

**PENGEMBANGAN SISTEM INFORMASI MANAJEMEN  
OPERASIONAL PT TATA METAL LESTARI BERDASARKAN  
HASIL EVALUASI MC CALL'S MODEL**

**LAPORAN PROYEK AKHIR**



Proposal Proyek Akhir ini ditulis untuk memenuhi Sebagian persyaratan mendapatkan gelar Ahli Madya Komputer

**Disusun Oleh:**

**DANUARTHA**

**NIM 10109013**

**PROGRAM STUDI DIPLOMA III SISTEM INFORMASI  
JURUSAN TEKNOLOGI INFORMASI DAN KOMPUTER  
POLITEKNIK NEGERI SUBANG  
2025**

**PENGEMBANGAN SISTEM INFORMASI MANAJEMEN  
OPERASIONAL PT TATA METAL LESTARI  
BERDASARKAN HASIL EVALUASI MCCALL'S *MODEL***

**DANUARTHA  
10109013**

**ABSTRAK**

Penelitian ini bertujuan untuk melakukan pengembangan pada Sistem Informasi Manajemen Operasional PT Tata Metal Lestari, khususnya pada fitur *form checklist crane* yang digunakan dalam Divisi Produksi. Permasalahan utama yang ditemukan meliputi keterbatasan dokumentasi kapasitas *crane*, pencatatan *shift leader* yang belum optimal, serta penerapan Sistem Perintah Kerja (SPK) yang tidak sesuai dengan kondisi operasional di lapangan. Selain itu, sistem juga memerlukan peningkatan pada sisi efisiensi dan penanganan *error (error handling)*.

Metodologi yang digunakan dalam pengembangan sistem ini adalah metode Agile dengan *framework* Scrum. Proses pengembangan dilakukan secara iteratif melalui tahapan *product backlog*, *sprint planning*, *sprint execution*, dan *sprint review*. Evaluasi kualitas sistem dilakukan dengan pendekatan McCall's Quality Model yang mencakup aspek *correctness*, *reliability*, *efficiency*, *integrity*, *usability*, *maintainability*, *flexibility*, *testability*, *portability*, *reusability*, dan *interoperability*.

Hasil yang diperoleh dari penelitian ini adalah sistem *checklist crane* yang lebih komprehensif dan sesuai kebutuhan pengguna, dengan pencatatan yang lebih baik, *error handling* yang lebih informatif, dan penghapusan fitur SPK berdasarkan evaluasi User Acceptance Testing (UAT). Namun, terdapat satu komponen pengembangan yang belum berhasil diterapkan, yaitu *upgrade server*, yang seharusnya dapat meningkatkan kenyamanan dan kecepatan boot sistem.

Hal baru yang diperoleh dari penelitian ini adalah integrasi model McCall ke dalam proses pengembangan sistem berbasis Scrum, serta penerapan *form checklist* yang disesuaikan dengan kebutuhan lintas divisi. Penelitian ini diharapkan dapat menjadi referensi bagi pengembangan sistem informasi serupa di lingkungan industri manufaktur.

**Kata Kunci:** *Checklist Crane*, Divisi Produksi, *Error Handling*, Manajemen Operasional, *Model McCall*, *Scrum*, Sistem Informasi.

**PENGEMBANGAN SISTEM INFORMASI MANAJEMEN  
OPERASIONAL PT TATA METAL LESTARI  
BERDASARKAN HASIL EVALUASI MCCALL'S MODEL**

**DANUARTHA  
10109013**

***ABSTRACT***

*This research aims to develop the Operational Management Information System at PT Tata Metal Lestari, specifically on the crane checklist form feature used by the Production Division. The main issues identified include limited documentation of crane capacity, inadequate shift leader logging, and the implementation of the Work Order System (SPK), which is unsuitable for real operational conditions. Additionally, the system requires improvements in efficiency and error handling.*

*The development methodology used is Agile with the Scrum framework. The system was developed iteratively through stages such as product backlog creation, sprint planning, sprint execution, and sprint review. System quality evaluation was carried out using McCall's Quality Model, which covers correctness, reliability, efficiency, integrity, usability, maintainability, flexibility, testability, portability, reusability, and interoperability.*

*The result of this study is a more comprehensive crane checklist system that better meets user needs, featuring improved documentation, more informative error handling, and removal of the SPK feature based on User Acceptance Testing (UAT). However, one enhancement was not successfully implemented—server upgrade—which was expected to improve system boot speed and user comfort.*

*The novelty of this research lies in the integration of McCall's model into the Scrum-based system development process and the adaptation of the checklist form feature to support cross-divisional operations. This study is expected to serve as a reference for similar system developments in the manufacturing industry environment.*

**Keywords:** *Crane Checklist, Error Handling, Information System, McCall's Model, Operational Management, Production Division, Scrum.*

## KATA PENGANTAR

Segala puji dan syukur dipanjatkan ke hadirat Allah SWT atas rahmat dan hidayah-Nya sehingga laporan Proyek Akhir ini dapat diselesaikan tepat waktu. Tujuan utama dari penyusunan laporan ini adalah untuk memperluas pemahaman mengenai tantangan yang ada di sektor industri, meningkatkan keterampilan dalam pemecahan masalah, serta menerapkan ilmu yang telah diperoleh selama studi ke dalam dunia kerja atau kegiatan organisasi. Dengan upaya yang telah dilakukan, semoga bimbingan dari dosen pembimbing senantiasa diberikan selama proses ini. Selain itu, laporan ini juga diharapkan dapat menjadi bentuk kontribusi dan pengabdian kepada perusahaan.

Harapan besar disematkan agar laporan ini dapat memberikan manfaat bagi para pemangku kepentingan, pembaca, serta masyarakat luas. Disadari bahwa laporan ini masih memiliki kekurangan, sehingga masukan berupa kritik dan saran yang membangun sangat diharapkan untuk menyempurnakan laporan di masa mendatang.

Pada kesempatan ini, penghargaan dan rasa terima kasih yang sebesar-besarnya disampaikan kepada:

1. Bapak Oyok Yudianto, S.T., M.T. selaku Direktur Utama Politeknik Negeri Subang.
2. Ibu Wiwik Endah Rahayu, S.T.P., M.Si. selaku Wakil Direktur Bidang Akademik dan Kemahasiswaan Politeknik Negeri Subang.
3. Bapak Nunu Nugraha Purnawan, S.Pd., M.Kom. selaku Wakil Direktur Bidang Umum dan Keuangan Politeknik Negeri Subang.
4. Bapak Dwi Vernanda, S.T., M.Pd. selaku Ketua Jurusan Teknologi Informasi dan Komputer Politeknik Negeri Subang.
5. Ibu Lani Nurlani, S.T., M.Kom. selaku Koordinator Prodi Diploma III Sistem Informasi.
6. Ibu Nurfitria Khoirunnisa, S.Tr.Kom., M.Kom. selaku Dosen Pembimbing 1 yang telah membimbing dan memotivasi untuk mengerjakan dan menyelesaikan Laporan Praktik Akhir ini.

7. Bapak Erick Febrianto S.Kom., M.TI. selaku Dosen Pembimbing 2 yang telah membimbing dan memotivasi untuk mengerjakan dan menyelesaikan Laporan Proyek Akhir ini.
8. Ibu Brigita Maria sebagai Manager divisi Warehouse yang telah memberikan kesempatan untuk melakukan magang di divisi Warehouse
9. Bapak Panggah selaku Mentor selama kegiatan Praktik Kerja Lapangan di PT. Tata Metal Lestari, yang telah memberikan pengarahan selama kegiatan Praktik Kerja Lapangan Berlangsung.
10. Kedua orang tua yang telah mendidik dan selalu mendoakan, serta tak pernah luput untuk selalu memberikan semangat.
11. Kepada pemilik NIM 1019053 yang selalu memberikan semangat kepada penulis untuk bisa menyelesaikan laporan ini.
12. Teman-teman SI 3A yang telah senantiasa membantu serta memotivasi selama kegiatan Praktik Kerja Lapangan berlangsung.
13. Serta pihak lain yang telah membantu dan memberikan motivasi dalam menyelesaikan laporan ini.

Akhir kata berharap laporan ini dapat berguna dan bermanfaat bagi semua kalangan.

Penulis,

**DANUARTHA**  
**NIM 10109013**

## DAFTAR ISI

ABSTRAK.....	i
<i>ABSTRACT</i> .....	ii
KATA PENGANTAR.....	iii
DAFTAR ISI .....	v
DAFTAR GAMBAR .....	vii
DAFTAR TABEL.....	ix
DAFTAR SINGKATAN .....	x
DAFTAR LAMPIRAN .....	xi
BAB 1 PENDAHULUAN .....	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah .....	3
1.3. Tujuan Penelitian .....	3
1.4. Relevansi atau Manfaat Hasil Penelitian .....	3
1.5. Batasan Masalah atau Ruang Lingkup .....	4
1.6. Sistematika Penulisan .....	7
BAB 2 LANDASAN TEORI.....	10
2.1. Pengembangan Sistem Informasi .....	10
2.2. Checklist digital dalam warehouse management .....	10
2.3. <i>Error handling</i> pada sistem <i>digital</i> .....	10
2.4. McCall's Quality Model .....	10
2.5. Sistem Pencarian Data dalam Warehouse.....	11
2.6. <i>Flowchart</i> .....	11
2.7. Unified Modeling Language (UML) .....	15
2.7.1. <i>Use case Diagram</i> .....	15
2.7.2. <i>Activity Diagram</i> .....	16
2.7.3. <i>Sequence Diagram</i> .....	17
2.7.4. <i>Class diagram</i> .....	19
2.7.5. UI/UX .....	20
2.8. <i>Black Box Testing</i> .....	21
2.9. <i>Entity Relationship Diagram</i> (ERD) .....	22
2.10. Penelitian yang Relevan .....	24

BAB 3	METODOLOGI PENELITIAN.....	26
3.1.	Evaluasi Mc Call's Model .....	26
3.2.	Metodologi.....	29
3.3.	Jadwal PA .....	33
BAB 4	ANALISIS DAN PERANCANGAN .....	34
4.1.	Analisis .....	34
4.2.	Mc Call's Model .....	34
4.3.	<i>Product Backlog</i> .....	38
4.4.	<i>Sprint Planning</i> .....	39
4.5.	Studi Literatur.....	40
4.6.	Perancangan.....	40
4.6.1.	Perancangan Sistem UML .....	41
4.6.2.	Perancangan <i>Basis Data</i> .....	50
4.6.3.	Perancangan Antar Muka.....	54
BAB 5	IMPLEMENTASI DAN EVALUASI.....	57
5.1.	Implementasi .....	57
5.1.1.	Implementasi Mc Call's Model.....	57
5.1.2.	Implementasi Sistem.....	73
5.1.3.	<i>Sprint Execution</i> .....	75
5.2.	Evaluasi.....	76
5.2.1.	<i>BlackBox</i> .....	76
5.2.2.	User Acceptance Testing.....	81
BAB 6	KESIMPULAN DAN SARAN.....	82
6.1.	Kesimpulan.....	82
6.2.	Saran .....	83
DAFTAR PUSTAKA	.....	84

## DAFTAR GAMBAR

Gambar 3.1. <i>Agile Methodology</i> .....	29
Gambar 3.2. <i>Scrum Framework</i> .....	31
Gambar 4.1. <i>Usecase Diagram</i> .....	41
Gambar 4.2. <i>Activity Diagram Mengelola Mapping</i> .....	43
Gambar 4.3. <i>Activity Diagram Mengelola Form Checklist</i> .....	44
Gambar 4.4. <i>Activity Diagram Mengelola Checklist Response</i> .....	45
Gambar 4.5. <i>Sequence Diagram Mengelola Mapping</i> .....	46
Gambar 4.6. <i>Sequence Diagram Mengelola Form Checklist</i> .....	47
Gambar 4.7. <i>Sequence Diagram Mengelola Form Checklist Response</i> .....	48
Gambar 4.8. <i>Class Diagram</i> .....	49
Gambar 4.9. <i>Entity Relationship Diagram</i> .....	50
Gambar 4.10. <i>Wire frame login page</i> .....	54
Gambar 4.11. <i>Wire frame Landing page</i> .....	55
Gambar 4.12. <i>Wire frame Mapping page</i> .....	55
Gambar 4.13. <i>Wire frame Error handling page</i> .....	55
Gambar 4.14. <i>Wire frame Form Check page</i> .....	56
Gambar 5.1. Pengujian <i>Realibility</i> dengan WAPT .....	58
Gambar 5.2. Pengujian <i>Realibility</i> dengan WAPT 1 .....	59
Gambar 5.3. Pengujian <i>Realibility</i> dengan WAPT 2 .....	59
Gambar 5.4. Pengujian <i>Efficiency</i> dengan G Lighthouse .....	60
Gambar 5.5. Pengujian <i>Efficiency</i> dengan G Lighthouse 1 .....	60
Gambar 5.6. Pengujian <i>Efficiency</i> dengan G Lighthouse 2 .....	60
Gambar 5.7. Pengujian <i>Efficiency</i> dengan G Lighthouse 3 .....	61
Gambar 5.8. Pengujian <i>Efficiency</i> dengan G Lighthouse 4 .....	61
Gambar 5.9. Pengujian <i>Efficiency</i> dengan G Lighthouse 5 .....	61
Gambar 5.10. Pengujian <i>Efficiency</i> dengan G Lighthouse 6 .....	62
Gambar 5.11. Pengujian <i>Efficiency</i> dengan G Lighthouse 7.....	62
Gambar 5.12. Pengujian <i>Efficiency</i> dengan G Lighthouse 8 .....	62
Gambar 5.13. Pengujian <i>Efficiency</i> dengan G Lighthouse 9 .....	63
Gambar 5.14. Pengujian <i>Efficiency</i> dengan G Lighthouse 10 .....	63



Gambar 5.15. Pengujian <i>Efficiency</i> dengan G Lighthouse 11.....	63
Gambar 5.16. Pengujian <i>Integrity</i> dengan Sucuri Sitehack .....	64
Gambar 5.17. Pengujian <i>Integrity</i> dengan Sucuri Sitehack 1 .....	64
Gambar 5.18. Halaman <i>Login</i> Sistem Informasi Manejemn Oeprasional .....	74
Gambar 5.19. Halaman Utama Sistem Informasi Manajemen Oeprasional .....	74
Gambar 5.20. Halaman <i>Form Checklist</i> Sistem Informasi Manejemn Oeprasional .....	74
Gambar 5.21. Halaman <i>Mapping</i> Sistem Informasi Manejemn Oeprasional .....	75
Gambar 5.22. Halaman <i>Error Handling</i> Sistem Informasi Manajemen Oeprasional .....	75
Gambar 5.23. Penggunaan sistem oleh divisi produksi .....	81

## DAFTAR TABEL

Tabel 2.1. <i>Symbol Flowchart</i> .....	11
Tabel 2.2. <i>Symbols Use Case Diagram</i> .....	15
Tabel 2.3. <i>Symbols Activity Diagram</i> .....	16
Tabel 2.4. <i>Symbols Sequence Diagram</i> .....	17
Tabel 2.5. <i>Symbols Class Diagram</i> .....	19
Tabel 2.6. Simbol ERD .....	22
Tabel 2.7. Penelitian yang Relevan .....	24
Tabel 3.1. Jadwal PA .....	33
Tabel 4.1. <i>Product Backlog</i> .....	39
Tabel 4.2. <i>Actor Description</i> .....	42
Tabel 5.1. Kategori pengujian Mc Call's Model .....	57
Tabel 5.2. Hasil Pengujian <i>Realibiliy</i> dengan WAPT .....	59
Tabel 5.3. Pengujian <i>Efficency</i> dengan Google Lighthouse .....	60
Tabel 5.4. Pengujian <i>Correctness</i> dengan Kuisisioner .....	64
Tabel 5.5. Pengujian <i>Realibility</i> dengan Kuisisioner .....	65
Tabel 5.6. Pengujian <i>Efficiency</i> dengan Kuisisioner .....	65
Tabel 5.7. Pengujian <i>Integrity</i> dengan Kuisisioner .....	66
Tabel 5.8. Pengujian <i>Usability</i> dengan Kuisisioner .....	66
Tabel 5.9. Pengujian <i>Maintainbility</i> dengan Kuisisioner .....	66
Tabel 5.10. Pengujian <i>Flexibility</i> dengan Kuisisioner .....	67
Tabel 5.11. Pengujian <i>Testability</i> dengan Kuisisioner .....	67
Tabel 5.12. Pengujian <i>Portability</i> dengan Kuisisioner .....	67
Tabel 5.13. Pengujian <i>Reusability</i> dengan Kuisisioner .....	68
Tabel 5.14. Pengujian <i>Interoperability</i> dengan Kuisisioner .....	68
Tabel 5.15. Penghitungan Presentasi dengan Interpretasi .....	69
Tabel 5.16. <i>Sprint Execution</i> .....	75
Tabel 5.17. Koresponden <i>Black Box Testing</i> .....	77
Tabel 5.18. Butir Uji <i>Blacxbox</i> .....	77

## DAFTAR SINGKATAN

1. PT : Perseroan Terbatas
2. SRS : *Software Requirements System*
3. SDD : *System Design Document*
4. STP : *Software Test Plan*
5. STC : *Software Test Case*
6. UI : *User Interface*
7. SPK : Sistem Pengambil Keputusan
8. UML : *Unified Modelling Language*
9. QA : Quality Assurance
10. CRUD : *Create, Read, Update, Delete*
11. ERD : *Entity Relationship Diagram*
12. PA : Proyek Akhir
13. UAT : *User Acceptance Test*

## **DAFTAR LAMPIRAN**

# **BAB 1**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

PT Tata Metal Lestari, sebagai bagian dari TATA LOGAM GROUP, merupakan perusahaan manufaktur terkemuka di Indonesia yang bergerak di bidang produksi baja ringan. Beberapa produk unggulan yang dihasilkan antara lain NEXALUME, NEXIUM, NEXACOLOR, dan SAKURA ROOF. Dalam mendukung kelancaran operasionalnya, perusahaan ini memiliki beberapa divisi penting, salah satunya adalah Divisi *Warehouse*. Divisi ini bertanggung jawab atas pengelolaan produk setelah melewati proses produksi dan pengujian kualitas (*Quality Assurance/QA*) hingga siap untuk distribusi ke berbagai tujuan, baik domestik maupun internasional.

Untuk meningkatkan efisiensi dalam pengelolaan warehouse, perusahaan telah mengimplementasikan Sistem Informasi Digital Warehouse sejak Oktober 2024. Sistem ini mencakup berbagai fitur, seperti pembuatan *shipping mark*, mapping & muat container, pengelolaan *form checklist*, pengelolaan *open pack*, pengelolaan packing list, checklist kendaraan, scan layout, coil damage, pengelolaan packing L08, hingga pembuatan surat izin keluar. Dengan adanya sistem ini, aktivitas *warehouse* menjadi lebih terdokumentasi dan terorganisir secara digital.

Namun, setelah sistem diterapkan dan dilakukan *User Acceptance Testing* (UAT), ditemukan beberapa kendala yang mengharuskan sistem mengalami pengembangan lebih lanjut. Beberapa fitur yang ada masih memerlukan penyesuaian agar dapat lebih optimal dalam menunjang kebutuhan operasional Divisi *Warehouse*. Selain itu, terdapat permintaan dari Divisi Produksi agar dapat mengakses fitur *form checklist*, karena divisi produksi juga memiliki kebutuhan yang sama dan memerlukan akses ke sistem tersebut.

Meskipun telah dilakukan UAT sebagai tahap akhir dalam pengujian sebelum implementasi penuh, evaluasi lebih lanjut tetap diperlukan untuk memastikan bahwa sistem benar-benar memenuhi standar kualitas perangkat lunak yang diharapkan. Menurut penelitian yang dilakukan oleh Pressman Suryani

(2015), UAT hanya menguji sistem dari sisi pengguna dan tidak selalu mencakup aspek teknis yang lebih dalam, seperti *maintainability*, *reliability*, dan *efficiency*. Oleh karena itu, penggunaan model McCall tetap diperlukan untuk menilai lebih lanjut kualitas sistem berdasarkan aspek yang lebih luas, guna mengidentifikasi kekurangan yang mungkin belum terlihat pada tahap UAT. Penelitian lain oleh Kitchenham & Pfleeger (2008) juga menegaskan bahwa model kualitas perangkat lunak seperti McCall mampu memberikan gambaran yang lebih komprehensif terkait permasalahan sistem yang tidak selalu terdeteksi dalam pengujian berbasis pengguna saja.

Berdasarkan analisis menggunakan model McCall, ditemukan beberapa permasalahan utama yang menunjukkan bahwa pengembangan lebih lanjut pada sistem sangat diperlukan. Kapasitas *crane* yang ada belum optimal dalam menunjang operasional, yang berpotensi menyebabkan keterlambatan dalam pergerakan barang di *warehouse*. Selain itu, sistem masih memiliki keterbatasan dalam menangani kesalahan atau *error* yang terjadi, sehingga beberapa kesalahan yang tidak terdeteksi dapat menghambat kelancaran operasional. Sistem Pendukung Keputusan (SPK) yang diterapkan juga belum sepenuhnya sesuai dengan kebutuhan, yang dapat mengakibatkan inefisiensi dalam pengambilan keputusan. Dari sisi pengelolaan kode, rendahnya *maintainability* menyebabkan kesulitan dalam pemeliharaan dan pengembangan sistem akibat kurangnya dokumentasi dan ketidakkonsistenan dalam standar pengkodean. Efisiensi sistem juga masih menjadi permasalahan, dengan kecepatan loading halaman yang belum optimal, yang menghambat akses informasi oleh pengguna. Selain itu, minimnya dukungan *usability*, seperti kurangnya dokumentasi dan bantuan *online*, membuat pengguna mengalami kesulitan dalam memahami cara kerja sistem secara mandiri.

Permasalahan-permasalahan tersebut menunjukkan bahwa sistem masih memerlukan pengembangan lebih lanjut agar dapat lebih optimal dalam mendukung aktivitas Divisi *Warehouse* dan Divisi Produksi. Oleh karena itu, pendekatan yang digunakan dalam pengembangan sistem selanjutnya adalah dengan tetap mengacu pada model McCall guna memastikan kualitas perangkat lunak yang lebih baik.

## **1.2. Perumusan Masalah**

Berdasarkan latar belakang tersebut, permasalahan yang dikaji dalam penelitian ini dapat dirumuskan sebagai berikut:

1. Bagaimana agar sistem informasi dapat mengatasi kendala operasional setelah diterapkannya evaluasi menggunakan Mc Call's Model?
2. Bagaimana sistem dapat dikembangkan agar Divisi Produksi dapat mengakses dan memanfaatkan fitur yang dibutuhkan secara efektif untuk mendukung proses operasional?
3. Bagaimana cara meningkatkan *maintainability* sistem agar memudahkan pemeliharaan, perbaikan, serta pengembangan lebih lanjut?
4. Bagaimana meningkatkan efisiensi dan performa sistem agar dapat berjalan lebih optimal dalam mendukung kebutuhan operasional *warehouse*?

## **1.3. Tujuan Penelitian**

Penelitian ini bertujuan untuk mengembangkan Sistem Informasi Manajemen Operasional di PT Tata Metal Lestari agar lebih optimal dalam mendukung operasional Divisi *Warehouse* serta memperluas penggunaannya ke Divisi Produksi. Dengan pengembangan ini, sistem diharapkan dapat memenuhi kebutuhan pengguna secara lebih efektif dan meningkatkan efisiensi dalam proses kerja.

1. Mengembangkan sistem informasi agar mampu mengatasi kendala operasional yang terjadi berdasarkan hasil evaluasi MC Call's Model
2. Menyempurnakan fitur-fitur sistem agar Divisi Produksi dapat mengakses dan memanfaatkannya secara optimal untuk mendukung proses operasional.
3. Meningkatkan *maintainability* sistem dengan pendekatan yang memudahkan pemeliharaan, perbaikan, serta pengembangan lebih lanjut.
4. Mengoptimalkan efisiensi dan performa sistem agar lebih responsif dan dapat berjalan lebih stabil dalam mendukung kebutuhan operasional *warehouse*.

## **1.4. Relevansi atau Manfaat Hasil Penelitian**

Hasil penelitian ini diharapkan dapat memberikan kontribusi yang signifikan dalam pengembangan sistem informasi, khususnya dalam mendukung

operasional *warehouse* dan produksi di PT Tata Metal Lestari. Adapun manfaat penelitian ini dapat dikategorikan ke dalam beberapa aspek berikut:

1) Manfaat Akademis

- a. Menambah wawasan dan referensi dalam pengembangan sistem informasi berbasis *digital* di lingkungan industri manufaktur.
- b. Memberikan contoh penerapan metode McCall dalam mengevaluasi dan meningkatkan kualitas perangkat lunak.
- c. Dapat menjadi dasar bagi penelitian lebih lanjut terkait pengembangan sistem informasi di bidang logistik dan produksi..

2) Manfaat Teknologi

- a. Menghasilkan sistem informasi yang lebih terintegrasi dan fungsional, sehingga dapat diakses oleh Divisi *Warehouse* dan Divisi Produksi secara lebih efektif.
- b. Meningkatkan efisiensi dan akurasi dalam pengelolaan *data warehouse*, seperti *form checklist*, *shipping mark*, *mapping container*, dan lainnya.
- c. Memberikan *insight* baru dalam pengembangan sistem berbasis web untuk manajemen *warehouse* dan produksi di industri manufaktur.

3) Manfaat Praktis untuk Perusahaan

- a. Mempermudah koordinasi antara Divisi *Warehouse* dan Divisi Produksi, sehingga alur kerja lebih efisien.
- b. Mengurangi risiko kesalahan dalam pencatatan dan pemantauan operasional dengan sistem yang lebih terstruktur.
- c. Mendukung proses pengambilan keputusan berbasis data dengan menyediakan informasi yang lebih akurat dan *real-time*.

