

**PERANCANGAN APLIKASI INVENTORY BAHAN LOGISTIK  
RUMAH SAKIT UMUM DAERAH SAWERIGADING PALOPO  
BERBASIS WEB MENGGUNAKAN METODE EXTREME  
PROGRAMMING**

**SKRIPSI**

“Diajukan untuk memenuhi salah satu persyaratan memperoleh Gelar Sarjana Komputer  
pada Program Studi Informatika”



Disusun oleh :

**FAIZAH**

**IK.22.11.018**

**FAKULTAS ILMU KOMPUTER  
UNIVERSITAS MEGA BUANA PALOPO**

**2024**

**HALAMAN PERSETUJUAN**

**SKRIPSI**

**PERANCANGAN APLIKASI INVENTORY BAHAN LOGISTIK RUMAH  
SAKIT UMUM DAERAH SAWERIGADING PALOPO BERBASIS WEB  
MENGUNAKAN METODE EXTREME PROGRAMMING**

yang disusun dan diajukan oleh

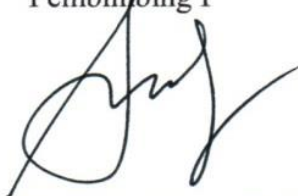


**Faizah**

**IK.22.11.018**

telah disetujui oleh Dosen Pembimbing Skripsi  
pada tanggal 08 Juni 2024

Pembimbing I



**Abdul Malik, S. Kom., M.Cs**  
NIDN : 0910049004

Pembimbing II



**Muh. Hajar Harike, S. Kom., M.Si.**  
NIDN : 0912069201

HALAMAN PENGESAHAN

SKRIPSI

PERANCANGAN APLIKASI INVENTORY BAHAN LOGISTIK RUMAH SAKIT UMUM DAERAH SAWERIGADING PALOPO BERBASIS WEB MENGGUNAKAN METODE EXTREME PROGRAMMING

yang disusun dan diajukan oleh



Faizah

IK.22.11.018

Telah dipertahankan di depan Dewan Penguji pada tanggal 08 Juni 2024

Dewan Penguji



**M. Hasanuddin, S. Kom., M.Kom.**

NIDN : 0927098905

Mengetahui

Pembimbing I



**Abdul Malik, S. Kom., M.Cs**

NIDN : 0910049004

Pembimbing II



**Muh. Hajar Harike, S. Kom., M.Si.**

NIDN : 0912069201

Skripsi ini telah diterima sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Komputer Tanggal 08 Juni 2024

DEKAN FAKULTAS ILMU KOMPUTER



**Abdul Malik, S. Kom., M.Cs.**

NIDN : 0910049004

## PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini :

Nama : Faizah  
NIM : IK. 22.11.018  
Tahun terdaftar : 2022  
Program Studi : S1 Informatika  
Fakultas : Ilmu Komputer  
Universitas : Universitas Mega Buana Palopo

Menyatakan bahwa dalam dokumen ilmiah Skripsi ini tidak terdapat bagian dari karya ilmiah lain yang telah diajukan untuk memperoleh gelar akademik di suatu Lembaga Pendidikan Tinggi, dan juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang/ lembaga lain, kecuali yang secara tertulis disitasi dalam rbitkan oleh orang/ lembaga lain, kecuali yang secara tertulis disitasi dalam dokumen ini dan disebutkan sumbernya secara lengkap dalam daftar Pustaka.

Dengan demikian saya menyatakan bahwa dokumen ilmiah ini bebas dari unsur-unsur plagiasi dan apabila dokumen ilmiah Skripsi ini di kemudian hari terbukti merupakan plagiasi dari hasil karya penulis lain dan/atau dengan sengaja mengajukan karya atau pendapat yang merupakan hasil karya penulis lain, maka penulis bersedia menerima sanksi akademik dan/atau sanksi hukum yang berlaku.

Palopo, 08 Juni 2024

  
**Faizah**  
IK. 22.11.018

## KATA PENGANTAR

*Assalamu'alaikum Warahmatullahi Wabarokatuh*

Alhamdulillah Rabbil 'Alamin. Segala puji dan syukur penulis panjatkan kepada Allah Subhanau wa Ta'ala karean atas rahmat dan karunia-Nya penulis dapat menyelesaikan Skripsi yang berjudul " Perancangan Aplikasi Inventory Bahan Logistik Rumah Sakit Umum Daerah Sawerigading Palopo Berbasis Web Menggunakan Metode Extreme Programming ". Shalawat serta salam semoga selalu tercurahkan kepada Rasulullah Shalallahu'alaihi wassalam yang telah membimbing umatnya dari zaman kegelapan ke zaman terang menderang seperti sekarang ini.

Skripsi ini disusun sebagai syarat yang harus dipenuhi menyelesaikan studi jenjang strata satu (S1) di Program Studi Teknik Informatika Universitas Mega Buana Palopo. Skripsi ini tidak akan berhasil tanpa peran serta dukungan dari berbagai pihak, yang senantiasa memberikan saran, dan motivasi yang sangat membantu untuk menyelesaikan skripsi ini, mengingat pengetahuan dan pengalaman penulis yang terbatas. Oleh karena itu, ucapan terimakasih disampaikan kepada:

1. Allah SWT yang telah mencurahkan rahmat dan hidayahnya hingga detik ini.
2. Kedua otang tua saya yaitu Ayahanda Alm. Abdullah dan Ibunda Siti Aminah yang senantiasa memanjatkan do'a, mencurahkan kasih sayang, dan memberikan dukungan berupa moril maupun materil hingga menjadi penyemangat dalam menyelesaikan skripsi ini.
3. Suami tercinta Ilmal, S.Pd dan anak-anakku sayang (Nadhifa Zahra Putri Alisah dan Muh. Dhafir Fakhriyan) terima kasih atas segala dukungan dan pengertiannya selama ibu (penulis) kuliah hingga menyelesaikan penyusunan skripsi ini.
4. Jumail, M. Sukri, Hermila, Irma, Wirnawati dan Fazilah selaku kakak kandung saya yang selalu mendo'akan saya dan memberikan dukungan dan motivasi kepada penulis dalam pengerjaan skripsi ini.
5. Bapak Abdul Malik, S. Kom., M. Cs. selaku dosen Pembimbing I dan Bapak M. Hajar Harike, S. Kom, MM., selaku dosen Pembimbing II saya. Terimakasih karena telah banyak meluangkan waktu, tenaga, dan kesabaran dalam membimbing. Serta kebijaksanaan dalam memberikan saran dan nasihat yang sangat bermanfaat bagi penulis dalam mengerjakan skripsi ini.

6. Seluruh dosen dan staf Teknik Informatika yang telah memberikan pendidikan serta ilmu yang bermanfaat bagi penulis selama mengampu pendidikan di Universitas Mega Buana Palopo.
7. Seluruh sahabat-sahabat saya (Risma Idris, Rena Jihad, Vivi Novianty, dan yang senantiasa memberikan motivasi serta membantu dalam menyelesaikan skripsi ini.
8. Sahabat-sahabat seperjuangan skripsi (Ira Widyastuti, Nur Wildayani, Musdy dan Faizal) yang senantiasa saling membantu satu sama lain untuk menyelesaikan skripsi ini.
9. Semua pihak yang telah membantu baik secara langsung maupun tidak langsung yang tidak bisa penulis sebutkan satu persatu.

