

PERANCANGAN SISTEM INFORMASI STOCK COMPOUND BERBASIS WEBSITE DENGAN PENDEKATAN *UNIFIED MODELLING LANGUAGE*

Ferdy Ali Darmawan¹⁾

Teknologi Industri, Politeknik Gajah Tunggal

Ferdydarmawan778@gmail.com

Ahmad Zohari²⁾

Teknik Mesin, Politeknik Gajah Tunggal

zohari@poltek-gt.ac.id

Dera Susilawati³⁾

Teknologi Informasi, Politeknik Gajah Tunggal

dera@poltek-gt.ac.id

ABSTRAK

PT RCI menghadapi tantangan dalam pengelolaan informasi FIFO *stock compound* di departemen mixing, sistem yang digunakan saat ini masih bersifat manual dengan media papan koin sebagai alat bantu pencatatan dan visualisasi data. Sistem tersebut memiliki keterbatasan yang hanya dapat menunjukkan lokasi dan item compound. Kondisi ini menyulitkan dalam pemantauan stok secara *real-time*, meningkatkan risiko kesalahan pengambilan *compound*, serta memerlukan waktu cukup lama untuk melakukan pembaruan data setiap harinya. Untuk mengatasi masalah tersebut, penelitian ini merancang sistem informasi *stock* berbasis *website* dengan menggunakan UML (*Unified Modelling Language*) sebagai bahasa pemodelan visual untuk memudahkan para pengguna dalam memahami dan berinteraksi dengan sistem. Perancangan UML dilakukan sebagai langkah awal sebelum pengembangan *website* dimulai. Sistem ini dirancang untuk membuat proses lebih efisien dan terstruktur. UML yang digunakan mencakup 4 jenis diagram, yaitu *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*. Perancangan sistem informasi *stock* berbasis *website* berhasil dibangun dengan menggunakan UML, yang melibatkan 2 aktor yaitu : Operator dan Staff.

Kata Kunci : *Unified Modelling Language (UML)*, Monitoring dan Kontrol, *stock information*, *update stock*

PT RCI faces challenges in managing FIFO stock compound information in the mixing department, the system currently used is still manual with a coin board as a tool for recording and visualizing data. This condition makes it difficult to monitor stock in real time, increases the risk of compound taking errors, and takes a long time to update data every day. To overcome this problem, this study designs a website-based stock information system using UML (Unified Modeling Language) as a visual modeling language to make it easier for users to understand and interact with the system. UML design is carried out as an initial step before website development begins. This system is designed to make the process more efficient and structured. The UML used includes 4 types of diagrams, namely Use Case Diagram, Activity Diagram, Sequence Diagram, and Class Diagram. The design of a website-based stock information system was successfully built using UML, which involved 2 actors, namely: Operator and Staff.

Keywords: *Unified Modelling Language (UML)*, Monitoring and Control, *stock information*, *update stock*

I. PENDAHULUAN

Latar Belakang

Teknologi digital semakin banyak dimanfaatkan dalam berbagai sektor industri, termasuk dalam pengelolaan informasi. Digitalisasi memungkinkan proses pengelolaan informasi menjadi lebih mudah, akurat, dan efisien dibandingkan dengan metode manual [1]. Penelitian yang dilakukan oleh zebua juga menunjukkan bahwa penerapan sistem informasi digital mampu meningkatkan efisiensi hingga 85% dibandingkan sistem manual [2]. Meski demikian, belum semua perusahaan sepenuhnya mengadopsi sistem digital dalam aktivitas operasionalnya, sehingga tantangan dalam efektivitas pengelolaan informasi masih terus dihadapi [3].

Salah satu perusahaan yang menghadapi tantangan tersebut, khususnya dalam pengelolaan informasi *stock* adalah PT. RCI, sebuah perusahaan manufaktur yang bergerak di bidang industri karet untuk kebutuhan berbagai sektor seperti pertanian, infrastruktur, transportasi dan otomotif. Dalam menjalankan operasionalnya, perusahaan ini mengandalkan berbagai departemen yang memiliki peran dan tanggung jawab spesifik untuk memastikan kelancaran produksi, salah satunya adalah Departemen *Mixing*.

Departemen *Mixing* merupakan departemen yang bertanggung jawab dalam mengolah bahan dasar menjadi *compound*, yang nantinya digunakan sebagai bahan baku utama pada tahap produksi selanjutnya. Departemen ini menerapkan prinsip manajemen stok berbasis FIFO (*First In First Out*), yaitu metode pengelolaan persediaan yang mengasumsikan bahwa barang yang pertama masuk akan menjadi yang pertama digunakan (Hamidy, 2024). Penerapan prinsip ini bertujuan untuk memastikan bahwa *compound* digunakan secara terstruktur sesuai urutan kedatangannya, sehingga meminimalkan risiko kedaluwarsa. Dalam mendukung penerapan prinsip tersebut, departemen *mixing* menggunakan sistem informasi manual berbasis papan koin sebagai acuan dalam proses pengambilan dan penyimpanan *stock compound*. Namun dalam penerapannya, sistem tersebut menimbulkan beberapa langkah kerja tambahan. Hal tersebut terjadi karena sistem berbasis papan FIFO koin tersebut tidak dapat mencatat dan memantau pergerakan *compound* secara *realtime* serta mendukung pengambilan *compound* secara terstruktur. Kondisi aktual dari sistem tersebut dapat dilihat pada Gambar 1



Gambar 1. Papan FIFO Compound

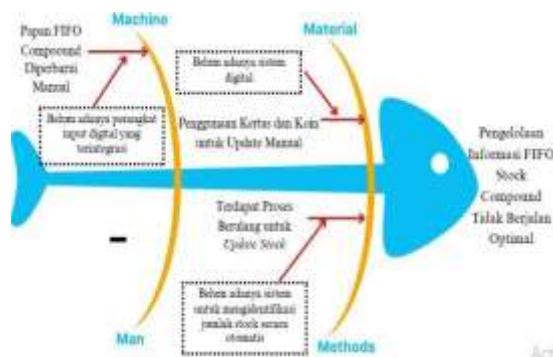
Papan FIFO tersebut hanya dapat menunjukkan lokasi dan item *compound*, tanpa adanya informasi jumlah stock. Selain itu, sistem ini belum memiliki fitur validasi otomatis untuk memastikan pengambilan *compound* sesuai prinsip FIFO. Akibatnya, jika terjadi kesalahan, sistem tidak dapat mencegah atau memberikan peringatan atas pengambilan yang tidak sesuai. Sistem ini juga tidak menyimpan riwayat aktivitas penyimpanan dan pengambilan *compound*, sehingga tingkat *traceability* dalam pengelolaan informasi *stock compound* menjadi rendah. Keterbatasan tersebut dapat berpengaruh langsung terhadap proses kerja di lapangan. Salah satunya yaitu pekerja harus melakukan proses *update stock* setiap hari untuk mencatat jumlah aktual *compound* yang tersedia. Untuk memberikan gambaran waktu yang diperlukan dalam aktivitas tersebut dapat dilihat pada Tabel 1 berikut.

Tabel 1. Waktu *Update Stock*

No	Waktu (detik/hari)
1	2332
2	3086
3	2527
4	2871
5	3404
6	2634
7	3157
8	3132
9	3193
10	2754
Total	29.090
Rata-rata	2.909

Berdasarkan Tabel 1 yang diperoleh melalui hasil observasi pada shift 1, proses *update stock* secara manual pada 10 sampel menunjukkan total waktu sebesar 29.090 detik, dengan rata-rata waktu 2.909 detik atau 48,5 menit per hari. Hal ini menunjukkan bahwa sistem informasi *stock* berbasis papan koin tidak lagi relevan dalam mendukung kebutuhan operasional dengan volume tinggi dan proses kerja yang dinamis. Keterbatasan ini menekankan pentingnya peralihan menuju sistem informasi *stock* yang lebih akurat, responsif, dan terintegrasi secara digital.

Dari permasalahan yang telah ditemukan, dilakukan analisis terhadap masalah tersebut dengan menggunakan *fishbone diagram*. Tujuannya adalah untuk mengidentifikasi dan mencari akar penyebab dari permasalahan yang terjadi.