### 1.5. Batasan Masalah atau Ruang Lingkup

Penelitian ini difokuskan pada pengembangan Sistem Informasi Manajemen Operasional di PT Tata Metal Lestari, dengan ruang lingkup yang mencakup peningkatan fungsionalitas sistem agar dapat digunakan oleh Divisi *Warehouse* dan Divisi Produksi. Untuk menjaga fokus penelitian, beberapa batasan dan asumsi yang digunakan adalah sebagai berikut:



1) Ruang Lingkup Penelitian

- a. Penelitian ini hanya mencakup pengembangan dan perbaikan Sistem Informasi Manajemen Operasional
- b. Fokus pengembangan difokuskan pada penyesuaian sistem berdasarkan hasil Mc Call's Model dan kebutuhan baru yang muncul selama implementasi.
- c. Penelitian ini juga mencakup integrasi sistem agar dapat diakses oleh Divisi Produksi, khususnya dalam penggunaan *form checklist*.

2) Pembatasan Permasalahan

- a. Penelitian ini tidak mencakup pengembangan sistem baru dari nol, tetapi hanya berfokus pada peningkatan dan modifikasi sistem yang sudah ada.
- b. Studi ini hanya berfokus pada pengguna di lingkungan PT Tata Metal Lestari, tanpa mempertimbangkan penggunaan sistem di perusahaan lain.
- c. Pengujian sistem dilakukan hanya dalam lingkungan *internal* perusahaan, tanpa uji coba eksternal di luar organisasi.
- d. Data yang digunakan dalam penelitian ini terbatas pada data operasional *warehouse* dan produksi yang relevan dengan sistem.

3) Asumsi-Asumsi yang Digunakan

- a. Sistem yang sedang dikembangkan akan tetap berbasis *web* dan diakses melalui *intranet* perusahaan.
- b. Divisi *Warehouse* dan Divisi Produksi memiliki kebutuhan yang selaras dalam penggunaan sistem, sehingga integrasi dapat dilakukan tanpa mengubah arsitektur sistem secara signifikan.
- c. Data yang digunakan dalam penelitian ini berasal dari informasi aktual di Divisi *Warehouse* PT Tata Metal Lestari, dengan asumsi bahwa *data* yang diberikan akurat dan *valid*.

4) Batasan Pengujian dengan menerapkan Mc Call's Model

- a. Pengujian *Product Operation*

- 1) Pengujian *realibility* menggunakan WAPT bertujuan untuk mengukur seberapa stabil dan konsistennya sistem saat menerima beban pengguna yang tinggi secara berkelanjutan.
  - 2) Pengujian *efficiency* menggunakan Google Lighthouse untuk mengevaluasi kinerja aplikasi dari segi kecepatan muat, waktu respons, dan penggunaan sumber daya.
  - 3) Pengujian *integrity* menggunakan Sucuri Sitehack untuk mendeteksi adanya kerentanan, *malware*, dan potensi celah keamanan pada sistem aplikasi.
  - 4) Pengujian *correctness* menggunakan kuisisioner untuk mengetahui sejauh mana sistem memenuhi kebutuhan fungsional pengguna sesuai dengan spesifikasi yang telah ditentukan.
  - 5) Pengujian *realibility* menggunakan kuisisioner untuk menilai persepsi pengguna terhadap kestabilan sistem dalam kondisi normal maupun beban tinggi.
  - 6) Pengujian *efficiency* menggunakan kuisisioner untuk mengetahui pendapat pengguna terkait kecepatan dan efisiensi penggunaan sistem dalam aktivitas sehari-hari.
  - 7) Pengujian *integrity* menggunakan kuisisioner untuk mengukur tingkat kepercayaan pengguna terhadap keamanan dan perlindungan data dalam sistem.
  - 8) Pengujian *usability* menggunakan kuisisioner untuk mengetahui tingkat kemudahan penggunaan, kejelasan tampilan, dan pengalaman pengguna saat berinteraksi dengan sistem.
- b. Product *revision*
- 1) Pengujian *maintainbility* menggunakan kuisisioner untuk mengevaluasi seberapa mudah sistem dapat diperbaiki atau dimodifikasi oleh tim pengembang ketika terjadi kesalahan atau diperlukan pembaruan.
  - 2) Pengujian *flexibility* menggunakan kuisisioner untuk mengetahui sejauh mana sistem dapat diadaptasi untuk kebutuhan atau lingkungan baru.

- 3) Pengujian *testability* menggunakan kuisioner untuk menilai kemudahan dalam melakukan pengujian terhadap sistem, baik secara manual maupun otomatis.

c. *Product Transition*

- 1) Pengujian *portability* menggunakan kuisioner untuk untuk menilai sejauh mana sistem dapat dijalankan di berbagai platform atau lingkungan yang berbeda.
- 2) Pengujian *reusability* menggunakan kuisioner untuk mengevaluasi kemungkinan bagian dari sistem dapat digunakan kembali untuk pengembangan aplikasi lain.
- 3) Pengujian *interoperability* menggunakan kuisioner untuk mengetahui kemampuan sistem dalam berkomunikasi atau bertukar data dengan sistem lain.

## 1.6. Sistematika Penulisan

Sistematika penulisan proyek akhir ini meliputi :

### **BAB 1 PENDAHULUAN**

Berisi pendahuluan yang menguraikan latar belakang masalah, yaitu sistem informasi digital warehouse yang masih memiliki keterbatasan dalam pencatatan kapasitas *crane*, pencatatan *shift leader*, dan penggunaan fitur SPK yang tidak sesuai dengan kondisi lapangan. Bab ini juga memuat rumusan masalah, tujuan penelitian, manfaat akademis dan praktis dari penelitian, batasan ruang lingkup, serta sistematika penulisan laporan secara keseluruhan.

### **BAB 2 LANDASAN TEORI**

Membahas landasan teori yang menjadi dasar pengembangan sistem. Di dalamnya dijelaskan teori-teori tentang pengembangan sistem informasi, fungsi *checklist digital* dalam manajemen gudang, pentingnya fitur *error handling*, serta pemaparan detail dari McCall's Quality Model yang dijadikan acuan utama dalam evaluasi sistem. Selain itu,

terdapat penjelasan mengenai metode pemodelan seperti UML (meliputi *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*), serta penggunaan *Entity Relationship Diagram* (ERD), UI/UX, dan metode pengujian *black box testing*. Bab ini juga mencantumkan beberapa penelitian terdahulu yang relevan sebagai pembanding

### **BAB 3      METODOLOGI PENELITIAN**

Menguraikan metodologi penelitian yang digunakan, yakni pendekatan Agile dengan *framework* Scrum. Proyek dibagi ke dalam beberapa sprint yang dilakukan secara bertahap. Evaluasi sistem dilakukan dengan dua pendekatan: pengujian berbasis alat (seperti WAPT, Google Lighthouse, dan Sucuri SiteCheck) dan pengujian berbasis kuisisioner. Penilaian dilakukan sesuai dengan kategori McCall's Model, yaitu *product operation*, *product revision*, dan *product transition*. Selain itu, bab ini juga memuat jadwal kegiatan proyek akhir secara rinci dalam bentuk tabel

### **BAB 4      ANALISIS DAN PERANCANGAN**

Menjelaskan proses analisis dan perancangan sistem berdasarkan hasil evaluasi pada sistem sebelumnya. Permasalahan utama yang ditemukan antara lain lambatnya kecepatan sistem, kurangnya *error handling* yang informatif, serta ketidaksesuaian fitur SPK. Untuk menjawab permasalahan tersebut, tim pengembang menyusun *product backlog* dan membaginya ke dalam empat sprint utama. Bab ini juga berisi perancangan sistem yang divisualisasikan dalam bentuk diagram UML, desain *basis data*, serta rancangan antarmuka pengguna yang lebih intuitif.

### **BAB 5      IMPLEMENTASI**

Menyajikan proses implementasi sistem sesuai dengan hasil *sprint* yang telah direncanakan. Setiap fitur diuji melalui metode *black box* dan kuisisioner UAT untuk memastikan

kesesuaian antara sistem dan kebutuhan pengguna. Hasil pengujian menunjukkan bahwa sistem telah mengalami peningkatan signifikan, khususnya pada aspek *correctness* (95%) dan *reliability* (90%), meskipun masih ditemukan kelemahan pada aspek *efficiency* dan *maintainability*. Evaluasi dilakukan secara sistematis dan hasilnya digunakan untuk memberikan rekomendasi pengembangan lanjutan.

## **BAB 6 KESIMPULAN DAN SARAN**

Memuat kesimpulan bahwa sistem yang dikembangkan telah berhasil meningkatkan performa dan fungsionalitas sesuai kebutuhan divisi pengguna. Saran yang diberikan meliputi perbaikan pada efisiensi sistem, dokumentasi kode untuk *maintainability*, serta peningkatan panduan penggunaan agar *usability* semakin baik. Secara keseluruhan, laporan ini menjadi kontribusi nyata dalam penerapan *model* kualitas perangkat lunak McCall dan pengembangan sistem informasi berbasis *Agile* dalam dunia industri.

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1. Pengembangan Sistem Informasi**

Pengembangan sistem informasi adalah suatu kumpulan proses yang terstruktur berupa metode, praktik, peralatan terotomasi yang digunakan oleh *stakeholder* untuk memelihara dan melakukan perubahan atau penyesuaian pada sistem informasi dan perangkat lunak (Hartono, 2021).

#### **2.2. Checklist digital dalam warehouse management**

*Checklist digital* merupakan alat bantu yang digunakan untuk memastikan bahwa setiap tahapan operasional gudang dilakukan sesuai prosedur, penggunaan *checklist digital* dapat meningkatkan kepatuhan terhadap *standard* oprasional dan memudahkan *monitoring* aktivitas gudang seara *real-time* (Richards & Grinsted, 2020).

#### **2.3. Error handling pada sistem digital**

*Error handling* adalah fitur penting dalam sistem *digital* untuk mencegah kesalahan pengguna dan meningkatkan keamanan data, implementasi *error handling* yang baik harus mencakup validasi *input*, notifikasi kesalahan yang informatif, serta mekanisme pemulihan data agar sistem tetap berjalan dengan optimal (Fadila dkk., 2024).

#### **2.4. McCall's Quality Model**

McCall's Quality Model adalah salah satu model kualitas perangkat lunak yang dikembangkan oleh Jim McCall pada tahun 1977 untuk Departemen Pertahanan AS. Model ini bertujuan untuk mengevaluasi dan meningkatkan kualitas perangkat lunak dengan pendekatan yang berfokus pada perspektif pengguna dan pengembang (Ramulu & Murhtyr, 2020). Model Mcall's terbagi menjadi beberapa komponen diantaranya :

- a. *Product Operation* (Operasi Produk) yang terdiri dari *correctness*, *reliability*, *efficiency*, *integrity*, dan *usability*.
- b. *Product Revision* (Revisi Produk) yang terdiri dari *maintainability*, *flexibility*, dan *testability*.

c. *Product Transition* (Transisi Produk) yang terdiri dari *portability*, *reusability*, dan *Interoperability*.

## 2.5. Sistem Pencarian Data dalam Warehouse

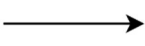
Penerapan metode pencarian data ini menerapkan metode LIFO (*Last In First Out*) yang dimana data yang terakhir ditambahkan akan keluar paling atas untuk memudahkan pencarian *data*, kemudian di terapkan juga pencarian dengan *index* yang lebih beragam pada setiap menu yang tersedia (Ramdani & Zakaria, 2023).

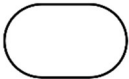
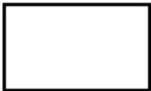
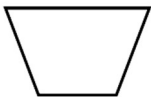
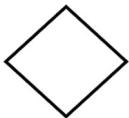

## 2.6. Flowchart

*Flowchart* adalah representasi grafis dari langkah-langkah dan urutan prosedur dalam suatu program. Diagram ini membantu analis dalam memecah masalah menjadi bagian-bagian yang lebih kecil serta mengevaluasi alternatif lain dalam operasional. *Flowchart* juga mempermudah pemahaman dan penyelesaian suatu masalah, terutama yang memerlukan analisis dan evaluasi lebih lanjut. Dalam lingkungan organisasi, proses kerja umumnya terdiri dari serangkaian aktivitas yang berulang. Setiap siklus aktivitas tersebut dapat diuraikan ke dalam langkah-langkah kecil, yang kemudian dapat dianalisis untuk menemukan bagian yang dapat ditingkatkan (*improve*) (Ridlo, 2017).





Secara khusus, American *National Standards Institute* (ANSI) menetapkan standar untuk diagram alur dan simbolnya pada 1960-an. Setelah itu, *International Organization for Standardization* (ISO) mengadopsi simbol ANSI pada tahun 1970. Secara umum, *flowchart* mengalir dari atas ke bawah dan dari kiri ke kanan (Zen Flowchart, 2022). Berikut merupakan simbol-simbol yang terdapat pada *flowchart*:



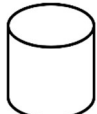
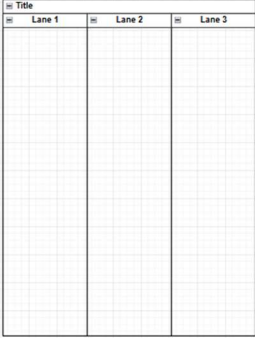
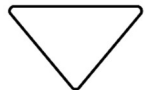

Tabel 2.1. *Symbol Flowchart*

No	Simbol	Nama	Keterangan
1		Simbol Arus	Digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain ( <i>connecting line</i> ). Simbol ini juga berfungsi untuk

No	Simbol	Nama	Keterangan
			menunjukkan garis alir dari proses.
2		Terminal	Menunjukkan permulaan ( <i>start</i> ) atau akhir ( <i>stop</i> ) dari suatu proses.
3		proses	Digunakan untuk menunjukkan kegiatan yang dilakukan oleh komputer. Pada bidang industri (proses produksi barang), simbol ini menggambarkan kegiatan inspeksi atau yang biasa dikenal dengan simbol inspeksi.
4		<i>Manual Operation</i>	Digunakan untuk menunjukkan kegiatan/proses yang tidak dilakukan dengan komputer.
5		<i>Decission/keputusan</i>	Merupakan simbol yang digunakan untuk memilih proses atau keputusan berdasarkan kondisi yang ada, simbol ini biasanya ditemui pada <i>flowchart</i> program.
6		<i>Input/ output</i>	Menunjukkan proses <i>input-output</i> yang terjadi tanpa bergantung dari jenis peralatannya.



No	Simbol	Nama	Keterangan
7		<i>Predefined process/</i> proses terdefinisi	Merupakan simbol yang digunakan untuk menunjukkan pelaksanaan suatu bagian prosedur (sub-proses). Dengan kata lain, prosedur yang terinformasi disini belum detail dan akan dirinci ditempat lain.
8		<i>Connector/On-page</i>	Berfungsi untuk menyederhanakan hubungan antar simbol yang letaknya berjauhan atau rumit bila dihubungkan dengan garis dalam satu halaman.
9		<i>Connector/Off-page</i>	Sama seperti <i>on-page connector</i> , hanya saja symbol ini digunakan untuk menghubungkan simbol dalam halaman berbeda. Label dari simbol ini dapat menggunakan huruf atau angka.
10		<i>Document</i>	Menunjukkan bahwa input berasal dari dokumen dalam bentuk kertas atau <i>output</i> yang perlu dicetak diatas kertas.

No	Simbol	Nama	Keterangan
11		<i>Multiple document</i>	Sama seperti <i>document</i> , hanya saja dokumen yang digunakan lebih dari satu dalam simbol ini.
12		Manual input	Digunakan untuk menunjukkan input data secara manual menggunakan <i>online keyboard</i> .
13		<i>Database</i>	Data yang disimpan secara elektronik dalam <i>database</i>
14		<i>Swimlane</i>	<i>Swimline</i> berguna untuk membagi proses atau langkah-langkah kedalam kategori untuk membedakan aktor yang bertanggungjawab untuk setiap rangkaian, tindakan atau proses.
15		<i>Arsip/storage</i>	Tempat penyimpanan dokumen yang kemungkinan akan diambil kembali untuk keperluan pengelolaan lebih lanjut.
16		<i>External data</i>	Menunjukkan <i>input</i> dan <i>output</i> menggunakan file

Sumber: (Vernanda, 2023)

## 2.7. Unified Modeling Language (UML)




*Unified Modelling Language* (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual. *Unified Modelling Language* (UML) adalah Bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek (Ronald dkk., 2022).

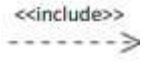



Salah satu dari *tahapan* SDLC atau *software development life cycle*, adalah desain. Desain bertujuan agar *software* yang akan dibuat dapat memenuhi kebutuhan *user* dan handal. Oleh karena itu, desain menjadi tahapan penting dalam proses pembuatan *software* (Sumirat dkk., 2023). Maka, perlu diketahui terdapat beberapa jenis diagram *Unified Model Language* (UML) yaitu *use case diagram*, *class diagram*, *activity diagram*, dan *sequence diagram* (Nistrina & Sahidah, 2022). Dapat disimpulkan bahwa, UML membantu dalam tahap desain perangkat lunak dengan memvisualisasikan struktur dan alur sistem. Berikut penjelasan lebih lanjut terkait jenis-jenis UML yang umum digunakan.

### 2.7.1. Use case Diagram

*Use case diagram* menggambarkan fungsional yang diharapkan dari sebuah sistem. *Use case* bertujuan untuk mempresentasikan interaksi antara *actor* dengan sistem. Aktor adalah suatu entitas manusia yang berinteraksi dengan sistem (Rahmatuloh & Revanda, 2022). Menurut (Sumirat dkk., 2023), *use case* ini bertujuan agar konsumen maupun pembuat dapat saling mengenal dan mengerti mengenai alur sistem yang akan dibuat.

Tabel 2.2. *Symbols Use Case Diagram*

No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Mewakili orang atau sistem lain yang berinteraksi dengan sistem yang akan dibuat.
2		<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, menggambarkan pekerjaan yang dapat memberikan <i>values</i> yang bermanfaat bagi aktor.
4		<i>Generalization</i>	Pewarisan sifat (ada Induk dan anak), anak mewarisi sifat dari induknya.



No	Simbol	Nama	Keterangan
			'Minimal ada 1 <i>use case</i> yang dijalankan'. Bisa juga digunakan untuk generalisasi <i>actor</i> , apapun yang bisa dilakukan oleh <i>actor</i> (induk) pasti bisa dilakukan oleh <i>actor</i> (anak).
5		<i>Include</i>	Sebuah hubungan Dimana 1 atau banyak <i>use case</i> dapat dikerjakan melalui <i>use case</i> sebelumnya.
6		<i>Extend</i>	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi itu terpenuhi. Bersifat <i>optional</i> , boleh dikerjakan boleh tidak
7		<i>Association</i>	Komunikasi antar aktor dan <i>use case</i> (induk/utama) yang berpartisipasi pada <i>use case</i> itu sendiri. Garis yang menghubungkan antara <i>actor</i> dengan <i>use case</i> /kegiatannya.
8		<i>System</i>	Untuk memperlihatkan batasan dari sistem pada diagram <i>use case</i> , <i>actor</i> berada diluar <i>symbol boundary</i> .


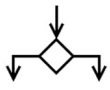
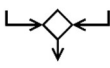

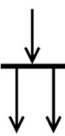
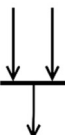
(Vernanda, 2023)

### 2.7.2. *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (Suharni dkk., 2023). Pengembangan dari *usecase* yang memiliki alur / aktivitas. *Activity* menggambarkan aktivitas dan juga entitas dari sistem yang akan dikembangkan (Vernanda, 2023). Berikut adalah elemen-elemen dari *activity diagram* menurut (Indriyani dkk., 2019):

Tabel 2.3. *Symbols Activity Diagram*

No	Simbol	Nama	Deskripsi
1		<i>Initial Node</i>	Menggambarkan awal dari serangkaian tindakan atau kegiatan
2		<i>Final Node</i>	Digunakan untuk menghentikan semua arus kontrol dan arus objek dalam suatu aktivitas (atau tindakan).



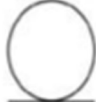

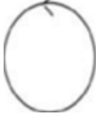

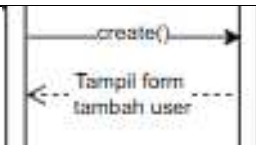
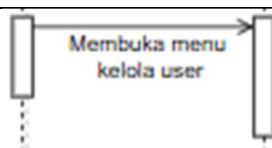
No	Simbol	Nama	Deskripsi
3		<i>Activity</i>	Digunakan untuk mewakili sekumpulan tindakan ( <i>action</i> ). Dengan menggunakan kata kerja.
4		<i>Decision Node</i>	Menunjukkan di mana keputusan akan dibuat
5		<i>Merge node</i>	Digunakan untuk menyatukan kembali berbagai jalur keputusan yang dibuat menggunakan simpul keputusan.
6		<i>Swimlane</i>	Digunakan untuk mengelompokkan aktivitas berdasarkan aktor yang bertanggung jawab.
7		<i>Fork Node</i>	Digunakan untuk membagi suatu proses menjadi beberapa aktivitas yang dapat berjalan secara paralel atau bersamaan.
8		<i>Join Node</i>	Digunakan untuk menyatukan kembali serangkaian arus aktivitas (atau tindakan) yang paralel atau bersamaan.



(Indriyani dkk., 2019)

### 2.7.3. Sequence Diagram

*Sequence diagram* adalah *diagram* yang digunakan untuk menggambarkan dan memodelkan interaksi antar komponen dalam suatu sistem secara beruntun berdasarkan waktu hidup (Sitorus, 2023). Sedangkan, menurut (Mardhia & Khusna, 2020) *Sequence diagram* adalah diagram interaksi yang menggambarkan alur komunikasi antara objek dalam suatu sistem. Diagram ini menampilkan objek sebagai lifeline (jalur kehidupan) yang mengalir ke bawah, dengan pesan yang digambarkan sebagai panah dari satu objek ke objek lainnya seiring waktu. Sequence diagram berguna untuk menunjukkan objek yang berkomunikasi serta pesan yang memicu interaksi tersebut.

Tabel 2.4. *Symbols Sequence Diagram*

No	Simbol	Nama	Keterangan
1		<i>Actor / Aktor</i>	Menggambarkan orang/pengguna yang sedang berinteraksi dengan sistem.
2		<i>Activation box</i>	Mewakili periode dimana suatu elemen melakukan operasi. Bagian atas dan bawah kotak aktivasi disejajarkan dengan inisiasi dan waktu penyelesaian masing-masing.
3		<i>Entity Class</i>	Menggambarkan sebuah penyimpanan data/ <i>table</i> ataupun yang berhubungan dengan <i>database</i> .
4		<i>Boundary Class</i>	Menggambarkan sebuah <i>form</i> / tampilan pada sistem informasi
5		<i>Control Class</i>	Menggambarkan hubungan antara <i>boundary class</i> dengan <i>control class</i>
6		Self Message	Jenis pesan yang mewakili permohonan pesan dari <i>lifeline</i> yang sama.
7		<i>Synchronous message/ Pesan sinkron</i>	Pesan ini digunakan untuk memanggil operasi atau method yang dimiliki oleh suatu objek. Pesan ini mengharuskan menyelesaikan 1 proses baru kemudian dapat memanggil proses berikutnya.
8		<i>Asynchronous message/ Pesan asinkron</i>	Pesan asinkron digunakan untuk memanggil operasi atau <i>method</i> yang dimiliki oleh suatu objek. Asinkron memberikan fasilitas untuk menjalankan

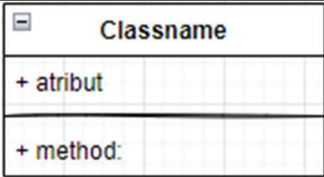
No	Simbol	Nama	Keterangan
			proses lain Ketika proses sebelumnya belum selesai.
9		<i>Reply message/</i> pesan balasan	Menunjukkan nilai Kembali dari obyek ke obyek yang mengirim pesan. Komponen <i>reply message</i> digambarkan dengan symbol panah ke kiri putus-putus yang berarti pesan ini merupakan balasan untuk sebuah panggilan tertentu
		<i>Life line</i>	Mewakili peserta individu dalam interaksi, sebagai respon dalam memproses.




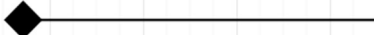
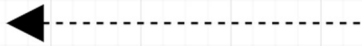
Sumber: (Vernanda, 2023)

#### 2.7.4. Class diagram

*Class diagram* adalah *diagram* yang menggambarkan kelas-kelas yang dibutuhkan sesuai dengan fungsionalitas serta menggambarkan interaksi antar kelas yang ada pada sistem (Ramdani & Zakaria, 2023). *Class diagram* merupakan salah satu jenis diagram struktur pada UML yang bersifat statis, maksudnya adalah diagram kelas tidak menjelaskan apa yang terjadi jika masing-masing kelas berhubungan, namun menjelaskan hubungan apa yang terjadi. *Class diagram* menggambarkan dengan jelas struktur serta deskripsi *class*, atribut, *method*, dan hubungan dari setiap objek (Vernanda, 2023). Dapat disimpulkan bahwa, *Class diagram* menggambarkan struktur sistem, termasuk kelas, atribut, metode, dan hubungan antar kelas tanpa menjelaskan alur interaksi.

Tabel 2.5. *Symbols Class Diagram*

No	Simbol	Nama	Keterangan
1		<i>Class</i>	Himpunan objek-objek dari berbagai atribut yang me-

No	Simbol	Nama	Keterangan
			miliki operasi yang sama
2		<i>Assosiation</i>	Relasi antar kelas dengan makna umum dan biasanya disertai <i>multiplicity</i>
3		<i>Directed Association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain.
4		<i>Aggregation</i>	Mengindikasikan keseluruhan bagian <i>relationship</i> disebut sebagai relasi
5		<i>Composition</i>	Relasi <i>Composition</i> terhadap class tempat dia bergantung
6		<i>Dependency</i>	Menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain

Sumber:(Suharni dkk., 2023)

#### 2.7.5. UI/UX

*User Interface* (UI) adalah elemen penting dalam sistem karena menjadi tampilan yang dioperasikan oleh pengguna. UI mencakup aspek visual seperti warna, animasi, dan tata letak yang dirancang agar mudah digunakan (Octavianti dkk., 2025). Seorang desainer UI merancang tampilan sesuai kebutuhan dasar pengguna, baik untuk aplikasi web maupun mobile. Di sisi lain, *User Experience* (UX) menekankan pada kenyamanan dan kepuasan pengguna selama berinteraksi



dengan aplikasi. UX bertujuan menciptakan pengalaman menarik dan menyenangkan. Dalam era sekarang, aplikasi tidak cukup hanya memiliki kegunaan, tetapi juga harus memberikan pengalaman pengguna yang baik.

