

Perancangan *Unified Modelling Language* Sistem Informasi Surat Jalan dan Lembaran Permintaan Perbaikan Berbasis Website

Teofilus Sunarsa¹⁾

Teknologi Informasi, Politeknik Gajah Tunggal
teofilus@poltek-gt.ac.id

M. Ridwan Arif Cahyono²⁾

Teknik Elektronika, Politeknik Gajah Tunggal
ridwan@poltek-gt.ac.id

Alwan Maulana³⁾

Teknik Elektronika, Politeknik Gajah Tunggal
alwanmaulana25@gmail.com

ABSTRAK

PT Ban Indonesia menghadapi tantangan dalam pengelolaan surat jalan dan Lembaran Permintaan Perbaikan (LPP) di gudang spare part, yang penting untuk kelancaran operasional mesin produksi. Proses pengelolaan yang berulang dan tidak terintegrasi menyebabkan ketidakefisienan dan potensi kesalahan. Penelitian ini merancang Unified Modelling Language untuk sistem informasi berbasis website. Perancangan menggunakan empat jenis diagram UML, yaitu use case diagram, activity diagram, sequence diagram, dan class diagram. Alat yang digunakan mencakup laptop serta dukungan software draw.io untuk pembuatan diagram UML. Hasil penelitian menunjukkan bahwa UML yang dirancang mampu memudahkan dalam perancangan sistem informasi surat jalan dan Lembaran Permintaan Perbaikan berbasis website. Use case diagram digunakan untuk menggambarkan fungsionalitas sistem dan interaksi antara aktor dan sistem. Activity diagram mengilustrasikan konsep aliran data atau kontrol, serta aksi-aksi terstruktur yang didesain dengan baik dalam suatu sistem. Sequence diagram menggambarkan kolaborasi antara objek-objek yang berinteraksi di dalam elemen-elemen suatu kelas. Class diagram menunjukkan hubungan antar kelas, yang mencakup atribut dan fungsi dari suatu objek.

Kata Kunci : *UML*, Monitoring dan Kontrol, Surat Jalan, Lembaran Permintaan Perbaikan

ABSTRACT

PT Ban Indonesia faces challenges in managing surat jalan and Lembaran Permintaan Perbaikan (LPP) in the spare parts warehouse, which are crucial for the smooth operation of production machinery. The repetitive and non-integrated management processes result in inefficiency and potential errors. This study designs a Unified Modelling Language (UML) for a web-based information system. The design uses four types of UML diagrams: use case diagram, activity diagram, sequence diagram, and class diagram. Tools used include a laptop and the draw.io software for UML diagram creation. The study's findings indicate that the designed UML facilitates the development of a web-based delivery order and Repair Request Sheets information system. The use case diagram is used to describe the system's functionality and the interaction between actors and the system. The activity diagram illustrates the concept of data or control flow, as well as well-structured actions designed within a system. The sequence diagram depicts the collaboration between objects interacting within elements of a class. The class diagram shows the relationships between classes, including the attributes and functions of an object.

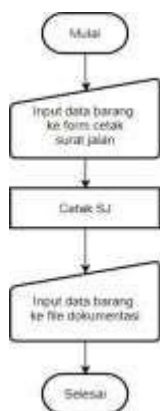
Keywords: UML, Monitoring and Control, Surat Jalan, Lembaran Permintaan Perbaikan

I. PENDAHULUAN

Latar Belakang

PT Ban Indonesia dikenal sebagai salah satu produsen ban terbesar di Asia Tenggara yang menghasilkan beragam jenis ban, mulai dari mobil penumpang, truk, offroad, industri, hingga sepeda motor. Pada struktur organisasinya, PT Ban Indonesia memiliki beberapa departemen yang bertanggung jawab atas berbagai aspek operasional perusahaan dan juga aset yang memiliki nilai tinggi sehingga memerlukan manajemen dan pemeliharaan yang efisien dan efektif, salah satunya adalah departemen Engineering. Departemen yang memiliki tanggung jawab penting dalam menjaga aset perusahaan adalah gudang spare part yang termasuk bagian dari departemen Engineering. Gudang spare part memiliki peran penting dalam menyimpan berbagai macam komponen dan suku cadang sesuai permintaan dari user dengan tujuan untuk menjaga kelancaran operasional mesin produksi pada perusahaan. Gudang spare part memiliki tugas seperti penerimaan barang (Receiving), pemeliharaan barang (House Keeping), perhitungan persediaan barang (Stock Opname), Melayani Permintaan Barang (Service), Sistem administrasi serta tugas lainnya. Tugas bagian sistem administrasi diantaranya yaitu membuat surat permintaan barang (Purchase Requisition), surat jalan, dan lembaran permintaan perbaikan.

Selain itu, bagian sistem administrasi juga bertugas mengelola surat jalan dan Lembaran Permintaan Perbaikan (LPP) yang jumlahnya tidak sedikit. Surat jalan dan LPP merupakan dokumen penting sebagai syarat suatu barang milik perusahaan bisa meninggalkan perusahaan. Pengelolaan surat jalan dan LPP dimulai dari pembuatan dokumen tersebut. Berikut adalah alur pembuatan surat jalan serta data surat jalan beserta statusnya sejak Januari hingga Desember tahun 2023:



Gambar 1. Alur Pembuatan Surat Jalan

Tabel 1. Data Penggunaan Surat Jalan Januari-Desember 2023

Bulan	Total Surat Jalan	Status Surat	
		Close	Open
Januari	195	160	35
Februari	187	145	42
Maret	196	150	46
April	188	140	48
Mei	198	160	38
Juni	195	155	40
Juli	197	165	32
Agustus	196	148	48
September	196	152	44
Oktober	197	153	44
November	196	142	54
Desember	196	138	58
Total	2.343	1.808	535

Berdasarkan gambar dan tabel, alur dan banyaknya penggunaan surat jalan menunjukkan bahwa pengelolaan surat jalan akan cukup memakan waktu. Tabel juga menunjukkan masih banyak surat jalan yang berstatus "open". Status "open" menandakan bahwa barang milik perusahaan yang keluar dari lingkungan perusahaan belum kembali, sedangkan status "close" menandakan bahwa barang telah kembali. Hal ini mengindikasikan adanya potensi kerugian atas barang yang belum dikembalikan atau diperbaiki, yang dapat mengganggu pekerjaan pengguna yang biasanya menggunakan barang tersebut untuk proses perbaikan atau pemeliharaan mesin. Jika terhambat, proses tersebut dapat mengganggu kinerja mesin, bahkan menyebabkan mesin berhenti. Oleh karena itu, perlu dilakukan proses monitoring pada surat jalan dan LPP. Jumlah penggunaan surat jalan dan LPP yang banyak serta memiliki kategori yang berbeda tentu akan menambah kesulitan dalam proses monitoring. Berikut adalah data status surat jalan yang berstatus "open" berdasarkan kategorinya:

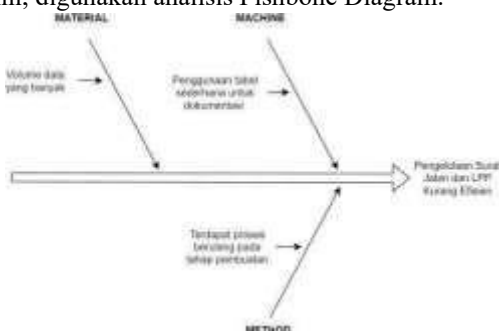
Tabel 2. Data Surat Jalan Open Berdasarkan Kategori

No	Kategori	Jumlah
1	Barang Ditolak	33
2	Barang Vendor	3
3	Claim	119

Tabel 2. Lanjutan

No	Kategori	Jumlah
4	Kalibrasi	10
5	Modifikasi	31
6	Perlengkapan Repair	33
7	Refill Gas	5
8	Rekondisi	9
9	Repair	201
10	Sampel User	91
Total		535

Berdasarkan data, surat jalan juga dibedakan berdasarkan kategori keperluannya. Selain banyaknya penggunaan surat jalan dan LPP, banyaknya kategori ini pasti akan memakan waktu lebih lama dalam proses monitoring. Hal ini membuat pengelolaan surat jalan dan LPP, yang dimulai dari pembuatan dokumen hingga proses monitoring, menjadi kurang efisien. Oleh karena itu, diperlukan tindakan lebih lanjut untuk mengatasi masalah ini. Untuk menganalisis penyebab masalah ini, digunakan analisis Fishbone Diagram.