Keterbatasan kemampuan, pengetahuan dan pengalaman penulis dalam pembuatan skripsi ini masih jauh dari kata sempurna. Untuk itu penulis akan selalu menerima segala masukan yang ditujukan untuk menyempurnakan skripsi ini. Akhir kata semoga Allah SWT membalas segala kebaikan yang penulis telah terima dan mengharapkan semoga skripsi ini dapat bermanfaat serta menambah wawasan pengetahuan baik bagi penulis sendiri maupun bagi pihak yang membutuhkan.

*Wassalamu`alaikum Warahmatullahi Wabarakatuh*

Palopo, 08 Juni 2024



Faizah

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PERSETUJUAN .....	I
HALAMAN PENGESAHAN .....	III
HALAMAN PERNYATAAN BEBAS PLAGIASI.....	III
KATA PENGANTAR .....	V
DAFTAR TABEL.....	IX
DAFTAR GAMBAR.....	X
DAFTAR LAMBANG DAN SINGKATAN .....	XII
DAFTAR ISTILAH.....	XIII
INTISARI .....	XIV
ABSTRACT.....	XV
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan Penelitian.....	2
1.5 Manfaat Penelitian.....	3
1.6 Metodologi Penelitian.....	3
1.6.1 Metode Pengumpulan Data .....	3
1.6.2 Metode Pengembangan Sistem .....	4
1.7 Sistematika Penulisan .....	5
BAB II TINJAUAN PUSTAKA .....	7
2.1 Studi Literatur.....	7
2.2 Dasar Teori .....	11
2.2.1 Profil RSUD Sawerigading Palopo.....	11
2.2.2 Pengertian <i>Inventory</i> /Persediaan.....	11
2.2.3 Sistem Informasi Berbasis Web.....	12
2.2.4 Metode <i>Extreme Programming (XP)</i> .....	12
2.2.5 <i>Framework Codeigniter</i> .....	14
2.2.6 Perancangan Sistem .....	16
2.2.7 Bahan Logistik Non Medis .....	29
2.2.8 Kerangka Berfikir Penelitian.....	30
BAB III ANALISIS DAN PERANCANGAN SISTEM.....	31
3.1 Analisis Sistem .....	31
3.2 Deskripsi Data .....	36
3.2.1 Data Input.....	36
3.2.2 Perancangan User Interface (Antarmuka Pengguna) .....	53
3.4 Rancangan Pengujian Sistem .....	57
BAB IV IMPLEMENTASI .....	58
4.1 Implementasi .....	58
4.1.1 Implementasi Sistem .....	58
4.1.2 Implementasi Database .....	58
4.1.4 Implementasi Program .....	59
4.3 Hasil Pengujian.....	65
4.3.1 Hasil Pengujian Black Box .....	65
4.3.2 Hasil Pengujian White Box.....	65
BAB V PENUTUP .....	67

5.1 Kesimpulan.....	67
5.2 Saran.....	67



## DAFTAR TABEL

Tabel 2.1 Class Diagram Inventory .....	9
Tabel 2.2 Simbol-Simbol Flowchart.....	19
Tabel 2.3 Simbol Use Case Diagram.....	20
Tabel 2.4 Simbol-Simbol Class Diagram (Indra Sukma, 2022).....	22
Tabel 2.5 Simbol Sequncial Diagram .....	23
Tabel 2.6 Simbol Activity Diagram.....	24
Tabel 2.7 Simbol-Simbol Entity Relationship Diagram.....	25
Tabel 2.8 Tabel Hubungan Cyclomatic Complexity Dengan Resiko.....	28
Tabel 3.1 Deskripsi Pendefinisian Use Case .....	40
Tabel 3.2 Szenario Use Case Login.....	42
Tabel 3.3 Szenario Use Case Data Barang .....	42
Tabel 3.4 Szenario Use Case Transaksi Barang Masuk .....	43
Tabel 3.5 Szenario Use Case Transaksi Barang Keluar .....	43
Tabel 3. 6 Szenario Use Case Transaksi Pengajuan Logistik.....	43
Tabel 3.7 Szenario Use Case Transaksi Pengajuan Logistik.....	44
Tabel 3.8 Szenario Use Case Laporan .....	44
Tabel 3.9 Szenario Use Case View Laporan .....	45
Tabel 3.10 Szenario Use Case Manajemen User.....	45
Tabel 3.11 Perancangan Tabel User .....	50
Tabel 3.12 Perancangan Tabel Barang .....	51
Tabel 3.13 Perancangan Tabel Barang Masuk .....	51
Tabel 3.14 Perancangan Tabel Barang Keluar .....	52
Tabel 3.15 Perancangan Tabel Pengajuan Logistik.....	52
Tabel 3.16 Perancangan Tabel Satuan Barang .....	52
Tabel 3.17 Perancangan Tabel Jenis Barang .....	53
Tabel 3.18 Tabel Perancangan Black Box Testing.....	57
Tabel 4.1 Tabel Pengujian Black Box .....	65
Tabel 4.2 Cyclomatic Complexity Form Login.....	68
Tabel 4.3 Graph Matrix Form Login .....	68
Tabel 4.4 Tabel Test Case Form Login .....	68
Tabel 4. 5 Tabel Cyclomatic Complexity Router Form Tambah Barang.....	70
Tabel 4.6 Graph Matrix Router Form Tambah Barang .....	71
Tabel 4.7 Tabel Test Case Router Form Tambah Barang .....	71
Tabel 4.8 Cyclomatic Complexity Router Form Tambah Transaksi Masuk.....	73
Tabel 4.9 Graph Matrix Router Form Tambah Transaksi Masuk .....	74

## DAFTAR GAMBAR

Gambar 2.1 Tahapan – tahapan model Extreme Programming.....	13
Gambar 2.2 Sistem Manajemen Database .....	17
Gambar 2.3 Sistem Manajemen Basis Data.....	17
Gambar 2. 4 Cara Kerja PHP.....	19
Gambar 2. 5 Black Box Testing.....	26
Gambar 2.6 White Box Testing .....	27
Gambar 2.7 Kerangka Penelitian .....	31
Gambar 3.1 Flowchart Sistem Barang Masuk.....	31
Gambar 3.2 Flowchart Sistem Pendistribusian Bahan Logistik .....	32
Gambar 3.3 Flowchart Analisis sistem yang diusulkan.....	36
Gambar 3.4 Diagram Arsitekur Sistem.....	39
Gambar 3.5 Use Case Diagram Inventory .....	40
Gambar 3.6 Activity Diagram Login.....	46
Gambar 3.7 Activity Diagram Transaksi Barang Masuk dan Keluar.....	47
Gambar 3.8 Activity Diagram Pengajuan Logistik .....	48
Gambar 3.9 Activity Diagram Bendahara Barang.....	49
Gambar 3.10 Class Diagram Inventory barang logistik .....	50
Gambar 3. 11 Interface Halaman Login .....	53
Gambar 3.12 Interface Halaman Dashboard.....	54
Gambar 3. 13 Interface Halaman Data Barang.....	54
Gambar 3.14 Interface Halaman Data Jenis Barang.....	55
Gambar 3. 15 Interface Halaman Transaksi Barang Masuk.....	55
Gambar 3.16 Interface Halaman Transaksi Barang Keluar .....	56
Gambar 3.17 Interface Halaman Laporan Barang Masuk .....	56
Gambar 3.18 Interface Halaman Laporan Barang Keluar .....	57
Gambar 3.19 Source Code Router Form Tambah Transaksi Masuk.....	72
Gambar 3.20 Flowgraph Router Form Tambah Transaksi Masuk .....	73
Gambar 4.1 Implementasi Database .....	59
Gambar 4.2 Tampilan Halaman Login .....	59
Gambar 4.3 Tampilan Halaman Utama .....	60
Gambar 4.4 Tampilan Halaman Data Barang.....	60
Gambar 4.5 Tampilan Halaman Jenis Barang .....	61
Gambar 4. 6 Tampilan Halaman Satuan Barang .....	61
Gambar 4.7 Tampilan Halaman Transaksi Barang Masuk.....	62
Gambar 4.8 Tampilan Halaman Transaksi Barang Keluar.....	62
Gambar 4.9 Tampilan Halaman Laporan Stok Barang.....	63
Gambar 4.10 Tampilan Halaman Laporan Barang Masuk .....	63
Gambar 4.11 Hasil Laporan Barang Masuk .....	64
Gambar 4.12 Tampilan Hasil Laporan Barang Keluar .....	64