Untuk mendesain antarmuka yang efektif, Ben Shneiderman dalam bukunya *Designing the User Interface* memperkenalkan 8 Golden Rules (Wong, 2025), antara lain: menjaga konsistensi, menyediakan pintasan bagi pengguna berpengalaman, memberikan umpan balik yang jelas, menciptakan penutupan dalam dialog, menangani kesalahan dengan sederhana, memungkinkan pembatalan tindakan, memberi pengguna kendali, dan mengurangi beban memori jangka pendek. Prinsip-prinsip ini membantu menciptakan UI/UX yang intuitif, efisien, dan ramah pengguna. *Tools* seperti Figma atau Corel sering digunakan untuk mendesain antarmuka yang memenuhi prinsip-prinsip tersebut.

## 2.8. *Black Box Testing*

*Black box testing* merupakan metode pengujian yang berfokus pada pengujian fungsionalitas sistem tanpa perlu melihat struktur *code* program (Annaufal dkk., 2025). Tahapan awal pada pengujian dengan metode *black box* adalah mengidentifikasi inputan lalu kemudian kita melakukan pengujian untuk mengetahui letak kesalahannya. Proses *black box testing* dilakukan dengan cara mencoba sebuah *software* yang telah selesai dibangun lalu pada setiap formnya akan dicoba memasukan *sample data* pengujian, ini sangat dibutuhkan untuk mengetahui apakah *software* yang dibangun telah berjalan sesuai dengan kebutuhan perusahaan (Shaleh dkk., 2021). Kesimpulannya, *black box testing* adalah metode pengujian yang berfokus pada fungsionalitas sistem tanpa melihat struktur kode program. Pengujian ini dilakukan dengan mengidentifikasi input, lalu menguji apakah sistem berfungsi sesuai dengan yang diharapkan. Dan menurut (Fikri dkk., 2024), untuk menghitung persentase validitas dan *non-validitas* yang nantinya akan ditemukan pada sistem, digunakanlah rumus sebagai berikut:

$$\text{Persentasi valid} = \left( \frac{\text{jumlah skenario valid}}{\text{total skenario uji}} \right) \times 100$$

$$\text{Persentasi tidak valid} = \left( \frac{\text{jumlah skenario tidak valid}}{\text{total skenario uji}} \right) \times 100$$

Keterangan :



- Presentasi *Valid* = Hasil persentase perhitungan yang sesuai harapan (*valid*)
- Persenasi tidak *valid* = Hasil persentase perhitungan yang tidak sesuai harapan (*valid*)
- Jumlah skenario *valid* = total skenario uji yang berhasil melewati pengujian dengan hasil *valid*
- Jumlah skenario tidak *valid* = total skenario uji yang menghasilkan ketidakvalidan dalam *respons* sistem
- Total skenario uji = Seluruh scenario uji yang dilakukan



## 2.9. *Entity Relationship Diagram (ERD)*

*Entity Relationship Diagram* adalah *diagram* yang menjelaskan dan mengidentifikasikan entitas hubungan antar rnititas *data* yang saling berkaitan (Kalsel, 2025). Pemodelan awal basis data yang paling banyak digunakan adalah *Entity Relationship Diagram (ERD)*. ERD digunakan untuk memodelkan basis data relasional. ERD adalah diagram yang menggunakan gammbar atau simbol untuk mengidentifikasi jenis entitas dalam suatu sistem, beserta atributnya, serta menjelaskan hubungan antar entitas tersebut. ERD juga dapat difahami sebagai model jaringan yang merepresentasikan susunan data dalam sistem secara abstrak, dengan fokus pada struktur dan hubungan data. (Hasanah, 2020)

Kesimpulannya, *Entity Relationship Diagram (ERD)* adalah alat yang digunakan untuk memodelkan basis data relasional. ERD menggambarkan entitas dalam suatu sistem, beserta atribut dan hubungannya, menggunakan diagram visual. Dengan demikian, ERD membantu memahami struktur dan keterkaitan data dalam sistem secara lebih jelas dan terstruktur.

Tabel 2.6. Simbol ERD

No	Simbol	Keterangan
1.		Entitas adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
2.		Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.

No	Simbol	Keterangan
3.		Atribut berfungsi mendeskripsikan karakter entitas (atribut yang berfungsi sebagai <i>key</i> diberi garis bawah).
4.		Garis sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.

Sumber:(Triaulia et al., 2021)

## 2.10. Penelitian yang Relevan

Adapun penelitian yang relevan sebagai berikut :

Tabel 2.7. Penelitian yang Relevan

NO	Nama Peneliti	Judul	Masalah	Metode Pengembangan	Hasil	Keterkaitan
1	Shovian, Muhammad Al Baihaqi, Hadi Putra Kharisma, Agi Santoso, Nurudin (Shovian dkk., 2024)	Pengembangan Aplikasi Sistem Informasi Manajemen Gudang Berbasis Web Menggunakan Metodologi Agile (Studi Kasus: CV. Jaya Laksa Lestari)	Sistem manajemen gudang kurang efisien dan tidak terintegrasi	<i>Agile Software Development</i>	MeAplikasi manajemen gudang berbasis <i>web</i> dengan fitur pengelolaan stok, penerimaan, pengiriman, dan laporan inventaris untuk meningkatkan efisiensi operasional dan pelayanan dalam pengelolaan gudang secara digital	Relevan dengan pengembangan sistem <i>checklist crane</i> , terutama dalam manajemen stok dan pencatatan data <i>crane</i>
2	Max, Rudy Gugat, Damara (Max & Gugat, 2023)	Rancang Bangun Sistem Revitalisasi Manajemen Gudang Logistik melalui Penerapan Sistem Informasi Persediaan Digital	Efisiensi dan pelayanan dalam manajemen gudang logistik masih kurang optimal	Studi Literatur	Studi Literatur sistem informasi <i>inventory</i> berbasis <i>web</i> yang meningkatkan efisiensi operasional dan manajemen keuangan toko	Relevan dalam hal optimasi pencatatan dan efisiensi manajemen <i>checklist crane</i>
3	Widiarta, I Made Mulyanto, Yudi Sutrianto, Arif (Widiarta dkk., 2023)	Rancang Bangun Sistem Informasi Inventory Menggunakan Metode Agile Software Development (Studi Kasus Toko Nada)	Pengelolaan persediaan barang yang tidak akurat dan efisien	<i>Agile Software Development</i>	Sistem informasi <i>inventory</i> berbasis <i>web</i> yang meningkatkan efisiensi operasional dan manajemen keuangan toko aplikasi manajemen gudang berbasis <i>web mobile</i> yang meningkatkan efektivitas pengelolaan data barang	Berkaitan dengan manajemen persediaan <i>crane</i> dan pemantauan alat yang digunakan dalam produksi

NO	Nama Peneliti	Judul	Masalah	Metode Pengembangan	Hasil	Keterkaitan
4	Wau, Kebenaran (Wau, 2022)	Pengembangan Sistem Informasi Persediaan Gudang Berbasis Website dengan Metode Waterfall	Pengelolaan persediaan gudang yang tidak terkomputerisas	<i>Waterfall</i>	Sistem informasi persediaan gudang berbasis <i>web</i> yang memudahkan pengelolaan data persediaan	Berkaitan dengan kebutuhan pencatatan kapasitas <i>crane</i> yang lebih sistematis dan <i>digital</i>
5	Novianti, A Sari, R P (Novianti & Sari, 2022)	Perancangan Sistem Gudang Material dengan Metode FAST pada PT. SAMCON	Pencatatan persediaan <i>manual</i> yang menyebabkan <i>human error</i> dan ketidakefisienan	<i>Metode Framework for the Application of Systems Thinking (FAST)</i>	Sistem informasi pergudangan yang mengurangi waktu pengambilan produk dan meminimalkan kesalahan manusia	Bisa diterapkan dalam pengelolaan dan <i>tracking data checklist crane</i> untuk menghindari <i>human error</i>
6	Farhan, Ahmad Wahab, Abdul Ri, Faisal M, Frederick Nehemia Alman, Hapifuddin (Farhan dkk., 2024)	Optimisasi Pagination dan Error Handling pada Portal Minat untuk Meningkatkan Performa Aplikasi Menggunakan Whitebox Testing	Performa aplikasi yang menurun akibat penanganan data besar dan penanganan kesalahan yang kurang optimal	<i>Whitebox Testing</i>	Implementasi pagination dan perbaikan mekanisme penanganan kesalahan yang meningkatkan performa aplikasi	Memberikan wawasan tentang teknik penanganan kesalahan yang efektif untuk meningkatkan performa sistem
7	Aini, Nurul Ratnawati, Fajar (Aini & Ratnawati, 2024)	Implementasi Metode McCall pada Pengujian Kualitas Website Diskominfo Kabupaten Bengkalis	Kualitas <i>website</i> yang belum terukur secara komprehensif	Metode McCall	Evaluasi kualitas <i>website</i> berdasarkan faktor-faktor McCall yang menghasilkan rekomendasi perbaikan	Menyediakan pendekatan untuk mengukur dan meningkatkan kualitas perangkat lunak menggunakan model McCall

## BAB 3

### METODOLOGI PENELITIAN

#### 3.1. Evaluasi Mc Call's Model

Evaluasi kualitas perangkat lunak dalam penelitian ini menggunakan McCall's Software Quality Model, yang mengkategorikan kualitas perangkat lunak ke dalam beberapa atribut utama, termasuk *Reliability*, *Efficiency*, dan *Integrity* (Boehm, 1978). Evaluasi dilakukan dengan dua pendekatan, yaitu pengujian berbasis perangkat lunak dan pengujian berbasis kuesioner.

1. Metode pertama adalah pengujian berbasis perangkat lunak, yang menggunakan beberapa alat bantu evaluasi sebagai berikut: *Reliability* Testing menggunakan WAPT, yang berfungsi untuk mengukur keandalan dan ketahanan sistem dalam menangani beban kerja secara berulang dan berkelanjutan (Kitchenham, 1996). Pengujian ini penting untuk memastikan bahwa sistem tetap berfungsi dengan baik dalam berbagai kondisi operasional.
2. *Efficiency* Testing menggunakan Google Lighthouse, yang mengevaluasi kinerja aplikasi berbasis web, termasuk kecepatan loading halaman, optimasi kode, dan efisiensi penggunaan sumber daya sistem (Al-Qutaish, 2010). Evaluasi ini membantu dalam mengidentifikasi potensi perbaikan dalam aspek performa aplikasi.
3. *Integrity* Testing menggunakan Sucuri SiteCheck, yang menganalisis keamanan aplikasi dan mendeteksi kemungkinan adanya ancaman eksternal yang dapat mengganggu integritas data atau menyebabkan kebocoran informasi (Boehm, 1978).

Selain pengujian berbasis perangkat lunak, metode kedua yang digunakan adalah pengujian berbasis kuesioner, yang bertujuan untuk memperoleh persepsi pengguna terkait kualitas perangkat lunak. Menurut Al-Qutaish (2010), pendekatan berbasis kuesioner merupakan metode yang efektif dalam mengukur kualitas perangkat lunak dari sudut pandang pengguna akhir. Pertanyaan dalam kuesioner difokuskan pada aspek reliabilitas, efisiensi, dan integritas sistem yang diuji.

Dalam penerapan Mc Call's Model ini diterapkan tiga kategori pengujian yaitu :

1. *Product Operation*

Aspek ini berfokus pada sejauh mana perangkat lunak dapat beroperasi secara optimal dan dapat diterima oleh pengguna dalam kondisi nyata.

a. *Reliability* (Keandalan)

Reliabilitas adalah kemampuan sistem untuk menjalankan fungsinya secara konsisten tanpa mengalami kegagalan. Pengujian ini bertujuan untuk mengetahui seberapa sering sistem mengalami kesalahan, *crash*, atau kehilangan data, baik dalam kondisi normal maupun saat berada di bawah tekanan atau beban tinggi. Sistem yang andal akan tetap berjalan stabil dalam waktu lama dan dalam berbagai kondisi penggunaan.

b. *Efficiency* (Efisiensi)

Efisiensi mengacu pada kemampuan perangkat lunak dalam menggunakan sumber daya seperti CPU, memori, dan waktu pemrosesan secara optimal. Pengujian ini bertujuan untuk menilai apakah sistem mampu merespons dengan cepat, memiliki waktu muat yang singkat, dan tidak membebani perangkat keras. Sistem yang efisien memberikan pengalaman pengguna yang lebih baik dan meringankan beban perangkat.

c. *Integrity* (Integritas)

Integritas berkaitan dengan keamanan dan perlindungan terhadap data serta akses sistem. Pengujian ini mengevaluasi apakah sistem mampu menjaga data dari akses yang tidak sah, manipulasi, atau perusakan. Tujuannya adalah memastikan bahwa data yang disimpan tetap utuh dan tidak mudah disalahgunakan.

d. *Correctness* (Kebenaran Fungsi)

*Correctness* atau ketepatan fungsi adalah sejauh mana sistem memenuhi kebutuhan dan spesifikasi yang telah ditentukan. Pengujian ini bertujuan untuk mengetahui apakah fitur-fitur yang

diimplementasikan sesuai dengan harapan dan berjalan dengan benar tanpa kesalahan logika maupun eksekusi.

e. *Reliability* (Persepsi Pengguna)

Selain pengujian teknis, persepsi pengguna juga penting dalam menilai keandalan sistem. Pengujian ini dilakukan dengan mengumpulkan pendapat pengguna terkait seberapa sering mereka mengalami masalah saat menggunakan sistem dan seberapa yakin mereka bahwa sistem akan berjalan baik setiap kali digunakan.

f. *Efficiency* (Persepsi Pengguna)

Pengujian ini mengevaluasi pengalaman pengguna terhadap kecepatan dan *respons* sistem saat digunakan. Ini mencakup waktu yang dibutuhkan untuk memuat halaman, memproses data, atau menyelesaikan suatu tugas, berdasarkan persepsi pengguna.

g. *Integrity* (Persepsi Pengguna)

Digunakan untuk menilai tingkat kepercayaan pengguna terhadap sistem, terutama dalam hal keamanan data pribadi, transaksi, dan aktivitas penting lainnya. Pengguna akan memberikan penilaian apakah mereka merasa aman menggunakan sistem tersebut.

h. *Usability* (Kemudahan Penggunaan)

*Usability* mengacu pada seberapa mudah dan nyaman sistem digunakan oleh pengguna. Pengujian ini mencakup aspek kemudahan navigasi, kejelasan antarmuka, kecepatan belajar menggunakan aplikasi, dan pengalaman secara umum. Tujuannya adalah memastikan bahwa sistem dapat digunakan oleh berbagai kalangan pengguna tanpa kesulitan berarti.

2. *Product Revision*

Aspek ini menilai kemampuan sistem dalam menghadapi perubahan dan pemeliharaan seiring berjalannya waktu.

a. *Maintainability* (Kemudahan Pemeliharaan)

Pengujian ini bertujuan untuk menilai seberapa mudah sistem dapat diperbaiki jika terjadi kesalahan atau *bug*, serta seberapa cepat pengembang dapat memahami dan memodifikasi kode sumber. Sistem



yang mudah dipelihara akan mempercepat proses *debugging* dan pengembangan di masa depan.

b. *Flexibility* (Fleksibilitas)

Fleksibilitas merujuk pada kemampuan sistem untuk beradaptasi terhadap perubahan, baik dari sisi kebutuhan pengguna, teknologi, maupun lingkungan operasional. Pengujian ini mengukur sejauh mana sistem dapat diubah tanpa menimbulkan gangguan besar terhadap fungsi yang sudah ada.

c. *Testability* (Kemudahan Pengujian)

Aspek ini menilai seberapa mudah sistem dapat diuji untuk memastikan kebenaran fungsi dan stabilitasnya. Sistem yang memiliki struktur yang modular dan terdokumentasi dengan baik akan lebih mudah diuji, baik secara manual maupun otomatis.

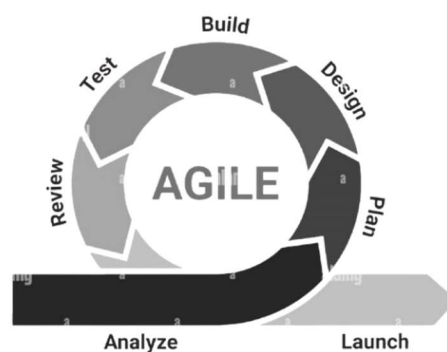
3. *Product Transition*

Aspek ini berkaitan dengan kemampuan sistem untuk diadaptasi atau diintegrasikan dengan lingkungan baru atau sistem lain.

a. *Maintainability* (Kemudahan Pemeliharaan)

Pengujian ini bertujuan untuk menilai seberapa mudah sistem dapat diperbaiki jika terjadi kesalahan atau bug, serta seberapa cepat pengembang dapat memahami dan memodifikasi kode sumber. Sistem yang mudah dipelihara akan mempercepat proses *debugging* dan pengembangan di masa depan.

### 3.2. Metodologi



Gambar 3.1. Agile Methodology

Dalam melakukan pengembangan Sistem Informasi Manajemen Operasional PT Tata Metal Lestari ini, metodologi yang digunakan adalah Agile dengan *framework* Scrum yang memiliki adaptabilitas dan fleksibilitas yang tinggi (Hilmyansyah dkk., 2022). Secara singkat metodologi scrum adalah Scrum adalah kerangka kerja dalam pengembangan perangkat lunak yang mengadopsi prinsip-prinsip *Agile* (Hafidhin dkk., 2025).

Pengaplikasian metode *Agile* meliputi beberapa tahapan yang perlu dilaksanakan diantaranya :

1) Perencanaan (*Planning*)

- a. Menentukan kebutuhan proyek.menggunakan metode Mc Call's
- b. Menyusun *backlog* produk yang berisi fitur-fitur yang akan dikembangkan.
- c. Mengestimasi waktu dan sumber daya yang dibutuhkan untuk setiap fitur.

2) Desain (*Design*)

- a. Merancang arsitektur sistem, proses bisnis, *basis data*, dan antarmuka pengguna.
- b. Menggunakan alat bantu seperti *Unified Modeling Language* (UML) dan *Entity Relationship Diagram* (ERD) untuk memvisualisasikan desain.

3) Pengembangan (*Development*)

- a. Menulis kode program sesuai dengan desain yang telah dibuat.
- b. Melakukan integrasi komponen-komponen sistem.

4) Pengujian (*Testing*)

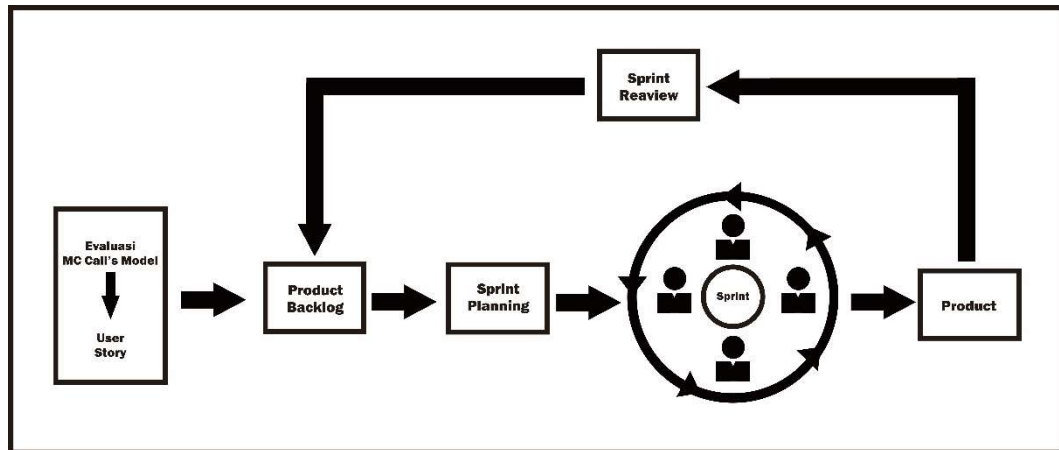
- a. Melakukan pengujian fungsional, integrasi, dan performa untuk memastikan sistem berfungsi sesuai dengan spesifikasi.
- b. Mengidentifikasi dan memperbaiki *bug* atau kesalahan yang ditemukan.

5) Penyebaran (*Deployment*)

- a. Mengimplementasikan sistem ke lingkungan produksi atau pengguna akhir.
- b. Memastikan sistem berjalan dengan baik di lingkungan operasional.

6) Pemeliharaan (*Maintenance*)

- a. Memantau kinerja sistem dan melakukan perbaikan atau pembaruan jika diperlukan.
- b. Menangani masalah yang muncul setelah sistem digunakan oleh pengguna.



Gambar 3.2. *Scrum Framework*

Berdasarkan metode diatas tahapan yang akan dilakukan dengan menerapkan metodologi *Agile* dengan *framework* Scrum adalah sebagai berikut :

1) *Product Backlog*

Pada tahap ini di buat perencanaan daftar kebutuhan sistem yang akan dikembangkan oleh pengembang. Mencakup list fitur yang akan dikembangkan.

2) *Sprint Planning*

Tahap ini dilakukan perencanaan pembagian backlog kedalam setiap sprintnya yang akan menghasilkan *sprint backlog*. Dalam *sprint planning* ditentukan *sprint backlog* yang akan di capai dalam setiap *sprint*

3) *Sprint Execution*

Tahap ini merupakan tahap eksekusi dari *sprint planning* yang mengacu kepada *sprint backlog*. Setelah setiap *sprint* selesai diadakan *sprint review* untuk menentukan *backlog* sudah tercapai atau belum.

4) *Increment*

Pada tahap ini *backlog* pada *sprint* telah selesai dikerjakan, apabila hasilnya telah sesuai maka *backlog* siap di rilis, namun apabila belum memenuhi *backlog* maka akan dilakukan *sprint* kembali setelah dilakukan

*sprint review* kepada *stakeholders*. Fitur akan dianggap *increment* apabila sudah memnuhi *criteria* yang ditentukan oleh *stakeholders*.

5) *Sprint Retrospective*

Pada tahap ini dilakukan evaluasi untuk setiap pengerjaan *sprint*. Tujuan dari dilakukannya tahap ini adalah untuk meningkatkan kefisiensi dan kualitas kerja dibandingkan dengan *sprint* sebelumnya.

### 3.3. Jadwal PA

Tabel 3.1. Jadwal PA

NO	Kegiatan	Februari		Maret				April			
		3	4	1	2	3	4	1	2	3	4
1	<b><i>Sprint 1 (Analisis &amp; Perencanaan)</i></b>										
	Evalasi Mc Call's Model										
	Analisis Kebutuhan berdasarkan hasil evaluasi										
	Penyusunan Product Backlog										
2	<b><i>Sprint 2 (Checklist Crane &amp; Shift Leader)</i></b>										
	Implementasi <i>Checklist Crane &amp; Shift Leader</i>										
	<i>Sprint review</i> dengan <i>stakeholder</i>										
	<i>Testing &amp;</i> evaluasi awal										
	Pemodelan uml										
3	<b><i>Sprint 3 (Error Handling &amp; Review)</i></b>										
	Implementasi <i>Error Handling</i>										
	<i>Sprint review</i> dengan <i>stakeholder</i>										
	Pemodelan uml										
	Testing										
4	<b><i>Sprint 4 (Penyempurnaan Mapping Muat &amp; Finalisasi)</i></b>										
	Menghilangkan penggunaan spk										
	<i>Sprint review</i> dengan <i>stakeholder</i>										
	Pembuatan user <i>manual</i>										
	Pemodelan uml										
	Testing										

## BAB 4

### ANALISIS DAN PERANCANGAN

#### 4.1. Analisis

Analisis merupakan suatu proses yang dilakukan untuk mencari tahu kebutuhan apa saja yang diperlukan dalam mengidentifikasi permasalahan sampai ke menentukan solusi. Analisis yang dilakukan bertujuan untuk mengidentifikasi permasalahan pada Sistem Informasi Manajemen Operasional dan menentukan solusi untuk pengembangan sistem tersebut.

Untuk melakukan analisis tersebut digunakan metode pendekatan Mc Call's Model. Dari pendekatan tersebut ditemukan beberapa masalah yang harus diselesaikan seperti *server* yang tidak bisa menyediakan data dengan cepat, *error handling* yang tidak tepat, penerapan spk yang kurang tepat dan masalah lainnya di bahas pada bagian implementasi Mc Call's Model.

#### 4.2. Mc Call's Model

Dalam penelitian ini, pendekatan dilakukan menggunakan Mc Call's Model untuk menemukan masalah masalah yang terdapat pada Sistem Informasi Digital Warehouse, berikut hasil dari penerapan Mc Call's Model :

- 1) Pengujian *reliability* menggunakan tool WAPT (Web Application Load, Stress, and Performance Testing) adalah sebesar 100%, jumlah keseluruhan dari sesi, halaman, dan hits yang berhasil adalah 3786, sementara jumlah total yang gagal adalah 0, maka dari itu pada pengujian ini mendapatkan kategori "Sangat Baik". Menunjukkan bahwa sistem memenuhi standar kehandalan perangkat lunak ketika diuji dengan tool WAPT.
- 2) Pengujian *efficiency* menggunakan tool Google Lighthouse adalah sebesar 53.3%, Selain itu, total *Largest Contentful Paint* (LCP) tercatat sebesar 44,8 detik, dengan rata-rata LCP per pengujian sebesar 3,78 detik. Dari hasil uji tersebut, dapat disimpulkan bahwa nilai *efficiency* yang diambil dari skor performa adalah sebesar 53.3%, maka dari itu pada pengujian ini mendapatkan kategori "Sangat Kurang". Menunjukkan bahwa sistem memiliki performa yang buruk.
- 3) Pengujian *Integrity* menggunakan Sucuri Sitehack mendapatkan medium *security risk*, dikarenakan penggunaan nama domain, namun menggunakan

IP langsung dari sistem tersebut, hal tersebut menurunkan tingkat keamanan yang ada pada sistem, namun hal tersebut sejalan dengan kebutuhan perusahaan yang tidak ingin sistem tersebut dapat diakses secara *public*.