Gambar 2. . Fishbone Diagram

Melalui analisis ini, diidentifikasi bahwa penyebab masalah pengelolaan surat jalan dan lpp yang kurang efisien adalah adanya proses yang berulang pada tahap pembuatan, yaitu penginputan informasi barang. Hal ini menyebabkan penambahan waktu pada proses pembuatan surat jalan. Proses input data untuk keperluan dokumentasi juga berpotensi menyebabkan terjadinya perbedaan data, karena data yang diinput ke file dokumentasi tidak terintegrasi dengan data yang diinput pada form cetak.

Selain itu, proses dokumentasi menggunakan tabel sederhana di Excel, menyulitkan untuk memonitor surat jalan dan LPP. Misalnya, dibutuhkan waktu tambahan untuk mengetahui mana surat jalan yang memiliki durasi open terlalu lama atau mengetahui vendor yang memiliki surat jalan berstatus open terbanyak.

Guna mengatasi permasalahan tersebut, diperlukan sistem informasi yang dapat memudahkan monitoring dan pengendalian terhadap surat jalan keluar dan LPP di gudang spare part PT Ban Indonesia. Perancangan sistem informasi dengan memanfaatkan teknologi berbasis website dapat menjadi solusi yang efektif dalam meningkatkan efisiensi pengelolaan surat jalan keluar dan LPP di Gudang Spare Part. UML dirancang untuk membantu dalam menggambarkan dan merancang struktur serta fungsi sistem secara menyeluruh.

Unified Modeling Language (UML) adalah sebuah bahasa yang digunakan untuk merinci, menggambarkan, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak. Artefak ini dapat berupa model, deskripsi, atau perangkat lunak itu sendiri. UML digunakan dalam pemodelan bisnis dan sistem non-perangkat lunak lainnya. Selain itu, UML juga merupakan bahasa pemodelan yang menerapkan konsep orientasi objek [1].

Dalam pemodelan perangkat lunak, terdapat 14 jenis diagram yang tersedia, namun dalam implementasinya, ada 4 jenis diagram yang paling umum digunakan. Keempat jenis diagram tersebut adalah Use Case Diagram, Activity Diagram, Sequence Diagram, dan Class Diagram. Secara umum, UML telah menjadi standar dalam pemodelan perangkat lunak, yang memungkinkan para pengembang untuk menggambarkan desain sistem perangkat lunak melalui berbagai jenis diagram [2].

Use case diagram adalah diagram yang menunjukkan hubungan antara aktor-aktor dan use case-use case dalam suatu sistem. Diagram ini digunakan untuk analisis dan desain sistem. Berikut adalah simbol pada use case diagram[3].

Activity diagram adalah diagram yang mengilustrasikan konsep aliran data atau kontrol, serta aksi-aksi terstruktur yang didesain dengan baik dalam suatu sistem.

Sequence diagram adalah diagram yang menggambarkan kolaborasi antara objek-objek yang berinteraksi di dalam elemen-elemen suatu kelas. Berikut ini adalah komponen-komponen dalam sequence diagram:

1. *Activations*: Menjelaskan eksekusi fungsi yang dimiliki oleh suatu objek.
2. *Actor*: Menjelaskan peran yang melakukan serangkaian aksi dalam suatu proses.
3. *Collaboration boundary*: Menjelaskan tempat untuk lingkungan percobaan dan digunakan untuk memonitor objek.
4. *Parallel vertical lines*: Menjelaskan garis proses yang menunjuk pada suatu status.
5. *Processes*: Menjelaskan tindakan atau aksi yang dilakukan oleh aktor dalam suatu waktu.

6. *Window*: Menjelaskan halaman yang sedang ditampilkan dalam suatu proses.
7. *Loop*: Menjelaskan model logika yang berpotensi diulang beberapa kali.

Class diagram adalah diagram yang menunjukkan hubungan antar kelas, yang mencakup atribut dan fungsi dari suatu objek. *Class* diagram memiliki tiga jenis relasi dalam penggunaannya, yaitu:

1. *Assosiation*: Hubungan yang menunjukkan interaksi antara kelas. Hubungan ini digambarkan dengan garis yang memiliki mata panah terbuka di ujungnya, menunjukkan aliran pesan dalam satu arah.
2. *Generalization*: Hubungan antara kelas yang bersifat dari khusus ke umum.
3. *Constraint*: Hubungan yang digunakan dalam sistem untuk memberikan batasan, sehingga didapatkan aspek yang tidak fungsional [4].

II. METODE PENELITIAN

Alur Penelitian

Alur penelitian merupakan tahapan yang dilakukan untuk merealisasikan serta menjelaskan penelitian yang sedang dilakukan agar terstruktur dengan baik. Tahapan penelitian dapat dilihat pada gambar 4.



Gambar 3. Alur Penelitian

Pengumpulan Data

Selama tahap ini, dilakukan observasi langsung di gudang spare part PT. Ban Indonesia untuk memahami secara detail proses pengelolaan surat jalan dan LPP yang telah ada. Wawancara juga dilakukan dengan staf gudang spare part dan pengguna sistem untuk mendapatkan wawasan yang lebih dalam mengenai kebutuhan dan kendala yang dihadapi dalam proses tersebut. Data yang dikumpulkan dari observasi dan wawancara akan menjadi dasar untuk merancang sistem yang tepat dan efektif.

Persiapan Alat dan Bahan

Pada tahap ini, dipilih alat dan bahan yang diperlukan berdasarkan konsep yang telah disusun sebelumnya. Pemilihan alat dan bahan dilakukan

dengan cermat untuk memastikan kelancaran pelaksanaan penelitian. Adapun alat yang diperlukan yaitu sebuah Laptop serta dukungan dari software Draw.io untuk membuat berbagai jenis diagram UML seperti Use Case Diagram, Activity Diagram, Sequence Diagram, dan Class Diagram.

