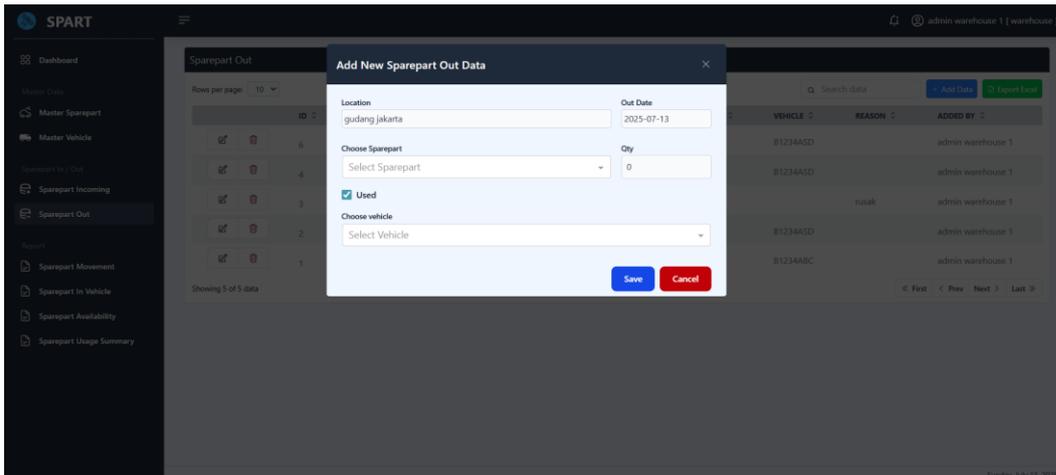
ID	LOCATION	OUT DATE	SPAREPART	QTY	USED	VEHICLE	REASON	ADDED BY
6	gudang jakarta	2025-07-11	Ban Michelin	12	<input checked="" type="checkbox"/>	B1234ASD		admin warehouse 1
4	gudang jakarta	2025-07-10	Busi A	2	<input checked="" type="checkbox"/>	B1234ASD		admin warehouse 1
3	gudang jakarta	2025-06-17	Ban Michelin	1	<input type="checkbox"/>		rusak	admin warehouse 1
2	gudang jakarta	2025-05-19	Busi A	2	<input checked="" type="checkbox"/>	B1234ASD		admin warehouse 1
1	gudang jakarta	2025-05-14	Ban Michelin	1	<input checked="" type="checkbox"/>	B1234ABC		admin warehouse 1

Gambar 4.76 Halaman *Sparepart Outcoming*

25. Halaman *Create Sparepart Outcoming*

Gambar ini menampilkan tampilan form tambah data *sparepart* keluar (*Add New Sparepart Outcoming Data*) pada menu *Outcoming Sparepart* aplikasi SPART. Form ini muncul sebagai *pop-up* ketika pengguna mengklik tombol "+ *Add Data*" dan digunakan untuk memasukkan informasi *sparepart* yang keluar dari gudang. Terdapat *field* input utama yang harus diisi: *Location*, *Out Date*, *Choose Sparepart* (nama *sparepart*), *Qty*, dan *Choose Vehicle / Reason*. Setelah data diisi, pengguna dapat menyimpannya dengan menekan tombol *Save* berwarna biru di bagian bawah.

Tampilan ini dirancang sederhana dan intuitif untuk memastikan pengguna dapat menambahkan data dengan cepat tanpa kebingungan. Tombol "X" di pojok kanan atas memungkinkan pengguna menutup form jika ingin membatalkan penambahan *sparepart* keluar. Fitur ini sangat berguna untuk memperbarui database transaksi secara langsung melalui antarmuka tanpa perlu melakukan input manual melalui backend atau file eksternal.