- 4) Pengujian *correctnes* berdasarkan hasil kuisioner mendapatkan nilai 95% yang mengindikasikan interpretasi “Sangat Baik”. Dalam penghitungan skor ideal yang didapatkan adalah 80, sementara skor aktual mendapatkan skor 76, skor ideal ditentukan dengan asumsi semua jawaban responden adalah “Sangat Setuju” (nilai 4), Hasil ini menunjukkan bahwa aspek *correctnes* sudah sangat baik namun masih diperlukan pemeliharaan untuk mendapatkan nilai yang sempurna.
- 5) Pengujian *Realibility* berdasarkan hasil kuisioner mendapatkan nilai 90% yang mengindikasikan interpretasi “Sangat Baik”. Dalam penghitungan skor ideal yang didapatkan adalah 80, sementara skor aktual mendapatkan skor 72, skor ideal ditentukan dengan asumsi semua jawaban responden adalah “Sangat Setuju” (nilai 4), Hasil ini menunjukkan bahwa aspek *Realibility* sudah sangat baik namun masih diperlukan pemeliharaan dan perhatian lebih lanjut untuk mendapatkan nilai yang sempurna.
- 6) Pengujian *efficiency* berdasarkan hasil kuisioner mendapatkan nilai 81.25% yang mengindikasikan interpretasi “Baik”. Dalam penghitungan skor ideal yang didapatkan adalah 80, sementara skor aktual mendapatkan skor 65, skor ideal ditentukan dengan asumsi semua jawaban responden adalah “Sangat Setuju” (nilai 4), Hasil ini menunjukkan bahwa aspek *efficiency* sudah sangat baik namun masih diperlukan perbaikan dan perhatian lebih lanjut untuk mendapatkan nilai yang sempurna.
- 7) Pengujian *integrity* berdasarkan hasil kuisioner mendapatkan nilai 93.75% yang mengindikasikan interpretasi “Sangat Baik”. Dalam penghitungan skor ideal yang didapatkan adalah 80, sementara skor aktual mendapatkan skor 75, skor ideal ditentukan dengan asumsi semua jawaban responden adalah “Sangat Setuju” (nilai 4), Hasil ini menunjukkan bahwa aspek *integrity* sudah sangat baik namun masih diperlukan perbaikan dan perhatian lebih lanjut untuk mendapatkan nilai yang sempurna.

- 8) Pengujian *usability* berdasarkan hasil kuisioner mendapatkan nilai 85% yang mengindikasikan intrpretasi “Baik”. Dalam penghitungan skor ideal yang didapatkan adalah 80, sementara skor aktual mendapatkan skor 68 , skor ideal ditentukan dengan asumsi semua jawaban responden adalah "Sangat Setuju" (nilai 4), Hasil ini menunjukkan bahwa aspek *usability* sudah sangat baik namun masih diperlukan perbaikan dan perhatian lebih lanjut untuk mendapatkan nilai yang sempurna.
- 9) Pengujian *Maintainability* berdasarkan hasil kuisioner mendapatkan nilai 72.5% yang mengindikasikan intrpretasi “Cukup Kurang”. Dalam penghitungan skor ideal yang didapatkan adalah 80, sementara skor aktual mendapatkan skor 58 , skor ideal ditentukan dengan asumsi semua jawaban responden adalah "Sangat Setuju" (nilai 4), Hasil ini menunjukkan bahwa aspek *Maintainability* sudah sangat baik namun masih diperlukan perbaikan dan perhatian lebih lanjut untuk mendapatkan nilai yang sempurna.
- 10) Pengujian *flexibility* berdasarkan hasil kuisioner mendapatkan nilai 82.5% yang mengindikasikan intrpretasi “Baik”. Dalam penghitungan skor ideal yang didapatkan adalah 80, sementara skor aktual mendapatkan skor 66 , skor ideal ditentukan dengan asumsi semua jawaban responden adalah "Sangat Setuju" (nilai 4), Hasil ini menunjukkan bahwa aspek *flexibility* sudah sangat baik namun masih diperlukan perhatian lebih lanjut untuk mendapatkan nilai yang sempurna.
- 11) Pengujian *testability* berdasarkan hasil kuisioner mendapatkan nilai 97.5% yang mengindikasikan intrpretasi “Sangat Baik”. Dalam penghitungan skor ideal yang didapatkan adalah 80, sementara skor aktual mendapatkan skor 78 , skor ideal ditentukan dengan asumsi semua jawaban responden adalah "Sangat Setuju" (nilai 4), Hasil ini menunjukkan bahwa aspek *testability* sudah sangat baik namun masih diperlukan pemeliharaan untuk mendapatkan nilai yang sempurna.
- 12) Pengujian *portability* berdasarkan hasil kuisioner mendapatkan nilai 96.5% yang mengindikasikan intrpretasi “Sangat Baik”. Dalam penghitungan skor ideal yang didapatkan adalah 80, sementara skor aktual mendapatkan skor 77 , skor ideal ditentukan dengan asumsi semua jawaban responden adalah



"Sangat Setuju" (nilai 4), Hasil ini menunjukkan bahwa aspek *portability* sudah sangat baik namun masih diperlukan pemeliharaan untuk mendapatkan nilai yang sempurna.

13) Pengujian *reusability* berdasarkan hasil kuisisioner mendapatkan nilai 100% yang mengindikasikan intrpretasi "Sangat Baik". Dalam penghitungan skor ideal yang didapatkan adalah 80, sementara skor aktual mendapatkan skor 80 , skor ideal ditentukan dengan asumsi semua jawaban responden adalah "Sangat Setuju" (nilai 4), Hasil ini menunjukkan bahwa aspek *reusability* sudah sangat baik.

14) Pengujian *interoperability* berdasarkan hasil kuisisioner mendapatkan nilai 82.5% yang mengindikasikan intrpretasi "Baik". Dalam penghitungan skor ideal yang didapatkan adalah 80, sementara skor aktual mendapatkan skor 6 , skor ideal ditentukan dengan asumsi semua jawaban responden adalah "Sangat Setuju" (nilai 4), Hasil ini menunjukkan bahwa aspek *interoperability* sudah sangat baik namun masih diperlukan perbaikan dan pemeliharaan untuk mendapatkan nilai yang sempurna.

15) Rata-rata kualitas dari hasil kuisioer

$$\text{Rata - Rata Kualitas} = \frac{95+90+81.25+93.75+85+72.5+82.5+97.5+96.25+100+82.5}{11} = 88.75\%$$

Rata-rata kualitas dari berbagai faktor yang diuji adalah sekitar 88.75%, yang menunjukkan bahwa secara keseluruhan, website tersebut memiliki kualitas yang baik. Faktor-faktor seperti *maintainability*, *testability*, *flexibility*, *reusability*, dan *interoperability* memberikan kontribusi positif yang signifikan terhadap kualitas keseluruhan perangkat lunak.

16) Saran Tambahan dari hasil kuisisioner

Disertakan juga untuk memberikan saran perbaikan pada form kuisisioner dan menghasil beberapa saran yaitu peningkatan kapasatisa crane, perbaikan *error handling*, dan penerapan SPK yang kurang sesuai.

Berdasarkan hasil pengujian yang telah dilakukan, beberapa aspek kualitas sistem telah dinilai dengan berbagai metode. Berikut adalah rekomendasi perbaikan yang difokuskan pada aspek dengan nilai rendah:

1. Peningkatan Kapasitas *Crane*: Disarankan untuk mengevaluasi dan meningkatkan kapasitas crane guna mengoptimalkan kinerja sistem.

2. Perbaikan *Error Handling*: Perlu diterapkan mekanisme *error handling* yang lebih baik untuk mengurangi potensi kesalahan yang tidak terdeteksi.
3. Evaluasi Penerapan SPK: Perlu dilakukan peninjauan ulang terhadap penerapan SPK agar lebih sesuai dengan kebutuhan operasional.
4. Peningkatan Maintainability: Nilai maintainability yang masih cukup rendah menunjukkan perlunya dokumentasi kode yang lebih baik, refactoring kode untuk meningkatkan keterbacaan, serta penerapan standar pengkodean yang lebih konsisten.
5. Optimasi Efisiensi: Nilai efisiensi masih kurang optimal, terutama dalam hal kecepatan loading halaman. Perbaikan dapat dilakukan dengan optimasi gambar, penggunaan *caching*, serta memperbaiki performa backend.
6. Peningkatan Usability: Website masih kurang dalam menyediakan dokumentasi dan bantuan *online*. Direkomendasikan untuk menambahkan halaman khusus yang berisi panduan penggunaan dan dokumentasi sistem agar lebih mudah diakses oleh pengguna.

Dengan menerapkan rekomendasi ini, diharapkan sistem dapat memiliki kualitas yang lebih baik dan memberikan pengalaman pengguna yang lebih optimal.

#### **4.3. Product Backlog**

Pada tahap ini di buat perencanaan daftar kebutuhan sistem yang akan dikembangkan oleh pengembang. Mencakup *list* fitur yang akan dikembangkan.

Untuk menentukan product backlog disusun *user story* sebagai berikut :

- a. *User Story* ;
  1. Sebagai pengguna sistem di divisi produksi
  2. Saya ingin dapat mencatat kapasitas *crane* yang digunakan dalam *checklist crane*,
  3. Sehingga saya dapat memastikan kapasitas *crane* terdokumentasi dengan baik untuk evaluasi dan pemeliharaan.
- b. *User Story* :
  1. Sebagai *shift leader*,
  2. Saya ingin dapat mencatat nama saya pada *checklist crane*,
  3. Sehingga saya memiliki tanggung jawab yang lebih jelas dalam proses *checklist crane*.

c. *User Story* :

1. Sebagai pengguna sistem,
2. Saya ingin mendapatkan pesan *error* yang lebih informatif dan mudah dipahami,
3. Sehingga saya dapat mengetahui penyebab kesalahan dan cara mengatasinya dengan lebih cepat.

d. *User Story* :

1. Sebagai pengguna sistem di divisi *warehouse*,
2. Saya *ingin* proses *mapping* muat dilakukan tanpa menggunakan SPK,
3. Sehingga sistem lebih sesuai dengan kondisi proses bisnis di lapangan berdasarkan hasil UAT.

*Acceptance Criteria* :

1. SPK tidak lagi menjadi bagian dari proses *Mapping* Muat.
2. Penghapusan SPK tidak menyebabkan *error* pada fitur lain yang terkait.
3. Sistem tetap berjalan sesuai kebutuhan operasional di lapangan.

Berdasarkan *user story* diatas dapat dirumuskan *product backlog* sebagai berikut :

Tabel 4.1. *Product Backlog*

No	<i>Backlog Item</i>	Prioritas	Keterangan
1	Penambahan kapasitas crane milik divisi produksi pada <i>checklist crane</i> di fitur <i>form checklist</i>	<i>High</i>	Menyesuaikan dengan kebutuhan operasional di lapangan.
2	Penambahan <i>shift leader</i> pada <i>checklist crane</i> di fitur <i>form checklist</i>	<i>High</i>	Mempermudah identifikasi pengawasan dan tanggung jawab.
3	Penambahan <i>error handling</i> yang lebih baik	<i>High</i>	Meningkatkan stabilitas dan keandalan sistem.
4	Menghilangkan penggunaan SPK pada <i>Mapping Muat</i>	<i>Medium</i>	Berdasarkan hasil <i>User Acceptance Testing</i> (UAT), SPK tidak sesuai dengan proses bisnis di divisi warehouse PT Tata Metal Lestari.

#### 4.4. *Sprint Planning*

Pada tahap ini dilakukan pembagian *product backlog* kedalam setiap *sprint* untuk nanti dikerjakan sesuai dengan *sprint* yang telah ditentukan. Setiap *Sprint*

dilakukan dalam jangka waktu antara 3-4 minggu dan dilakukan dalam total 4 sprint, berikut pembagian *product backlog* kedalam *sprint* :

- 1) Sprint 1 Analisis & Perencanaan
- 2) Sprint 2 *Checklist Crane & Shift Leader*
  1. Penambahan kapasitas *crane* milik divisi produksi pada *checklist crane* di fitur *form checklist*
  2. Penambahan *shift leader* pada *checklist crane* di fitur *form checklist*
- 3) Sprint 3 *Error Handling & Review*
  1. Penambahan *error handling* yang lebih baik
- 4) Sprint 4 Penyempurnaan *Mapping Muat & Finalisasi*
  1. Menghilangkan penggunaan SPK pada *Mapping Muat*

#### **4.5. Studi Literatur**

Pada tahap ini, kegiatan yang dilakukan adalah mempelajari dan meneliti berbagai sumber bacaan yang memiliki relevansi dengan permasalahan yang diangkat dalam penelitian. Sumber-sumber tersebut mencakup buku, jurnal ilmiah, hingga referensi penelitian terdahulu yang berkaitan dengan pengembangan sistem informasi. Berdasarkan pemahaman yang didapatkan penelitian ini dilakukan dengan menerapkan metode penelitian *Agile* dan menggunakan *framework* scrum untuk alur kerja yang tidak terpaku dengan urutan dan lebih flexible, serta digunakannya metode pendekatan menggunakan Mc Call's Model untuk mendeskripsikan masalah masalah yang ada pada Sistem Manajemen Operasional

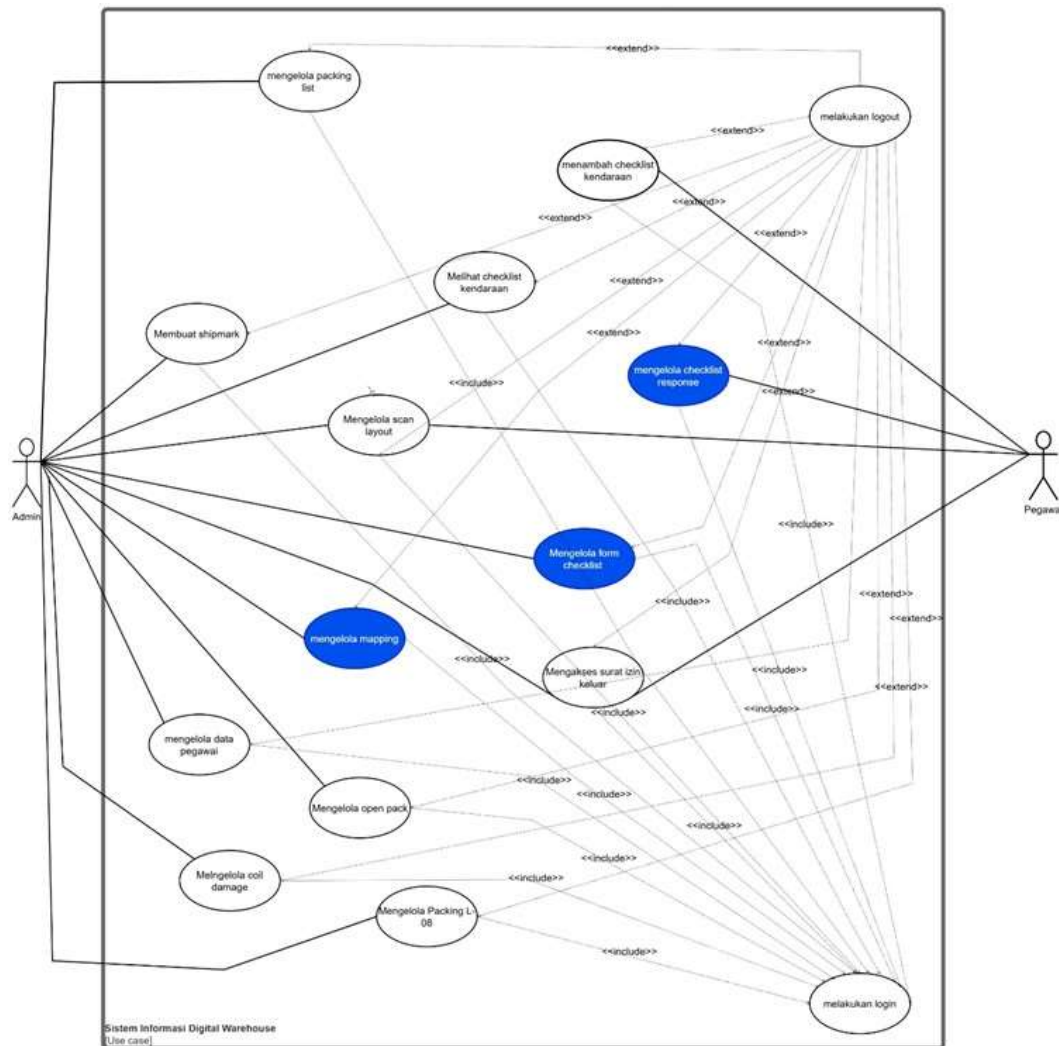
#### **4.6. Perancangan**

Pada tahap ini dilakukan perancangan sistem untuk membuat acuan bagaimana sistem akan dikembangkan. Perancangan ini dilakukan dengan menggunakan diagram diagram yang ada pada UML (*Unified Modeling Language*) diantaranya, yaitu : *usecase diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*. Kemudian di lakukan juga perancangan mengenai basis data dan perancangan antar muka sistem.

#### 4.6.1. Perancangan Sistem UML

##### 4.6.1.1. *Usecase Diagram*

Gambar Dibawah ini menunjukan *usecase* diagram dari Sistem Informasi Manajemen Operasional. *Usecase* diagram tersebut mencakup keseluruhan dari sistem namun yang akan dikembangkan di tandai dengan warna latar *usecase* yang berbeda yaitu berwarna biru.



Gambar 4.1. *Usecase Diagram*

Pada Gambaran use case tersebut terdapat 15 usecase dengan 2 aktor yaitu admin dan pegawai. Usecase yang akan di perbarui pada Sistem Informasi Manajemen Operasional adalah mengelola *mapping* yaitu untuk melakukan pengecekan kondisi koil pada saat dimuat ke dalam *container* ataupun *trailer*, kemudian *form checklist* yaitu untuk melakukan pengecekan alat alat berat yang

digunakan oleh pegawai, dan mengelola *checklist* response yaitu untuk melakukan pengeolahan *data* terhadap respons yang di kirimkan oleh pegawai pada fitur *form checklist*. Berikut adalah penjelasan mengenai actor yang terlibat pada sistem informasi manajemen operasional.

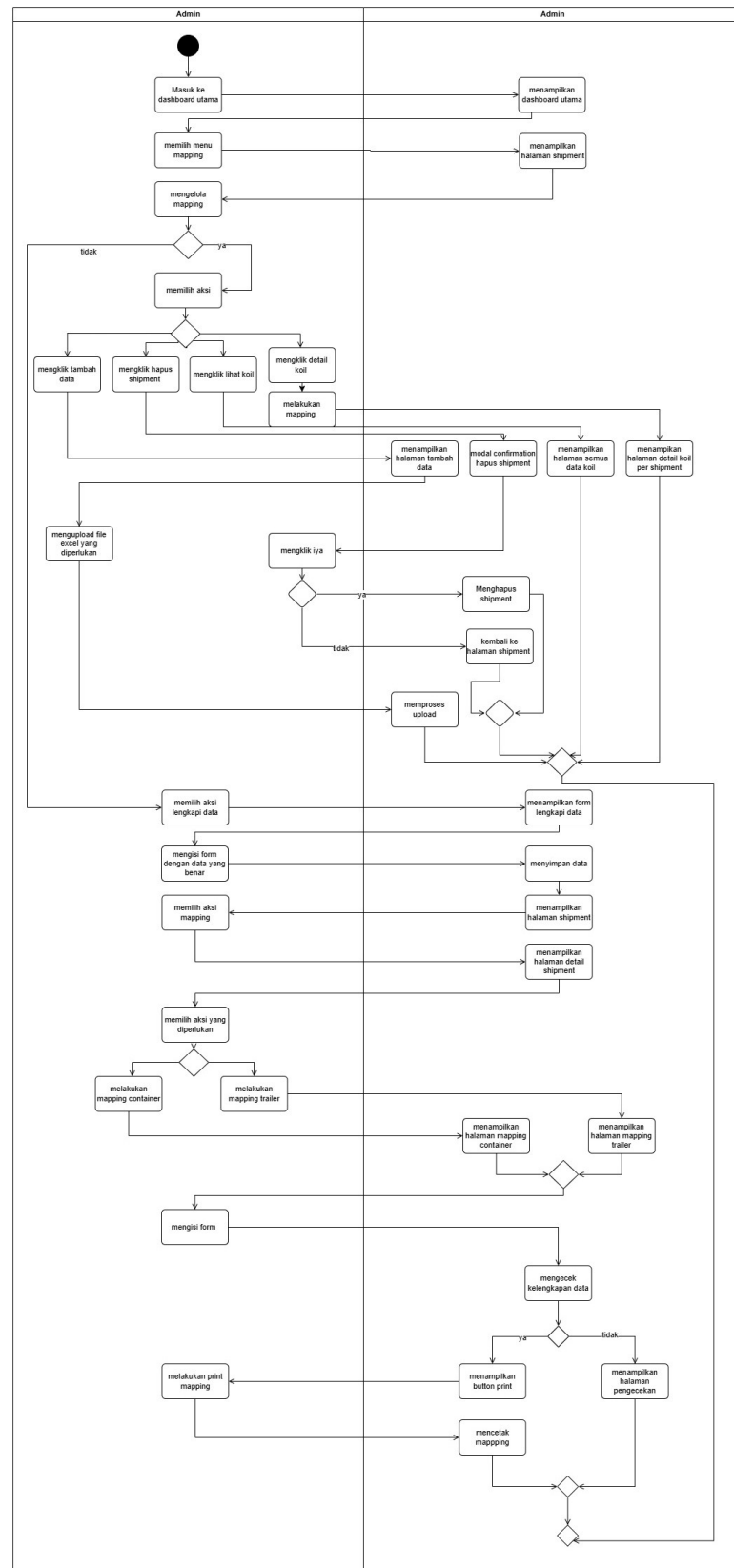
Tabel 4.2. *Actor Description*

No.	Aktor	Deskripsi
1.	Admin	Admin adalah actor yang dapat mengakses semua fungsionalitas yang ada pada system (kecuali fungsi menambahkan checklist kendaraan dan mengelola checklist response), dengan melakukan login terlebih dahulu, namun pada fungsi mengakses surat izin keluar, admin tidak perlu melakukan login.
2.	Pegawai	Pegawai adalah actor yang dapat mengakses dan mengelola fungsi menambahkan checklist kendaraan checklist response, scan layout dengan melakukan login terlebih dahulu, dan fungsi mengakses surat izin keluar, pegawai tidak perlu melakukan login.

#### 4.6.1.2. *Activity Diagram*

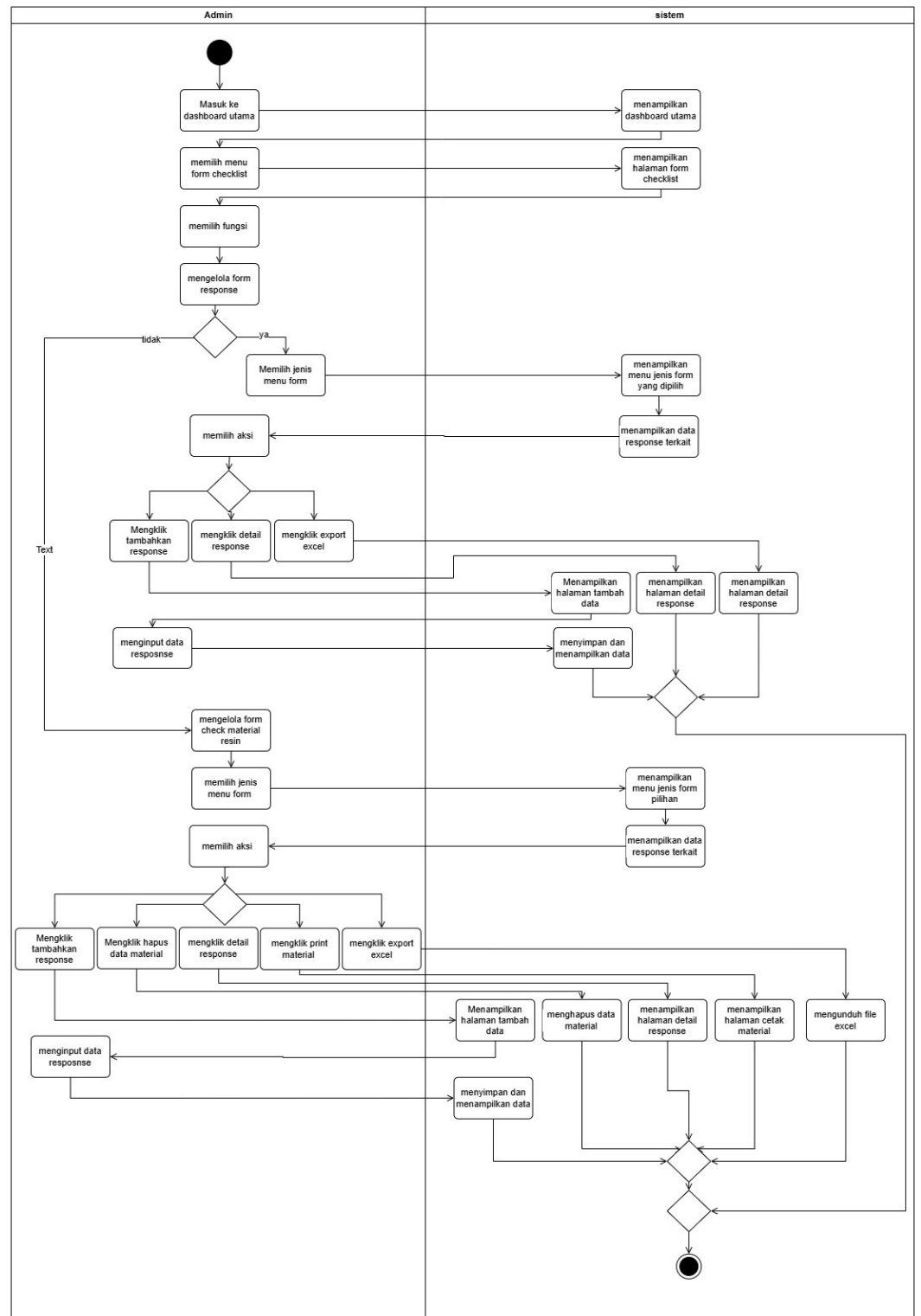
Di bawah ini adalah *activity* diagram dari sistem informasi manajemen operasional yang digunakan untuk menggambarkan alir interaksi sistem dengan user pada bagian sistem yang akan dikembangkan ditandai dengan warna yang berbeda pada *usecase* diagram.