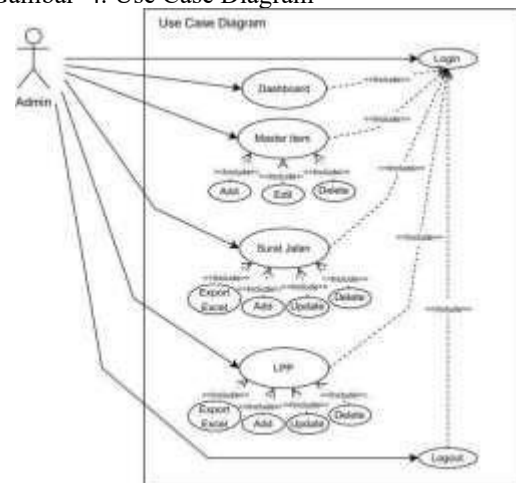
Perancangan UML

UML (Unified Modeling Language) adalah sebuah bahasa standar yang digunakan untuk memodelkan, merancang, dan mendokumentasikan sistem perangkat lunak secara visual. Dalam merancang UML yang sesuai, digunakan empat jenis diagram, yaitu use case diagram, activity diagram, sequence diagram, dan class diagram. Pemilihan diagram ini dilakukan karena mampu memvisualisasikan model aplikasi secara rinci dan mendukung kelancaran proses pengembangan aplikasi dengan menggambarkan alur aktivitas dan tindakan dalam sistem secara lengkap.

III. HASIL DAN PEMBAHASAN

Use Case Diagram

Berikut Use Case Diagram untuk sistem informasi yang dirancang yang dapat dilihat pada Gambar 4. Use Case Diagram



Gambar 4. Use Case Diagram

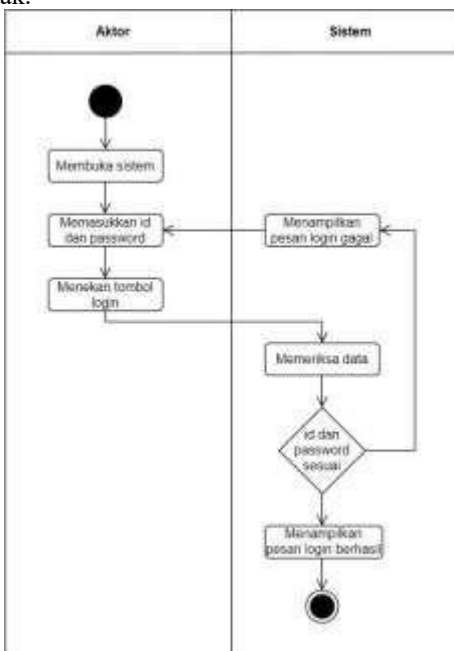
Berdasarkan Gambar 4. Use Case Diagram, dalam diagram tersebut, terdapat aktor yang berperan sebagai admin. Fungsi atau fitur yang dapat dilakukan oleh admin antara lain: melihat dashboard; melihat, menambah, mengubah, dan menghapus data master item; melihat dan mengekspor data surat jalan ke dalam bentuk Excel; membuat, mengubah, dan menghapus surat jalan; melihat dan mengekspor data LPP ke dalam bentuk Excel; serta membuat, mengubah, dan menghapus LPP. Selain itu, admin juga dapat melakukan logout dari sistem. Sebelumnya admin harus terlebih dahulu melakukan login menggunakan akun untuk mengakses fungsi-fungsi yang ada dalam sistem informasi. Proses login ini merupakan langkah awal untuk

memastikan bahwa hanya pengguna yang berwenang yang dapat melanjutkan ke tahap berikutnya dalam sistem.

Activity Diagram

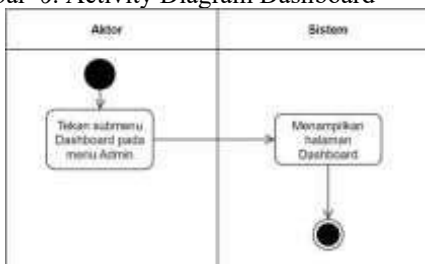
Fitur-fitur dalam sistem memiliki fungsi dan cara kerjanya masing-masing. Untuk memahami bagaimana cara kerja setiap fitur, dapat dilihat melalui Activity Diagram.

Pada halaman login, aktor memasukkan informasi yang diperlukan untuk login, seperti email dan kata sandi (password). Selanjutnya sistem menerima data login dari aktor dan memeriksa informasi tersebut. Ini mencakup pengecekan apakah username dan password sesuai dengan data yang ada dalam sistem, serta menentukan apakah aktor dapat meneruskan ke halaman berikutnya atau tidak.

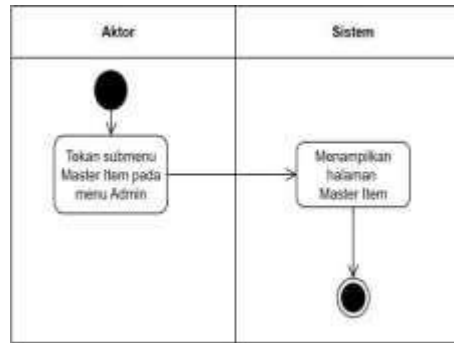


Gambar 5. Activity Diagram Login

Pada halaman Dashboard, tidak terdapat banyak aktivitas karena halaman ini hanya menampilkan informasi yang diambil dari database. Hal ini dapat dilihat pada activity diagram di Gambar 6. Activity Diagram Dashboard

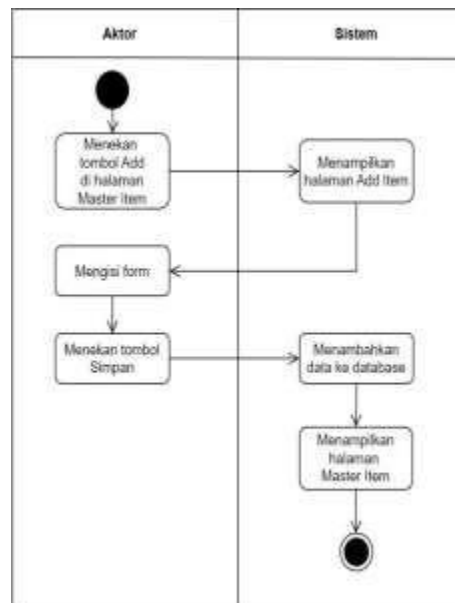


Gambar 6. Activity Diagram Dashboard



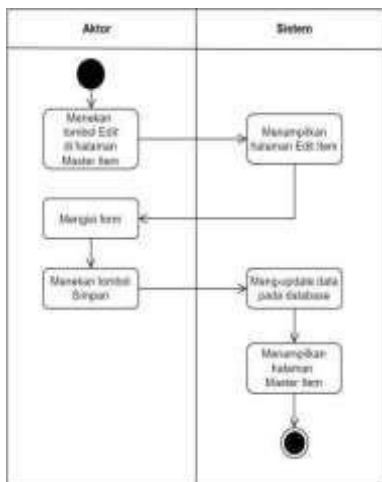
Gambar 7. Activity Diagram Master Item

Berdasarkan Gambar 7. Activity Diagram Master Item, untuk mengakses halaman master item, aktor perlu terlebih dahulu menekan submenu Master Item yang terdapat pada submenu admin. Selanjutnya, sistem akan menampilkan halaman Master Item.



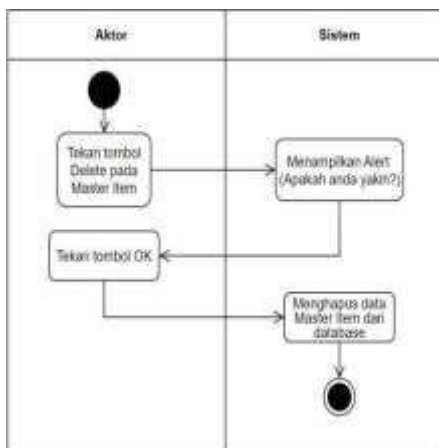
Gambar 8. Activity Diagram Add Master Item

Berdasarkan activity diagram pada , aktor dapat menggunakan fitur tambah master item dengan menekan tombol Add yang ada pada halaman Master Item. Setelah itu, sistem akan menampilkan halaman Add Item. Selanjutnya, aktor mengisi form dan menekan tombol Simpan. Sistem kemudian akan menambahkan data yang diisi aktor ke dalam database, dan setelah itu, sistem akan mengembalikan halaman ke Master Item. Saat halaman Master Item kembali tampil data yang digunakan actor untuk mengisi form sudah masuk ke database dan dapat dilihat pada halaman Master Item.