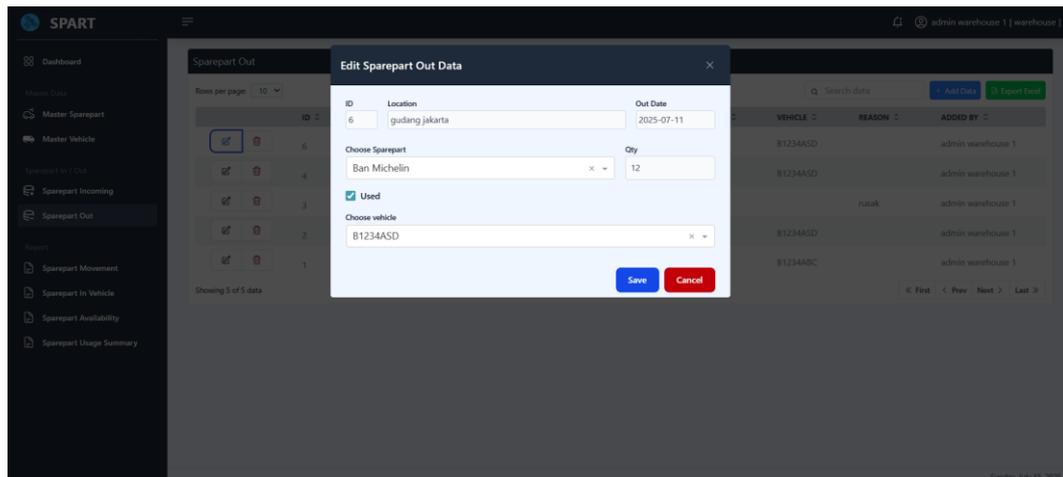


Gambar 4.77 Halaman *Create Sparepart Outcoming*

26. Halaman *Edit Sparepart Outcoming*

Gambar ini menampilkan tampilan form *edit sparepart* keluar (*Edit Sparepart Outcoming Data*) dalam aplikasi SPART. Form ini ditampilkan dalam bentuk *pop-up* ketika pengguna mengklik ikon pensil pada baris data *sparepart* masuk tertentu. Form ini berisi *field* yang bisa diubah: *Out Date*, *Choose Sparepart* (nama *sparepart*), *Qty*, dan *Vehicle / Reason*, yang memungkinkan *user* untuk memperbarui data *sparepart* keluar yang sudah ada di sistem.

Tampilan ini dirancang agar pengguna dapat melakukan pembaruan data dengan mudah tanpa meninggalkan halaman utama. Setelah melakukan perubahan yang diperlukan, pengguna cukup menekan tombol *Save* berwarna biru untuk menyimpan pembaruan tersebut. Fitur ini penting untuk menjaga akurasi dan konsistensi data *sparepart* keluar dalam aplikasi manajemen *inventory sparepart*.



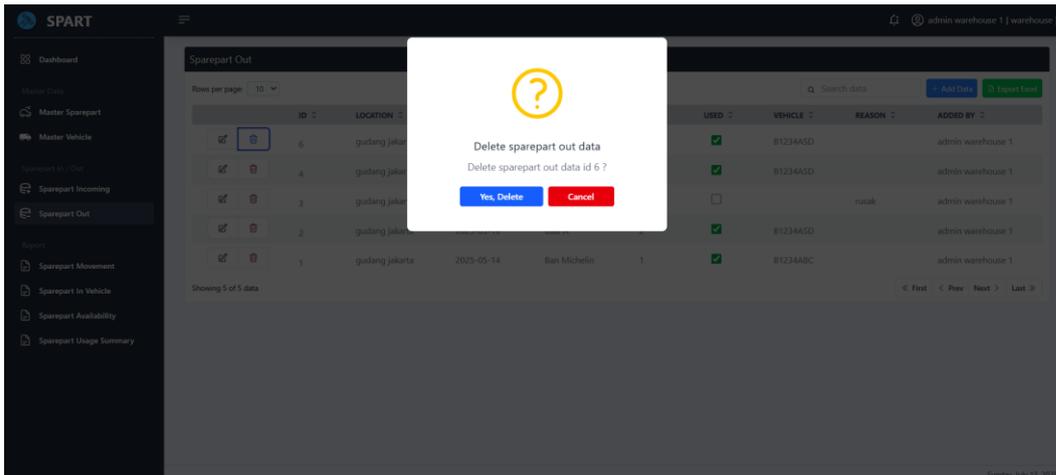
Gambar 4.78 Halaman *Edit Sparepart Outcoming*

27. Halaman *Delete Sparepart Outcoming*

Gambar yang ditampilkan menunjukkan tampilan *pop-up* konfirmasi "*Delete Sparepart Out*" pada menu *Sparepart Outcoming* dalam aplikasi SPART. Dialog ini muncul ketika *user* menekan tombol ikon tempat sampah (hapus) pada salah satu baris data *Sparepart Out*. Dalam kasus ini, sistem menanyakan apakah benar ingin menghapus *Sparepart Out* dengan id "3". *Pop-up* ini menyediakan dua opsi tindakan:

1. Tombol biru "*Yes, Delete*" untuk mengonfirmasi penghapusan *Sparepart Out*, yang akan secara permanen menghapus data *Sparepart Out* dari sistem.
2. Tombol merah "*Cancel*" untuk membatalkan proses dan menutup jendela konfirmasi.