## 1. Activity Diagram Mengelola Mapping



Gambar 4.2. Activity Diagram Mengelola Mapping

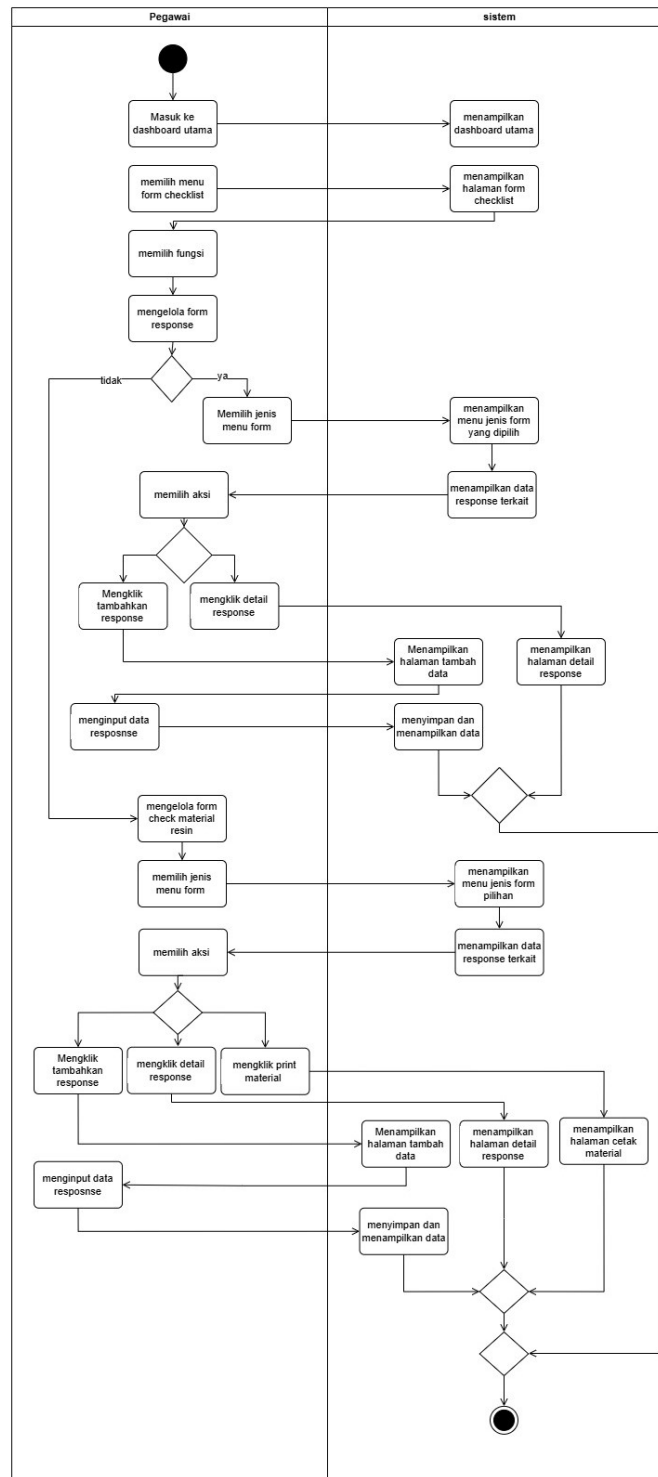
## 2. Activity Diagram Mengelola Form Checklist



Gambar 4.3. Activity Diagram Mengelola Form Checklist



### 3. Activity Diagram Mengelola Checklist Response

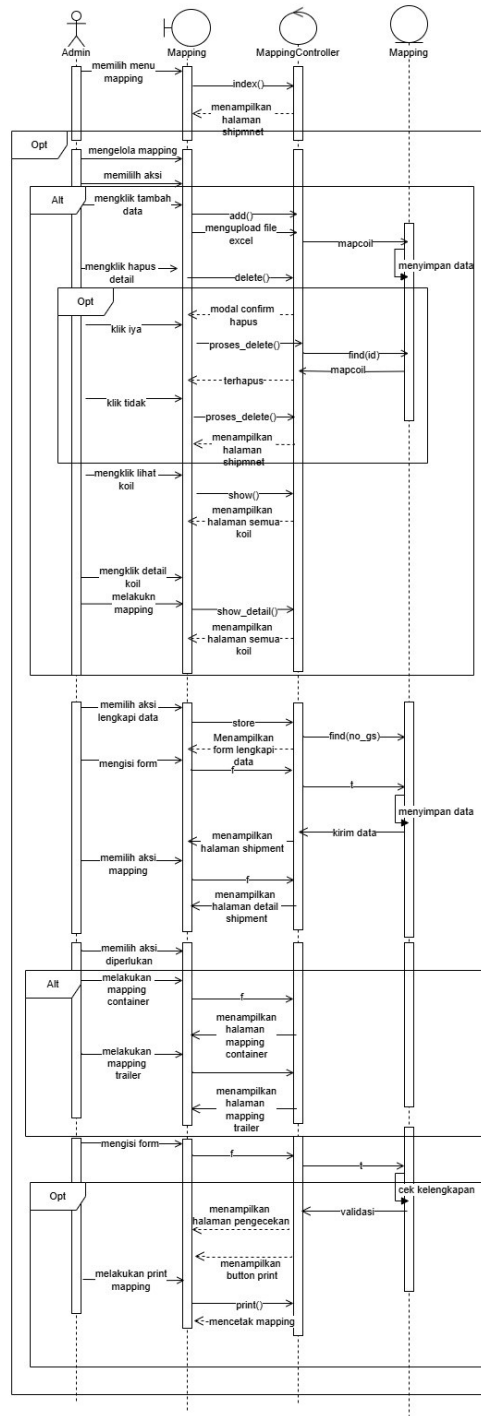


Gambar 4.4. Activity Diagram Mengelola Checklist Response

#### 4.6.1.3. Sequence Diagram

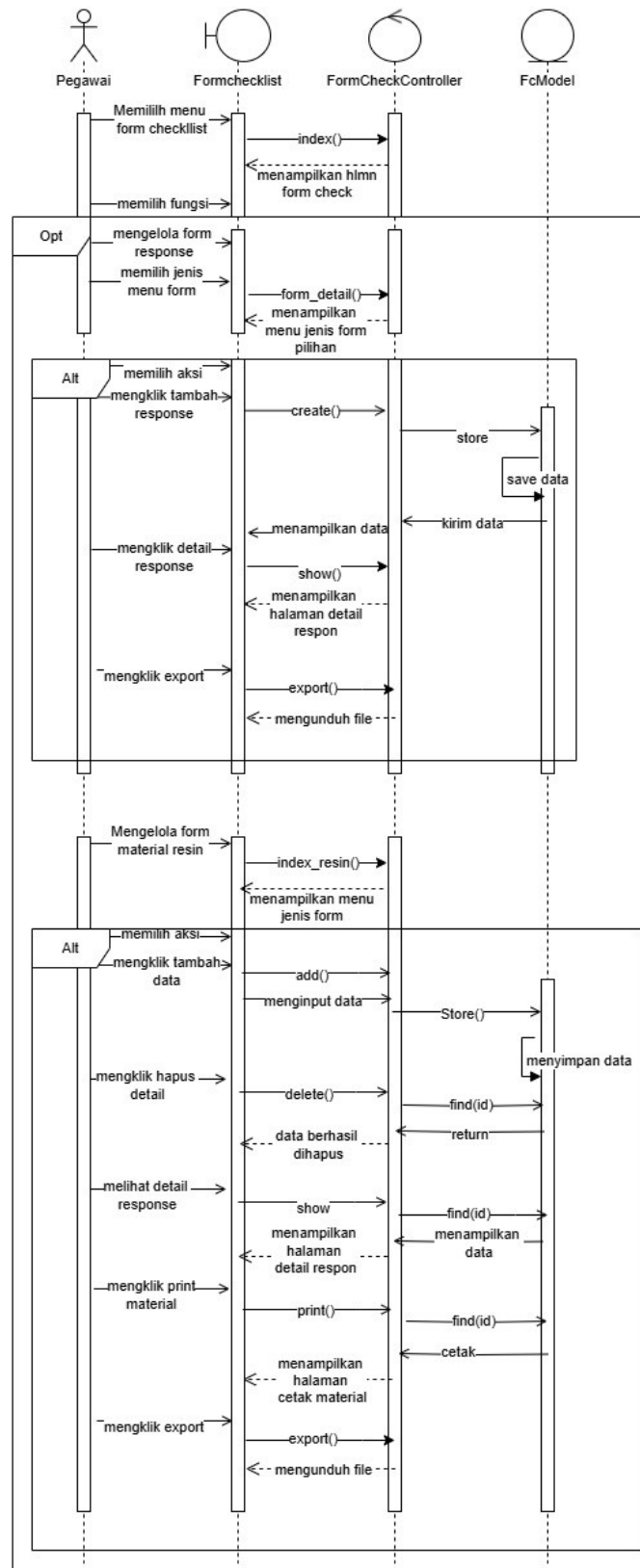
Dibawah ini adalah *sequence* diagram dari sistem informasi manajemen operasional yang digunakan untuk menggambarkan alir proses sistem berdasarkan waktu yang ditentukan untuk menyelesaikan suatu proses.

##### 1. Sequence Diagram Mengelola Mapping



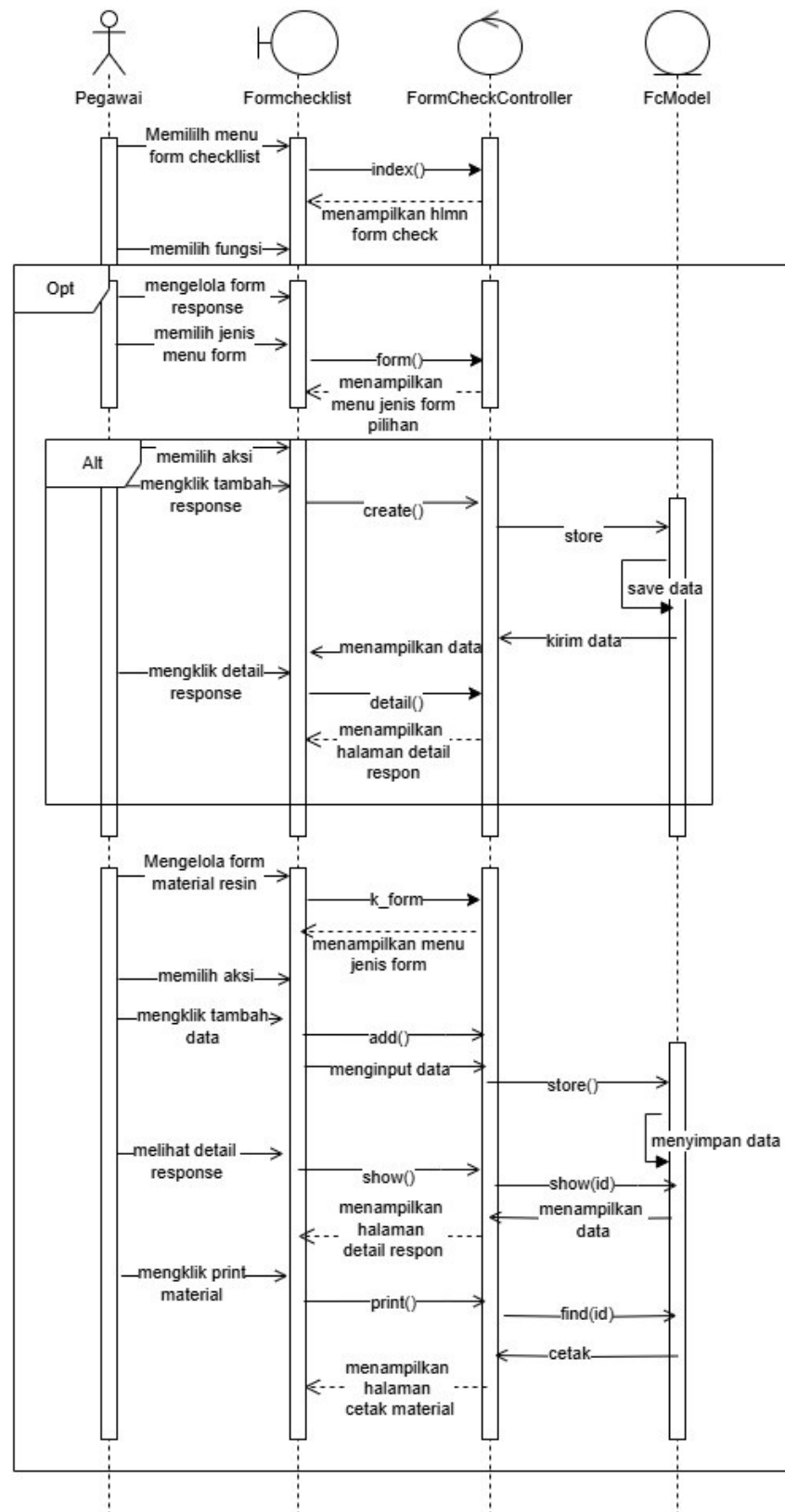
Gambar 4.5. Sequence Diagram Mengelola Mapping

## 2. Sequence Diagram Mengelola Form Checklist



Gambar 4.6. Sequence Diagram Mengelola Form Checklist

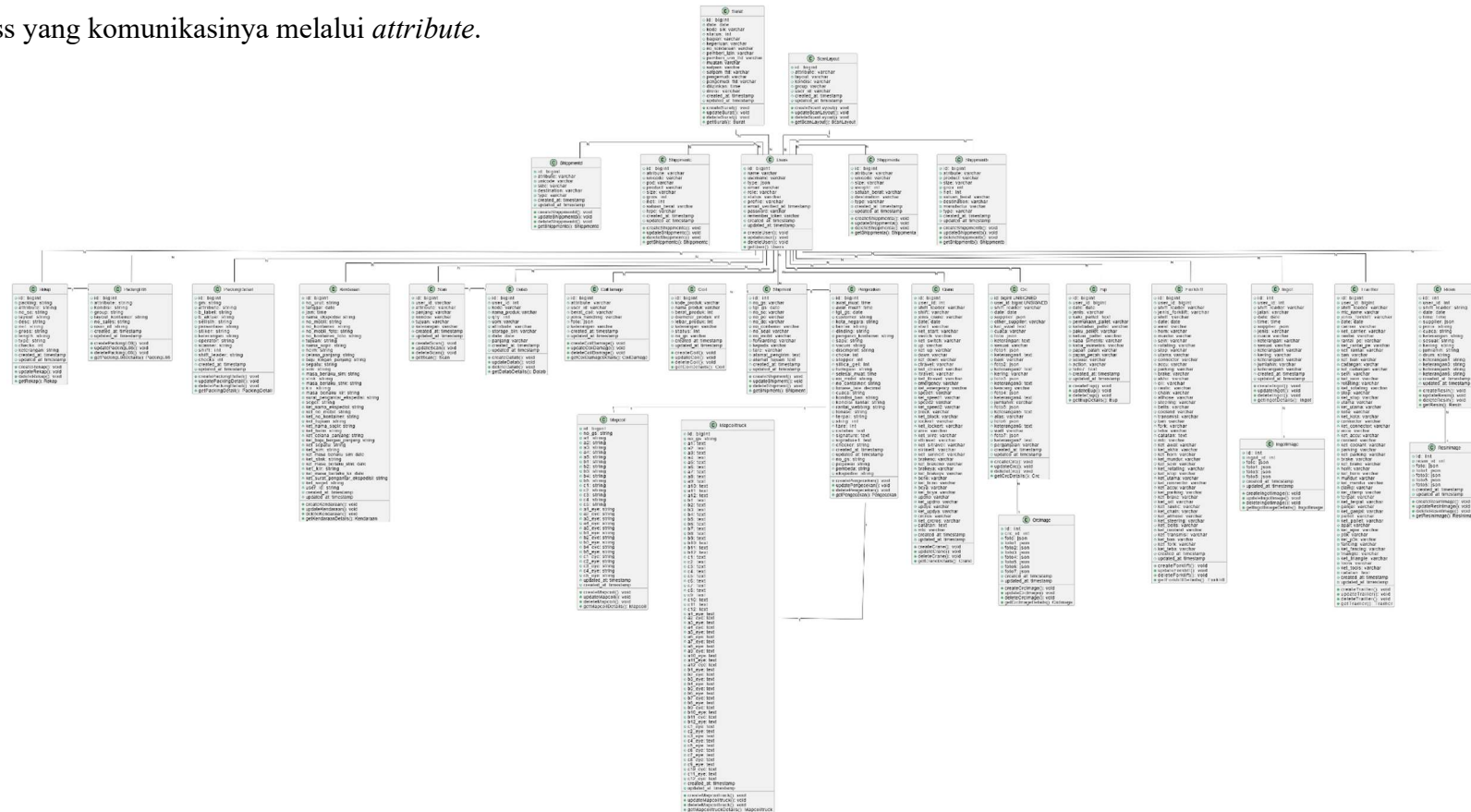
### 3. Sequence Diagram Mengelola Checklist Response



Gambar 4.7. Sequence Diagram Mengelola Form Checklist Response

#### 4.6.1.4. *Class Diagram*

Dibawah ini adalah *class diagram* dari sistem informasi manajemen operasional yang digunakan untuk menggambarkan hubungan antar class yang komunikasinya melalui *attribute*.



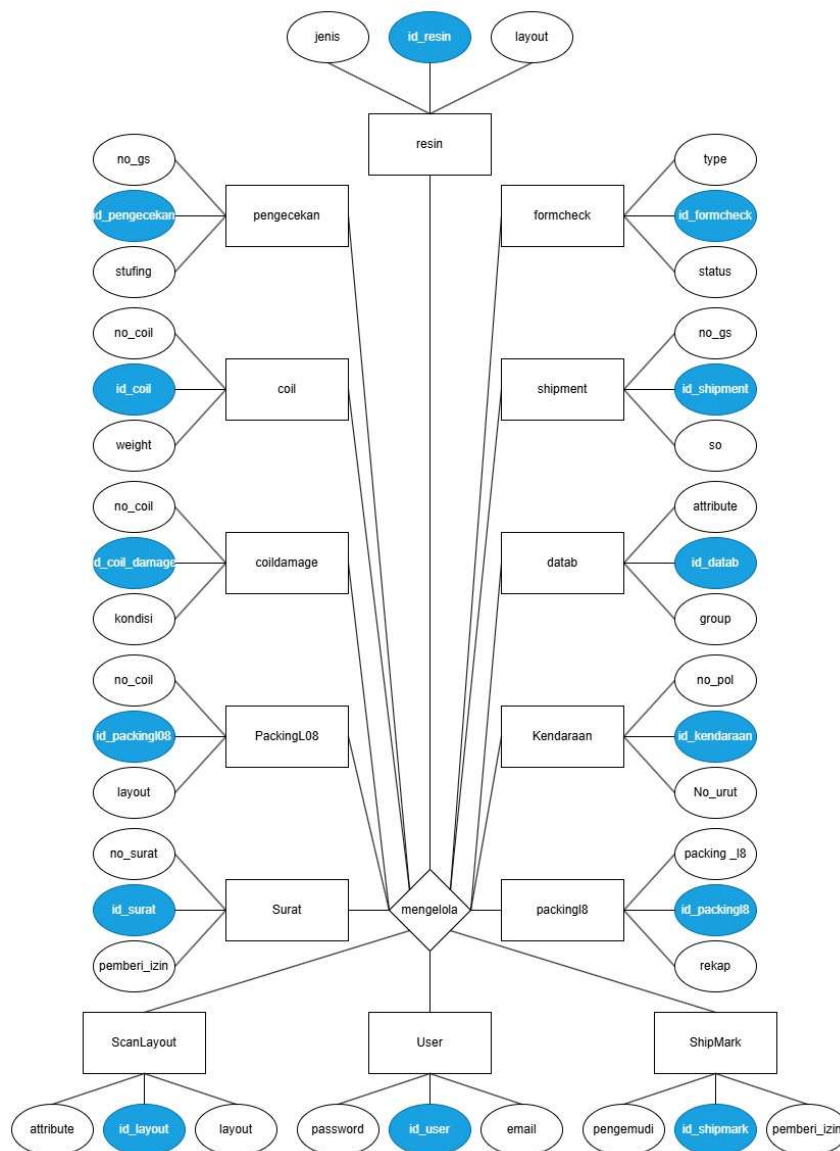
Gambar 4.8. *Class Diagram*

#### 4.6.2. Perancangan Basis Data

Perancangan *basis data* pada sistem ini mencakup pembuatan *Entity Relationship Diagram* (ERD) yang berfungsi untuk memvisualisasikan hubungan antar entitas, penyusunan kamus data yang mendeskripsikan secara rinci atribut-atribut dari masing-masing entitas, serta perancangan struktur tabel yang menjadi dasar dalam implementasi fisik *database*.

##### 4.6.2.1. Entity Relationship Diagram

Pada penelitian ini disusun *Entity Relationship Diagram* (ERD) diagram sebagai berikut.



Gambar 4.9. Entity Relationship Diagram

#### 4.6.2.2. Kamus Data

1. coil{id, attribute, user\_id, berat\_coil, jenis\_handling, foto, keterangan, created\_at, updated\_at}
2. crane{id, user\_id, shift\_leader, shift, jenis\_crane, date, start, ket\_start, switch, ket\_switch, up, ket\_up, down, ket\_down, ctravel, ket\_ctravel, ltravel, ket\_ltravel, emergency, ket\_emergency, speed1, ket\_speed1, speed2, ket\_speed2, block, ket\_block, lockert, ket\_lockert, wire, ket\_wire, sltravel, ket\_sltravel, sirinelt, ket\_sirinelt, brakeno, ket\_brakeno, brakeya, ket\_brakeya, bcno, ket\_bcno, bcya, ket\_bcya, updno, ket\_updno, updya, ket\_updya, crcros, ket\_crcros, catatan, mtc, created\_at, updated\_at}
3. crc{id, user\_id, shift\_leader, date, supplier, other\_supplier, ket\_awal, cuaca, foto, keterangan, sesuai, foto1, keterangan1, baik, foto2, keterangan2, kering, foto3, keterangan3, kencang, foto4, keterangan4, jumlahin, foto5, keterangan5, alas, foto6, keterangan6, wall, foto7, keterangan7, perganjalan, created\_at, updated\_at}
4. crc\_image{id, crc\_id, foto, foto1, foto2, foto3, foto4, foto5, foto6, foto7, created\_at, updated\_at}
5. datab{id, user\_id, kode, nama\_produk, qty, uom, attribute, storage\_bin, date, panjang, created\_at, updated\_at}
6. eup{id, user\_id, date, jenis, kaki\_pallet, permukaan\_pallet, ketebalan\_pallet, paku\_pallet, keluar\_pallet, kaba\_simetris, kaba\_asimetris, papan\_patah, papan\_pecah, sesuai, action, foto7, created\_at, updated\_at}
7. failed\_jobs{id, uuid, connection, queue, payload, exception, failed\_at}
8. forklift{id, user\_id, shift\_leader, jenis\_forklift, shift, date, awal, horn, mundur, sein, rotating, stop, utama, connector, accu, parking, brake, akhir, oil, raulic, chain, allhose, steering, belts, cooland, transmisi, ban, fork, teba, catatan, mtc, ket\_awal, ket\_akhir, ket\_horn, ket\_mundur, ket\_sein, ket\_rotating, ket\_stop, ket\_utama, ket\_connector, ket\_accu, ket\_parking, ket\_brake, ket\_oil, ket\_raulic, ket\_chain, ket\_allhose, ket\_steering, ket\_belts, ket\_cooland, ket\_transmisi, ket\_ban, ket\_fork, ket\_teba, created\_at, updated\_at}

9. hasil\_list{id, kode, attribute, nama\_produk, qty, uom, storage\_bin, date, user\_id, panjang, kondisi, tujuan, created\_at, updated\_at}
10. ingot{id, user\_id, shift\_leader, jalan, date, time, supplier, jenis, cuaca, keterangan, sesuai, keterangan1, kering, keterangan3, jumlahin, keterangan5, created\_at, updated\_at}
11. ingot\_image{id, ingot\_id, foto, foto1, foto3, foto5, created\_at, updated\_at}
12. job\_batches{id, name, total\_jobs, pending\_jobs, failed\_jobs, failed\_job\_ids, options, cancelled\_at, created\_at, finished\_at}
13. jobs{id, queue, payload, attempts, reserved\_at, available\_at, created\_at}
14. kendaraan{id, no\_urut, tanggal, jam, nama\_ekspedisi, no\_mobil, no\_kontainer, no\_mobil\_foto, no\_kontainer\_foto, tujuan, nama\_sopir, helm, celana\_panjang, baju\_lengan\_panjang, sepatu, sim, masa\_berlaku\_sim, stnk, masa\_berlaku\_stnk, kir, masa\_berlaku\_kir, surat\_pengantar\_ekspedisi, segel, ket\_nama\_ekspedisi, ket\_no\_mobil, ket\_no\_kontainer, ket\_tujuan, ket\_nama\_sopir, ket\_helm, ket\_celana\_panjang, ket\_baju\_lengan\_panjang, ket\_sepatu, ket\_sim, ket\_masa\_berlaku\_sim, ket\_stnk, ket\_masa\_berlaku\_stnk, ket\_kir, ket\_masa\_berlaku\_kir, ket\_surat\_pengantar\_ekspedisi, ket\_segel, user\_id, created\_at, updated\_at}
15. mapcoil{id, no\_gs, a1, a2, a3, a4, a5, b1, b2, b3, b4, b5, c1, c2, c3, c4, c5, a1\_eye, a2\_eye, a3\_eye, a4\_eye, a5\_eye, b1\_eye, b2\_eye, b3\_eye, b4\_eye, b5\_eye, c1\_eye, c2\_eye, c3\_eye, c4\_eye, c5\_eye, updated\_at, created\_at}
16. mapcoiltruck{id, no\_gs, a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11, a12, b1, b2, b3, b4, b5, b6, b7, b8, b9, b10, b11, b12, c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11, c12, a1\_eye, a2\_eye, a3\_eye, a4\_eye, a5\_eye, a6\_eye, a7\_eye, a8\_eye, a9\_eye, a10\_eye, a11\_eye, a12\_eye, b1\_eye, b2\_eye, b3\_eye, b4\_eye, b5\_eye, b6\_eye, b7\_eye, b8\_eye, b9\_eye, b10\_eye, b11\_eye, b12\_eye, c1\_eye, c2\_eye, c3\_eye, c4\_eye, c5\_eye, c6\_eye, c7\_eye, c8\_eye, c9\_eye, c10\_eye, c11\_eye, c12\_eye, created\_at, updated\_at}
17. migrations{id, migration, batch}
18. packing{id, gm, shift\_leader, operator, jenis, shift, created\_at, updated\_at}