Gambar 9. Activity Diagram Add Master Item

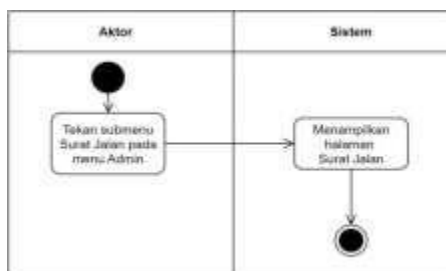
Berdasarkan activity diagram pada Gambar 9. Activity Diagram Add Master Item, cara menggunakan fitur edit master item dimulai dengan aktor menekan tombol Edit pada item yang ingin diedit. Selanjutnya, sistem akan menampilkan halaman Edit Item. Aktor kemudian mengisi form sesuai dengan perubahan yang ingin dilakukan. Setelah selesai, aktor menekan tombol Simpan. Sistem akan meng-update data item pada database dan kemudian menampilkan kembali halaman Master Item dengan perubahan yang telah dilakukan.



Gambar 10. Activity Diagram Delete Master Item

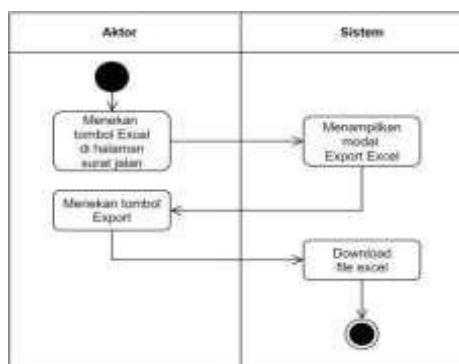
Berdasarkan activity diagram pada Gambar 10. Activity Diagram Delete Master Item, cara menggunakan fitur delete master item dimulai dengan aktor menekan tombol Delete pada item yang ingin dihapus di halaman Master Item. Selanjutnya, sistem akan menampilkan alert dengan pesan "Apakah Anda yakin?". Setelah aktor menekan tombol OK, sistem akan menghapus item yang dipilih aktor dari database.

Terdapat beberapa fitur yang dapat dilakukan oleh aktor pada halaman Surat Jalan, untuk mengaksesnya, aktor perlu menuju halaman Surat Jalan.



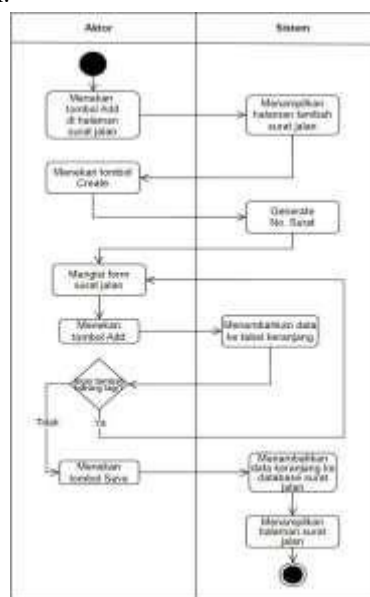
Gambar 11. Activity Diagram Surat Jalan

Berdasarkan activity diagram pada Gambar 11. Activity Diagram Surat Jalan, untuk menuju halaman Surat Jalan actor perlu menekan submenu surat jalan, lalu halaman surat jalan akan ditampilkan oleh sistem.



Gambar 12. Activity Diagram Export Excel Surat Jalan

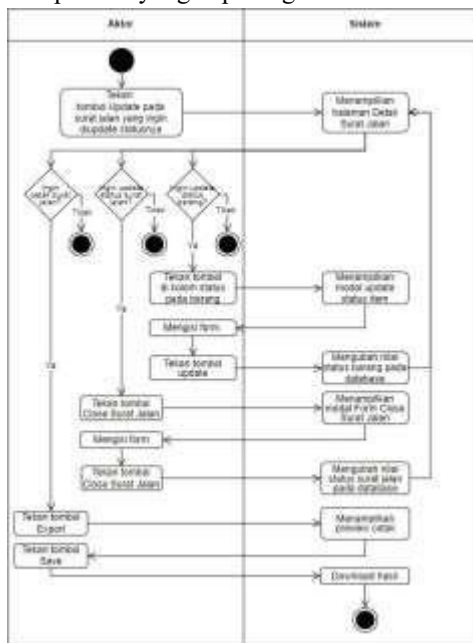
Berdasarkan activity diagram pada Gambar 12. Activity Diagram Export Excel Surat Jalan, tidak banyak langkah yang diperlukan untuk mengekspor data. Pada langkah terakhir, file akan terunduh saat aktor menekan tombol "Export" pada modal yang muncul.



Gambar 13. Activity Diagram Add Surat Jalan

Berdasarkan activity diagram pada Gambar 13. Activity Diagram Add Surat Jalan, aktor dapat membuat surat jalan yang berisi lebih dari satu jenis barang atau item yang berbeda. Proses ini dimulai dengan menekan tombol "Create" terlebih dahulu agar sistem dapat menghasilkan nomor surat jalan sebagai identitas surat jalan yang akan dibuat. Setelah itu, aktor mengisi data surat jalan yang diperlukan pada form dan menekan tombol Add. Sistem akan menambahkan data yang sudah diisi ke dalam tabel keranjang. Selanjutnya, aktor dapat memilih apakah ingin menambahkan barang lagi atau tidak. Jika masih ingin menambahkan barang, aktor mengisi kembali data pada form dengan nomor surat yang sama lalu menekan tombol Add kembali untuk menambahkan data ke dalam tabel keranjang. Jika sudah tidak ingin menambahkan barang lagi, aktor menekan tombol Save. Sistem kemudian akan menambahkan data yang ada pada tabel keranjang ke dalam database, dan setelah itu, sistem akan membawa aktor kembali ke halaman Surat Jalan.

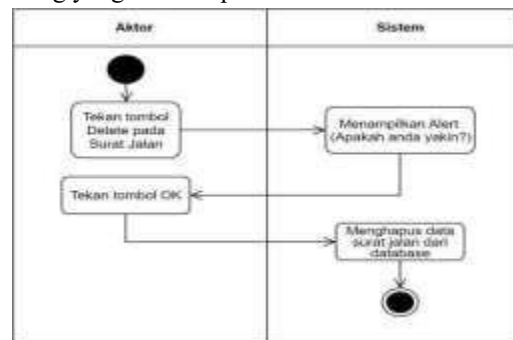
Pada halaman Detail Surat Jalan terdapat beberapa fitur yang dapat digunakan aktor.



Gambar 14. Activity Diagram Update Status Surat Jalan

Pada activity diagram pada Gambar 14. Activity Diagram Update Status Surat Jalan, aktor terlebih dahulu perlu menekan tombol Update. Selanjutnya, sistem akan menampilkan halaman detail surat jalan. Di halaman ini, aktor dapat menggunakan beberapa fitur, yaitu cetak surat jalan, update status surat jalan, dan update status barang. Pada fitur cetak surat jalan, pertama-tama aktor menekan tombol Export. Sistem kemudian akan menampilkan preview cetak. Setelah itu, aktor menekan tombol Save, dan sistem akan mendownload file tersebut. Pada fitur update status surat jalan, aktor menekan tombol Close Surat Jalan.