Fitur ini penting untuk mencegah penghapusan data secara tidak sengaja dan memberikan kontrol penuh kepada administrator dalam pengelolaan data *Sparepart Out*.



Gambar 4.79 Halaman *Delete Sparepart Outcoming*

28. Halaman *Report Sparepart Movement*

Gambar ini menampilkan halaman *Report Sparepart Movement* pada aplikasi SPART. Halaman ini digunakan untuk memantau pergerakan *sparepart*, baik masuk maupun keluar, berdasarkan lokasi dan periode waktu tertentu. Di bagian atas terdapat filter berupa dropdown untuk memilih *Location*, *Sparepart*, serta kolom isian *Start Date* dan *End Date* yang digunakan untuk menyaring data laporan sesuai kebutuhan. Setelah itu, pengguna dapat menekan tombol *Load Report* untuk menampilkan hasilnya.

Tabel di bawah filter menampilkan data laporan pergerakan *sparepart*, seperti pada contoh: *sparepart* bernama “Ban Michelin” yang keluar (*Action: Out*) dari “gudang jakarta” dengan catatan “rusak”, jumlah sebanyak 1 unit, dan dilakukan pada tanggal 2025-06-17. Pengguna juga dapat mengekspor laporan ini ke dalam file *Excel* menggunakan tombol *Export Excel*, sehingga memudahkan pengarsipan atau analisis lebih lanjut.

LOCATION	SPAREPART NAME	ACTION	NOTE	QTY	DATE
gudang jakarta	Busi A	In	Incoming Sparepart	5	2025-07-10
gudang jakarta	Busi A	Used	B1234ASD	2	2025-07-10
gudang bogor	Ban Enduro	In	Incoming Sparepart	10	2025-07-11
gudang jakarta	Ban Michelin	In	Incoming Sparepart	20	2025-07-11
gudang bogor	Ban Enduro	Used	B1234ABC	5	2025-07-11
gudang jakarta	Ban Michelin	Used	B1234ASD	12	2025-07-11

Gambar 4.80 Halaman *Report Sparepart Movement*

29. Halaman *Report Sparepart In Vehicle*

Gambar ini memperlihatkan halaman *Report Sparepart in Vehicle* pada aplikasi SPART, yang berfungsi untuk menampilkan data *sparepart* yang sedang terpasang atau digunakan di kendaraan tertentu. Di bagian atas halaman, terdapat filter pencarian berupa pilihan *Location*, *Sparepart*, dan *Vehicle* untuk menyaring laporan berdasarkan gudang, nama *sparepart*, dan kendaraan yang diinginkan. Setelah filter dipilih, pengguna dapat menekan tombol *Load Report* untuk menampilkan data sesuai kriteria yang dipilih.

Tabel di bawah filter menampilkan hasil laporan yang mencakup informasi lokasi, nama *sparepart*, nomor kendaraan (*Number Plate*), dan jumlah *sparepart* (*Qty*) yang terpasang pada kendaraan tersebut. Sebagai contoh, *sparepart* “Ban Michelin” dengan jumlah 1 unit dipasang di kendaraan dengan pelat nomor B1234ABC, dan “Busi A” dengan jumlah 2 unit di kendaraan B1234ASD, keduanya berada di gudang jakarta. Pengguna juga dapat mengekspor data ini ke *Excel* dengan mudah melalui tombol *Export Excel*.

Report Sparepart in vehicle

Choose Location: All | Choose Sparepart: All | Choose Vehicle: All | Load Report

Rows per page: 10 | Search data | Export Excel

LOCATION	SPAREPART NAME	NUMBER PLATE	QTY
gudang bogor	Ban Enduro	B1234ABC	5
gudang jakarta	Ban Michelin	B1234ABC	1
gudang jakarta	Ban Michelin	B1234ASD	12
gudang jakarta	Busi A	B1234ASD	4

Showing 4 of 4 data | << First | Prev | Next | Last >>

Sunday, July 13, 2025

Gambar 4.81 Halaman *Report Sparepart In Vehicle*

30. Halaman *Report Sparepart Availability*

Gambar ini menampilkan halaman *Report Sparepart Availability* dari aplikasi SPART yang berguna untuk memantau ketersediaan *sparepart* di masing-masing lokasi gudang. Di bagian atas terdapat filter pencarian berdasarkan *Location* dan *Sparepart*, serta tombol *Load Report* untuk menampilkan data berdasarkan filter yang dipilih. Laporan juga bisa diekspor ke format *Excel* melalui tombol *Export Excel* di sebelah kanan.