Gambar 2. *Fishbone Diagram*

Melalui analisis ini ditemukan beberapa faktor penyebab belum optimalnya pengelolaan informasi *stock* FIFO di Departemen Mixing PT RCI. Permasalahan ini disebabkan oleh tiga faktor utama, yaitu *machine*, *material*, dan *methods*. Pertama, dari faktor mesin (*machine*), pengelolaan informasi *stock* masih dilakukan melalui papan FIFO yang diperbarui secara manual karena belum tersedianya perangkat input digital maupun sistem digital yang terintegrasi. Kedua, dari aspek material, pencatatan informasi *stock* masih bergantung pada media fisik berupa kertas dan koin, yang rawan menimbulkan kesalahan dan ketidaksesuaian data. Ketiga, dari aspek metode (*methods*), tidak adanya sistem yang mampu mendeteksi perubahan *stock* secara *real-time* menyebabkan proses pembaruan data harus dilakukan berulang dengan rata-rata durasi mencapai 48,5 menit per hari.

Dalam mengatasi permasalahan tersebut, diperlukan sistem digital berbasis *website* untuk meningkatkan efisiensi dan memudahkan dalam pengelolaan informasi *stock* di Departemen Mixing PT RCI. Sistem ini dirancang menggunakan metode *Unified Modelling Language* (UML), sebuah pendekatan pemodelan visual yang membantu menggambarkan alur kerja dan struktur sistem secara sistematis.

Relevansi dan efektivitas penggunaan UML dalam pengembangan sistem juga telah dibuktikan dalam berbagai penelitian sebelumnya. Penelitian yang dilakukan oleh Khairunnisa dan Voutama menerapkan UML pada perancangan sistem informasi peminjaman inventaris berbasis website di BEM Fasilkom Unsika, yang terbukti mampu mempermudah visualisasi kebutuhan sistem secara terstruktur dan mempercepat proses pengembangan. Implementasi sistem yang dihasilkan juga meningkatkan efisiensi layanan serta mengurangi beban administrasi dalam pengelolaan inventaris [4]. Selain itu, penelitian yang dilakukan oleh Hariansyah juga menunjukkan bahwa penggunaan UML dalam perancangan aplikasi inventaris laboratorium dapat meningkatkan efisiensi pengembangan sistem. Perencanaan yang terstruktur sejak awal memudahkan programmer dalam mengembangkan aplikasi sesuai kebutuhan [5].

Unified Modeling Language (UML) merupakan suatu pemodelan visual yang digunakan untuk merancang, menggambarkan, dan mengkomunikasikan desain suatu sistem secara terstruktur dan sistematis [6]. UML yang akan digunakan mencakup empat jenis diagram, yaitu *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram*.

Use Case Diagram adalah bagian dari diagram UML yang digunakan untuk menunjukkan interaksi pengguna (*user*) terhadap sistem dalam suatu lingkup. *Use case diagram* biasa diterapkan untuk mempresentasikan fungsi dari sistem secara visual, sehingga memudahkan pemahaman dan komunikasi dari sudut pengembang dengan klien/ pengguna [6].

Activity diagram adalah diagram yang dibuat berdasarkan beberapa rancangan dari *use case diagram* dan menggambarkan proses aktivitas yang berjalan dalam sebuah sistem, mulai dari awal proses hingga keputusan yang akan terjadi pada sistem dan bagaimana proses tersebut berakhir [7].

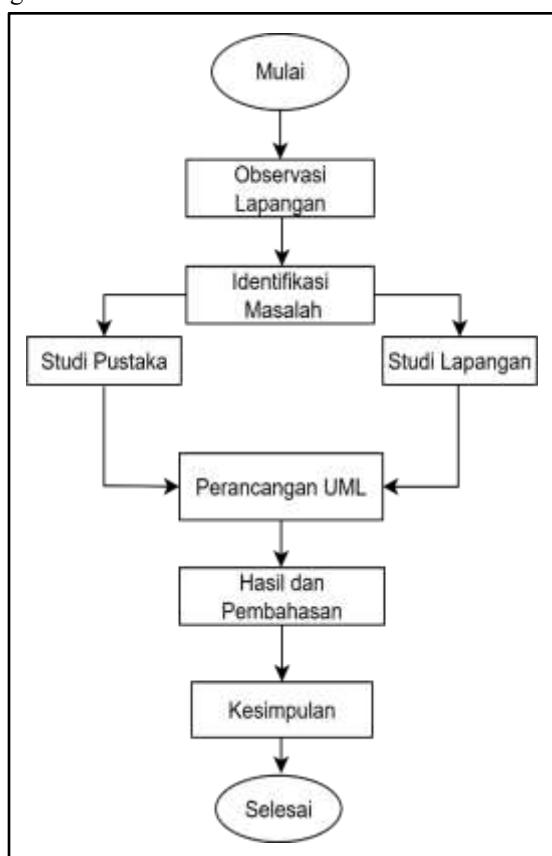
Sequence diagram adalah diagram yang menggambarkan kolaborasi antara objek objek yang berinteraksi di dalam elemen-elemen suatu kelas [8].

Black box testing adalah metode pengujian yang berfokus pada verifikasi fungsionalitas sistem dari perspektif pengguna. Pengujian ini memastikan bahwa sistem beroperasi dengan benar tanpa menguji proses internalnya. Metode ini hanya mengevaluasi hasil output yang terlihat oleh pengguna tanpa memperhitungkan bagaimana proses di dalam sistem bekerja [9].

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 3.



Gambar 3. Alur Penelitian

Observasi Lapangan

Hal yang pertama dilakukan dalam memulai penelitian adalah observasi. Observasi ini dilakukan di Departemen *Mixing*. Hal yang diamati adalah bagaimana sistem informasi *stock* di departemen tersebut. Observasi ini sangat untuk memahami alur kerja aktual dan mengidentifikasi permasalahan yang terjadi di lapangan.

Identifikasi Masalah

Setelah melakukan observasi secara langsung, selanjutnya dilakukan identifikasi terhadap masalah yang terjadi di departemen *mixing*. Dalam proses tersebut, terungkap bahwa salah satu permasalahan yang dihadapi yaitu terkait dengan sistem pengelolaan informasi *stock compound* yang masih menggunakan koin.

Studi Pustaka

Pada tahap ini, dilakukan analisis mendalam terhadap riset sebelumnya yang relevan dengan penelitian yang dilakukan, khususnya terkait perancangan sisrem menggunakan UML. Tujuan dari analisis ini adalah untuk memberikan landasan teoritis yang kuat dan berbasis bukti untuk mendukung proses perancangan dan pengembangan *website* pada tahap selanjutnya.

Studi Lapangan

Pada tahap ini, dilakukan untuk mengumpulkan data dan informasi secara langsung dari lapangan. Data yang diperoleh yaitu data produksi *mixing* periode 2024 dan data *cycle time* dari penggunaan pengelolaan informasi papan FIFO koin.

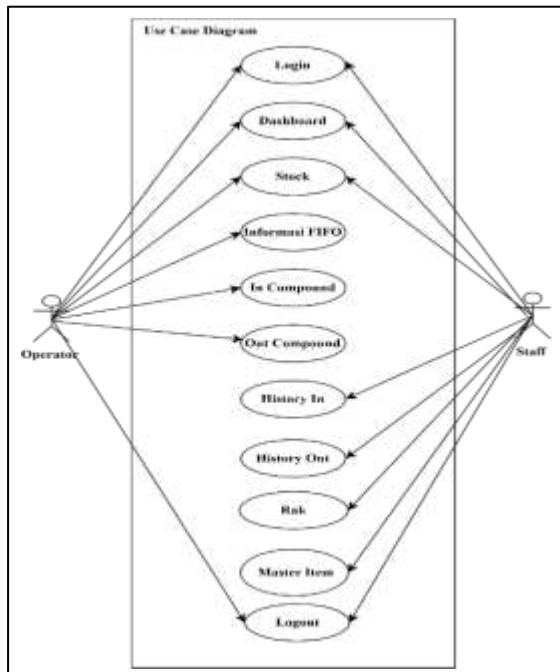
Perancangan UML

Pada tahap ini, dilakukan perancangan UML (*Unified Modeling Language*) sebagai langkah awal sebelum pengembangan *website* dimulai. Proses ini mencakup pembuatan berbagai diagram yang akan membantu dalam visualisasi struktur dan perilaku sistem yang akan dibuat. UML yang akan digunakan mencakup empat jenis diagram, yaitu *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram*.