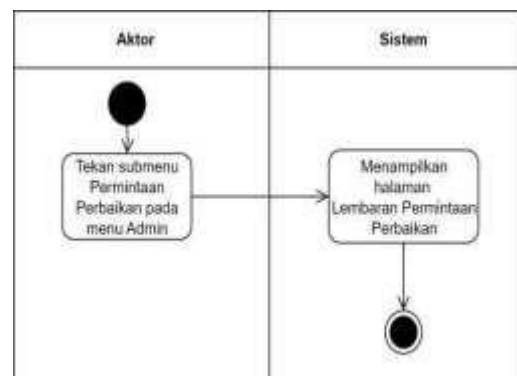
Selanjutnya, sistem akan menampilkan modal form Close Surat Jalan. Aktor mengisi form dan menekan tombol Close Surat Jalan. Sistem kemudian akan mengubah nilai status surat jalan pada database dan menampilkan kembali halaman detail surat jalan dengan status yang telah diperbarui. Pada fitur update status barang, aktor menekan tombol yang ada di kolom status pada barang yang ingin di-update. Sistem akan menampilkan modal update status item. Aktor mengisi form dan menekan tombol Update. Sistem kemudian akan mengubah nilai status barang pada database dan menampilkan kembali halaman detail surat jalan dengan status barang yang sudah diperbarui.



Gambar 15. Delete Surat Jalan

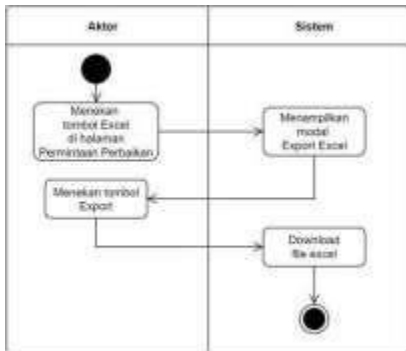
Berdasarkan activity diagram pada Gambar 15. Delete Surat Jalan, cara menggunakan fitur hapus surat jalan dimulai dengan aktor menekan tombol Hapus pada surat jalan yang akan dihapus. Sistem kemudian akan menampilkan alert dengan pesan "Apakah Anda yakin?". Setelah itu, aktor menekan tombol OK, dan sistem akan menghapus data surat jalan dari database.

Terdapat beberapa fitur yang dapat dilakukan oleh aktor pada halaman Permintaan Perbaikan, untuk mengaksesnya, aktor perlu menuju Permintaan Perbaikan.



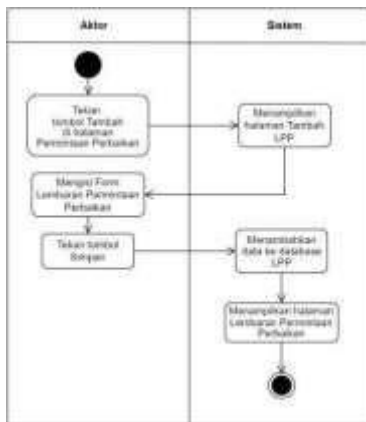
Gambar 16. Activity Diagram LPP

Berdasarkan activity diagram pada gambar 16, untuk menuju halaman Surat Jalan actor perlu menekan submenu Permintaan Perbaikan, lalu halaman Permintaan Perbaikan akan ditampilkan oleh sistem.



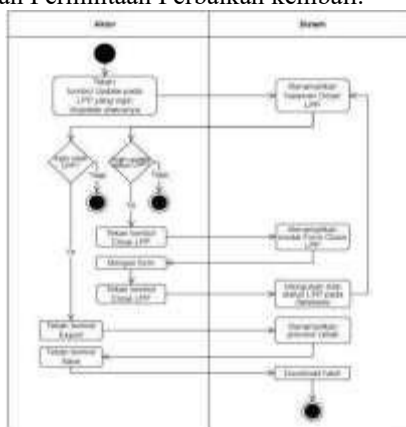
Gambar 17. Activity Diagram Export Excel LPP

Berdasarkan activity diagram pada Gambar 17. Activity Diagram Export Excel LPP, tidak banyak langkah yang diperlukan untuk mengekspor data. Pada langkah terakhir, file akan terunduh saat aktor menekan tombol "Export" pada modal yang muncul.



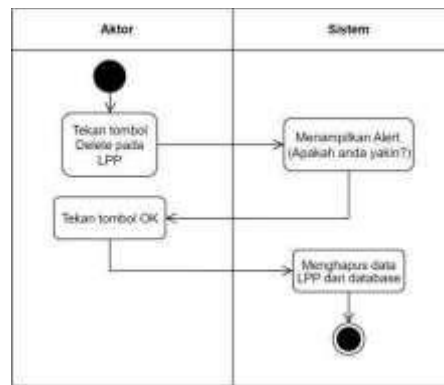
Gambar 18. Activity Diagram Tambah LPP

Berdasarkan activity diagram pada Gambar 18. Activity Diagram Tambah LPP, untuk menggunakan fitur tambah LPP, aktor pertama-tama menekan tombol tambah pada halaman Permintaan Perbaikan. Sistem kemudian akan menampilkan halaman tambah LPP, di mana aktor mengisi form lembaran permintaan perbaikan. Setelah mengisi form, aktor menekan tombol Simpan. Sistem akan menambahkan data tersebut ke dalam database, dan selanjutnya menampilkan halaman Permintaan Perbaikan kembali.



Gambar 19. Activity Diagram Update LPP

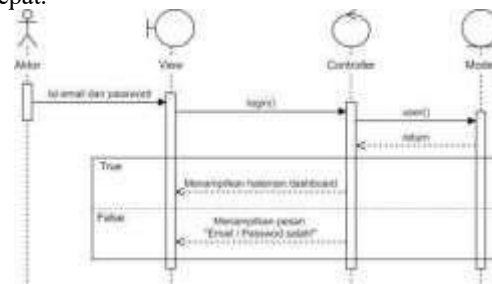
Berdasarkan activity diagram pada Gambar 19. Activity Diagram Update LPP, aktor dapat menggunakan dua fitur pada LPP, yaitu cetak dan update status. Untuk fitur update status LPP, aktor pertama-tama menekan tombol Update pada LPP yang ingin di-update. Setelah itu, halaman Detail LPP akan ditampilkan. Di halaman ini, aktor menekan tombol Close LPP untuk memulai proses update status. Modal form akan muncul, di mana aktor mengisi form yang diperlukan dan menekan tombol Close LPP. Sistem akan mengubah nilai status pada database dan kemudian menampilkan kembali halaman Detail LPP dengan status yang telah diperbarui. Selanjutnya, untuk fitur cetak LPP, aktor menekan tombol Export. Sistem akan mendownload hasil cetak.