19. packing\_detail{id, gm, attribute, b\_label, b\_aktual, selisih, persentase, stiker, keterangan, operator, scanner, shift, shift\_leader, created\_at, updated\_at}
20. packingl08{id, attribute, kondisi, group, layout\_kontainer, no\_sales, user\_id, created\_at, updated\_at}
21. password\_reset\_tokens{email, token, created\_at}
22. pengecekan{id, awal\_muat, awal\_muat1, tgl\_gs, customer, kota\_negara, lantai, dinding, pengunci\_kontainer, sapu, vacuum, disemprot, choke, stopper, silica\_gel, fumigasi, selesai\_muat, no\_mobil, no\_container, tonase\_tare, cuaca, kondisi\_ban, kondisi\_lantai, rantai\_webbing, tonase, terpal, sling, tare, catatan, signature, signature1, checker, created\_at, updated\_at, no\_gs, pegawai, pembeda}
23. rekap{id, packing, attribute, no\_so, layout, desc, net, gross, length, type, created\_at, updated\_at}
24. resin{id, user\_id, shift\_leader, date, time, supplier, jenis, cuaca, keterangan, sesuai, kering, jumlahin, drum, keterangan1, keterangan3, keterangan5, keterangan6, created\_at, updated\_at}
25. resin\_image{id, resin\_id, foto, foto1, foto3, foto5, foto6, created\_at, updated\_at}
26. scan{id, user\_id, attribute, panjang, kondisi, tujuan, keterangan, created\_at, updated\_at}
27. scan\_layout{id, attribute, layout, kondisi, group, user\_id, created\_at, updated\_at}
28. sessions{id, user\_id, ip\_address, user\_agent, payload, last\_activity}
29. shipment{id, no\_gs, tgl\_gs, no\_so, no\_po, no\_do, no\_container, no\_seal, no\_mobil, forwarding, kepada, tare, alamat\_pengirim, alamat\_tujuan, created\_at, updated\_at}
30. shippmenta{id, atribute, unicode, size, weight, satuan\_berat, destination, type, created\_at, updated\_at}
31. shippmentb{id, atribute, product, size, gros, net, satuan\_berat, destination, manufactur, type, created\_at, updated\_at}

32. shippmentc{id, atribute, unicode, pod, product, size, gros, net, satuan\_berat, type, created\_at, updated\_at}
33. shippmentd{id, atribute, unicode, size, destination, type, created\_at, updated\_at}
34. supply{id, shift\_leader, shift, supply, foto, user\_id, created\_at, updated\_at}
35. surat{id, date, kode\_sik, status, bagian, keperluan, no\_kendaraan, pemberi\_izin, pemberi\_izin\_ttd, muatan, satpam, satpam\_ttd, pengemudi, pengemudi\_ttd, diizinkan, divisi, created\_at, updated\_at}
36. trailler{id, user\_id, shift\_leader, mtc\_name, jenis\_forklift, date, carrier, ket\_carrier, rantai, rantai\_pe, ket\_rantai\_pe, ket\_rantai, ban, ket\_ban, cadangan, ket\_cadangan, sein, ket\_sein, rotating, ket\_rotating, stop, ket\_stop, utama, ket\_utama, kota, ket\_kota, connector, ket\_connector, accu, ket\_accu, coolant, ket\_coolant, parking, ket\_parking, brake, ket\_brake, horn, ket\_horn, mundur, ket\_mundur, clamp, ket\_clamp, terpal, ket\_terpal, ganjal, ket\_ganjal, pallet, ket\_pallet, apar, ket\_apar, p3k, ket\_p3k, fancing, ket\_fancing, triangle, ket\_triangle, tools, ket\_tools, catatan, created\_at, updated\_at}
37. users{id, name, username, type, email, role, status, profile, email\_verified\_at, password, remember\_token, created\_at, updated\_at}

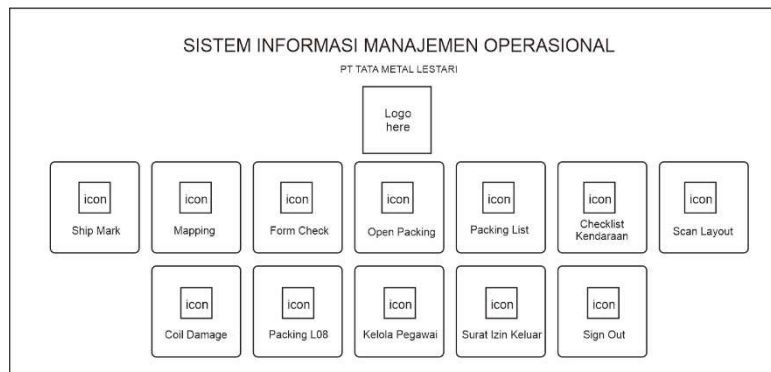
#### 4.6.3. Perancangan Antar Muka

Berikut adalah perncangan antar muka (*user interface*) dalam bentuk *wire frame* dari Sistem Informasi Manajemen Operasional yang akan dibangun:

##### 1. Login Page

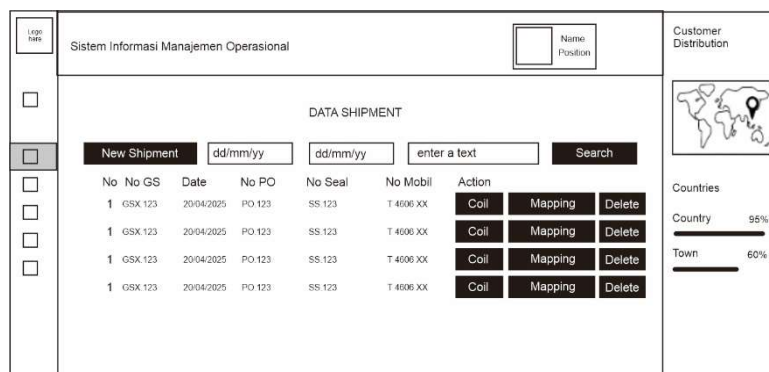
Gambar 4.10. *Wire frame login page*

## 2. Landing Page



Gambar 4.11. Wire frame Landing page

## 3. Mapping Page



Gambar 4.12. Wire frame Mapping page

## 4. Error Handling Page



Gambar 4.13. Wire frame Error handling page

## 5. Form Check Page

The wireframe illustrates the layout of the Form Check page. It features a sidebar on the left with a logo and navigation links. The header includes a title and user information. The main content area displays an overview section with three cards: an empty card, a 'Total Response' card showing 287, and a 'Total Pegawai' card showing 39. The right sidebar contains a 'Customer Distribution' section with a world map and two progress bars for 'Country' (95%) and 'Town' (60%).

Section	Item	Value/Status
Sidebar	Home	<input type="checkbox"/>
	Dashboard	<input type="checkbox"/>
	Form Checklist	<input type="checkbox"/>
	Crane	<input type="checkbox"/>
	Forklift	<input type="checkbox"/>
	Trailer	<input type="checkbox"/>
Main Content	Overview	<input type="checkbox"/>
	Total Response	287
	Total Pegawai	39
	Empty Card	<input type="checkbox"/>
Right Sidebar	Customer Distribution	Map
	Countries	<input type="checkbox"/>
	Country	95%
	Town	60%

Gambar 4.14. *Wire frame Form Check page*

## BAB 5

### IMPLEMENTASI DAN EVALUASI

#### 5.1. Implementasi

##### 5.1.1. Implementasi Mc Call's Model

Dalam proses penelitian ini dilakukan pengujian dengan metode pendekatan Mc Call's Model untuk menemukan permasalahan yang ada pada sistem yang akan dikembangkan. Berikut adalah hasil yang diperoleh dari penerapan Mc Call's Model.

Tabel 5.1. Kategori pengujian Mc Call's Model

Kategori	Jenis	Pertanyaan
<i>Product Operation</i>	<i>Correctness</i>	Sistem Mampu Menampilkan data dengan benar
		Sistem Mampu Menampilkan Kesesuaian Informasi
	<i>Reliability</i>	Keseluruhan Menu dan Fitur dapat berjalan dengan baik
		Menu yang tersedia dapat memenuhi fungsinya masing-masing ?
	<i>Efficiency</i>	Seberapa cepat sistem dalam menampilkan data yang dibutuhkan?
		Apakah sistem dapat menangani jumlah pengguna yang banyak tanpa mengalami penurunan performa?
	<i>Integrity</i>	Sistem mampu melakukan pelacakan perubahan data
		Apakah sistem memiliki mekanisme untuk mencegah akses tidak sah ?
	<i>Usability</i>	Sistem Memiliki desain dan tampilan yang jelas ?
		Bahasa Yang digunakan pada sistem konsisten ?
<i>Product Revisions</i>	<i>Maintainability</i>	Apakah sistem mudah diperbaiki jika terjadi kesalahan?
		Apakah sistem memiliki dokumentasi yang memudahkan pemeliharaan?
	<i>Flexibility</i>	Apakah sistem dapat dengan mudah disesuaikan dengan kebutuhan baru?
		Apakah sistem mendukung berbagai konfigurasi sesuai dengan kebutuhan pengguna?
	<i>Testability</i>	Sistem memiliki informasi yang cukup baik untuk diuji?

		Apakah sistem mendukung pengujian otomatis atau manual dengan mudah?
<i>Product Transition</i>	<i>Portability</i>	Apakah sistem dapat dijalankan di berbagai platform atau perangkat?
		Seberapa mudah sistem dapat dipindahkan ke infrastruktur atau server lain?
	<i>Reusability</i>	Apakah ada komponen dalam sistem yang dapat digunakan kembali untuk pengembangan sistem lain?
		Apakah sistem menggunakan modul atau library yang bersifat reusable?
	<i>Interoperability</i>	Apakah sistem dapat berintegrasi dengan aplikasi lain dengan mudah?
		Seberapa baik sistem dapat bertukar data dengan sistem eksternal?
<i>Other</i>	<i>other</i>	Saran atau kebutuhan yang perlu ditambahkan pada sistem

#### 5.1.1.1. *Product Operation*

##### 1. *Reliability* menggunakan WAPT

Test execution parameters:

Test status: finished

Test started at: 7/03/2025 9:54:21 PM

Scenario name:

Test run comment:

Test executed by: odanu (CC)

Test executed on: CC

Test duration: 0:05:00

Test result: SUCCESS

Pass/Fail Criteria

Name	Result	Comment
Session error rate for each profile	SUCCESS	

Summary

Profile	Successful sessions	Failed sessions	Successful pages	Failed pages	Successful hits	Failed hits	Other errors	Total KBytes sent	Total KBytes received	Avg response time, sec (with page resources)
SID Warehouse	76	0	427	0	3283	0	0	3305	157824	0.65(2.18)

Number of active users

Profile	0:00:00 - 0:00:30	0:00:30 - 0:01:00	0:01:00 - 0:01:30	0:01:30 - 0:02:00	0:02:00 - 0:02:30	0:02:30 - 0:03:00	0:03:00 - 0:03:30	0:03:30 - 0:04:00	0:04:00 - 0:04:30	0:04:30 - 0:05:00
SID Warehouse	5	5	8	11	12	12	12	12	12	12
Total	5	5	8	11	12	12	12	12	12	12

Successful sessions (Failed sessions)

Profile	0:00:00 - 0:00:30	0:00:30 - 0:01:00	0:01:00 - 0:01:30	0:01:30 - 0:02:00	0:02:00 - 0:02:30	0:02:30 - 0:03:00	0:03:00 - 0:03:30	0:03:30 - 0:04:00	0:04:00 - 0:04:30	0:04:30 - 0:05:00	Total
SID Warehouse	0(0)	3(0)	2(0)	8(0)	13(0)	9(0)	10(0)	11(0)	8(0)	12(0)	76(0)
Total	0(0)	3(0)	2(0)	8(0)	13(0)	9(0)	10(0)	11(0)	8(0)	12(0)	76(0)

Gambar 5.1. Pengujian *Reliability* dengan WAPT

Successful pages (Failed pages)											
Profile	0:00:00 - 0:00:30	0:00:30 - 0:01:00	0:01:00 - 0:01:30	0:01:30 - 0:02:00	0:02:00 - 0:02:30	0:02:30 - 0:03:00	0:03:00 - 0:03:30	0:03:30 - 0:04:00	0:04:00 - 0:04:30	0:04:30 - 0:05:00	Total
SID Warehouse	8(0)	24(0)	23(0)	51(0)	54(0)	53(0)	57(0)	37(0)	58(0)	62(0)	427(0)
Total	8(0)	24(0)	23(0)	51(0)	54(0)	53(0)	57(0)	37(0)	58(0)	62(0)	427(0)

Successful hits (Failed hits)											
Profile	0:00:00 - 0:00:30	0:00:30 - 0:01:00	0:01:00 - 0:01:30	0:01:30 - 0:02:00	0:02:00 - 0:02:30	0:02:30 - 0:03:00	0:03:00 - 0:03:30	0:03:30 - 0:04:00	0:04:00 - 0:04:30	0:04:30 - 0:05:00	Total
SID Warehouse	180(0)	181(0)	180(0)	416(0)	391(0)	450(0)	419(0)	339(0)	379(0)	462(0)	3283(0)
Total	180(0)	181(0)	180(0)	416(0)	391(0)	450(0)	419(0)	339(0)	379(0)	462(0)	3283(0)

Successful sessions per second											
Profile	0:00:00 - 0:00:30	0:00:30 - 0:01:00	0:01:00 - 0:01:30	0:01:30 - 0:02:00	0:02:00 - 0:02:30	0:02:30 - 0:03:00	0:03:00 - 0:03:30	0:03:30 - 0:04:00	0:04:00 - 0:04:30	0:04:30 - 0:05:00	Total
SID Warehouse	0	0.10	0.07	0.27	0.43	0.30	0.33	0.37	0.27	0.40	0.29
Total	0	0.10	0.07	0.27	0.43	0.30	0.33	0.37	0.27	0.40	0.29

Successful pages per second											
Profile	0:00:00 - 0:00:30	0:00:30 - 0:01:00	0:01:00 - 0:01:30	0:01:30 - 0:02:00	0:02:00 - 0:02:30	0:02:30 - 0:03:00	0:03:00 - 0:03:30	0:03:30 - 0:04:00	0:04:00 - 0:04:30	0:04:30 - 0:05:00	Total
SID Warehouse	0.27	0.80	0.77	1.70	1.80	1.77	1.90	1.23	1.93	2.07	1.42
Total	0.27	0.80	0.77	1.70	1.80	1.77	1.90	1.23	1.93	2.07	1.42

Successful hits per second											
Profile	0:00:00 - 0:00:30	0:00:30 - 0:01:00	0:01:00 - 0:01:30	0:01:30 - 0:02:00	0:02:00 - 0:02:30	0:02:30 - 0:03:00	0:03:00 - 0:03:30	0:03:30 - 0:04:00	0:04:00 - 0:04:30	0:04:30 - 0:05:00	Total
SID Warehouse	3.20	6.03	6	13.9	13.0	15	14.0	11.3	12.6	15.4	10.9
Total	3.20	6.03	6	13.9	13.0	15	14.0	11.3	12.6	15.4	10.9

Gambar 5.2. Pengujian *Reliability* dengan WAPT 1



Gambar 5.3. Pengujian *Reliability* dengan WAPT 2

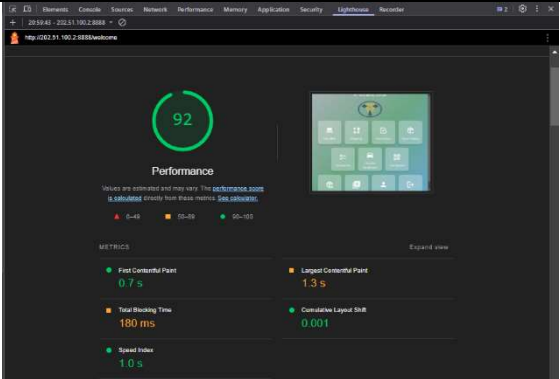
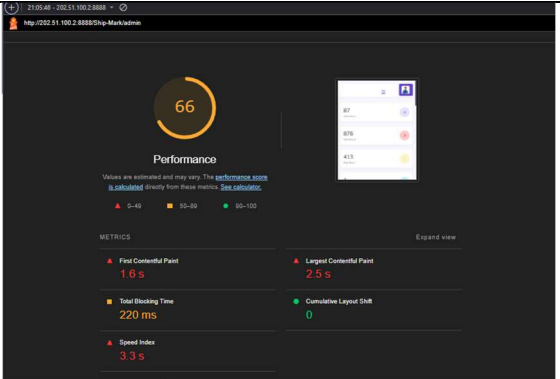
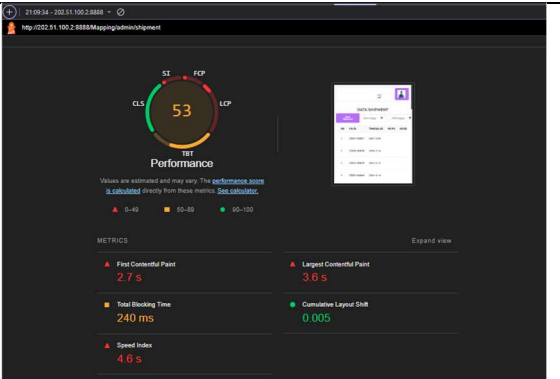
Tabel 5.2. Hasil Pengujian *Reliability* dengan WAPT

NO	Metrik	Sukses	Gagal
1	<i>Sessions</i>	76	0
2	<i>Page</i>	427	0
3	<i>Hits</i>	3283	0
Total		3786	0

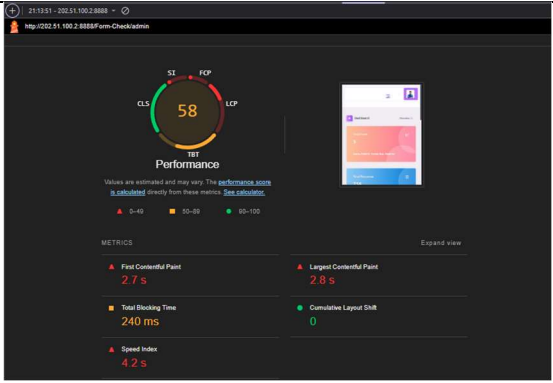
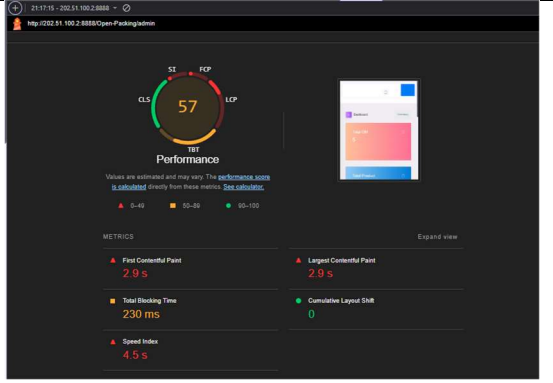
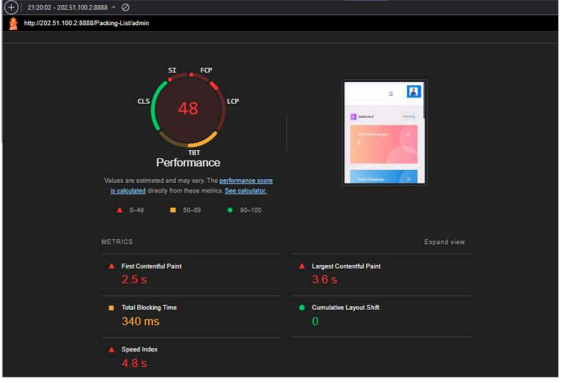
$$P = \frac{3786}{3786} \times 100\% = 100\%$$

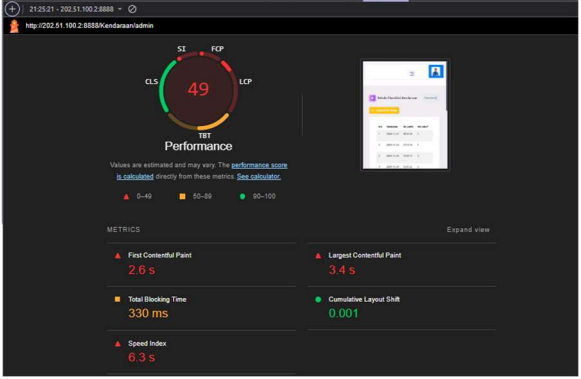
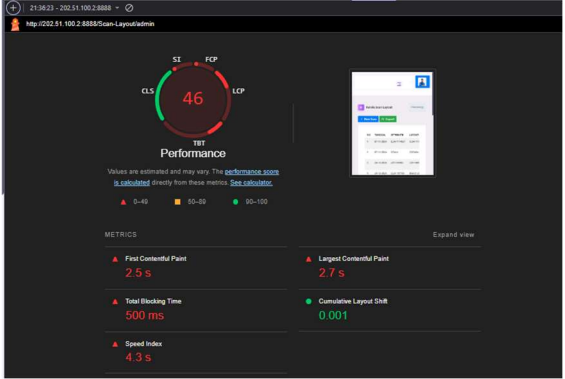
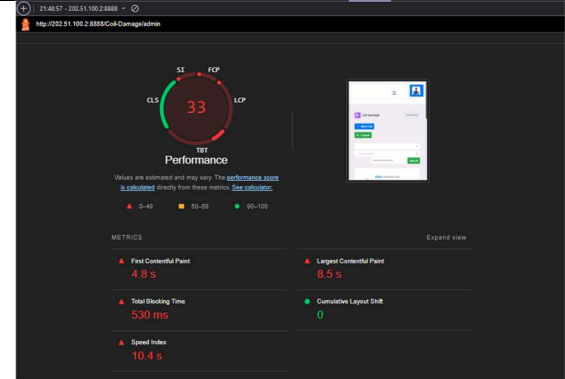
## 2. *Efficiency* menggunakan Google Lighthouse

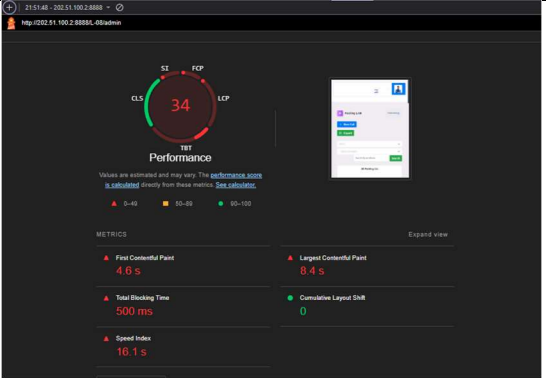
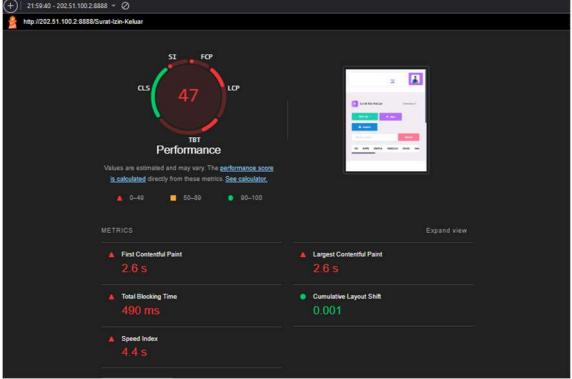
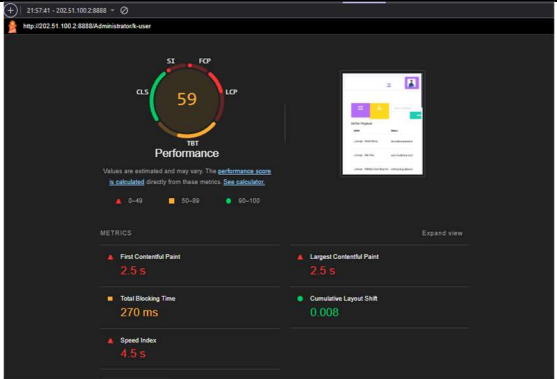
Tabel 5.3. Pengujian *Efficiency* dengan Google Lighthouse

No	Halaman	Hasil
1	<i>Welcome</i>	 <p>Gambar 5.4. Pengujian <i>Efficiency</i> dengan G Lighthouse</p> <p><i>Performance</i> : 92 LCP : 1.3 S</p>
2	<i>Ship Mark</i>	 <p>Gambar 5.5. Pengujian <i>Efficiency</i> dengan G Lighthouse 1</p> <p><i>Performance</i> : 66 LCP : 2.5 S</p>
3	<i>Mapping</i>	 <p>Gambar 5.6. Pengujian <i>Efficiency</i> dengan G Lighthouse 2</p> <p><i>Performance</i> : 53</p>