Tabel laporan menampilkan informasi detail seperti nama lokasi, nama *sparepart*, jumlah minimum stok (Minimum Qty), jumlah *sparepart* yang masuk (*In*), yang sudah digunakan (*Used*), yang keluar (*Out*), serta jumlah *sparepart* yang masih tersedia (*Available*). Warna pada kolom "Available" menandakan status stok, misalnya angka 0 berwarna oranye menandakan stok kosong yang perlu segera ditindaklanjuti. Fitur ini membantu pengguna dalam pengambilan keputusan terkait pengadaan atau distribusi *sparepart*.

Report Sparepart Availability

Choose Location: All | Choose Sparepart: All | Load Report

Rows per page: 10 | Search data | Export Excel

LOCATION	SPAREPART NAME	MINIMUM QTY	IN	USED	OUT	AVAILABLE
gudang jakarta	Ban Aspira	1	0	0	0	0
gudang jakarta	Ban Michelin	1	22	13	1	8
gudang jakarta	Buzi A	2	7	4	0	3
gudang bogor	Ban Aspira	1	0	0	0	0
gudang bogor	Ban Michelin	1	4	0	0	4
gudang bogor	Ban Enduro	2	10	5	0	5

Showing 6 of 6 data | << First < Prev Next > Last >>

Sunday, July 13, 2025

Gambar 4.82 Halaman *Report Sparepart Availability*

31. Halaman *Report Sparepart Usage Summary*

Gambar ini menampilkan halaman *Report Sparepart Usage Summary* pada aplikasi SPART. Halaman ini digunakan untuk memantau pergerakan *sparepart* yang paling terbanyak terpakai dalam kurun waktu tertentu. Di bagian atas terdapat filter berupa dropdown untuk memilih *Location* serta kolom isian *Start Date* dan *End Date* yang digunakan untuk menyaring data laporan sesuai kebutuhan. Setelah itu, pengguna dapat menekan tombol *Load Report* untuk menampilkan hasilnya.

Tabel di bawah filter menampilkan data laporan pergerakan *sparepart*, seperti pada contoh: *sparepart* bernama “Ban Michelin” yang keluar atau terpakai dari “gudang jakarta” dengan jumlah sebanyak 12 unit yang merupakan *sparepart* terbanyak pada kurun waktu yang ditentukan. Pengguna juga dapat mengekspor laporan ini ke dalam file *Excel* menggunakan tombol *Export Excel*, sehingga memudahkan pengarsipan atau analisis lebih lanjut.

Report Sparepart Usage Summary

Choose Location: All | Start Date: 2025-07-01 | End Date: 2025-07-31 | Load Report

Rows per page: 10 | Search data | Export Excel

LOCATION	SPAREPART NAME	QTY
gudang jakarta	Ban Michelin	12
gudang bogor	Ban Enduro	5
gudang jakarta	Busi A	2

Showing 3 of 3 data | << First | Prev | Next | Last >>

Sunday, July 13, 2025

Gambar 4.83 Halaman *Report Sparepart Usage Summary*

4.4.2 Tahapan Pengujian

Tahap pengujian aplikasi adalah proses penting pada siklus pengembangan *software* dengan tujuan agar memastikan bahwa sistem telah berjalan sesuai dengan kebutuhan dan fungsionalitas yang telah dirancang. Pengujian dilakukan untuk mendeteksi adanya bug, kesalahan logika, atau kekurangan pada fitur sebelum aplikasi digunakan secara penuh oleh pengguna. Dalam konteks aplikasi *inventory* seperti SPART, pengujian meliputi seluruh modul mulai dari input data *master* (*user*, lokasi, *sparepart*), transaksi masuk dan keluar barang, hingga pelaporan.