Gambar 20. Activity Diagram Delete LPP

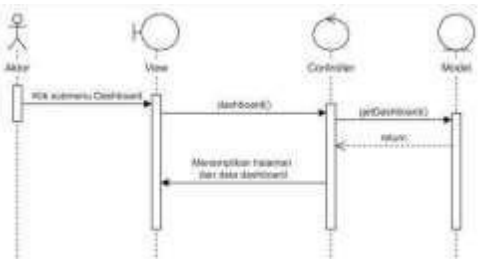
Sequence Diagram

Berikut sequence diagram yang menggambarkan interaksi antar objek dalam sistem yang dirancang, disini objeknya berupa aktor, view, controller, dan model. Aktor, dalam konteks ini, mewakili pengguna atau entitas eksternal yang berinteraksi dengan sistem. View merujuk pada bagian dari sistem yang bertanggung jawab untuk menampilkan informasi kepada pengguna dan menerima input dari mereka. Controller mengatur alur kontrol dan koordinasi antara view dan model, mengontrol bagaimana data diolah dan diperbarui. Model merepresentasikan struktur data dan mengelola data, serta berkomunikasi dengan database. Sequence diagram membantu dalam memvisualisasikan bagaimana aktor berinteraksi dengan view untuk memicu aksi, bagaimana controller mengatur proses, dan bagaimana model memproses data untuk menghasilkan output yang tepat.



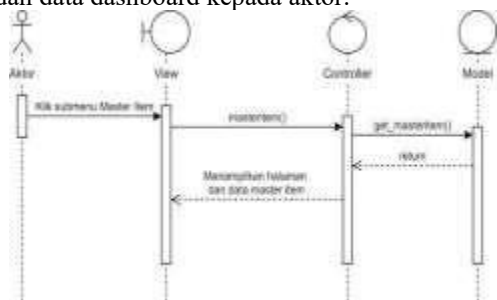
Gambar 21. Sequence Diagram Login

Berdasarkan sequence diagram pada Gambar 21. Sequence Diagram Login, Aktor, yang merupakan pengguna, memulai dengan memasukkan email dan password. Input ini dikirim ke View, yang kemudian memanggil fungsi "login()" pada Controller. Controller berinteraksi dengan Model dengan memanggil "user()" untuk dilakukan verifikasi. Jika hasil verifikasi benar (True), Controller mengirimkan sinyal kembali ke View dan menampilkan "halaman dashboard". Jika hasil verifikasi salah (False), akan ditampilkan pesan kesalahan "Email/Password salah!".



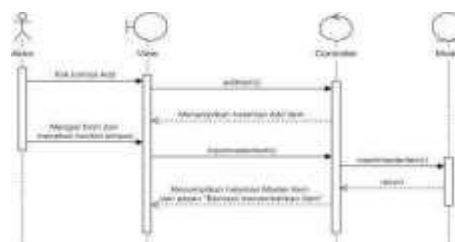
Gambar 22. Sequence Diagram Dashboard

Berdasarkan sequence diagram pada Gambar 22. Sequence Diagram Dashboard, proses interaksi untuk mengakses halaman dashboard dimulai dengan aktor yang mengklik submenu Dashboard. Tindakan ini dikirim ke View, yang kemudian memanggil fungsi dashboard() pada Controller. Controller, pada gilirannya, memanggil fungsi getDashboard() pada Model untuk mendapatkan data yang diperlukan. Setelah data diperoleh, Model mengirimkan kembali data tersebut ke Controller. Controller kemudian mengarahkan View untuk menampilkan halaman dan data dashboard kepada aktor.



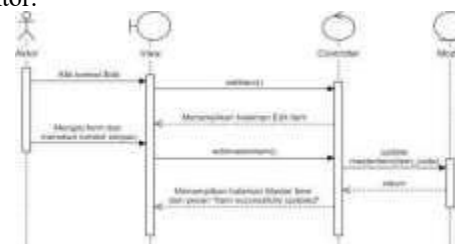
Gambar 23. Sequence Diagram Master Item

Berdasarkan sequence diagram pada Gambar 23. Sequence Diagram Master Item, proses interaksi untuk mengakses halaman Master Item dimulai dengan aktor yang mengklik submenu Master Item. Tindakan ini dikirim ke View, yang kemudian memanggil fungsi masteritem() pada Controller. Controller, pada gilirannya, memanggil fungsi get_masteritem() pada Model untuk mendapatkan data master item yang diperlukan. Setelah data diperoleh, Model mengirimkan kembali data tersebut ke Controller. Controller kemudian mengarahkan View untuk menampilkan halaman dan data master item kepada aktor.



Gambar 24. Sequence Diagram Add Master Item

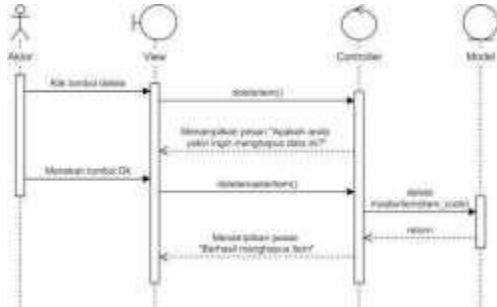
Berdasarkan sequence diagram pada Gambar 24. Sequence Diagram Add Master Item tersebut, proses interaksi untuk menggunakan fitur tambah master item dimulai dengan aktor yang mengklik tombol "Add" pada halaman Master Item. Tindakan ini dikirim ke View, yang kemudian memanggil fungsi additem() pada Controller. Controller merespons dengan menampilkan halaman Add Item kepada aktor melalui View. Selanjutnya, aktor mengisi form yang tersedia pada halaman Add Item dan menekan tombol "Simpan". Tindakan ini dikirim kembali ke View, yang kemudian memanggil fungsi inputmasteritem() pada Controller. Controller melanjutkan dengan memanggil fungsi insertmasteritem() pada Model untuk memasukkan data item baru ke dalam database. Setelah data berhasil dimasukkan, Model mengirimkan respons kembali ke Controller. Controller kemudian mengarahkan View untuk menampilkan halaman Master Item beserta pesan konfirmasi "Berhasil menambahkan Item" kepada aktor.



Gambar 25. Sequence Diagram Edit Master Item

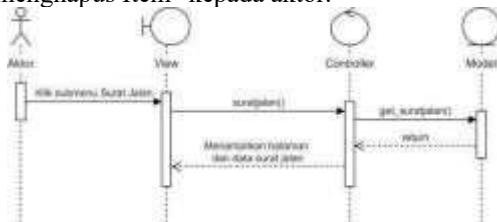
Berdasarkan sequence diagram pada Gambar 25. Sequence Diagram Edit Master Item, proses interaksi untuk menggunakan fitur edit master item dimulai dengan aktor yang mengklik tombol "Edit" pada halaman Master Item. Tindakan ini dikirim ke View, yang kemudian memanggil fungsi edititem() pada Controller. Controller merespons dengan menampilkan halaman Edit Item kepada aktor melalui View. Selanjutnya, aktor mengisi form yang tersedia pada halaman Edit Item dan menekan tombol "Simpan". Tindakan ini dikirim kembali ke View, yang kemudian memanggil fungsi editmasteritem() pada Controller. Controller melanjutkan dengan memanggil fungsi updatemasteritem(item_code) pada Model untuk memperbarui data item dalam database. Setelah data berhasil diperbarui, Model mengirimkan respons kembali ke Controller.

Controller kemudian mengarahkan View untuk menampilkan halaman Master Item beserta pesan konfirmasi "Item successfully updated" kepada aktor.