Gambar 4.13 Tampilan Hasil Laporan Barang Keluar .....	65
Gambar 4.14 Tampilan Halaman Manajemen User .....	65
Gambar 4.15 Source Code Form Login.....	67
Gambar 4.16 Source Code Form Login.....	67
Gambar 4.17 Flowgraph Form Login .....	67
Gambar 4.18 Source Code Router Form Tambah Barang.....	69
Gambar 4.19 Flowgraph Form Tambah Data Barang .....	70

## DAFTAR LAMBANG DAN SINGKATAN

RSUD	:	Rumah Sakit Umum Daerah
XP	:	Extreme Programming
UML	:	Unified Modelling Language
ERD	:	Entity Relationship Diagram
PHP	:	Hypertext Preprocessor
MySQL	:	My Structured Query Language
RUP	:	Rational Unified Process
RAD	:	Rapid Application Development
FEFO	:	First Expired First Out
GUI	:	Antarmuka Pengguna Grafis
KIS	:	Keep it Simple
CSS	:	Cascading Style Sheets
MVC	:	Model View Controller
DBMS	:	Sistem Manajemen Database
HTML	:	HyperText Markup Language
CRUD	:	Create, Read, Update and Delete
CC	:	Cyclomatic Complexity

## DAFTAR ISTILAH

Inventory : persediaan barang  
Konvensional : pengelola data masih sederhana

## INTISARI

Rumah Sakit Umum Daerah (RSUD) merupakan salah satu lembaga kesehatan yang memiliki peran penting dalam memberikan pelayanan kesehatan kepada masyarakat. Untuk menjaga kualitas layanan yang optimal, RSUD membutuhkan manajemen yang efektif dalam pengelolaan bahan logistiknya. *Inventory* atau persediaan memiliki peran yang krusial dalam operasional sehari-hari sebuah RSUD. *Inventory* yang tidak terkelola dengan baik dapat menyebabkan berbagai masalah, seperti kelangkaan bahan medis penting, pemborosan sumber daya, dan kesulitan dalam pemantauan stok. Pada RSUD Sawerigading Kota Palopo di bagian logistik masih mengelola data barang secara manual dengan mencatat barang masuk dan keluar di buku. Setelah pencatatan manual ini, data tersebut dimasukkan ke dalam Microsoft Excel. Karena masih menggunakan metode manual untuk pengolahan data barang, sering terjadi kesalahan input data yang berulang, dan persediaan barang tidak tercatat secara real-time. Hal ini membuat informasi tentang stok atau persediaan barang menjadi tidak efisien. Oleh karena itu, solusi terbaik untuk mengatasi masalah ini adalah dengan membangun sistem *inventory* persediaan barang yang memonitor proses masuk dan keluarnya stok serta persediaan barang di Rumah Sakit Umum Sawerigading Palopo. Sistem ini akan dirancang berbasis web dan menggunakan metode *Extreme Programming*. Tujuan penelitian ini adalah merancang sistem informasi persediaan bahan logistik berbasis web menggunakan metode *Extreme Programming (XP)* untuk mengatasi masalah pengelolaan persediaan bahan logistik di Rumah Sakit. Metode XP dipilih karena memungkinkan pembangunan sistem yang lebih cepat dan sangat fleksibel terhadap perubahan yang terjadi selama proses pengembangan perangkat lunak. Hasil dan manfaat dari perancangan aplikasi *inventory* ini adalah menghasilkan suatu aplikasi yang efisien, tepat dan fleksibel dalam membantu dan memudahkan proses pendataan barang logistik pada Rumah Sakit Sawerigading Kota Palopo.

**Kata kunci:** *inventory, logistik, web.*

## **ABSTRACT**

*The Regional General Hospital (RSUD) is a key healthcare institution that plays a vital role in providing health services to the community. To maintain optimal service quality, RSUD requires effective management of its logistical supplies. Inventory management is crucial for the daily operations of an RSUD. Poorly managed inventory can lead to various issues, such as shortages of essential medical supplies, resource wastage, and difficulties in stock monitoring. At RSUD Sawerigading Kota Palopo, the logistics department still manages inventory data manually by recording incoming and outgoing items in a book, after this manual recording, the data is then entered into Microsoft Excel. Due to the reliance on manual methods for inventory data processing, frequent data entry errors occur, and inventory levels are not tracked in real-time. This situation makes stock information inefficient. Therefore, the best solution to address this issue is to develop an inventory system that monitors the flow of stock and supplies at Sawerigading General Hospital in Palopo. This system will be web-based and use the Extreme Programming (XP) methodology. The goal of this research is to design a web-based inventory management system using the Extreme Programming (XP) methodology to address the logistical supply management issues at the hospital. The XP methodology is chosen because it allows for faster system development and is highly adaptable to changes that occur during the software development process. The results and benefits of designing this inventory application include producing an efficient, accurate, and flexible application that facilitates and simplifies the logistics inventory management process at Sawerigading General Hospital in Palopo.*

**Keywords:** *inventory, logistic, web.*

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Teknologi memiliki dampak yang signifikan pada semua aspek kehidupan, termasuk persaingan di pasar kerja yang mendorong individu untuk memperoleh pengetahuan lebih lanjut tentang kemajuan teknologi. Dalam pengolahan data, strategi diperlukan bersama dengan sarana pendukung untuk mempercepat pengolahan dan menghasilkan laporan yang diperlukan perusahaan. Sistem informasi berbasis komputer dan teknologi informasi sangat penting karena kemampuannya menyajikan informasi dengan cepat dan akurat. *Inventory* mencerminkan sumber daya yang tersedia tetapi belum digunakan sepenuhnya. Ini dapat diinterpretasikan sebagai stok yang tersedia pada saat itu, daftar rincian barang yang tersedia, atau jumlah barang yang dimiliki oleh suatu organisasi pada waktu tertentu. Fungsi utama dari *Inventory* adalah memenuhi permintaan pelanggan dengan persediaan barang yang minimal (Limbong & Sianipar, 2022).