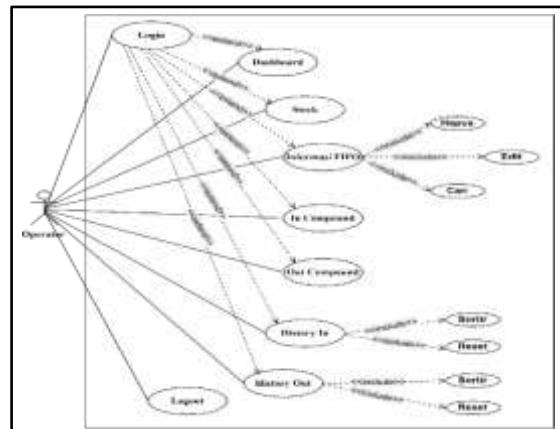
III. HASIL DAN PEMBAHASAN

Use Case Diagram

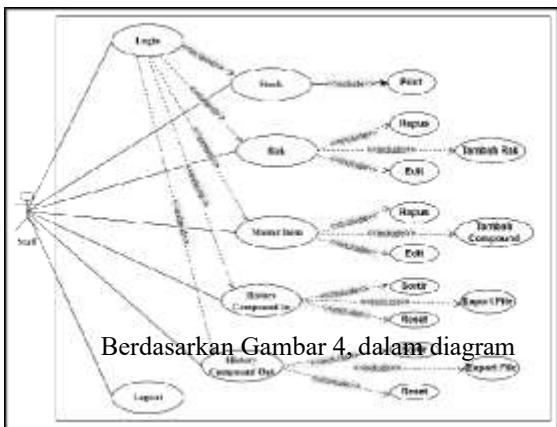
Berikut *Use Case Diagram* untuk sistem informasi *stock* yang dirancang dapat dilihat pada Gambar dibawah ini.



Gambar 4. Use Case Diagram Website



Gambar 5. Use Case Diagram Operator



Gambar 6. Use Case Diagram Staff

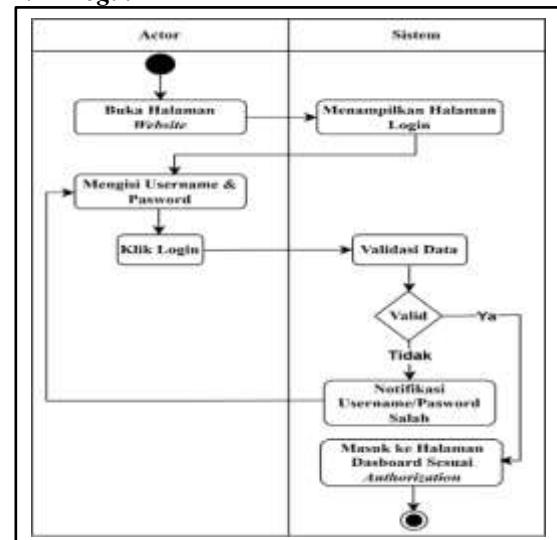
Berdasarkan *use case diagram* yang ditampilkan memperlihatkan dua aktor utama dalam sistem, yaitu operator dan staff. Diagram ini menggambarkan interaksi masing-masing aktor dengan sistem serta menu yang dapat diakses. Operator memiliki akses untuk *login*, masuk ke *dashboard*, menu *stock*, menu informasi *FIFO* yang terdapat tombol fungsi *edit*, *hapus* dan *cari* *compound*, menu *In* dan *Out* untuk melakukan proses *scan in/out compound*, menu *history compound In* dan *compound out* yang terdapat tombol fungsi *sortir* dan *reset*, serta akses untuk *logout*.

Sementara itu, staff memiliki akses yaitu *login*, melihat dan mencetak data *stock* melalui menu *stock* yang terdapat fungsi tombol *print*, melihat dan *mengexport history compound* melalui menu *history compound in* dan *out* yang terdapat tombol fungsi *export*, menambahkan item baru melalui menu *master item* yang terdapat tombol tambah *compound*, dan menambahkan rak baru melalui menu *rak* yang terdapat tombol tambah *rak*, serta *logout* dari 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* berikut.

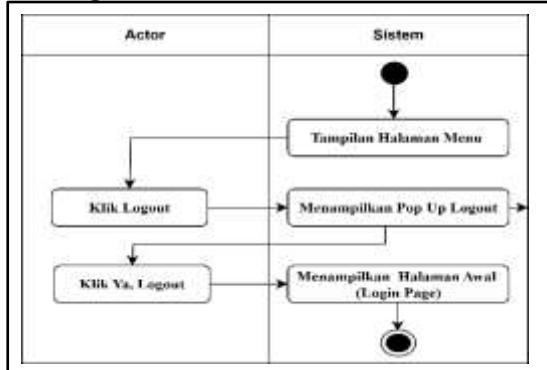
1. Login



Gambar 7. Activity Diagram Login

Berdasarkan *activity diagram* pada Gambar 7, aktor melakukan *login* dengan memasukan *username* dan *password*. Klik “Login” kemudian sistem memproses dan melakukan validasi. Apabila data valid maka aktor akan masuk ke halaman sesuai otorisasinya. Namun, apabila data tidak tervalidasi maka sistem akan menampilkan pesan kesalahan dan mengisi kembali *username* dan *password* dengan benar.

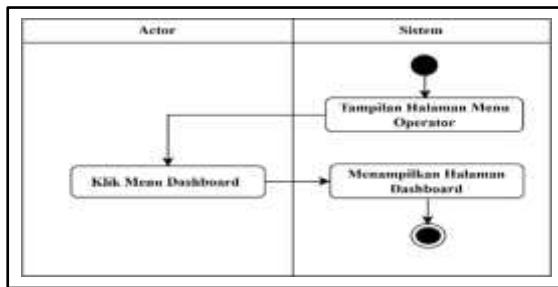
2. Logout



Gambar 8. Activity Diagram Logout

Berdasarkan *activity diagram* pada Gambar 8, apabila aktor ingin *logout* dari website, aktor dapat mengklik tombol *logout*. Kemudian, Sistem akan menampilkan halaman *login*.

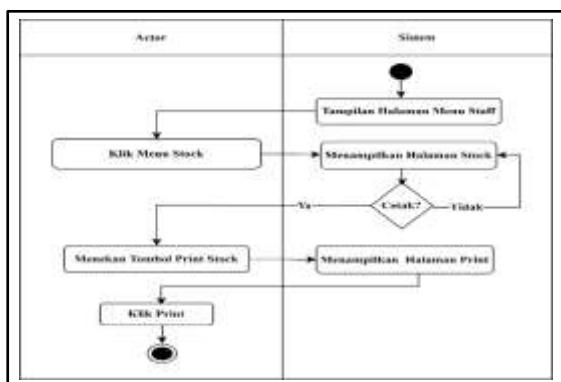
3. Dashboard



Gambar 9. Activity Diagram Dashboard Operator

Berdasarkan *activity diagram* pada Gambar 9, proses dimulai dengan aktor menekan menu *dashboard*. Kemudian, sistem akan menampilkan halaman *dashboard* yang memuat informasi mengenai rak kosong maupun yang sudah terisi, sehingga memudahkan dalam menentukan lokasi penyimpanan *compound*.

4. Stock

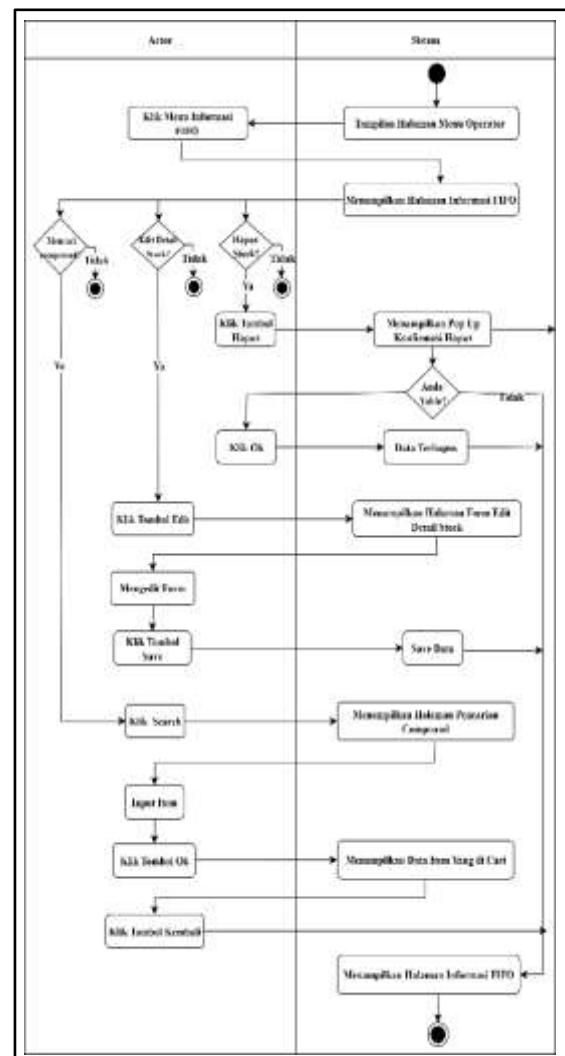


Gambar 10. Activity Diagram Stock Staff

Berdasarkan *activity diagram* pada Gambar 10, aktor dapat mengakses informasi *stock* dengan menekan menu *Stock* pada halaman utama. Sistem

kemudian akan menampilkan halaman berisi data *stock* yang tersedia. Jika aktor ingin mencetak data tersebut, aktor dapat menekan tombol *Print Stock*. Kemudian sistem akan menampilkan *preview cetak*. Selanjutnya, aktor menekan tombol *Print* dan sistem akan memproses permintaan dan mencetak data stok sesuai yang ditampilkan.