Proses pengujian ini dapat dilakukan dengan beberapa metode, seperti pengujian fungsional untuk memastikan semua tombol, form, dan alur kerja berjalan sebagaimana mestinya, serta pengujian integrasi untuk melihat hubungan antar modul dan data. Selain itu, pengujian pengguna (*user acceptance testing*) dilakukan untuk mendapatkan umpan balik dari pengguna akhir terkait kenyamanan dan pengalaman dari menggunakan aplikasi. Hasil dari pengujian ini akan menjadi acuan untuk melakukan perbaikan atau penyempurnaan sistem agar aplikasi benar-benar siap digunakan dalam lingkungan operasional yang sesungguhnya. Berikut ini merupakan hasil dari pengujian aplikasi dalam bentuk table

Tabel 4.14. Table Pengujian Aplikasi

No	Menu	Nama Pengujian	Langkah Uji	Hasil yang Diharapkan	Status
1	<i>Login</i>	<i>Login</i>	Masukkan <i>username</i> dan <i>password</i> valid	Berhasil <i>login</i>	✓
2	<i>Login</i>	<i>Login - invalid</i>	Masukkan <i>username</i> atau <i>password</i> salah	Pesan error	✓
3	<i>Master User</i>	<i>Create user</i>	Klik ' <i>Add Data</i> ', isi form, simpan <i>user</i> baru	Data tersimpan	✓
4	<i>Master User</i>	<i>Create user - invalid</i>	Isi form kosong, klik simpan	Pesan error wajib isi	✓
5	<i>Master User</i>	<i>Read user</i>	Lihat daftar <i>user</i>	Data tampil	✓
6	<i>Master User</i>	<i>Update user</i>	Klik <i>edit</i> , ubah data, simpan data <i>user</i>	Data diperbarui	✓
7	<i>Master User</i>	<i>Update user - invalid</i>	<i>Edit</i> tanpa mengisi <i>field</i> wajib terisi	Validasi gagal	✓
8	<i>Master User</i>	<i>Delete user</i>	Klik <i>delete</i> , konfirmasi <i>delete user</i>	Data hilang	✓
9	<i>Master User</i>	<i>Delete user - invalid</i>	Data <i>user</i> tidak ada	Data tidak ditemukan	✓
10	<i>Master Location</i>	<i>Create location</i>	Klik ' <i>Add Data</i> ', isi form, simpan <i>location</i> baru	Data tersimpan	✓
11	<i>Master Location</i>	<i>Create location - invalid</i>	Isi form kosong, klik simpan	Pesan error wajib diisi	✓
12	<i>Master Location</i>	<i>Read location</i>	Lihat daftar <i>location</i>	Data tampil	✓
13	<i>Master Location</i>	<i>Update location</i>	Klik <i>edit</i> , ubah data, simpan data <i>location</i>	Data diperbarui	✓

14	<i>Master Location</i>	Update <i>location - invalid</i>	<i>Edit</i> tanpa mengisi <i>field</i> wajib terisi	Validasi gagal	✓
15	<i>Master Location</i>	Delete <i>location</i>	Klik <i>delete</i> , konfirmasi <i>delete location</i>	Data hilang	✓
16	<i>Master Location</i>	Delete <i>location - invalid</i>	Data <i>location</i> tidak ada	Data tidak ditemukan	✓
17	<i>Master Sparepart</i>	Create <i>sparepart</i>	Klik ' <i>Add Data</i> ', isi form, simpan <i>sparepart</i> baru	Data tersimpan	✓
18	<i>Master Sparepart</i>	Create <i>sparepart - invalid</i>	Isi form kosong, klik simpan	Pesan error wajib isi	✓
19	<i>Master Sparepart</i>	Read <i>sparepart</i>	Lihat daftar <i>sparepart</i>	Data tampil	✓
20	<i>Master Sparepart</i>	Update <i>sparepart</i>	Klik <i>edit</i> , ubah data, simpan data <i>sparepart</i>	Data diperbarui	✓
21	<i>Master Sparepart</i>	Update <i>sparepart - invalid</i>	<i>Edit</i> tanpa mengisi <i>field</i> wajib terisi	Validasi gagal	✓
22	<i>Master Sparepart</i>	Delete <i>sparepart</i>	Klik <i>delete</i> , konfirmasi <i>delete sparepart</i>	Data hilang	✓
23	<i>Master Sparepart</i>	Delete <i>sparepart - invalid</i>	Data <i>sparepart</i> tidak ada	Data tidak ditemukan	✓
24	<i>Master Vehicle</i>	Create <i>vehicle</i>	Klik ' <i>Add Data</i> ', isi form, simpan kendaraan baru	Data tersimpan	✓