Rumah Sakit menjadi salah satu lembaga kesehatan yang memiliki peran penting dalam memberikan pelayanan kesehatan kepada masyarakat. Untuk menjaga kualitas layanan yang optimal, rumah sakit membutuhkan manajemen yang efektif dalam pengelolaan bahan logistiknya. Salah satu aspek yang menjadi perhatian dalam manajemen tersebut adalah pengelolaan *inventory*. *Inventory* atau persediaan memiliki peran yang krusial dalam operasional sehari-hari sebuah rumah sakit. *Inventory* yang tidak terkelola dengan baik dapat menyebabkan berbagai masalah, seperti kelangkaan bahan penting, pemborosan sumber daya, dan kesulitan dalam pemantauan stok. Oleh karena itu, pengelolaan *inventory* yang efektif menjadi suatu kebutuhan yang tidak dapat diabaikan.

Dalam konteks RSUD Daerah Sawerigading Palopo, pengelolaan *inventory* masih dilakukan secara konvensional, yang cenderung rentan terhadap kesalahan dan kurang efisien dalam pengelolaannya. Pada RSUD Sawerigading Kota Palopo bagian logistik masih mengelola data barang secara manual dengan mencatat barang masuk dan keluar di buku. Setelah pencatatan manual ini, data tersebut dimasukkan ke dalam Microsoft Excel. Hal ini membuat informasi tentang stok atau persediaan barang menjadi tidak efisien. Dengan demikian dapat menyebabkan kesalahan dalam pencatatan barang, tidak akurat nya laporan



barang, dan kesulitan dalam pencarian barang yang dibutuhkan. Untuk itu peneliti melakukan penelitian sebagai alternatif dengan judul “**Perancangan Aplikasi Inventory Bahan Logistik Rumah Sakit Umum Daerah Sawerigading Palopo Berbasis Web Menggunakan Metode Extreme Programming**”. Penggunaan teknologi informasi, khususnya aplikasi berbasis *web*, dapat menjadi solusi yang efektif untuk meningkatkan efisiensi dan akurasi dalam pengelolaan *inventory*.

Dengan memanfaatkan teknologi berbasis *web*, diharapkan RSUD Daerah Sawerigading Palopo dapat memiliki sistem pengelolaan *inventory* yang mudah diakses oleh seluruh pihak yang berkepentingan. Aplikasi yang dirancang akan memungkinkan pengguna untuk melakukan pemantauan stok secara *real-time*, melakukan pengelolaan bahan logistik, serta menghasilkan laporan yang akurat untuk mendukung proses pengambilan keputusan.

Dengan demikian, diharapkan penelitian ini dapat memberikan kontribusi yang signifikan dalam pengembangan sistem *inventory* di RSUD Daerah Sawerigading Palopo dan dapat menjadi referensi bagi penelitian serupa di masa mendatang.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang dan identifikasi masalah, maka dapat dirumuskan bahwa permasalahan pada penelitian ini adalah bagaimana merancang Aplikasi *Inventory* Bahan Logistik Berbasis Web Pada Rumah Sakit Umum Daerah Sawerigading Palopo Menggunakan Metode *Extreme Programming*.

## **1.3 Batasan Masalah**

Penulis menentukan batasan dan ruang lingkup permasalahan yang akan diterapkan pada penelitian ini, yaitu :

1. Rancangan aplikasi pengolahan *inventory* data bahan logistik yang dibuat berbasis website.
2. Rancangan aplikasi yang dibuat mencakup pengelolaan barang masuk, pendistribusian bahan logistik
3. Rancangan aplikasi *inventory* menghasilkan output berupa laporan barang masuk, laporan persediaan barang dan laporan barang keluar.

## **1.4 Tujuan Penelitian**

Berdasarkan uraian latar belakang dan rumusan masalah, maka dapat disampaikan tujuan dari penelitian ini adalah sebagai berikut :

1. Terciptanya sebuah aplikasi inventory bahan logistik rumah sakit berbasis web.
2. Membuat sebuah aplikasi yang mencakup pengelolaan barang masuk, stok barang dan pendistribusian bahan logistik rumah sakit.
3. Membuat sebuah aplikasi yang dapat menghasilkan laporan barang masuk, laporan persediaan barang dan laporan barang keluar pada rumah sakit.

## **1.5 Manfaat Penelitian**

Manfaat yang diharapkan dari penelitian ini adalah :

1. Bagi RSUD Daerah Sawerigading Palopo diantaranya :
  - a. Aplikasi ini memungkinkan rumah sakit untuk melacak dan mengelola barang mereka dengan lebih efisien.
  - b. Aplikasi ini memungkinkan untuk pemantauan persediaan secara real-time, yang memungkinkan staf rumah sakit untuk mengetahui kapan mereka perlu memesan ulang atau mengelola persediaan dengan lebih efisien.
  - c. Dengan pengelolaan aset yang lebih baik, rumah sakit dapat fokus lebih banyak pada memberikan perawatan berkualitas kepada pasien. Peralatan yang tersedia dan berfungsi dengan baik akan membantu dalam memberikan pelayanan yang lebih baik.
2. Bagi Universitas :

Kesuksesan proyek implementasi teknologi informasi dapat meningkatkan reputasi universitas dalam hal inovasi, pengabdian kepada masyarakat, dan kemampuan untuk menciptakan solusi praktis untuk masalah nyata. Ini dapat meningkatkan daya tarik universitas bagi calon mahasiswa, dosen, dan mitra potensial lainnya.
3. Untuk peneliti dan pengembang sistem berikutnya  

Proyek ini dapat menjadi sumber referensi yang berharga bagi peneliti dan pengembang sistem berikutnya dalam hal perancangan, implementasi, dan evaluasi sistem informasi. Mereka dapat mempelajari pengalaman, tantangan, dan solusi yang ditemukan selama proses implementasi untuk menghindari kesalahan yang sama dan meningkatkan efisiensi proses pengembangan.

## **1.6 Metodologi Penelitian**

### **1.6.1 Metode Pengumpulan Data**

Metode pengumpulan data yang digunakan oleh penulis dalam melakukan pengumpulan data untuk pembuatan skripsi adalah :

#### 1. Studi Pustaka

Penulis menggunakan studi pustaka untuk merujuk pada teori dan pendapat yang terdapat dalam berbagai sumber seperti buku, jurnal, atau dokumen lainnya, dengan tujuan mendukung penelitian ini melalui analisis beberapa literatur yang relevan dengan topik yang dibahas.

#### 2. Studi Lapangan

Studi lapangan dilakukan secara langsung di Rumah Sakit Umum Daerah Sawerigading Kota Palopo pada bagian logistik untuk mendapatkan data-data yang diperlukan pada penelitian ini, dengan melakukan :

##### a. *Observasi*

Observasi dilakukan Penulis melakukan pengamatan langsung pada RSUD Daerah Sawerigading Palopo terhadap kegiatan sistem *inventory* barang.

##### b. *Wawancara*

Untuk mendapatkan informasi yang lebih lengkap maka penulis melakukan proses pengumpulan data-data yang dilakukan dengan cara tanya jawab pada admin bagian gudang dan bagian barang.