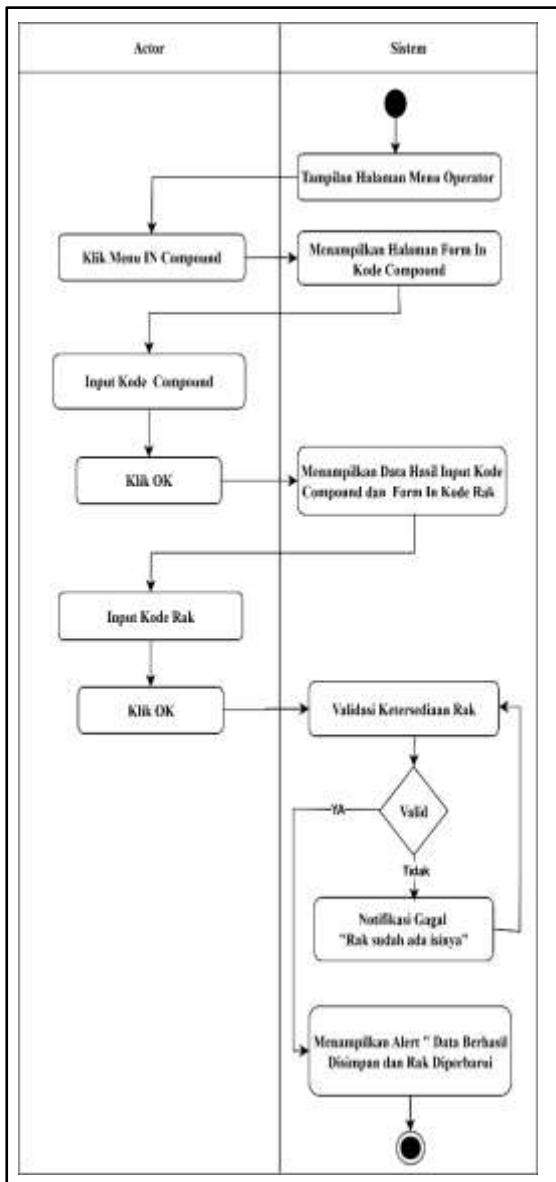
5. Informasi FIFO



Gambar 11. Activity Diagram Informasi FIFO

Berdasarkan *activity diagram* pada Gambar 11, aktor memulai dengan masuk ke halaman *Informasi FIFO*. Jika ingin mencari *compound*, aktor mengklik tombol *cari compound* dan sistem akan menampilkan halaman pencarian. Aktor memasukan item yang dicari, kemudian sistem menampilkan hasil pencarian sesuai urutan FIFO. Jika ingin mengedit, aktor menekan tombol *Edit*, mengubah data pada form, lalu menekan *Save*, dan sistem menyimpan perubahan serta menampilkan notifikasi keberhasilan. Jika ingin menghapus data stok, aktor menekan tombol *Hapus*, mengonfirmasi *pop up*, dan sistem akan menghapus data serta kembali ke halaman *informasi FIFO*.

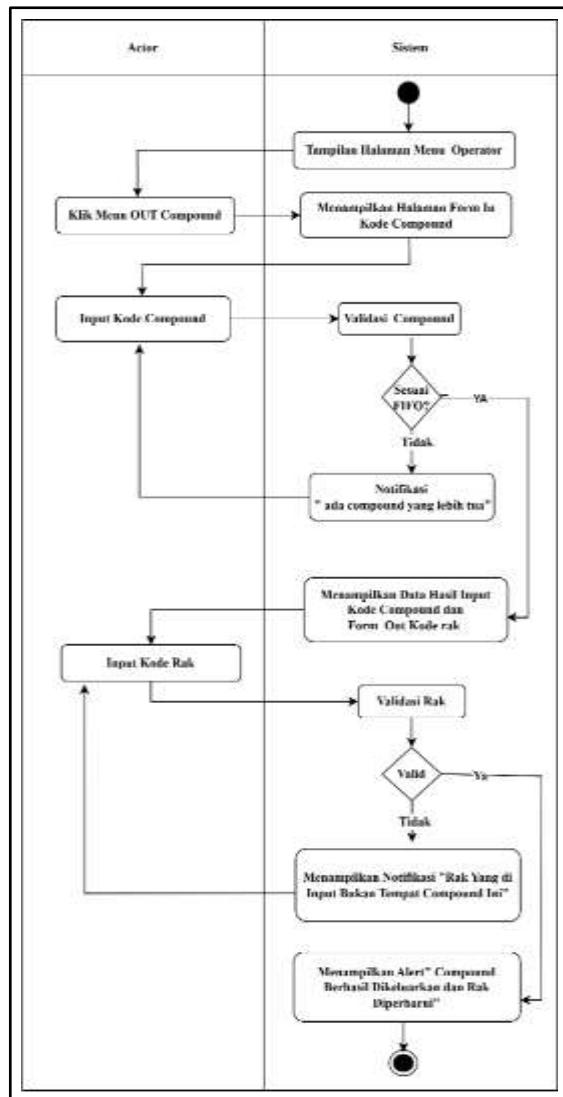
6. In Compound



Gambar 12. Activity Diagram In Compound

Berdasarkan *activity diagram* pada Gambar 12, aktor memulai proses penyimpanan *compound* dengan mengakses menu *In Compound*. Sistem kemudian menampilkan halaman *form* untuk menginput kode *compound*. Aktor dapat memasukkan kode tersebut secara manual atau menggunakan alat pemindai. Selanjutnya, sistem akan menampilkan data *compound* yang telah diinput, beserta *form* untuk memasukkan kode rak. Setelah aktor menginput kode rak, baik secara manual maupun dengan alat pemindai, sistem akan memverifikasi ketersediaan rak. Jika rak masih kosong, sistem akan menampilkan notifikasi pop-up "Data berhasil disimpan dan rak diperbarui." Namun, jika rak sudah terisi, sistem akan menampilkan notifikasi alert "Rak sudah ada isinya" dan mengarahkan kembali ke halaman *input* kode rak.

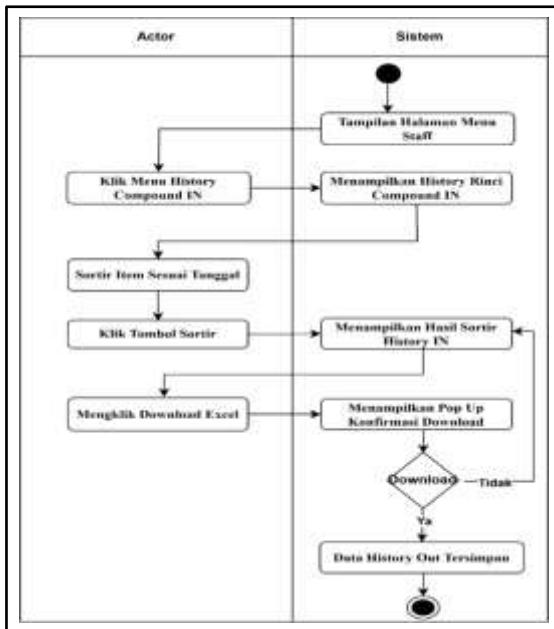
7. Out Compound



Gambar 13. Activity Diagram Out Compound

Berdasarkan *activity diagram* pada Gambar 13, aktor memulai proses pengambilan atau *transfer compound* dengan mengakses menu *Out Compound*. Sistem kemudian akan menampilkan halaman *form* untuk menginput kode *compound*. Aktor dapat menginput kode tersebut secara manual atau menggunakan *scanner*. Setelah itu, sistem akan memverifikasi apakah kode *compound* yang diinput sesuai dengan prinsip FIFO. Jika tidak sesuai, aktor diminta untuk menginput ulang data. Namun, jika sesuai, sistem akan menampilkan data *compound* beserta *form* untuk memasukkan kode rak. Selanjutnya, aktor melakukan pemindai kode rak. Sistem akan memverifikasi apakah rak yang diinput benar merupakan tempat penyimpanan *compound* yang akan diambil. Jika benar, sistem akan menampilkan notifikasi "Compound berhasil dikeluarkan dan rak diperbarui". Namun, jika tidak sesuai, sistem akan menampilkan notifikasi alert "Rak yang diinput bukan tempat *compound* ini".