25	<i>Master Vehicle</i>	<i>Create vehicle - invalid</i>	Isi form kosong, klik simpan	Pesan error wajib isi	✓
26	<i>Master Vehicle</i>	<i>Read vehicle</i>	Lihat daftar kendaraan	Data tampil	✓
27	<i>Master Vehicle</i>	<i>Update vehicle</i>	Klik <i>edit</i> , ubah data, simpan data kendaraan	Data diperbarui	✓
28	<i>Master Vehicle</i>	<i>Update vehicle - invalid</i>	<i>Edit</i> tanpa mengisi <i>field</i> wajib terisi	Validasi gagal	✓
29	<i>Master Vehicle</i>	<i>Delete vehicle</i>	Klik <i>delete</i> , konfirmasi <i>delete</i> kendaraan	Data hilang	✓
30	<i>Master Vehicle</i>	<i>Delete vehicle - invalid</i>	Data kendaraan tidak ada	Data tidak ditemukan	✓
31	<i>Sparepart Incoming</i>	<i>Create incoming</i>	Klik ' <i>Add Data</i> ', isi form, simpan <i>sparepart</i> masuk baru	Data tampil dan stok bertambah	✓
32	<i>Sparepart Incoming</i>	<i>Create incoming - invalid</i>	Isi form kosong, klik simpan	Pesan error wajib isi	✓
33	<i>Sparepart Incoming</i>	<i>Read incoming</i>	Lihat data <i>sparepart</i> masuk	Data tampil	✓
34	<i>Sparepart Incoming</i>	<i>Update incoming</i>	Klik <i>edit</i> , ubah data, simpan data <i>sparepart</i> masuk	Data diperbarui	✓
35	<i>Sparepart Incoming</i>	<i>Update incoming - invalid</i>	<i>Edit</i> tanpa mengisi <i>field</i> wajib terisi	Validasi gagal	✓

36	<i>Sparepart Incoming</i>	<i>Delete incoming</i>	Klik <i>delete</i> , konfirmasi <i>delete sparepart</i> masuk	Data hilang dan stok berkurang	✓
37	<i>Sparepart Incoming</i>	<i>Delete incoming - invalid</i>	Data <i>sparepart</i> masuk tidak ada	Data tidak ditemukan	✓
38	<i>Sparepart Out</i>	<i>Create outcoming</i>	Klik ' <i>Add Data</i> ', isi form, simpan <i>sparepart</i> keluar baru	Data tersimpan	✓
39	<i>Sparepart Out</i>	<i>Create outcoming - invalid</i>	Isi form kosong, klik simpan	Pesan error wajib isi	✓
40	<i>Sparepart Out</i>	<i>Read outcoming</i>	Lihat data <i>sparepart</i> keluar	Data tampil	✓
41	<i>Sparepart Out</i>	<i>Update outcoming</i>	Klik <i>edit</i> , ubah data, simpan data <i>sparepart</i> keluar	Data diperbarui	✓
42	<i>Sparepart Out</i>	<i>Update outcoming - invalid</i>	<i>Edit</i> tanpa mengisi <i>field</i> wajib terisi	Validasi gagal	✓
43	<i>Sparepart Out</i>	<i>Delete outcoming</i>	Klik <i>delete</i> , konfirmasi <i>delete sparepart</i> keluar	Data hilang dan stok bertambah	✓
44	<i>Sparepart Out</i>	<i>Delete outcoming - invalid</i>	Data <i>sparepart</i> keluar tidak ada	Data tidak ditemukan	✓
45	<i>Report Movement</i>	<i>Generate report</i>	Pilih tanggal, <i>sparepart</i> , dan lokasi	Data pergerakan masuk dan keluar <i>sparepart</i> tampil	✓

46	<i>Report Movement</i>	<i>Export report</i>	Klik <i>Export Excel</i>	File diunduh	✓
47	<i>Report in Vehicle</i>	<i>View report</i>	Pilih lokasi, <i>sparepart</i> dan kendaraan	Tampil <i>sparepart</i> terpasang di kendaraan	✓
48	<i>Report in Vehicle</i>	<i>Export</i>	Klik <i>Export</i>	File diunduh	✓
49	<i>Report Availability</i>	Lihat ketersediaan	Pilih lokasi dan <i>sparepart</i>	Tampil data stok <i>sparepart</i> terkini	✓
50	<i>Report Availability</i>	<i>Export</i>	Klik <i>Export Excel</i>	File diunduh	✓