#### 3. Studi Literatur

Studi literatur dilakukan oleh penulis untuk mendapatkan gambaran penyelesaian masalah yang dapat dilakukan dari penelitian sejenis sebelumnya. Dari hasil studi literatur ini, peneliti dapat mengambil kesimpulan dari penelitian sejenis sebelumnya untuk membuat usulan sistem yang lebih baik. Penelitian sejenis tersebut dapat berasal dari skripsi dan jurnal yang ada pada internet.

### **1.6.2 Metode Pengembangan Sistem**

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode *Extreme Programming (XP)* yang menggabungkan berbagai ide sederhana tanpa mengorbankan kualitas perangkat lunak yang akan dikembangkan.

Pengembangan sistem yang digunakan adalah *Extreme Programming (XP)* yang terdiri dari 4 tahapan yaitu :

#### 1. *Planning*

Pada tahap perencanaan ini, langkah awal dalam pembangunan sistem dimana dalam tahapan ini dilakukan beberapa kegiatan perencanaan yaitu identifikasi permasalahan, menganalisa kebutuhan bisnis, kebutuhan pengguna dan kebutuhan sistem.

## 2. *Design*

Tahapan ini merupakan bagian dari perancangan sistem dimana dilakukan pemodelan sistem dan pemodelan arsitektur sistem menggunakan diagram Unified Modelling Language (UML), serta pemodelan basis data menggunakan Entity Relationship Diagram (ERD)

## 3. *Coding*

Pada tahap ini adalah kegiatan penerapan pemodelan yang sudah dibuat kedalam bentuk user interface dengan menggunakan bahasa pemrograman. Adapun bahasa pemrograman yang digunakan adalah PHP dengan metode terstruktur. Untuk sistem manajemen basis data menggunakan piranti lunak MySQL.

## 4. *Testing*

Setelah tahapan pengkodean selesai, kemudian dilakukan tahapan pengujian sistem untuk mengetahui kesalahan apa saja yang timbul saat aplikasi sedang berjalan serta mengetahui apakah sistem yang dibangun sudah sesuai dengan kebutuhan pengguna. Metode pengujian yang digunakan pada tahapan ini adalah metode *whitebox testing* dan *blackbox testing*, dimana pengujian yang dilakukan terhadap form beberapa masukkan apakah sudah berjalan sesuai dengan fungsinya masing – masing.

### 1.7 Sistematika Penulisan

Penelitian ini memiliki susunan penelitian dengan pembahasan yang terbagi menjadi lima bab yakni :

#### **BAB I                    PENDAHULUAN**

Pada bab ini, penulis memaparkan penelitian “Perancangan Aplikasi Inventory Bahan Logistik Rumah Sakit Umum Daerah Sawerigading Palopo Berbasis Web Menggunakan Metode Extreme Programming” secara padat dan singkat menjadi latar belakang, identifikasi masalah, rumusan masalah, batasan

masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penelitian.

## **BAB II                    TINJAUAN PUSTAKA**

Pada bab ini, penulis memaparkan seluruh teori yang penulis gunakan sebagai dasar landasan penelitian “Perancangan Aplikasi Inventory Bahan Logistik Rumah Sakit Umum Daerah Sawerigading Palopo Berbasis Web Menggunakan Metode Extreme Programming”.

## **BAB III                  ANALISIS DAN PERANCANGAN SISTEM**

Pada bab ini, penulis memaparkan analisis sistem dan perancangan sistem serta yang penulis gunakan di dalam penelitian “Perancangan Aplikasi Inventory Bahan Logistik Rumah Sakit Umum Daerah Sawerigading Palopo Berbasis Web Menggunakan Metode Extreme Programming” yang terdiri dari metode pengumpulan data dan metode pengembangan sistem.

## **BAB IV                  IMPLEMENTASI**

Pada bab ini, penulis memaparkan hasil penelitian dan penjelasan tahapan Extreme Programming yang penulis gunakan dalam penelitian “Perancangan Aplikasi Inventory Bahan Logistik Rumah Sakit Umum Daerah Sawerigading Palopo Berbasis Web Menggunakan Metode Extreme Programming”.

## **BAB V                    PENUTUP**

Pada bab ini, penulis memaparkan kesimpulan dari isi dan hasil penelitian “Perancangan Aplikasi Inventory Bahan Logistik Rumah Sakit Umum Daerah Sawerigading Palopo Berbasis Web Menggunakan Metode Extreme Programming” serta saran penulis terhadap pengembangan penelitian dan penelitian yang sejenis di masa depan.

## BAB II TINJAUAN PUSTAKA

### 2.1 Studi Literatur

Salah satu data pendukung yang perlu dijadikan sebagai acuan adalah penelitian terdahulu yang relevan dengan permasalahan yang diangkat. Tujuannya adalah sebagai penguat teori dan sebagai acuan ilmiah dalam menyusun kerangka pikir penyelesaian masalah.

1. Penelitian yang dilakukan oleh Widodo, (2022) dengan judul “Sistem Informasi Logistik Menggunakan Metode *Prototype* pada Palang Merah Indonesia Kota Tasikmalaya” menemukan bahwa kegiatan logistik di Palang Merah Indonesia Kota Tasikmalaya dilakukan dengan monitoring berkala terhadap permintaan dan penyampaian kepada kepala markas. Terdapat masalah dalam pengajuan logistik dan pengelolaan data barang yang masih menggunakan media kertas dan Microsoft Excel, yang kurang efektif dan memakan waktu. Tujuan penelitian ini adalah merancang sistem informasi logistik berbasis web menggunakan metode *Prototype*. Hasilnya adalah sistem informasi yang efisien, tepat, dan fleksibel dalam membantu proses pendataan barang logistik.
2. Dan penelitian yang dilakukan oleh Firdaus, (2021) dengan judul “Perancangan Aplikasi Inventory Barang Pada Puskesmas Balida Berbasis Web” menemukan bahwa pengolahan data inventory secara konvensional di Puskesmas Balida menyebabkan sering terjadinya selisih saldo barang dan hambatan dalam pelaporan persediaan barang. Sistem yang terkomputerisasi diharapkan dapat mencatat proses barang masuk secara cepat dan akurat serta meminimalkan kesalahan. Penelitian ini menggunakan *PHP* dengan *framework Laravel* dan *MySQL* sebagai basis datanya. Metode pengembangan yang digunakan adalah *RUP (Rational Unified Process)*, dan hasilnya menunjukkan bahwa aplikasi inventory dapat berfungsi dengan baik dan meningkatkan pengolahan inventory barang..
3. Selanjutnya Penelitian yang dilakukan oleh Limbong & Sianipar, (2022) dengan judul “Sistem Informasi *Inventory* Aset dan Logistik Rumah Sakit Bina Kasih Berbasis Website dengan Menggunakan Metode *Rapid Application Development (RAD)*” mengidentifikasi bahwa Rumah Sakit Bina

Kasih belum memiliki sistem informasi untuk inventaris aset dan logistik. Pengembangan sistem informasi ini menggunakan metode *Rapid Application Development (RAD)* untuk mendukung produktivitas departemen logistik dalam pengadaan, pencatatan, dan pengelolaan laporan inventaris. Hasilnya adalah sistem informasi berbasis web yang efisien dan akurat