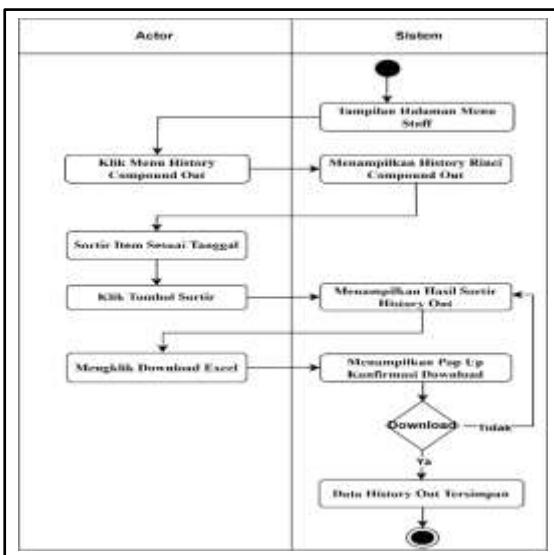
8. History In



Gambar 14. Activity Diagram History In Staff

Berdasarkan *activity diagram* pada Gambar 14, aktor dapat mengakses menu *History Compound In* untuk melihat dan mengunduh riwayat penyimpanan *compound* di area *storage*. Setelah mengklik menu tersebut, sistem akan menampilkan halaman detail riwayat *compound In*. Aktor dapat menyortir riwayat berdasarkan tanggal yang diinginkan dan mengunduh riwayat tersebut dalam format file Excel.

9. History Out

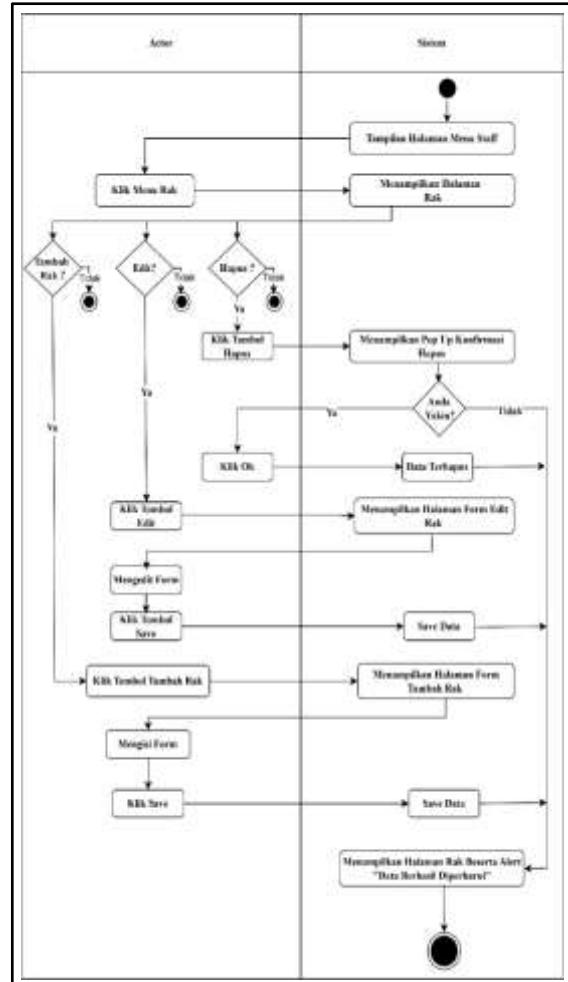


Gambar 15. Activity Diagram History Out Staff

Berdasarkan *activity diagram* pada Gambar 15, aktor dapat mengakses menu *History Compound Out* untuk melihat dan mengunduh riwayat pengambilan *compound* dari area *storage*. Setelah mengklik menu tersebut, sistem akan menampilkan

halaman detail riwayat *compound out*. Aktor dapat menyortir riwayat berdasarkan tanggal yang diinginkan dan mengunduh riwayat tersebut dalam format file Excel.

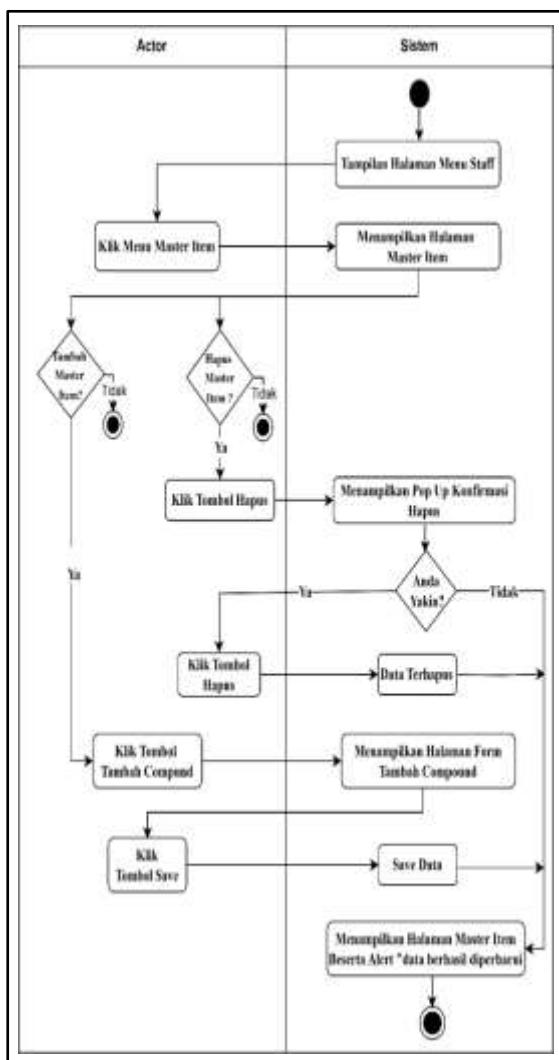
10. Rak



Gambar 16. Activity Diagram Rak Staff

Berdasarkan *activity diagram* pada Gambar 16, aktor dapat mengakses beberapa fitur pada menu Rak, yaitu menambahkan rak baru, mengedit kapasitas rak, dan menghapus rak yang sudah ada. Proses dimulai dengan masuk ke halaman Rak. Untuk menambahkan rak, aktor menekan tombol Tambah Rak, mengisi data pada *form* yang tersedia, lalu menekan tombol Save. Sistem akan menyimpan data dan menampilkan kembali halaman Rak disertai notifikasi "Data berhasil diperbarui". Jika ingin mengedit kapasitas rak, aktor menekan tombol Edit, melakukan perubahan pada *form*, lalu menyimpan. Sistem akan memperbarui data dan menampilkan notifikasi yang sama. Sementara itu, untuk menghapus rak, aktor menekan tombol Hapus pada rak yang dipilih. Sistem akan menampilkan *pop-up* konfirmasi, dan setelah aktor menekan OK, sistem menghapus data tersebut dan kembali menampilkan halaman Rak beserta notifikasi keberhasilan.

11. Master Item

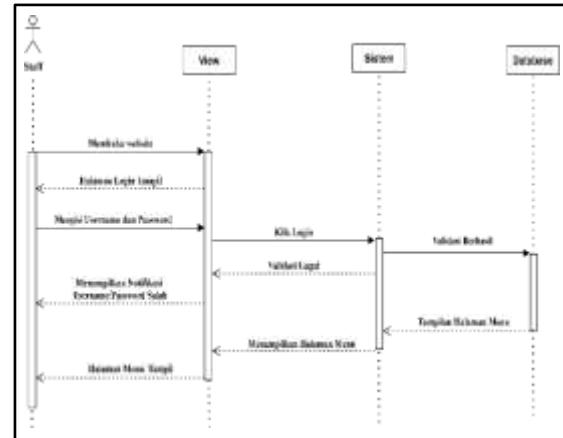


Gambar 17. Activity Diagram Master Item Staff

Berdasarkan *activity diagram* pada Gambar 17, terdapat beberapa fitur yang dapat diakses oleh aktor pada menu ini seperti, menambahkan dan menghapus master item, untuk mengaksesnya aktor terlebih dahulu masuk ke halaman *Master Item*. Apabila ingin menambahkan *master item*, aktor dapat menekan tombol tambah *compound* yang ada pada halaman *Master Item*. Setelah itu, sistem akan menampilkan halaman *form tambah compound*. Selanjutnya, aktor mengisi *form* dan menekan tombol *Save*. Sistem kemudian menyimpan data yang telah di masukan, dan setelah itu sistem akan menampilkan kembali halaman *Master item* beserta alert “Data berhasil di perbarui”. Sementara itu, jika ingin menghapus *master item*, aktor dapat menekan tombol *hapus* pada item yang ingin dihapus di halaman *master item*. Selanjutnya, sistem akan menampilkan alert dengan pesan “Apakah Anda yakin ingin menghapus *compound* ini?”. Setelah aktor menekan tombol *OK*, sistem akan menghapus item yang dipilih aktor.