		LCP : 3.6 S
4	Form Check	 <p>Gambar 5.7. Pengujian <i>Efficiency</i> dengan G Lighthouse 3</p> <p>Performance : 58 LCP : 2.8 S</p>
5	Open Packing	 <p>Gambar 5.8. Pengujian <i>Efficiency</i> dengan G Lighthouse 4</p> <p>Performance : 57 LCP : 2.9 S</p>
6	Packing List	 <p>Gambar 5.9. Pengujian <i>Efficiency</i> dengan G Lighthouse 5</p> <p>Performance : 48 LCP : 3.6 S</p>

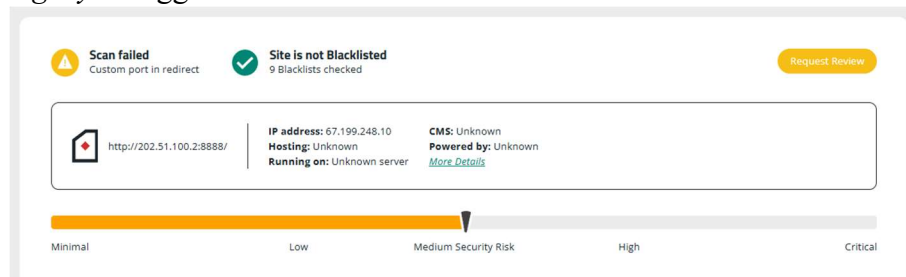
7	<i>Checklist Kendaraan</i>	 <p>Gambar 5.10. Pengujian Efficiency dengan <i>G Lighthouse 6</i></p> <p><i>Performance : 49</i> <i>LCP : 3.4 S</i></p>
8	<i>Scan Layout</i>	 <p>Gambar 5.11. Pengujian Efficiency dengan <i>G Lighthouse 7</i></p> <p><i>Performance : 46</i> <i>LCP : 2.7 S</i></p>
9	<i>Coil Damage</i>	 <p>Gambar 5.12. Pengujian Efficiency dengan <i>G Lighthouse 8</i></p> <p><i>Performance : 33</i> <i>LCP : 8.5 S</i></p>

10	Packing L08	 <p>Gambar 5.13. Pengujian Efficiency dengan <i>G Lighthouse 9</i></p> <p><i>Performance : 34</i> <i>LCP : 8.4 S</i></p>
11	Surat Izin Keluar	 <p>Gambar 5.14. Pengujian Efficiency dengan <i>G Lighthouse 10</i></p> <p><i>Performance : 47</i> <i>LCP : 2.6 S</i></p>
12	Kelola Pegawai	 <p>Gambar 5.15. Pengujian Efficiency dengan <i>G Lighthouse 11</i></p> <p><i>Performance : 59</i> <i>LCP : 2.5 S</i></p>

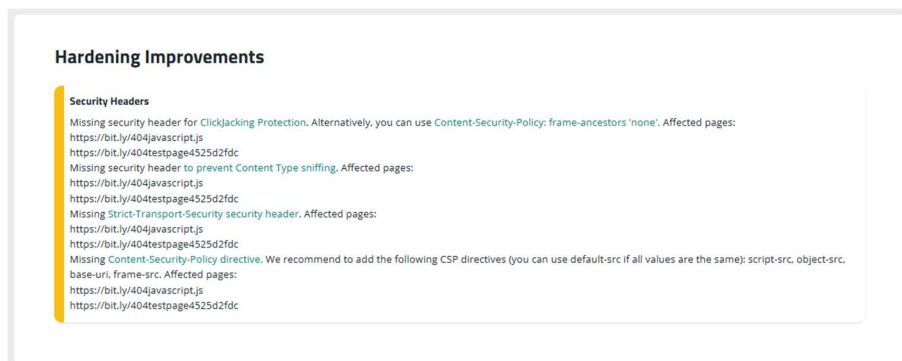
Total <i>Performance</i>	642
Rata-rata <i>Performance</i>	642 / 12 = 53.3
Total LCP	44.8
Rata-rata LCP	44.8 / 12 = 3.78

Dari hasil uji diatas kemudian dapat disimpulkan bahwa nilai *efficiency* yang diambil dari skor *performance* adalah sebesar 53.3%, serta dengan nilai rata rata *time load* sebesar 3.78 detik

### 3. *Integrity* menggunakan Sucuri Sitecheck



Gambar 5.16. Pengujian *Integrity* dengan Sucuri Sitehack



Gambar 5.17. Pengujian *Integrity* dengan Sucuri Sitehack 1

### Pengujian dengan kuisioner

$$P = \frac{(\text{Skor Aktual})}{(\text{Skor Ideal})} \times 100\%$$

Dimana :

P = Presntase Kualitas

Skor Aktual = Skor akutal dari responden

Skor Ideal = Skor yang diharapkan

### 4. *Correctness*

Tabel 5.4. Pengujian *Correctness* dengan Kuisioner

Pertanyaan	Skor Aktual	Skor Ideal
Sistem mampu menampilkan data dengan benar ?	38	4 X 10 = 40
Sistem Mampu Menampilkan Kesesuaian Informasi ?	38	4 X 10 = 40
<b>Total</b>	<b>76</b>	<b>80</b>

$$P = \frac{76}{80} \times 100\% = 95\%$$

#### 5. *Reliability*

Tabel 5.5. Pengujian *Reliability* dengan Kuisisioner

Pertanyaan	Skor Aktual	Skor Ideal
Keseluruhan Menu dan Fitur dapat berjalan dengan baik ?	35	4 X 10 = 40
Menu yang tersedia dapat memenuhi fungsinya masing-masing ?	37	4 X 10 = 40
<b>Total</b>	<b>72</b>	<b>80</b>

$$P = \frac{72}{80} \times 100\% = 90\%$$

#### 6. *Efficiency*

Tabel 5.6. Pengujian *Efficiency* dengan Kuisisioner

Pertanyaan	Skor Aktual	Skor Ideal
Seberapa cepat sistem dalam menampilkan data yang dibutuhkan?	32	4 X 10 = 40
Apakah sistem dapat menangani jumlah pengguna yang banyak tanpa mengalami penurunan performa?	33	4 X 10 = 40
<b>Total</b>	<b>65</b>	<b>80</b>

$$P = \frac{65}{80} \times 100\% = 81,25\%$$

#### 7. *Integrity*

Tabel 5.7. Pengujian *Integrity* dengan Kuisisioner

Pertanyaan	Skor Aktual	Skor Ideal
Sistem mampu melakukan pelacakan perubahan data ?	35	4 X 10 = 40
Apakah sistem memiliki mekanisme untuk mencegah akses tidak sah ?	40	4 X 10 = 40
<b>Total</b>	75	80

$$P = \frac{75}{80} \times 100\% = 93,75\%$$

#### 8. *Usability*

Tabel 5.8. Pengujian *Usability* dengan Kuisisioner

Pertanyaan	Skor Aktual	Skor Ideal
Sistem Memiliki desain dan tampilan yang jelas ?	37	4 X 10 = 40
Bahasa Yang digunakan pada sistem konsisten ?	31	4 X 10 = 40
<b>Total</b>	68	80

$$P = \frac{68}{80} \times 100\% = 85\%$$

#### 5.1.1.2. Product Revisions

##### 1. *Maintainability*

Tabel 5.9. Pengujian *Maintainability* dengan Kuisisioner

Pertanyaan	Skor Aktual	Skor Ideal
Apakah sistem mudah diperbaiki jika terjadi kesalahan?	19	4 X 10 = 40
Apakah sistem memiliki dokumentasi yang memudahkan pemeliharaan?	39	4 X 10 = 40
<b>Total</b>	58	80

$$P = \frac{58}{80} \times 100\% = 72,5\%$$

## 2. *Flexibility*

Tabel 5.10. Pengujian *Flexibility* dengan Kuisisioner

Pertanyaan	Skor Aktual	Skor Ideal
Apakah sistem dapat dengan mudah disesuaikan dengan kebutuhan baru?	27	4 X 10 = 40
Apakah sistem mendukung berbagai konfigurasi sesuai dengan kebutuhan pengguna?	39	4 X 10 = 40
<b>Total</b>	66	80

$$P = \frac{66}{80} \times 100\% = 82,5\%$$

## 3. *Testability*

Tabel 5.11. Pengujian *Testability* dengan Kuisisioner

Pertanyaan	Skor Aktual	Skor Ideal
Sistem memiliki informasi yang cukup baik untuk diuji?	39	4 X 10 = 40
Apakah sistem mendukung pengujian otomatis atau manual dengan mudah?	39	4 X 10 = 40
<b>Total</b>	78	80

$$P = \frac{78}{80} \times 100\% = 97,5\%$$

### 5.1.1.3. *Product Transition*

#### 1. *Portability*

Tabel 5.12. Pengujian *Portability* dengan Kuisisioner

Pertanyaan	Skor Aktual	Skor Ideal
Apakah sistem dapat dijalankan di berbagai	37	4 X 10 = 40

Pertanyaan	Skor Aktual	Skor Ideal
platform atau perangkat?		
Seberapa mudah sistem dapat dipindahkan ke infrastruktur atau server lain?	40	4 X 10 = 40
<b>Total</b>	77	80

$$P = \frac{77}{80} \times 100\% = 96,25\%$$

## 2. Reusability

Tabel 5.13. Pengujian *Reusability* dengan Kuisisioner

Pertanyaan	Skor Aktual	Skor Ideal
Apakah ada komponen dalam sistem yang dapat digunakan kembali untuk pengembangan sistem lain?	40	4 X 10 = 40
Apakah sistem menggunakan modul atau library yang bersifat <i>reusable</i> ?	40	4 X 10 = 40
<b>Total</b>	80	80

$$P = \frac{80}{80} \times 100\% = 100\%$$

## 3. Interoperability

Tabel 5.14. Pengujian *Interoperability* dengan Kuisisioner

Pertanyaan	Skor Aktual	Skor Ideal
Apakah sistem dapat berintegrasi dengan aplikasi lain dengan mudah?	27	4 X 10 = 40
Seberapa baik sistem dapat bertukar data dengan sistem eksternal?	39	4 X 10 = 40
<b>Total</b>	66	80



$$P = \frac{66}{80} \times 100\% = 82,5\%$$

#### 5.1.1.4. Hasil dan Rekomendasi

Dalam langkah ini, hasil dari semua langkah penelitian yang telah dilaksanakan akan dilengkapakan, dan kemudian akan digunakan sebagai dasar untuk merumuskan saran perbaikan yang diperlukan berdasarkan hasil uji pengujian kualitas perangkat lunak pada *website* Sistem Informasi Manajemen Operasional PT Tata Metal Lestari. Untuk menentukan interpretasi setiap hasil pengujian, maka rentang nilai dilihat pada tabel dibawah ini.

Tabel 5.15. Penghitungan Presentasi dengan Interpretasi

Presentasi	Interpretasi
$90 \leq x \leq 100$	Sangat Baik
$80 \leq x < 90$	Baik
$70 \leq x < 80$	Cukup Kurang
$60 \leq x < 70$	Kurang
$x < 60$	Sangat Kurang

#### 5.1.1.5. Hasil Evaluasi

1. Pengujian *reliability* menggunakan tool WAPT (*Web Application Load, Stress, and Performance Testing*) adalah sebesar 100%, jumlah keseluruhan dari sesi, halaman, dan hits yang berhasil adalah 3786, sementara jumlah total yang gagal adalah 0, maka dari itu pada pengujian ini mendapatkan kategori “Sangat Baik”. Menunjukan bahwa sistem memenuhi standar kehandalan perangkat lunak ketika diuji dengan *tool* WAPT.
2. Pengujian *efficiency* menggunakan *tool* Google Lighthouse adalah sebesar 53.3%, Selain itu, total *Largest Contentful Paint* (LCP) tercatat sebesar 44,8 detik, dengan rata-rata LCP per pengujian sebesar 3,78 detik. Dari hasil uji tersebut, dapat disimpulkan bahwa nilai *efficiency* yang diambil dari skor performa adalah sebesar 53.3%, maka dari itu pada pengujian ini mendapatkan kategori “Sangat Kurang”. Menunjukan bahwa sistem memiliki performa yang buruk.
3. Pengujian *Integrity* menggunakan Sucuri Sitehack mendapatkan *medium security risk*, dikarenakan penggunaan nama domain, namun menggunakan

IP langsung dari sistem tersebut, hal tersebut menurunkan tingkat keamanan yang ada pada sistem, namun hal tersebut sejalan dengan kebutuhan perusahaan yang tidak ingin sistem tersebut dapat diakses secara *public*.

4. Pengujian *correctnes* berdasarkan hasil kuisioner mendapatkan nilai 95% yang mengindikasikan interpretasi “Sangat Baik”. Dalam penghitungan skor ideal yang didapatkan adalah 80, sementara skor aktual mendapatkan skor 76, skor ideal ditentukan dengan asumsi semua jawaban responden adalah “Sangat Setuju” (nilai 4), Hasil ini menunjukkan bahwa aspek *correctnes* sudah sangat baik namun masih diperlukan pemeliharaan untuk mendapatkan nilai yang sempurna.
5. Pengujian *Realibility* berdasarkan hasil kuisioner mendapatkan nilai 90% yang mengindikasikan interpretasi “Sangat Baik”. Dalam penghitungan skor ideal yang didapatkan adalah 80, sementara skor aktual mendapatkan skor 72, skor ideal ditentukan dengan asumsi semua jawaban responden adalah “Sangat Setuju” (nilai 4), Hasil ini menunjukkan bahwa aspek *Realibility* sudah sangat baik namun masih diperlukan pemeliharaan dan perhatian lebih lanjut untuk mendapatkan nilai yang sempurna.
6. Pengujian *efficiency* berdasarkan hasil kuisioner mendapatkan nilai 81.25% yang mengindikasikan interpretasi “Baik”. Dalam penghitungan skor ideal yang didapatkan adalah 80, sementara skor aktual mendapatkan skor 65, skor ideal ditentukan dengan asumsi semua jawaban *responden* adalah “Sangat Setuju” (nilai 4), Hasil ini menunjukkan bahwa aspek *efficiency* sudah sangat baik namun masih diperlukan perbaikan dan perhatian lebih lanjut untuk mendapatkan nilai yang sempurna.
7. Pengujian *integrity* berdasarkan hasil kuisioner mendapatkan nilai 93.75% yang mengindikasikan interpretasi “Sangat Baik”. Dalam penghitungan skor ideal yang didapatkan adalah 80, sementara skor aktual mendapatkan skor 75, skor ideal ditentukan dengan asumsi semua jawaban *responden* adalah “Sangat Setuju” (nilai 4), Hasil ini menunjukkan bahwa aspek *integrity* sudah sangat baik namun masih diperlukan perbaikan dan perhatian lebih lanjut untuk mendapatkan nilai yang sempurna.

8. Pengujian *usability* berdasarkan hasil kuisioner mendapatkan nilai 85% yang mengindikasikan intrpretasi “Baik”. Dalam penghitungan skor ideal yang didapatkan adalah 80, sementara skor aktual mendapatkan skor 68 , skor ideal ditentukan dengan asumsi semua jawaban *responden* adalah "Sangat Setuju" (nilai 4), Hasil ini menunjukkan bahwa aspek *usability* sudah sangat baik namun masih diperlukan perbaikan dan perhatian lebih lanjut untuk mendapatkan nilai yang sempurna.
9. Pengujian *Maintainability* berdasarkan hasil kuisioner mendapatkan nilai 72.5% yang mengindikasikan intrpretasi “Cukup Kurang”. Dalam penghitungan skor ideal yang didapatkan adalah 80, sementara skor aktual mendapatkan skor 58 , skor ideal ditentukan dengan asumsi semua jawaban *responden* adalah "Sangat Setuju" (nilai 4), Hasil ini menunjukkan bahwa aspek *Maintainability* sudah sangat baik namun masih diperlukan perbaikan dan perhatian lebih lanjut untuk mendapatkan nilai yang sempurna.
10. Pengujian *flexibility* berdasarkan hasil kuisioner mendapatkan nilai 82.5% yang mengindikasikan intrpretasi “Baik”. Dalam penghitungan skor ideal yang didapatkan adalah 80, sementara skor aktual mendapatkan skor 66 , skor ideal ditentukan dengan asumsi semua jawaban *responden* adalah "Sangat Setuju" (nilai 4), Hasil ini menunjukkan bahwa aspek *flexibility* sudah sangat baik namun masih diperlukan perhatian lebih lanjut untuk mendapatkan nilai yang sempurna.
11. Pengujian *testability* berdasarkan hasil kuisioner mendapatkan nilai 97.5% yang mengindikasikan intrpretasi “Sangat Baik”. Dalam penghitungan skor ideal yang didapatkan adalah 80, sementara skor aktual mendapatkan skor 78 , skor ideal ditentukan dengan asumsi semua jawaban *responden* adalah "Sangat Setuju" (nilai 4), Hasil ini menunjukkan bahwa aspek *testability* sudah sangat baik namun masih diperlukan pemeliharaan untuk mendapatkan nilai yang sempurna.
12. Pengujian *portability* berdasarkan hasil kuisioner mendapatkan nilai 96.5% yang mengindikasikan intrpretasi “Sangat Baik”. Dalam penghitungan skor ideal yang didapatkan adalah 80, sementara skor aktual mendapatkan skor 77 , skor ideal ditentukan dengan asumsi semua jawaban *responden* adalah

"Sangat Setuju" (nilai 4), Hasil ini menunjukkan bahwa aspek *portability* sudah sangat baik namun masih diperlukan pemeliharaan untuk mendapatkan nilai yang sempurna.

13. Pengujian *reusability* berdasarkan hasil kuisioner mendapatkan nilai 100% yang mengindikasikan intrpretasi "Sangat Baik". Dalam penghitungan skor ideal yang didapatkan adalah 80, sementara skor aktual mendapatkan skor 80 , skor ideal ditentukan dengan asumsi semua jawaban *responden* adalah "Sangat Setuju" (nilai 4), Hasil ini menunjukkan bahwa aspek *reusability* sudah sangat baik.

14. Pengujian *interoperability* berdasarkan hasil kuisioner mendapatkan nilai 82.5% yang mengindikasikan intrpretasi "Baik". Dalam penghitungan skor ideal yang didapatkan adalah 80, sementara skor aktual mendapatkan skor 6 , skor ideal ditentukan dengan asumsi semua jawaban *responden* adalah "Sangat Setuju" (nilai 4), Hasil ini menunjukkan bahwa aspek *interoperability* sudah sangat baik namun masih diperlukan perbaikan dan pemeliharaan untuk mendapatkan nilai yang sempurna.

15. Rata-rata kualitas dari hasil kuisioner

$$\text{Rata – Rata Kualitas} = \frac{95+90+81.25+93.75+85+72.5+82.5+97.5+96.25+100+82.5}{11}=88.75\%$$

Rata-rata kualitas dari berbagai faktor yang diuji adalah sekitar 88.75%, yang menunjukkan bahwa secara keseluruhan, *website* tersebut memiliki kualitas yang baik. Faktor-faktor seperti *maintainability*, *testability*, *flexibility*, *reusability*, dan *interoperability* memberikan kontribusi positif yang signifikan terhadap kualitas keseluruhan perangkat lunak.

16. Saran Tambahan dari hasil kuisioner

Disertakan juga untuk memberikan saran perbaikan pada form kuisioner dan menghasil beberapa saran yaitu peningkatan kapasatisa crane, perbaikan *error handling*, dan penerapan SPK yang kurang sesuai.

#### 5.1.1.6. Rekomendasi

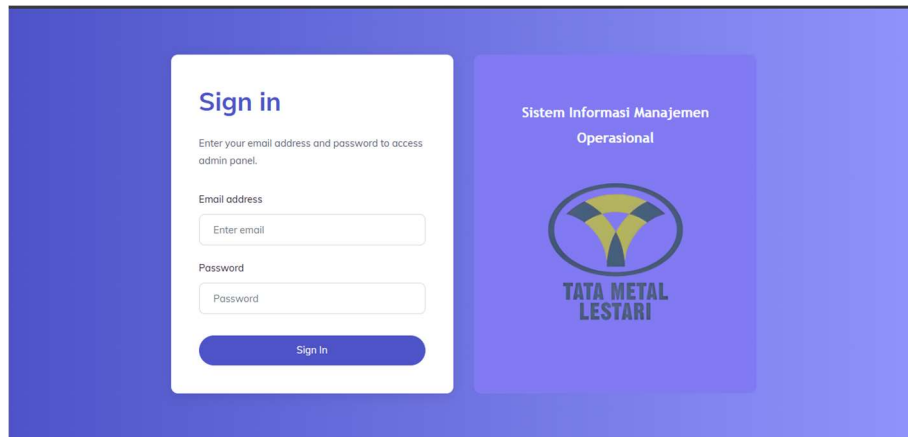
Berdasarkan hasil pengujian yang telah dilakukan, beberapa aspek kualitas sistem telah dinilai dengan berbagai metode. Berikut adalah rekomendasi perbaikan yang difokuskan pada aspek dengan nilai rendah:

1. Peningkatan Kapasitas *Crane*: Disarankan untuk mengevaluasi dan meningkatkan kapasitas crane guna mengoptimalkan kinerja sistem.
2. Perbaikan *Error Handling*: Perlu diterapkan mekanisme *error handling* yang lebih baik untuk mengurangi potensi kesalahan yang tidak terdeteksi.
3. Evaluasi Penerapan SPK: Perlu dilakukan peninjauan ulang terhadap penerapan SPK agar lebih sesuai dengan kebutuhan operasional.
4. Peningkatan *Maintainability*: Nilai *maintainability* yang masih cukup rendah menunjukkan perlunya dokumentasi kode yang lebih baik, refactoring kode untuk meningkatkan keterbacaan, serta penerapan standar pengkodean yang lebih konsisten.
5. Optimasi Efisiensi: Nilai efisiensi masih kurang optimal, terutama dalam hal kecepatan loading halaman. Perbaikan dapat dilakukan dengan optimasi gambar, penggunaan *caching*, serta memperbaiki performa *backend*.
6. Peningkatan *Usability*: Website masih kurang dalam menyediakan dokumentasi dan bantuan online. Direkomendasikan untuk menambahkan halaman khusus yang berisi panduan penggunaan dan dokumentasi sistem agar lebih mudah diakses oleh pengguna.

Dengan menerapkan rekomendasi ini, diharapkan sistem dapat memiliki kualitas yang lebih baik dan memberikan pengalaman pengguna yang lebih optimal.

#### 5.1.2. Implementasi Sistem

1. Implementasi halaman *login*



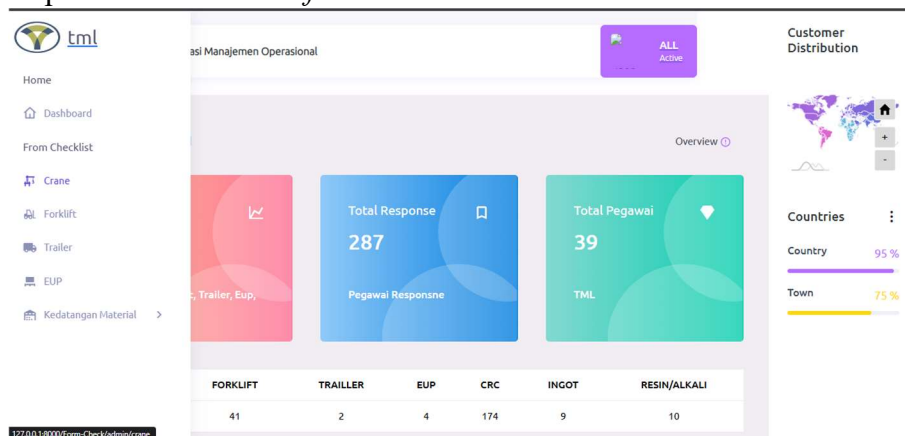
Gambar 5.18. Halaman *Login* Sistem Informasi Manejemn Oeprasional

## 2. Implementasi halaman utama



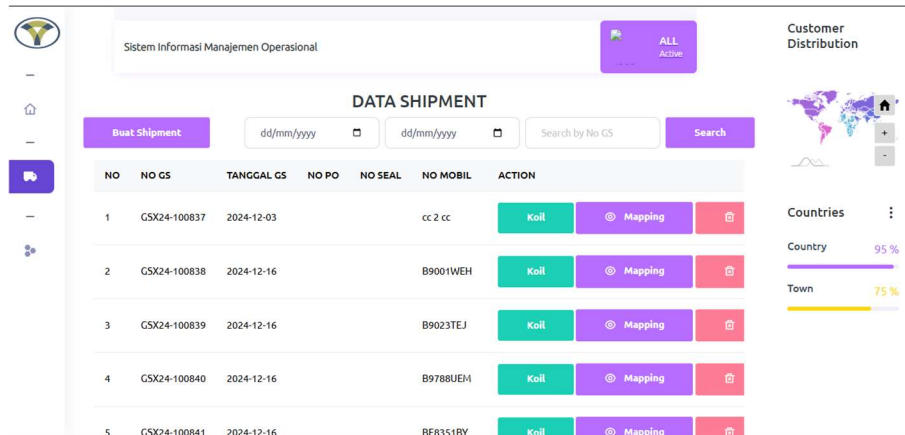
Gambar 5.19. Halaman Utama Sistem Informasi Manajemen Oeprasional

## 3. Implementasi halaman *form checklist*



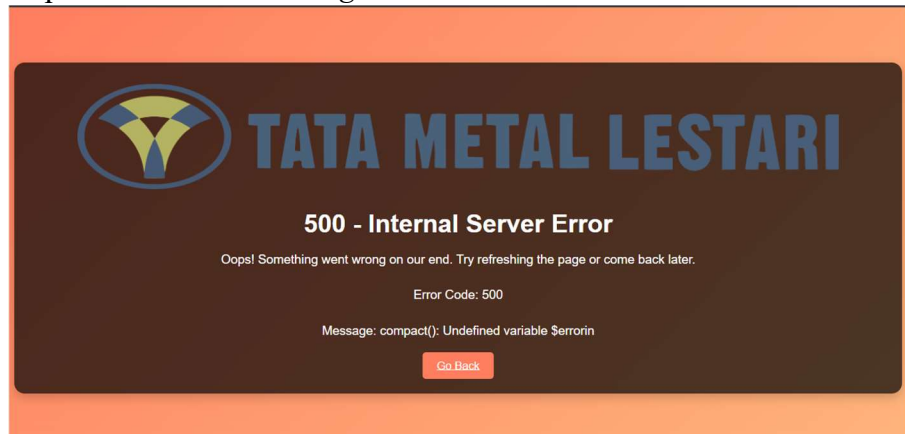
Gambar 5.20. Halaman *Form Checklist* Sistem Informasi Manejemn Oeprasional

## 4. Implemntasi halaman *mapping*



Gambar 5.21. Halaman *Mapping* Sistem Informasi Manejemn Oeprasional

## 5. Impemntasi *error handling*



Gambar 5.22. Halaman *Error Handling* Sistem Informasi Manajemen Oeprasional

### 5.1.3. *Sprint Execution*

Berdasarkan sprint pannnig yang dijelaskan pada bab 4 telah disusun *sprint execution* menyesuaikan dengan penimplemntasian yang sebenarnya dilakukan. Berikut adalah tabel dari *sprint execution*.