4. Dan selanjutnya penelitian yang dilakukan oleh Kesuma et al., (2022) dengan judul “Sistem Informasi *Inventory* Alat Kesehatan Menggunakan Metode *FEFO*” menemukan bahwa PT. Teluk Pursel Sejahtera Purwokerto masih melakukan pencatatan stok secara manual, yang sering menyebabkan kesalahan dan keterlambatan dalam pelaporan. Penelitian ini mengembangkan sistem informasi *inventory* menggunakan metode *FEFO (First Expired First Out)* dengan pendekatan pengembangan perangkat lunak *waterfall*. Hasilnya adalah sistem informasi yang lebih cepat dan akurat dalam pencatatan data dan penyusunan laporan, serta meminimalkan kesalahan.
5. Dan yang terakhir penelitian yang dilakukan oleh Amdi Rizal et al., (2022) dengan judul “Aplikasi *Inventory* Persediaan Barang Berbasis Web Menggunakan Metode *Extreme Programming* (Studi Kasus : Esha 2 Cell)” Esha 2 Cell yang bergerak dalam bidang penjualan *smartphone*, dalam kegiatan bisnisnya, Esha Cell 2 masih melakukan pengolahan data barang secara manual di buku seperti mencatat barang masuk dan keluar, setelah dicatat secara manual di buku, data tersebut di inputkan dalam Microsoft Excel. Karena masih menggunakan pencatat manual dan Microsoft Excel dalam pengolahan data saat barang masuk dan keluar maka sering terjadi kesalahan *input* data yang berulang, dan persediaan barang tidak tercatat secara realtime. Hal ini, membuat informasi tentang stok atau persediaan barang belum efisien. Oleh karena itu solusi terbaik untuk mengatasi kekurangan yang sering terjadi sebelumnya, penulis akan membangun sebuah sistem *inventory* persediaan barang untuk memonitoring proses masuk dan keluarnya stok barang serta persediaan barang di Esha Cell. Perancangan sistem ini akan berbasis *web* dan menggunakan metode *Extreme Programming*. Tujuan penelitian ini adalah merancang sebuah sistem informasi persediaan (stok) barang berbasis web menggunakan metode *Extreme Programming (XP)* untuk mengatasi permasalahan pengelolaan

persediaan barang yang ada ditoko. Metode *Extreme Programming (XP)* dipilih karena pembangunan sistem yang lebih cepat dan sangat fleksibel dengan perubahan yang terjadi pada proses pembangunan perangkat lunak. Pengujian pada sistem aplikasi ini menggunakan metode ISO-25010, dan hasil pengujian sistem dari 3 koresponden berdasarkan kriteria konversi kelayakan sistem terhadap fungsional admin diperoleh nilai 87%.

Tabel 2.1 Class Diagram Inventory

No	Referensi	Metode	Sumber Data	Keterangan	Perbedaan Penelitian
1.	(Widodo, 2022)	Metode <i>Prototype</i>	Palang Merah Indonesia Kota Tasikmalaya	Mengembangkan sistem informasi logistik menggunakan metode <i>prototype</i> . Ini bisa berupa peningkatan efisiensi dalam rantai pasok, pengurangan biaya operasional, peningkatan visibilitas stok, atau peningkatan manajemen risiko	Metode yang digunakan berbeda
2.	(Firdaus, 2021)	<i>RUP (Rational Unified Process)</i>	Puskesmas Balida 2021	Merancang aplikasi untuk manajemen <i>inventory</i> barang di Puskesmas Balida yang berbasis web dengan menggunakan <i>framework Laravel</i> dan	Metode yang digunakan berbeda



				<i>MySQL (My Structured Query Language)</i> sebagai databasenya	
3.	(Limbong & Sianipar, 2022)	Metode <i>Rapid Application Development (RAD)</i>	Rumah Sakit Bina Kasih	Mengembangkan sebuah sistem untuk mengelola <i>Inventory</i> aset dan logistik dengan menggunakan metode <i>RAD</i>	Metode yang digunakan berbeda
4.	(Kesuma et al., 2022)	Metode <i>Waterfall</i>	PT. Teluk Pursel Sejahtera Purwokerto	Menggunakan metode <i>FEFO (First Expired, First Out)</i> dalam mengelola inventaris alat kesehatan. Metode <i>FEFO</i> digunakan untuk mengatur pengeluaran barang berdasarkan tanggal kedaluwarsa, dengan barang-barang yang akan kedaluwarsa lebih awal akan dikeluarkan terlebih dahulu.	Metode yang digunakan dan data yang dikelola berbeda
5.	(Amdi Rizal et al., 2022)	Metode <i>XP (Extreme Programm</i>	Esha 2 Cell Kota Bandar Lampung	Merancang sebuah sistem informasi persediaan (stok) barang berbasis	Data yang dikelola berbeda

		ing)		web menggunakan metode <i>Extreme Programming (XP)</i> untuk mengatasi permasalahan pengelolaan persediaan barang yang ada ditoko.	
--	--	------	--	--	--

## 2.2 Dasar Teori

### 2.2.1 Profil RSUD Sawerigading Palopo

RSUD Sawerigading Kota Palopo sebelumnya adalah Rumah Sakit milik Pemerintah Kabupaten Luwu yang dibangun pada masa pemerintahan Belanda pada tahun 1920. Merupakan salah satu bangunan bersejarah yang ada pada pusat Pemerintahan Kerajaan Luwu, dalam perjalanan telah mengalami dua kali renovasi yakni renovasi pertama dilaksanakan pada tahun 1981 – 1982, di masa pemerintahan Bupati Luwu Drs. Abdullah Suara dan peresmiannya dilakukan oleh Gubernur Sulawesi Selatan Andi Oddang. Renovasi kedua pada tahun 2001-2002 di masa pemerintahan Bupati Dr.H.Kamrul Kasim,SH.MH. Banyak bagian bangunan banyak tidak layak digunakan untuk sebuah Rumah Sakit sehingga memungkinkan sulit untuk dipertahankan keasliannya sebagai bangunan bersejarah.

Rumah Sakit yang sebelumnya memiliki status Rumah Sakit Tipe D, dan tahun 1994 ditingkatkan statusnya menjadi Rumah Sakit kelas C, berdasarkan SK Menteri Kesehatan Republik Indonesia Nomor : 396/ Menkes/KS/IV/1994 (sebagai kantor). Kemudian berdasarkan Peraturan Daerah Kabupaten Luwu Nomor : 9 tahun 2002 RSUD Sawerigading Kota Palopo yang sebelumnya sebagai kantor berubah menjadi Badan Pengelola (Mawardi, 2020).

RSUD Sawerigading Kota Palopo terletak di Jl. Dr. Ratulangi KM.7 Rampoang Palopo, Sulawesi Selatan, Indonesia (Sakit et al., 2018).

### 2.2.2 Pengertian *Inventory*/Persediaan

*Inventory* adalah sebuah konsep yang mencerminkan sumber daya yang tersedia namun belum dimanfaatkan sepenuhnya. Definisi *inventory* dapat

diinterpretasikan dalam berbagai konteks, seperti stok yang tersedia saat itu, daftar rincian barang yang tersedia, atau jumlah barang yang dimiliki oleh suatu organisasi pada suatu waktu dalam konteks keuangan dan akuntansi. Fungsi utama dari *inventory* adalah memenuhi permintaan pelanggan dengan menyediakan persediaan barang seefisien mungkin. *Inventory*, atau sering disebut persediaan, merupakan kumpulan barang mentah, material, atau barang jadi yang disimpan untuk digunakan di masa mendatang atau dalam jangka waktu tertentu. Persediaan barang memegang peranan penting dalam perusahaan untuk menghadapi perubahan dalam pasar produksi dan mengantisipasi fluktuasi harga dalam permintaan barang. Dengan demikian, *inventory* dapat dijelaskan sebagai sumber daya yang berbentuk barang mentah atau barang jadi yang tersedia oleh perusahaan untuk memenuhi kebutuhan konsumen. (Amdi Rizal et al., 2022).