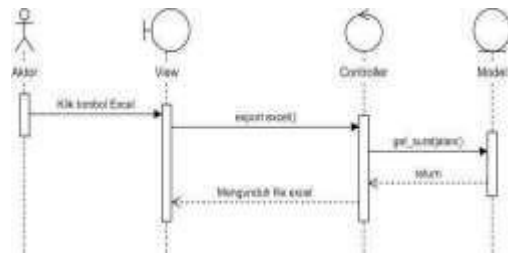
Gambar 26. Sequence Diagram Delete Master Item

Berdasarkan *sequence* diagram pada Gambar 26. Sequence Diagram Delete Master Item, proses interaksi untuk menggunakan fitur delete master item dimulai dengan aktor yang mengklik tombol "Delete" pada halaman *Master Item*. Tindakan ini dikirim ke View, yang kemudian memanggil fungsi `deleteitem()` pada Controller. Controller merespons dengan menampilkan pesan konfirmasi kepada aktor melalui View, yang berbunyi "Apakah anda yakin ingin menghapus data ini?". Selanjutnya, aktor menekan tombol "OK" untuk mengonfirmasi penghapusan item. Tindakan ini dikirim kembali ke View, yang kemudian memanggil fungsi `deletemasteritem()` pada Controller. Controller melanjutkan dengan memanggil fungsi `deletemasteritem(item_code)` pada Model untuk menghapus data item dari database. Setelah data berhasil dihapus, Model mengirimkan respons kembali ke Controller. Controller kemudian mengarahkan View untuk menampilkan pesan konfirmasi "Berhasil menghapus Item" kepada aktor.



Gambar 27. Sequence Diagram Surat Jalan

Berdasarkan *sequence* diagram pada Gambar 27. Sequence Diagram Surat Jalan, proses interaksi untuk mengakses halaman surat jalan dimulai dengan aktor yang mengklik submenu "Surat Jalan". Tindakan ini dikirim ke View, yang kemudian memanggil fungsi `suratjalan()` pada Controller. Controller kemudian berinteraksi dengan Model dengan memanggil fungsi `get_suratjalan()` untuk mengambil data surat jalan dari database. Setelah data diperoleh, Model mengirimkan respons kembali ke Controller. Controller kemudian mengarahkan View untuk menampilkan halaman surat jalan beserta data yang telah diambil.



Gambar 28. Sequence Diagram Export Excel Surat Jalan

Berdasarkan *sequence* diagram pada Gambar 28. Sequence Diagram Export Excel Surat Jalan, proses interaksi untuk melakukan ekspor surat jalan ke excel dimulai dengan aktor mengklik tombol "Excel" pada antarmuka pengguna. Tindakan ini dikirim ke View, yang kemudian memanggil fungsi `exportexcel()` pada Controller. Controller berinteraksi dengan Model dengan memanggil fungsi `get_suratjalan()` untuk mengambil data surat jalan yang diperlukan dari database. Setelah data diperoleh, Model mengirimkan data tersebut kembali ke Controller. Controller kemudian mengarahkan View untuk memulai proses pengunduhan file Excel yang berisi data surat jalan. Hasil akhir dari proses ini adalah file Excel yang dapat diunduh oleh aktor, berisi data surat jalan yang telah diambil dari database, sehingga aktor dapat mengakses dan menggunakan data tersebut dalam format Excel.



Gambar 29. Sequence Diagram Add Surat Jalan

Sequence diagram pada Gambar 29. Sequence Diagram Add Surat Jalan untuk menambahkan surat jalan dimulai ketika aktor mengklik tombol "Add" pada antarmuka pengguna, yang kemudian meminta View untuk menjalankan fungsi `tambah_surat_jalan()`. Fungsi ini menampilkan halaman Form Surat Jalan kepada aktor. Selanjutnya, aktor mengklik tombol "Create" untuk memulai proses pembuatan surat jalan. View memanggil fungsi `generate_no_surat()` pada Controller, yang kemudian berinteraksi dengan Model untuk mendapatkan nomor surat terakhir melalui fungsi `get_nosurat_terakhir()`. Model mengembalikan nomor surat terakhir kepada Controller, yang kemudian menggunakan nomor ini untuk menghasilkan nomor surat baru dan mengisi nomor surat tersebut pada form. Aktor kemudian mengisi form dan menekan tombol "Add", yang menginstruksikan View untuk menambahkan data

ke tabel Detail Penerimaan melalui fungsi tambah data ketabel Detail Penerimaan().

Controller melakukan validasi input dan jika ada data yang tidak lengkap, View menampilkan pesan "Please fill out all the required fields" kepada aktor. Jika data lengkap, data ditambahkan ke tabel Detail Penerimaan. Setelah data ditambahkan, aktor menekan tombol "Save", yang meminta View untuk menjalankan fungsi tambah_surat_jalan(). Controller kemudian memanggil fungsi insertsuratjalan() pada Model untuk menyimpan data surat jalan baru ke database. Model mengembalikan hasil penyimpanan kepada Controller, yang kemudian menginstruksikan View untuk menampilkan halaman surat jalan beserta pesan "Berhasil menambahkan surat jalan" kepada aktor.

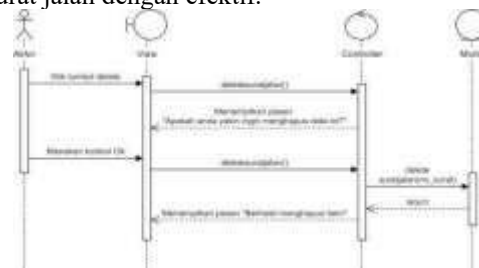


Gambar 30. Sequence Diagram Update Surat Jalan

Sequence diagram pada Gambar 30. Sequence Diagram Update Surat Jalan menggambarkan proses yang dilakukan oleh aktor untuk memperbarui status surat jalan, memperbarui status barang, serta mencetak atau mengekspor surat jalan. Proses dimulai ketika aktor mengklik tombol "Update", yang memicu View untuk memanggil fungsi `update_suratjalan()` pada Controller guna mendapatkan detail surat jalan dari Model melalui fungsi `get_detailsuratjalan()`. Data detail surat jalan dikembalikan oleh Model ke Controller, yang kemudian diteruskan ke View untuk menampilkan halaman Detail Surat Jalan.

Selanjutnya, untuk memperbarui status barang, aktor mengklik tombol pada status barang, yang memicu View untuk menampilkan modal update status barang dengan memanggil fungsi `modal update status barang()`. Aktor kemudian mengisi form di modal tersebut dan menekan tombol "Update", yang memanggil fungsi `update status barang()` pada Controller. Controller melakukan validasi input, dan jika valid, memanggil fungsi `update status suratjalan()` pada Model untuk memperbarui status barang. Model mengembalikan hasilnya ke Controller, yang

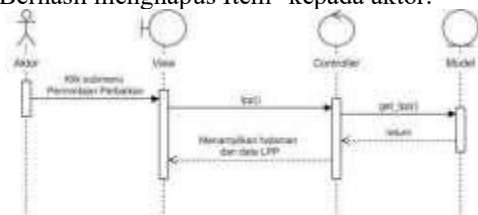
kemudian memerintahkan View untuk menampilkan pesan keberhasilan "Status barang berhasil diperbarui" atau pesan error jika ada input yang tidak valid. Untuk menutup surat jalan, aktor mengklik tombol "Close Surat Jalan" yang memicu View untuk menampilkan modal form close surat jalan dengan memanggil fungsi `tampil modal form close surat jalan()`. Aktor mengisi form di modal tersebut dan menekan tombol "Close Surat Jalan", yang memanggil fungsi `update status suratjalan()` pada Controller. Controller melakukan validasi input, dan jika valid, memanggil fungsi `update status suratjalan()` pada Model untuk memperbarui status surat jalan. Model mengembalikan hasilnya ke Controller, yang kemudian memerintahkan View untuk menampilkan pesan keberhasilan "Status surat jalan berhasil diperbarui" atau pesan error jika ada input yang tidak valid. Untuk mengekspor surat jalan, aktor mengklik tombol "Export", yang memicu View untuk memanggil fungsi `export suratjalan()` pada Controller. Controller memanggil fungsi `get detailsuratjalan()` pada Model untuk mendapatkan data surat jalan yang akan diekspor. Model mengembalikan data tersebut ke Controller, yang kemudian diteruskan ke View untuk menampilkan preview export. Aktor kemudian mengklik tombol "Save" untuk mengunduh file, dan View menjalankan fungsi `mengunduh file()` untuk menyelesaikan proses pengunduhan. Dengan mengikuti langkah-langkah dalam sequence diagram ini, aktor dapat memperbarui status surat jalan dan barang, serta mencetak atau mengekspor surat jalan dengan efektif.