Tabel 5.16. *Sprint Execution*

Sprint	Durasi	Sprint Goal	Item Backlog	Increament Hasil	Sprint Reaview	Increament Delivery
Sprint 1	3 minggu	Analisis & perencanaan, dilakukan dengan menerapkan Mc	Penerapan Mc Call's Model & perencanaan pengembangan berdasarkan hasil dari Mc Call's Model	Semua selesai	-	

Sprint	Durasi	Sprint Goal	Item Backlog	Increment Hasil	Sprint Review	Increment Delivery
		Call's Model				
Sprint 2	4 Minggu	<i>Checklist Crane &amp; Shift Leader</i>	1. Penambahan kapasitas <i>crane</i> milik divisi produksi pada <i>checklist crane</i> di fitur <i>form checklist</i> 2. Penambahan <i>shift leader</i> pada <i>checklist crane</i> di fitur <i>form checklist</i> 3. perancangan sistem	Semua selesai	-	1. <i>checklist crane &amp; shift leader</i> pada <i>form check</i>
Sprint 3	4 Minggu	<i>Error Handling &amp; Review</i>	1. Penambahan error handling yang lebih baik 2. perancangan sistem	Semua selesai	-	1. <i>Error handling</i>
Sprint 4	3 Minggu	Penyempurnaan Mapping Muat & Finalisasi	1. Menghilangkan penggunaan SPK pada Mapping Muat 3. perancangan sistem	Semua selesai	-	1. <i>Mapping</i>

## 5.2. Evaluasi

Pengujian sistem dilakukan terhadap program yang telah dirancang dengan tujuan untuk memastikan bahwa sistem yang dibangun sesuai dengan hasil analisis dan perancangan yang telah dilakukan. Pengujian ini bertujuan untuk mencapai sebuah kesimpulan akhir yang *valid*. Dalam pengujian sistem ini, digunakan dua model atau metode pengujian utama, yaitu pengujian model *Black Box* dan UAT (*User Acceptance Testing*).

### 5.2.1. *BlackBox*

Pengujian model *Black Box* dilakukan dengan mempertimbangkan sistem sebagai suatu kotak hitam, di mana *input* diberikan ke sistem dan *output* yang dihasilkan dievaluasi tanpa memperhatikan bagaimana sistem mengolah *input*



tersebut. Metode ini fokus pada fungsi dan perilaku eksternal sistem. Tujuannya adalah untuk memastikan bahwa sistem memberikan hasil yang diharapkan tanpa perlu mengetahui rincian implementasinya.

Pengujian dilakukan oleh 3 orang responden yang merupakan *user* yang berkaitan dengan sistem ini. Nama penguji dapat dilihat pada tabel 5.1 di bawah ini:

Tabel 5.17. Koresponden *Black Box Testing*

No	Nama	Status
1.	Sheva Virtalioka	<i>User (Admin)</i>
2.	Asep Saepulloh	<i>User (Pegawai)</i>
3.	Rifky Frimanda	<i>User (Admin)</i>

Berikut merupakan tabel identifikasi dan perancangan pengujian yang digunakan pada pengujian *Black Box*, terdapat 17 kelas uji dengan 94 butir uji dan 21 *test case*. Berikut identifikasi dan perancangan pengujian dapat dilihat pada Tabel 5.2. di bawah ini:

Tabel 5.18. Butir Uji *Blacxbox*

Kelas Uji	Butir Uji	Tingkat Pengujian	Nomor Identifikasi	Jenis Pengujian
Melakukan Login	Validasi Login	Pengujian Sistem	LOG-1	<i>Blackbox</i>
	Pengujian Sesi	Pengujian Sistem	LOG-2	<i>Blackbox</i>
Melakukan Logout	Validasi Logout	Pengujian Sistem	LGT-1	<i>Blackbox</i>
	Pengujian Sesi	Pengujian Sistem	LGT-2	<i>Blackbox</i>
	Auto lagout	Pengujian Sistem	LGT-3	<i>Blackbox</i>
Mengelola data pegawai	Tampilkan halaman mengelola data pegawai	Pengujian Sistem	PW-1	<i>Blackbox</i>
	Tambah Pegawai	Pengujian Sistem	PW-2	<i>Blackbox</i>
	Edit Pegawai	Pengujian Sistem	PW-3	<i>Blackbox</i>
	Hapus Pegawai	Pengujian Sistem	PW-4	<i>Blackbox</i>
	Print Data Pegawai	Pengujian Sistem	PW-5	<i>Blackbox</i>
mengelola packing list	Menampilkan database	Pengujian Sistem	PL-1	<i>Blackbox</i>

Kelas Uji	Butir Uji	Tingkat Pengujian	Nomor Identifikasi	Jenis Pengujian
	Menambah database	Pengujian Sistem	PL-2	Blackbox
	Mengapus data di database	Pengujian Sistem	PL-3	Blackbox
	Menghapus Semua data di database	Pengujian Sistem	PL-4	Blackbox
	Menambah file	Pengujian Sistem	PL-5	Blackbox
	Mengedit data di database	Pengujian Sistem	PL-6	Blackbox
	Menampilkan hasil scan	Pengujian Sistem	PL-7	Blackbox
	Membuat pack baru	Pengujian Sistem	PL-8	Blackbox
	Mengedit hasil scan	Pengujian Sistem	PL-9	Blackbox
	Menghapus hasil scan	Pengujian Sistem	PL-10	Blackbox
	Menampilkan hasil packing list	Pengujian Sistem	PL-11	Blackbox
	Melihat detail hasil packing list	Pengujian Sistem	PL-12	Blackbox
	Mengexport hasil packing list	Pengujian Sistem	PL-13	Blackbox
Membuat shipmark	Menampilkan data shipmark	Pengujian Sistem	SP-1	Blackbox
	Menambahkan data shipmark	Pengujian Sistem	SP-2	Blackbox
	Mengupload file excel	Pengujian Sistem	SP-3	Blackbox
	Melihat detail shipmark	Pengujian Sistem	SP-4	Blackbox
	Mencetak shipmark	Pengujian Sistem	SP-5	Blackbox
	Mencetak semua shipmark	Pengujian Sistem	SP-6	Blackbox
	Mengedit shipmark	Pengujian Sistem	SP-7	Blackbox
	Menghapus shipmark	Pengujian Sistem	SP-8	Blackbox
Melihat checklist kendaraan	Menampilkan checklist kendaraan	Pengujian Sistem	CK-1	Blackbox

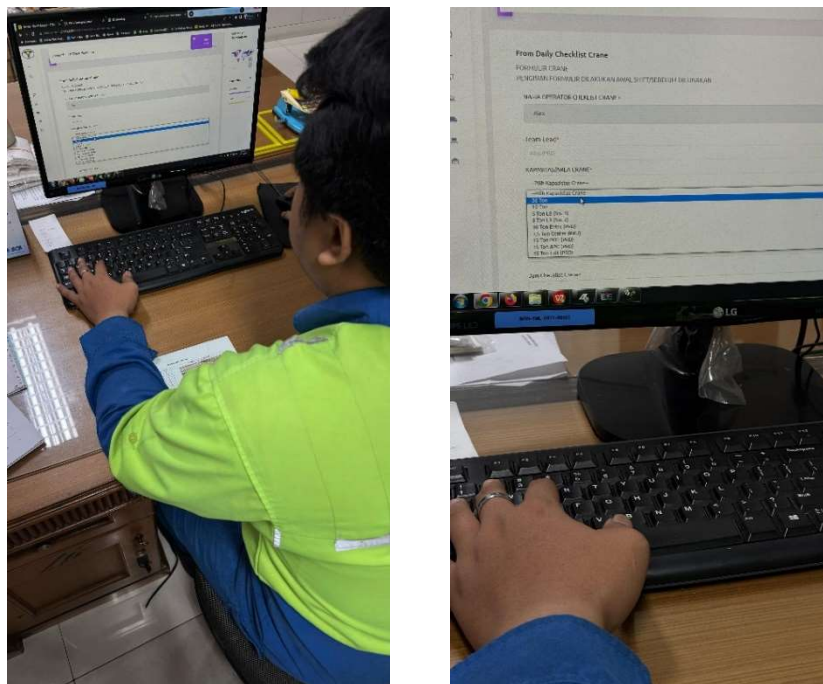
Kelas Uji	Butir Uji	Tingkat Pengujian	Nomor Identifikasi	Jenis Pengujian
	Melihat detail checklist	Pengujian Sistem	CK-2	Blackbox
	Mencetak checklist kendaraan	Pengujian Sistem	CK-3	Blackbox
	Mengekport checklist kendaraan	Pengujian Sistem	CK-4	Blackbox
	Mengedit detail checklist kendaraan	Pengujian Sistem	CK-5	Blackbox
	Menghapus detail checklist	Pengujian Sistem	CK-6	Blackbox
Menambah checklist kendaraan	Menampilkan Menambah data checklisr	Pengujian Sistem	CKP-1	Blackbox
mengelola scan layout	Menampilkan scan layout	Pengujian Sistem	SL-1	Blackbox
	membuat scan	Pengujian Sistem	SL-2	Blackbox
	Menghapus scan layout	Pengujian Sistem	SL-3	Blackbox
	Mengedit scan layout	Pengujian Sistem	SL-4	Blackbox
	Mengexport scan	Pengujian Sistem	SL-5	Blackbox
mengelola checklist response	Menampilkan checklist response	Pengujian Sistem	FCP-1	Blackbox
	Menambahkan response	Pengujian Sistem	FCP-2	Blackbox
	Menambahkan response resin	Pengujian Sistem	FCP-3	Blackbox
	Melihat detail response	Pengujian Sistem	FCP-4	Blackbox
	Melihat detail response resin	Pengujian Sistem	FCP-5	Blackbox
mengelola form checklist	Mengexport excel	Pengujian Sistem	FC-1	Blackbox
	Mengexport excel resin	Pengujian Sistem	FC-2	Blackbox
	Mencetak form checklist	Pengujian Sistem	FC-3	Blackbox
	menghapus form materia resin	Pengujian Sistem	FC-4	Blackbox

Kelas Uji	Butir Uji	Tingkat Pengujian	Nomor Identifikasi	Jenis Pengujian
mengelola mapping	Menampilkan mengelola mapping	Pengujian Sistem	MP-1	Blackbox
	Melihat data koil	Pengujian Sistem	MP-2	Blackbox
	Membuat shippment	Pengujian Sistem	MP-3	Blackbox
	Melihat detail koil	Pengujian Sistem	MP-4	Blackbox
	Melakukan Mapping	Pengujian Sistem	MP-5	Blackbox
	Mencetak hasil mapping	Pengujian Sistem	MP-6	Blackbox
Mengelola open pack	Menampilkan open pack	Pengujian Sistem	OP-1	Blackbox
	Menambahkan data GM	Pengujian Sistem	OP-2	Blackbox
	Membuat scan produk baru	Pengujian Sistem	OP-3	Blackbox
	Mengexport open pack	Pengujian Sistem	OP-4	Blackbox
	Menghapus open pack	Pengujian Sistem	OP-5	Blackbox
	Melihat detail open pack	Pengujian Sistem	OP-6	Blackbox
.Mengelola coil damage	Menampilkan coil damage	Pengujian Sistem	CD-1	Blackbox
	Menambahkan coil baru	Pengujian Sistem	CD-2	Blackbox
	Mengexport data coil	Pengujian Sistem	CD-3	Blackbox
	mengedit data coil	Pengujian Sistem	CD-4	Blackbox
	Menghapus data coil	Pengujian Sistem	CD-5	Blackbox
mengelola packing L-08	Menampilkan Packing L-08	Pengujian Sistem	L8-1	Blackbox
	Melihat detail rekap	Pengujian Sistem	L8-2	Blackbox
	Menghapus data rekap	Pengujian Sistem	L8-3	Blackbox
	Menambah rekap baru	Pengujian Sistem	L8-4	Blackbox
	Menghapus data detail	Pengujian Sistem	L8-5	Blackbox
	Mengexport data	Pengujian Sistem	L8-6	Blackbox
	Menambah koil	Pengujian Sistem	L8-7	Blackbox

Kelas Uji	Butir Uji	Tingkat Pengujian	Nomor Identifikasi	Jenis Pengujian
Mengakses surat izin keluar	Menampilkan surat izin keluar	Pengujian Sistem	SIK-1	Blackbox
	menambah surat	Pengujian Sistem	SIK-2	Blackbox
	mengedit surat	Pengujian Sistem	SIK-3	Blackbox
	menghapus surat	Pengujian Sistem	SIK-4	Blackbox
	mengexport surat	Pengujian Sistem	SIK-5	Blackbox
	mencetak surat	Pengujian Sistem	SIK-6	Blackbox
	menyetujui surat keluar	Pengujian Sistem	SIK-7	Blackbox
	mengizinkan pengajuan surat	Pengujian Sistem	SIK-8	Blackbox

### 5.2.2. User Acceptance Testing

Hasil pengujian *Black Box* yang telah dilakukan, menunjukkan bahwa *Software* yang dibangun telah memenuhi kebutuhan fungsional. Namun, pada prosesnya ada kemungkinan masih terjadi kesalahan maka dari itu diperlukan adanya *User Acceptance Testing* (UAT) untuk menguji *Software* yang dibangun sehingga telah sesuai dengan kebutuhan dan kenyamanan *user*.



Gambar 5.23. Penggunaan sistem oleh divisi produksi

## BAB 6

### KESIMPULAN DAN SARAN

#### 6.1. Kesimpulan

Berdasarkan hasil pengembangan dan evaluasi terhadap Sistem Informasi Manajemen Operasional PT Tata Metal Lestari, dapat disimpulkan bahwa:

1. Sistem informasi yang dikembangkan mampu meningkatkan dokumentasi dan efisiensi operasional, khususnya dalam fitur *checklist crane*, dengan menambahkan pencatatan kapasitas *crane* dan nama *shift leader*.
2. Penghapusan Sistem Perintah Kerja (SPK) pada fitur *Mapping Muat* berdasarkan hasil *User Acceptance Testing* (UAT) menjadikan sistem lebih sesuai dengan kondisi bisnis di lapangan.
3. Penerapan metode *Agile* dengan *framework* Scrum memungkinkan pengembangan sistem yang lebih terstruktur, fleksibel, dan terfokus pada kebutuhan pengguna melalui tahapan *product backlog*, *sprint planning*, *sprint execution*, hingga *sprint review*.
4. Evaluasi kualitas perangkat lunak dengan McCall's Model menunjukkan hasil sangat baik pada sebagian besar aspek, seperti *reliability*, *testability*, *reusability*, dan *interoperability*. Namun, masih terdapat beberapa aspek yang perlu ditingkatkan, seperti *maintainability* dan *efficiency*.
5. Salah satu kendala yang belum berhasil diatasi dalam penelitian ini adalah *upgrade server*. Kegagalan dalam meningkatkan spesifikasi *server* berdampak pada performa sistem, terutama dalam hal kecepatan boot dan kenyamanan pengguna saat mengakses aplikasi. Hal ini menjadi perhatian penting untuk pengembangan di masa mendatang agar sistem dapat berjalan lebih responsif dan stabil.

Dengan demikian, sistem yang telah dikembangkan telah memberikan kontribusi signifikan dalam mendukung kebutuhan Divisi *Warehouse* dan Produksi, meskipun masih diperlukan perbaikan lanjutan di beberapa aspek teknis.

## 6.2. Saran

1. *Upgrade server* perlu segera diprioritaskan agar sistem dapat memberikan performa yang lebih optimal, termasuk peningkatan kecepatan akses, kecepatan booting, dan kenyamanan pengguna dalam penggunaan harian.
2. Optimasi efisiensi sistem dapat dilakukan dengan menerapkan teknik kompresi gambar, *caching*, serta *refactoring* kode untuk mengurangi waktu muat halaman (*loading time*) dan penggunaan sumber daya yang berlebihan.
3. Perbaikan dokumentasi teknis dan standar pengkodean sangat penting untuk meningkatkan *maintainability* sistem, sehingga memudahkan pengembang dalam proses perawatan dan pengembangan lanjutan.
4. Penambahan fitur bantuan (*help/FAQ*) atau manual pengguna berbasis digital akan meningkatkan aspek *usability*, terutama bagi pengguna baru yang belum terbiasa dengan sistem.
5. Perlu dilakukan pengujian performa secara berkala untuk memastikan bahwa sistem tetap stabil dan dapat menyesuaikan dengan peningkatan jumlah pengguna serta kebutuhan bisnis yang berkembang.
6. Melibatkan pengguna dari berbagai divisi secara aktif dalam setiap tahapan pengembangan selanjutnya agar sistem benar-benar sesuai dengan kebutuhan dan memberikan dampak yang maksimal terhadap proses operasional perusahaan.

## DAFTAR PUSTAKA

- Aini, N., & Ratnawati, F. (2024). *Implementasi Metode Mccall Pada Pengujian Kualitas Website Diskominfo Kabupaten Bengkalis dengan alamat <https://diskominfo.kab.bengkalis.go.id> website ini mencakup informasi lunak McCall , Juliane , dkk ( 2021 ) Mengatakan bahwa teori kualitas McCall. 4.*
- Annaufal, A. N., Ardhani, M. N., Bintang, S., & Abetnego, G. (2025). *EXPLORE – Volume 15 No 1 Tahun 2025 Terakreditasi Sinta 5 SK No : 23 / E / KPT / 2019 Analisis Kerentanan pada Situs Buatinkamu . id dengan Pengujian Penetrasi Metode Black Box EXPLORE – Volume 15 No 1 Tahun 2025 Terakreditasi Sinta 5 SK No : 23 / E / KPT. 15(1), 11–17.*
- Fadila, R. H., Sabrina, P. N., & Ashaury, H. (2024). *Refactoring Untuk Meningkatkan Kualitas Reliabilitas Dan Correctness Pada Sistem Manajemen Sekolah Berbasis Web. 321–326.*
- Farhan, A., Wahab, A., Ri, F., M, F. N., & Alman, H. (2024). *Optimisasi Pagination dan Error Handling pada Portal Minat untuk Meningkatkan Performa Aplikasi Menggunakan Whitebox Testing Universitas Indo Global Mandiri , Indonesia Optimization of Pagination and Error Handling in Portal Minat to Improve Application P. 4(12), 603–611.*
- Fikri, R. R. N., Indera, I., Rahardi, A., & Agus, I. (2024). *Pengujian Blackbox pada Sistem Informasi Komunitas Pecinta Kucing di Bandar Lampung. Jurnal Teknika, 18(1), 25–34.*
- Hafidhin, R. A., Fitri, A. S., Fitri, S., Wati, A., Informasi, S., Raya, J., Madya, R., Anyar, G., Scrum, D., Review, S., & Karyawan, K. (2025). *Rancang bangun sistem rekomendasi kandidat karyawan berbasis website menggunakan metode scrum. 9(1), 492–499.*
- Hartono, B. (2021). *Cara Mudah dan Cepat Sistem Informasi.*
- Hasanah, F. N. (2020). *Buku Ajar Rekayasa Perangkat Lunak. In Buku Ajar Rekayasa Perangkat Lunak. <https://doi.org/10.21070/2020/978-623-6833-89-6>*
- Hilmyansyah, M., Malabay, M., Simorangkir, H., & Yulhendri, Y. (2022).



- Implementasi Metode Scrum Pada Pembangunan Sistem Informasi Monitoring Progress Proyek Berbasis Web (Studi Kasus: PT Quatra Engineering Mandiri). *Ikraith-Informatika*, 6(3), 30–40. <https://doi.org/10.37817/ikraith-informatika.v6i3.2198>
- Indriyani, F., Yunita, Muthia, D. A., Surniandari, A., & Sriyadi. (2019). 20. *Buku-Ajar-APSI\_2*. 1–90.
- Kalsel, P. (2025). *Desain sistem informasi pendaftaran pasien rawat jalan berbasis*. 3(1).
- Mardhia, M. M., & Khusna, A. N. (2020). *Petunjuk Praktikum Analisis Perancangan Perangkat Lunak*. 1–72.
- Max, R., & Gugat, D. (2023). Revitalisasi Manajemen Gudang Logistik melalui Penerapan Sistem Informasi Persediaan Digital. *Jurnal Pendidikan Tambusai*, 7, 18909–18914.
- Nistrina, K., & Sahidah, L. (2022). Unified Modelling Language (Uml) Untuk Perancangan Sistem Informasi Penerimaan Siswa Baru Di Smk Marga Insan Kamil. *Jurnal Sistem Informasi, J-SIKA*, 4(1), 17.
- Novianti, A., & Sari, R. P. (2022). Perancangan Sistem Gudang Material dengan Metode FAST pada PT. Samcon. *Jurnal Teknologi dan Informasi*, 12(1), 93–105. <https://doi.org/10.34010/jati.v12i1.6574>
- Octavianti, N. S., Hadiwiyan, R., & Efrat, R. (2025). *JTIM : Jurnal Teknologi Informasi dan Multimedia Penerapan User Centered Design untuk Optimisasi User Experience Aplikasi Virtusee*. 7(1), 11–22.
- Rahmatuloh, M., & Revanda, M. R. (2022). Rancang Bangun Sistem Informasi Jasa Pengiriman Barang Pada PT. Haluan Indah Transporindo Berbasis Web. *Jurnal Teknik Informatika*, 14(1), 54–59.
- Ramdani, S. K., & Zakaria, H. (2023). Penerapan Framework Laravel Dalam Rancangan Aplikasi Data Warehouse Untuk Optimalisasi Pencarian Barang Dengan Metode Lifo ( Studi Kasus : Kickoff Sports ). *JURIHUM : Jurnal Inovasi dan Humaniora*, 1(4), 486–498.
- Ramulu, K., & Murhtyr, B. V. R. (2020). Importance of Software Quality Models in Software Engineering. *International Journal of Engineering Technologies and Management Research*, 5(3), 200–218.

<https://doi.org/10.29121/ijetmr.v5.i3.2018.192>

- Richards, G., & Grinsted, S. (2020). The logistics and supply chain toolkit. In *Logistics and supply chain toolkit : over 100 tools and guides for supply chain, transport, warehousing and inventory management*.
- Ridlo, I. A. (2017). Pedoman Pembuatan Flowchart. *Academia.Edu*, 27. [academia.edu/34767055/Pedoman\\_Pembuatan\\_Flowchart](https://www.academia.edu/34767055/Pedoman_Pembuatan_Flowchart)
- Ronal, Yunita, & Yuliana. (2022). Desain Unified Modeling Language (UML) Dalam Perancangan Aplikasi Hauling Trip Di Industri Tambang Batubara. *Jurnal Teknik Informatika dan Sistem Informas*, 9(4), 3038–3050.
- Shaleh, I. A., Yogi, J. P., Pirdaus, P., Syawal, R., & Saifudin, A. (2021). Pengujian Black Box pada Sistem Informasi Penjualan Buku Berbasis Web dengan Teknik Equivalent Partitions. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 4(1), 38. <https://doi.org/10.32493/jtsi.v4i1.8960>
- Shovian, M., Al Baihaqi, H., Putra Kharisma, A., & Santoso, N. (2024). Pengembangan Aplikasi Sistem Informasi Manajemen Gudang Berbasis Web Menggunakan Metodologi Agile (Studi Kasus: CV. Jaya Laksa Lestari). 1(1), 2548–2964. <http://j-ptiik.ub.ac.id>
- Sitorus, B. A. (2023). Rancang Bangun Sistem Pengolahan Data Gaji Guru Berbasis Web. *Circle Archive*, 3(1). <http://circle-archive.com/index.php/carc/article/view/13%0Ahttp://circle-archive.com/index.php/carc/article/download/13/13>
- Suharni, Susilowati, E., & Pakusadewa, F. (2023). Perancangan Website Rumah Makan Ninik Sebagai Media Promosi Menggunakan Unified Modelling Language. *Rekayasa Informasi*, 12(1), 1–12. <https://ejournal.istn.ac.id/index.php/rekayasainformasi/article/view/1527/1021>
- Sumirat, L. P., Cahyono, D., Kristyawan, Y., & Kacung, S. (2023). *DASAR-DASAR Rekayasa Perangkat Lunak*.
- Vernanda, D. (2023). *Analisis dan Perancangan Sistem Informasi*. POLSUB PRESS.
- Wau, K. (2022). Pengembangan Sistem Informasi Persediaan Gudang Berbasis Website Dengan Metode Waterfall. *Jurnal Teknik, Komputer, Agroteknologi*

*Dan Sains*, 1(1), 10–23. <https://doi.org/10.56248/marostek.v1i1.8>

Widiarta, I. M., Mulyanto, Y., & Sutrianto, A. (2023). Rancang Bangun Sistem Informasi Inventory Menggunakan Metode Agile Software Development (Studi Kasus Toko Nada). *Digital Transformation Technology (Digitech)*, 3(Maret), 20.

Wong, E. (2025). *Delapan Aturan Emas Shneiderman Akan Membantu Anda Mendesain Antarmuka yang Lebih Baik*.

Zen Flowchart. (2022). *Jajaran Genjang dalam Diagram Alir - Simbol Diagram Alir Input / Output*.