Sequence Diagram

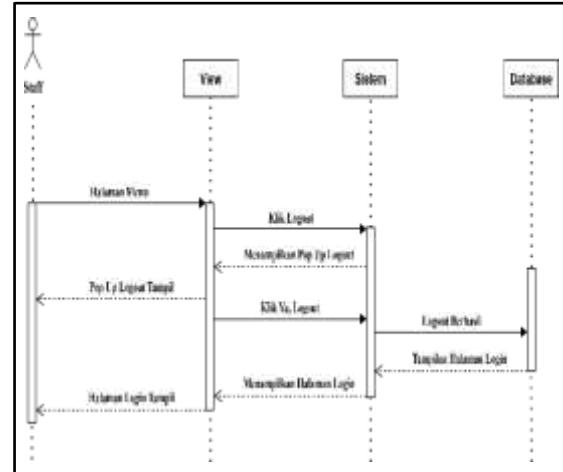
1. Login



Gambar 18. Sequence Diagram Login

Sequence diagram pada Gambar 18 menggambarkan proses *login website*. Pertama-tama operator akan masuk ke halaman *login* dengan memasukan *username* dan *Password*. Kemudian sistem akan mengirim data tersebut untuk divalidasi. Apabila data yang dimasukan benar kemudian *database* akan menyimpan dan sistem akan menampilkan halaman menu operator, namun jika data yang dimasukkan salah maka akan menampilkan notifikasi *username/password salah*.

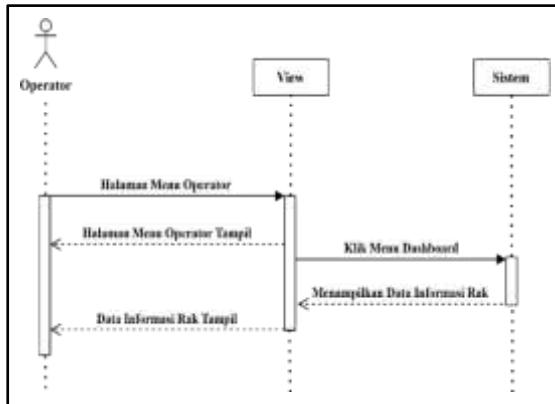
2. Logout



Gambar 19. Sequence Diagram Logout

Berdasarkan *sequence diagram* pada Gambar 19, aktor dapat keluar dari aplikasi dengan menekan tombol *logout*. Kemudian sistem akan memproses dan *database* menyimpan lalu sistem mengarahkan ke halaman *login* yang menandakan *logout* berhasil.

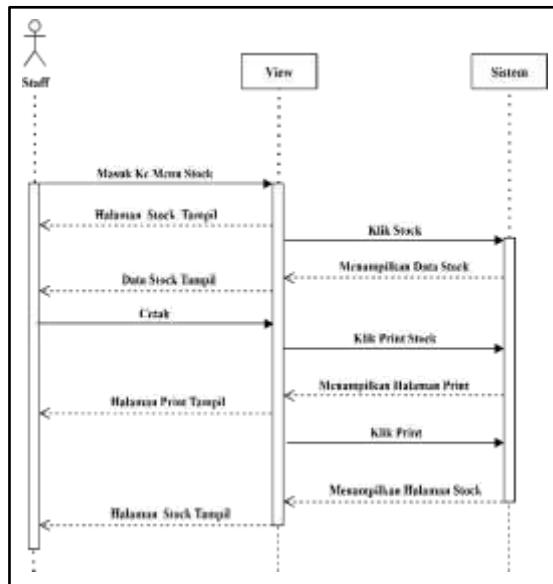
3. Dashboard



Gambar 20. Sequence Diagram Dashboard

Berdasarkan *sequence diagram* pada Gambar 20, ketika ingin masuk ke halaman *Dashboard*, operator harus menekan menu *Dashboard*, Kemudian sistem akan menampilkan data informasi rak yang kosong dan terisi.

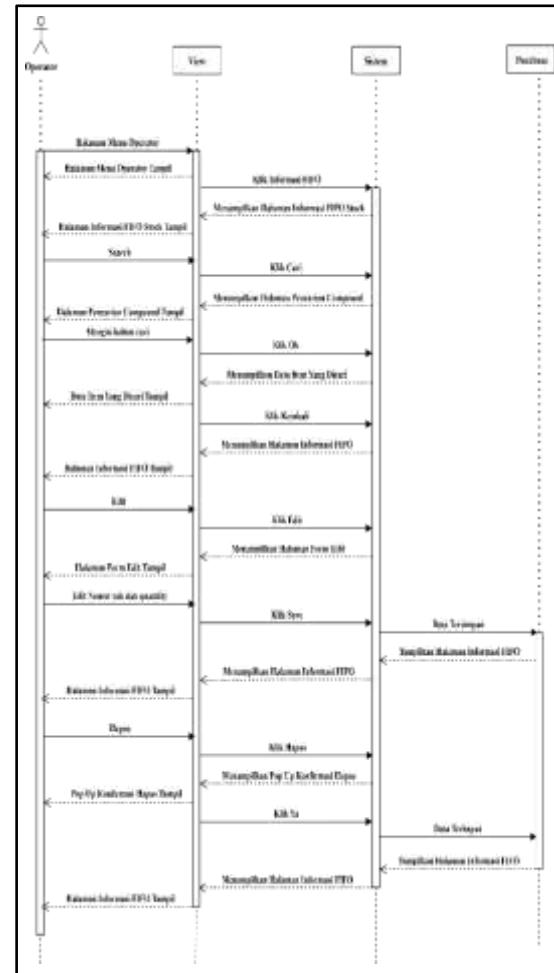
4. Stock



Gambar 21. Sequence Diagram Stock Staff

Berdasarkan *sequence diagram* pada Gambar 21, ketika ingin mencetak data stok sebagai laporan. Staff terlebih dahulu masuk ke menu *Stock*, kemudian menekan tombol *Print Stock*. Setelah itu, sistem akan menampilkan halaman *Print Preview* sebagai antarmuka konfirmasi sebelum pencetakan dilakukan. Jika staff menekan tombol *Print*, sistem akan langsung memproses perintah tersebut dan mencetak laporan stok sesuai data yang tersedia. Setelah proses cetak selesai, kemudian sistem akan menampilkan kembali halaman *Stock*.

5. Informasi FIFO



Gambar 22. Sequence Diagram Informasi FIFO

Berdasarkan *sequence diagram* pada Gambar 22, operator mulai dengan masuk ke halaman *Informasi FIFO*. Pada saat ingin mencari *compound*, operator menekan tombol *Cari Compound*, lalu sistem menampilkan halaman pencarian. Setelah operator menginput nama *compound* yang dicari, sistem akan menampilkan data yang sesuai. Jika telah selesai, menekan tombol *Kembali*, kemudian akan menampilkan halaman *Informasi FIFO*. Ketika ingin memperbarui data stok, operator menekan tombol *Edit*. Kemudian sistem menampilkan halaman *Edit*, dan setelah data diubah, sistem menyimpan perubahan ke *database* lalu kembali ke halaman *Informasi FIFO*. Sementara itu, saat operator ingin menghapus data stok, operator menekan tombol *Hapus* pada data *stock* yg ingin dihapus dan sistem akan memproses sekaligus menghapus data dari *database*. Jika penghapusan berhasil, sistem kembali menampilkan halaman *Informasi FIFO*.