Gambar 31. Sequence Diagram Delete Surat Jalan

Sequence diagram pada Gambar 31. Sequence Diagram Delete Surat Jalan menggambarkan proses penghapusan surat jalan yang dilakukan oleh aktor. Proses dimulai ketika aktor mengklik tombol "delete" pada antarmuka pengguna, yang kemudian meminta View untuk menjalankan fungsi `deletesuratjalan()`. Fungsi ini menampilkan pesan konfirmasi kepada aktor dengan pertanyaan "Apakah anda yakin ingin menghapus data ini?". Setelah aktor menekan tombol "Ok" untuk mengonfirmasi penghapusan, View kembali memanggil fungsi `delete surat jalan()` pada Controller. Controller kemudian berinteraksi dengan Model untuk menghapus surat jalan yang dimaksud dengan memanggil fungsi `delete suratjalan(no surat)` dan memberikan nomor

surat yang akan dihapus sebagai parameter. Model melakukan proses penghapusan dan mengembalikan hasil operasi tersebut kepada Controller. Setelah mendapatkan hasil dari Model, Controller menginstruksikan View untuk menampilkan pesan "Berhasil menghapus Item" kepada aktor.



Gambar 32. Sequence Diagram LPP

Berdasarkan Gambar 32. Sequence Diagram LPP, menggambarkan proses interaksi untuk mengakses halaman "Permintaan Perbaikan". Proses ini diawali dengan Aktor yang mengklik submenu "Permintaan Perbaikan", memicu View untuk memanggil metode "lpp()" pada Controller. Controller kemudian berinteraksi dengan Model dengan memanggil metode "get_lpp()" untuk mengambil data. Setelah data berhasil diambil, Controller menampilkan kembali halaman dan data LPP pada View untuk Aktor.



Gambar 33. Sequence Diagram Export Excel LPP

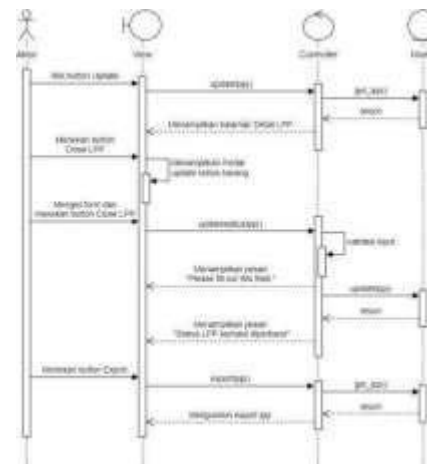
Berdasarkan sequence diagram pada Gambar 33. Sequence Diagram Export Excel LPP, Proses ekspor LPP diawali dengan Aktor yang mengklik tombol "Excel", memicu View untuk mengirim pesan "export excel()" ke Controller. Controller kemudian meminta data dari Model dengan "get_lpp()" untuk ekspor. Setelah data diperoleh, Model mengirimkannya kembali ke Controller, yang selanjutnya menginformasikan View bahwa file Excel telah siap untuk diunduh dengan "Mengunduh file excel".



Gambar 34. Sequence Diagram Add LPP

Berdasarkan sequence diagram pada Gambar 34. Sequence Diagram Add LPP, fitur tambah LPP dimulai dengan aktor yang mengklik tombol "Add". Tindakan ini memicu View untuk menjalankan fungsi tambah lpp(), yang menampilkan halaman "Tambah LPP" kepada aktor.

Aktor kemudian mengisi form dan menekan tombol "Save", menginstruksikan View untuk menambahkan data ke tabel melalui fungsi tambah lpp(). Controller melakukan validasi input dan, jika lengkap, memanggil fungsi insertlpp() pada Model untuk menyimpan data LPP baru ke dalam database. Setelah penyimpanan berhasil, Model memberikan sinyal kembali ke Controller, yang kemudian menginstruksikan View untuk menampilkan halaman LPP dengan pesan "Berhasil menambahkan LPP", menandakan penambahan LPP telah berhasil dilakukan.



Gambar 35. Sequence Diagram Update LPP

Sequence diagram Gambar 35. Sequence Diagram Update LPP ini menggambarkan proses yang dilakukan oleh aktor untuk memperbarui status Lembaran Permintaan Perbaikan (LPP) serta mencetak atau mengekspornya. Proses dimulai ketika aktor mengklik tombol "Update", yang memicu View untuk memanggil fungsi 'updatelpp()' pada Controller. Controller kemudian memanggil fungsi 'get_lpp()' pada Model untuk mendapatkan detail LPP. Model mengembalikan data detail LPP ke Controller, yang kemudian diteruskan ke View untuk menampilkan halaman Detail LPP. Selanjutnya, aktor menekan tombol "Close LPP", yang memicu View untuk menampilkan modal update status barang dengan memanggil fungsi 'modal update status barang()'. Aktor kemudian mengisi form di modal tersebut dan menekan tombol "Close LPP", yang memanggil fungsi 'update status lpp()' pada Controller. Controller melakukan validasi input, dan jika valid, memanggil fungsi 'update lpp()' pada Model untuk memperbarui status LPP. Model mengembalikan hasilnya ke Controller, yang kemudian memerintahkan View untuk menampilkan pesan keberhasilan "Status LPP berhasil diperbarui" atau pesan error "Please fill out this field" jika ada input yang tidak valid. Untuk mengekspor LPP, aktor mengklik tombol "Export", yang memicu View untuk memanggil fungsi 'exportlpp()' pada Controller. Controller memanggil fungsi 'get_lpp()' pada Model untuk mendapatkan data LPP yang akan diekspor. Model mengembalikan data tersebut ke

menggambarkan kolaborasi antara objek-objek yang berinteraksi di dalam elemen-elemen suatu kelas. *Class* diagram menunjukkan hubungan antar kelas, yang mencakup atribut dan fungsi dari suatu objek.

DAFTAR PUSTAKA

- [1] A. Feby Prasetya and U. Lestari Dewi Putri, "Perancangan Aplikasi Rental Mobil Menggunakan Diagram UML (Unified Modelling Language)," 2022.
- [2] M. Sumiati, R. Abdillah, and A. Cahyo, "Pemodelan UML untuk Sistem Informasi Persewaan Alat Pesta," 2021.
- [3] M. Yusuf, "APLIKASI PENGAJUAN PROPOSAL PENELITIAN BERBASIS WEB PADA LPPM UNIVERSITAS TAMAN SISWA PALEMBANG DENGAN METODE UML," 2021.
- [4] M. Nazir, S. Fajariani Putri, and D. Malik, "Perancangan Aplikasi E-VOTING Menggunakan Diagram UML (Unified Modelling Language)," 2022.