### **2.2.3 Sistem Informasi Berbasis Web**

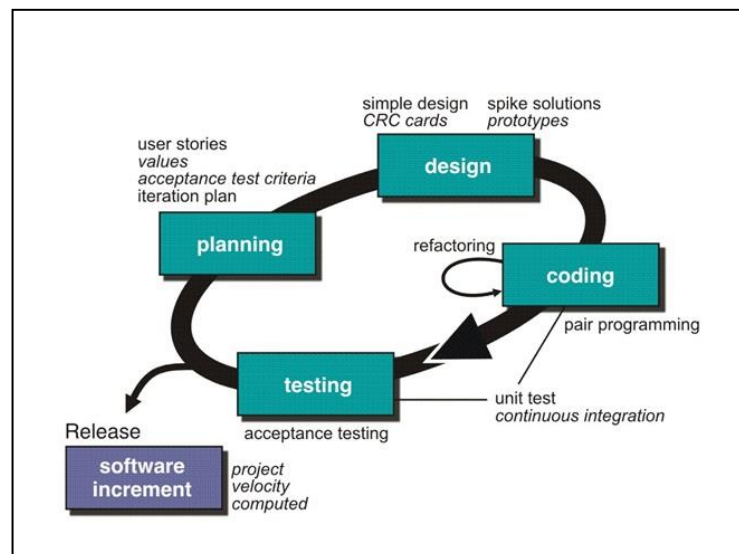
Sistem informasi berbasis web umumnya digunakan sebagai standar untuk penyimpanan data, memperoleh informasi, memformat data, dan menampilkan informasi melalui arsitektur client/server. Sistem ini menangani berbagai jenis informasi digital, termasuk teks, hypermedia, grafis, dan suara. Penggunaannya sangat mudah bagi pengguna karena mengadopsi Antarmuka Pengguna Grafis (GUI).

Sistem informasi berbasis web mirip dengan sistem informasi lainnya, namun berbeda dalam penggunaan bahasa pemrograman web dan sifatnya yang online. Sistem informasi berbasis web berfungsi sebagai media untuk menyajikan informasi melalui berbagai media interaktif seperti gambar, video, audio, atau kombinasi dari semua jenis data tersebut (Prasetio, 2021).

### **2.2.4 Metode *Extreme Programming (XP)***

Penelitian ini memanfaatkan metode pengembangan sistem *Extreme Programming (XP)*. Metode ini sangat cocok untuk pengembangan perangkat lunak karena secara intrinsik dilakukan dengan cepat dan mengadopsi pendekatan berorientasi objek, serta melibatkan tim yang dibentuk dalam skala kecil. Tahapan-tahapan utama dalam metode ini meliputi perencanaan, perancangan, pengkodean, dan pengujian. *Extreme Programming (XP)* adalah metode pengembangan perangkat lunak yang bertujuan untuk meningkatkan efisiensi dan fleksibilitas dalam pengembangan perangkat lunak (Amdi Rizal et al., 2022).

*Extreme Programming (XP)* sangat responsif terhadap perubahan dan menawarkan fase pengembangan cepat yang selaras dengan fokus yang diinginkan. Tahapan pengembangan di *Extreme Programming (XP)* meliputi perencanaan, desain, pengkodean, dan pengujian. Ini berasal dari pengembangan tangkas, suatu teknik untuk memenuhi kebutuhan perangkat lunak atau sistem informasi dengan cepat yang melibatkan pengguna untuk meminimalkan kesalahan pengembangan (Septiani & Habibie, 2022).



Gambar 2.1 Tahapan – tahapan model *Extreme Programming*

Tahapan – tahapan model *Extreme Programming* adalah sebagai berikut :

### 1. *Planning*

Pada tahap perencanaan ini, dilakukan pengumpulan kebutuhan awal dari pengguna, yang dalam *Extreme Programming (XP)* disebut sebagai *user stories*. Hal ini diperlukan agar pengembang memahami konteks bisnis, *output* yang dibutuhkan oleh sistem, dan fitur inti dari perangkat lunak yang sedang dikembangkan. Tahap ini bertujuan untuk menganalisis kebutuhan sistem sehingga dapat memenuhi persyaratan pengguna atau cerita pengguna dengan baik.

### 2. *Design*

Desain dari sistem pada penelitian ini digambarkan dengan model *UML* berupa *use case diagram*, *activity diagram*, dan *relation table*. Pembuatan desain pada *Extreme Programming* tetap mengedepankan prinsip *Keep it Simple (KIS)*. Desain disini merupakan representasi dari sistem guna

mempermudah pengembang dalam membangun sistem. Desain ini dimaksudkan untuk mempermudah pengembangan sistem nantinya.

### 3. Coding

Pengkodean adalah fase dimana gagasan yang direncanakan dan dirancang diwujudkan ke dalam bahasa yang dapat diproses oleh mesin. Proses ini memanfaatkan pendekatan pemrograman berbasis objek dan sejumlah bahasa pemrograman web seperti *HTML*, *CSS*, *PHP*, dan *Javascript*. Selain itu, pengkodean juga melibatkan penggunaan teknologi database seperti *MySQL*, framework seperti *Laravel*, *Codeigniter*, *Firebase Realtime Database*, dan lingkungan pengembangan *Android Studio*.

### 4. Testing

Tahap ini akan menggunakan unit test yang sebelumnya telah dibuat. Karena pembuatan dari unit test adalah pendekatan utama dari *Extreme Programming (XP)*. Dalam melakukan pengujian, penulis menggunakan 2 teknik pengujian yaitu pengujian *white box*. Pada tahap pengujian *black box*, dilakukan pengujian setiap unit test, maksudnya melakukan pengujian integrasi antara *input* dan hasil *output* yang sesuai semestinya terjadi.

#### 2.2.5 Framework Codeigniter

*CodeIgniter* adalah sebuah web application network yang bersifat open source yang digunakan untuk membangun aplikasi php dinamis.

*CodeIgniter* menjadi sebuah framework PHP dengan model *MVC (Model, View, Controller)* untuk membangun website dinamis dengan menggunakan PHP yang dapat mempercepat pengembang untuk membuat sebuah aplikasi web. Selain ringan dan cepat, *CodeIgniter* juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kodenya. Dokumentasi yang lengkap inilah yang menjadi salah satu alasan kuat mengapa banyak orang memilih *CodeIgniter* sebagai framework pilihannya. Karena kelebihan-kelebihan yang dimiliki oleh *CodeIgniter*, pembuat PHP Rasmus Lerdorf memuji *CodeIgniter* di frOSCon (Agustus 2008) dengan mengatakan bahwa dia menyukai *CodeIgniter* karena “it is faster, lighter and the least like a framework.”