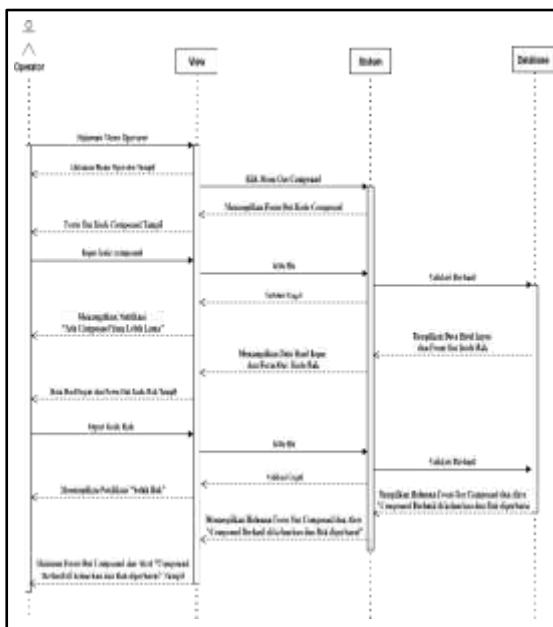
6. *In Compound*



Gambar 23. Sequence Diagram In Compound

Berdasarkan *sequence diagram* pada Gambar 23, proses pada menu *In Compound* diawali saat operator mengakses halaman *input* kode *compound*. Di halaman ini, operator memasukkan kode *compound* menggunakan *scanner* atau secara manual. Selanjutnya, sistem menampilkan halaman *input* kode rak, lalu operator memasukkan kode rak. Setelah itu, sistem melakukan validasi data ke database untuk memastikan bahwa kode rak yang dimasukkan tersedia. Apabila validasi berhasil, data *compound* dan lokasi rak akan langsung tersimpan di *database* dan sistem akan menampilkan kembali halaman *input* kode *compound* dengan notifikasi alert “Data berhasil disimpan dan rak diperbarui.

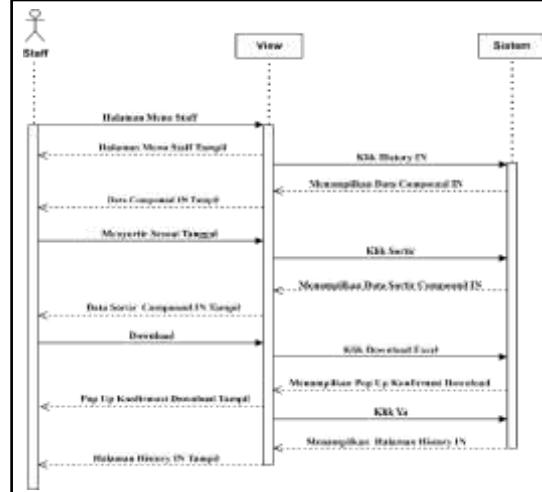
7. *Out Compound*



Gambar 24. Sequence Diagram Out Compound

Berdasarkan *sequence diagram* pada Gambar 24, proses ini dimulai ketika operator menekan menu *Out Compound*. Sistem kemudian akan menampilkan halaman *form* input kode *compound*. Pada halaman ini, operator memasukkan kode *compound* menggunakan scanner atau secara manual. Setelah itu, sistem melakukan validasi data ke *database* untuk memastikan bahwa kode *compound* yang dimasukkan adalah *compound* yang harus diambil sesuai prinsip FIFO. Jika validasi gagal, sistem akan menampilkan notifikasi “Ada *compound* yang lebih lama.” Namun, jika validasi berhasil, sistem akan menampilkan data hasil input dan halaman *form* input kode rak. Selanjutnya, operator memasukkan kode rak, baik dengan scanner maupun manual. Sistem kemudian memvalidasi kode rak ke *database* untuk memastikan bahwa rak tersebut memang tempat penyimpanan *compound* yang diambil. Setelah validasi berhasil, sistem akan menampilkan kembali halaman *form* *Out Compound*.

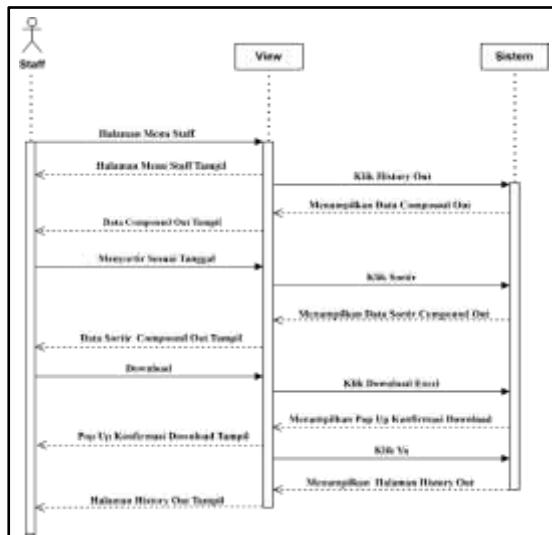
8. *History In*



Gambar 25. Sequence Diagram History In Staff

Berdasarkan *sequence diagram* pada Gambar 25, ketika ingin melihat dan *mengexport* data *history compound In* ke dalam format excel, staff terlebih dahulu memilih menu *History Compound In* pada halaman menu staff. Setelah itu, sistem akan menampilkan seluruh data *compound* yang tersimpan. Selanjutnya staff dapat melakukan penyortiran data berdasarkan tanggal tertentu untuk mempermudah pencarian riwayat. setelah data tersortir ditampilkan, staff menekan tombol Download untuk mengunduh data tersebut. Sistem kemudian akan menampilkan *pop-up* konfirmasi download, dan apabila staff menekan tombol Ya, sistem akan *mengexport* data ke file excel dan mengarahkan kembali ke halaman *History Compound In*.

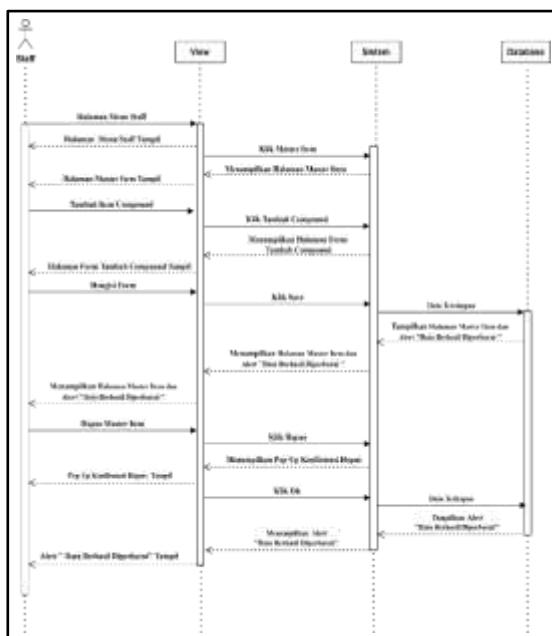
9. History Out



Gambar 26. Sequence Diagram History Out Staff

Berdasarkan *sequence diagram* pada Gambar 26, ketika ingin melihat dan *export* data *history compound Out* ke dalam *format excel*, staff terlebih dahulu memilih menu *History Compound Out* pada halaman menu staff. Setelah itu, sistem akan menampilkan seluruh data *compound* yang telah di keluarkan. Selanjutnya staff dapat melakukan penyortiran data berdasarkan tanggal tertentu untuk mempermudah pencarian riwayat. setelah data tersortir ditampilkan, staff menekan tombol Download untuk mengunduh data tersebut. Sistem kemudian akan menampilkan *pop-up* konfirmasi download, dan apabila staff menekan tombol Ya, sistem akan *export* data ke file excel.

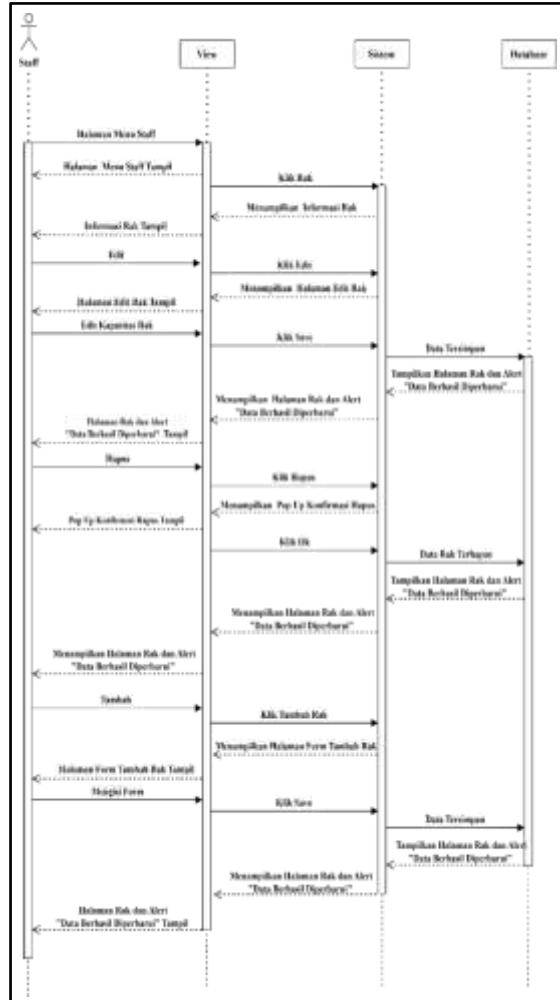
10. Master Item



Gambar 27. Sequence Diagram Master Item Staff

Berdasarkan *sequence diagram* pada Gambar 27, operator mulai dengan masuk ke halaman master item yang menyediakan beberapa fitur yaitu menambahkan dan menghapus *master item*. Ketika staff ingin menambahkan *master item*, operator menekan tombol *Tambah Compound*. Sistem akan menampilkan halaman *form* tambah *compound*, kemudian operator mengisi *form* yang tersedia dan menekan tombol *Save*. Setelah itu, sistem memproses dan menyimpan data *compound* baru tersebut ke dalam *database*. Sedangkan, jika ingin menghapus *master item*, staff menekan tombol *Hapus* pada *compound* yang ingin dihapus. Sistem akan memproses dan menghapus data *compound* tersebut dari *database*. Setelah penghapusan berhasil, sistem kembali menampilkan halaman *master item*.

11. Rak

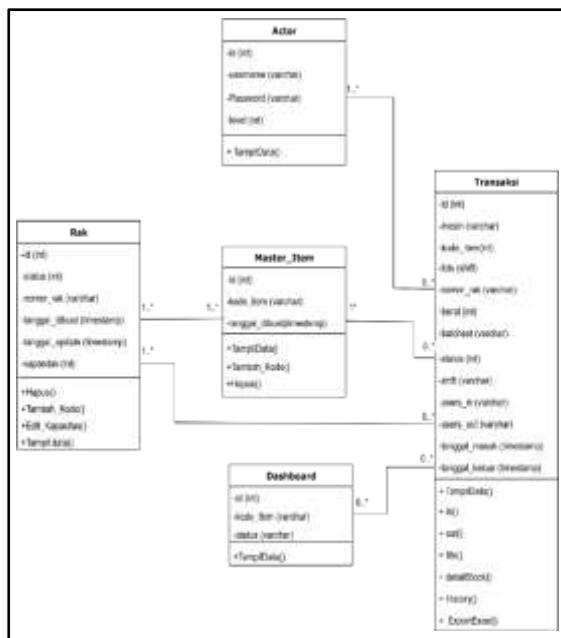


Gambar 28. Sequence Diagram Rak Staff

Berdasarkan *sequence diagram* pada Gambar 28, staff mulai dengan masuk ke halaman rak yang menyediakan beberapa fitur mengedit, menghapus dan menambahkan rak. Jika staff ingin mengedit rak, staff menekan tombol *Edit* pada rak yang diinginkan. Sistem kemudian menampilkan

halaman Edit Rak, dimana staff dapat melakukan perubahan yang diperlukan dan menekan tombol *Save*. Setelah itu, sistem akan memproses data yang diperbarui dan menyimpannya ke dalam *database*. Ketika ingin menghapus rak, staff menekan tombol Hapus pada rak yang ingin dihapus. Sistem akan memproses dan menghapus data rak tersebut dari *database*. Setelah penghapusan berhasil, sistem kembali menampilkan halaman Rak. Sementara itu, jika staff ingin menambahkan rak baru, staff menekan tombol Tambah Rak. Sistem akan menampilkan halaman *form* tambah rak, kemudian staff mengisi *form* yang tersedia dan menekan tombol *Save*. Setelah itu, sistem memproses dan menyimpan data rak baru tersebut ke dalam *database*.

Class Diagram



Gambar 29. Class Diagram

Berdasarkan *class diagram* pada Gambar 29, terdapat lima *class* utama yang terdiri dari *Actor*, *Rak*, *Master_Item*, *Dashboard*, dan *Transaksi*. Secara keseluruhan *class diagram* ini menggambarkan alur proses yang terstruktur dalam kegiatan penyimpanan, pengelolaan stok, dan pemantauan sistem FIFO (*First In First Out*) berbasis *website*. *Class Actor* berfungsi sebagai pengelola sistem dengan atribut *username*, *password*, dan *level* akses untuk mengoperasikan seluruh fitur sistem. *Class Rak* merepresentasikan lokasi penyimpanan fisik dengan atribut *status*, *nomor rak*, dan *kapasitas* untuk melacak lokasi spesifik setiap *item*. *Class Master_Item* berperan sebagai *database* master barang dengan kode *item* untuk memastikan standarisasi data produk. *Class Transaksi* merupakan inti sistem FIFO yang mencatat aktivitas keluar masuk *compound* dengan

atribut lengkap seperti kode item, nomor rak, berat, tanggal masuk, dan tanggal keluar untuk memastikan rotasi stok yang tepat. *Class Dashboard* berfungsi sebagai pusat monitoring dengan tampilan visual untuk memantau status *inventory* secara *real-time*.

IV. KESIMPULAN

Unified Modelling Language (UML) berhasil dirancang untuk memudahkan dalam perancangan sistem informasi *stock* berbasis *website*. Diagram UML yang digunakan yaitu *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*, yang melibatkan 2 aktor yaitu : Operator dan Staff.

DAFTAR PUSTAKA

- [1] B. Adzkiya, "Perancangan Sistem Informasi Warehouse Berbasis Web Dalam Menunjang Pengadaan Persediaan Material Di PT . X Design Of A Web-Based Warehouse Information System In Supporting The Procurement Of Material Inventory At PT X Program Studi Logistik Minyak Dan Ga," Pp. 78–87, 2024.
- [2] N. J. D. K. Zebua, E. Waruwu, D. S. Zebua, And Y. Mendorfa, "Implementasi Sistem Pencatatan Laporan Persediaan Barang Berbasis Digital Di Satuan Polisi Pamong Praja Kota Gunungsitoli," *Tuhanori: Jurnal Ilmiah Multidisiplin*, Vol. 2, No. 4, Pp. 269–291, Nov. 2024, Doi: 10.62138/Tuhanori.V2i4.85.
- [3] S. Klaudia, K. Blitar, J. Timur, J. Candi, W. Lawang, And J. Timur, "Integrasi Digitalisasi Keuangan Dalam Meningkatkan Efisiensi Operasional Perusahaan," Pp. 224–232, 2024.
- [4] G. Khairunnisa, A. Voutama Sistem Informasi, And U. H. Singaperbangsa Karawang Jalan Ronggo Waluyo Karawang, "Penerapan Uml Dalam Perancangan Sistem Informasi Peminjaman Inventaris Berbasis Web Di Bem Fasilkom Uniska," 2024.
- [5] F. Fiani And P. Aditya, "Perancangan Aplikasi Inventaris Laboratorium Komputer Menggunakan Uml (Unified Modelling Language)," 2021.
- [6] S. Pranoto, S. Sutiono, Sarifudin, And D. Nasution, "Penerapan Uml Dalam Perancangan Sistem Informasi Pelaporan Dan Evaluasi Pembangunan Pada Bagian Administrasi Pembangunan Sekretariat Daerah Kota Tebing Tinggi," *Surplus: Jurnal Ekonomi Dan Bisnis*, Vol. 2, No. 2, Pp. 384–401, 2024.

- [7] F. N. Hasanah, *Buku Ajar Rekayasa Perangkat Lunak*. 2020. Doi: 10.21070/2020/978-623-6833-89-6.
- [8] M. Nazir, S. F. Putri, And D. Malik, “Perancangan Aplikasi E-Voting Menggunakan Diagram Uml (Unified Modelling Language),” *Jurnal Ilmiah Komputer Terapan Dan Informasi*, Vol. 1, No. 1, Pp. 5–9, 2022, [Online]. Available: <Http://Journal.Polita.Ac.Id/Index.Php/Politati/Article/View/99/92>
- [9] Y. F. Achmad And A. Yulfitri, “Pengujian Sistem Pendukung Keputusan Menggunakan Black Box Testisng Studi Kasus E-Wisudawan Di Institus Sains Dan Teknologi Al-Kamal,” *Jurnal Ilmu Komputer*, Vol. 5, P. 42, 2020.