*CodeIgniter* pertamakali dikembangkan pada tahun 2006 oleh Rick Ellis. Dengan logo api yang menyala, *CodeIgniter* dengan cepat “membakar” semangat

para web developer untuk mengembangkan web dinamis dengan cepat dan mudah menggunakan framework PHP yang satu ini (PT Cloud Hosting, 2019).

Salah satu kerangka kerja yang digunakan dalam penulisan tugas akhir ini adalah *Codeigniter*. Penulis memilih menggunakan kerangka kerja *Codeigniter* karena memungkinkan pengembangan program tanpa harus membuat kode dari awal, sehingga proses kerjanya menjadi lebih cepat. *Codeigniter* merupakan kerangka kerja yang dibangun menggunakan bahasa pemrograman *PHP* (*Hypertext Preprocessor*) dengan tujuan mempermudah para pengembang web dalam membuat atau mengembangkan aplikasi berbasis web. *Codeigniter* dikenal memiliki kinerja eksekusi yang cepat dibandingkan dengan kerangka kerja lainnya. Selain itu, *Codeigniter* bersifat *open source* dan mengadopsi model basis *MVC* (*Model View Controller*), yang merupakan konsep model yang modern saat ini.

Menurut Yesputra, Rolly, Marpaung Nasrun (2018), metode *MVC* (*Model View Controller*) terdiri dari tiga komponen, yaitu:

1. Model, yang bertanggung jawab untuk mengelola basis data seperti *MySQL* (*My Structured Query Language*) atau *Oracle RDMS* (*Relational Database Management System*). Model ini berinteraksi dengan database dan biasanya berisi kelas atau fungsi untuk melakukan operasi seperti membuat, memperbarui, menghapus, mencari, dan mengambil data dari database. Model juga berhubungan dengan perintah-perintah query sebagai tindak lanjut dari operasi-operasi tersebut.
2. View, merupakan antarmuka pengguna atau tampilan yang ditujukan untuk pengguna akhir. View dapat berupa halaman *HTML*, *CSS*, *JavaScript*, *jQuery*, dan *AJAX*. Karena menggunakan metode *MVC*, view tidak boleh melakukan pemrosesan data atau akses ke database.

Tugas view adalah menampilkan data hasil dari Model dan *Controller*. *Controller*, bertindak sebagai penghubung antara view dan model. *Controller* digunakan sebagai jembatan antara keduanya karena model tidak dapat berinteraksi langsung dengan view dan sebaliknya. Tugas *controller* meliputi pemrosesan data, penyediaan variabel yang akan ditampilkan di view, pemanggilan model untuk mengakses database, penanganan kesalahan validasi, dan pengecekan terhadap *input* data (Ridwan et al., 2022).

Dari pengertian di atas, dapat disimpulkan bahwa *Codeigniter* adalah sebuah kerangka kerja *PHP (Hypertext Preprocessor)* yang menyediakan fitur lengkap untuk pengembangan aplikasi web.

### **2.2.6 Perancangan Sistem**

Perancangan Sistem dilakukan untuk agar kebutuhan pengguna dapat terakomodir dengan baik pada saat proses implementasi sistem. Perancangan sistem dimulai dengan membuat pemodelan menggunakan user case diagram untuk mendeskripsikan interaksi antar aktor dengan fungsi-fungsi atau fitur yang akan berjalan dalam sistem nantinya (Malik, Hasanuddin, et al., 2023).

Perancangan merupakan proses penentuan prosedur dan data yang dibutuhkan oleh sistem baru. Tahap perancangan sistem memberikan gambaran rancangan yang lengkap sebagai panduan bagi pengembang aplikasi. Sesuai dengan komponen sistem yang akan dikomputerisasi, dalam tahap ini, perlu didesain hardware atau software, database, dan aplikasi. Menurut Sommerville dalam buku Agus Mulyanto (2009: 259), proses perancangan dapat melibatkan pengembangan beberapa model sistem pada tingkat abstraksi yang berbeda.

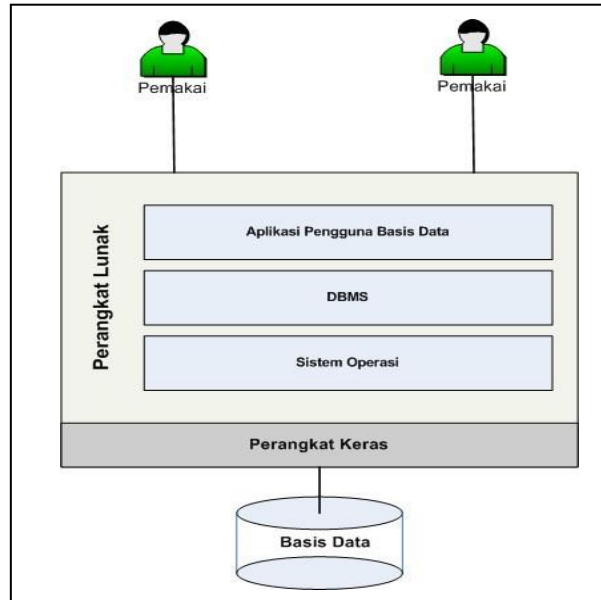
Perancangan adalah proses untuk menentukan apa yang akan dikerjakan dengan menggunakan berbagai teknik, yang melibatkan deskripsi mengenai arsitektur dan detail komponen, serta keterbatasan yang akan ditemui dalam proses pengerjaannya (Hanadhito Riswantoro, 2019).

Berdasarkan beberapa pendapat tersebut, dapat disimpulkan bahwa perancangan merupakan tahapan setelah analisis sistem yang bertujuan untuk menghasilkan rancangan yang memenuhi kebutuhan yang telah ditetapkan selama tahap analisis.

#### **1. Database**

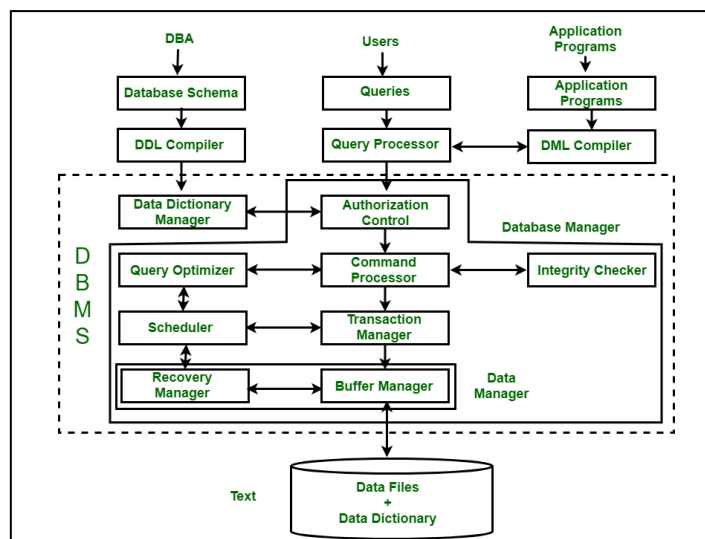
Database adalah kumpulan data atau basis data yang dikelola dengan cara tertentu dan terkait satu sama lain agar dapat dikelola dengan mudah. Manajemen database memfasilitasi pencarian, penyimpanan, dan penghapusan informasi bagi pengguna.

Database juga bisa diartikan sebagai sebuah sistem yang berfungsi mengumpulkan data, arsip, atau tabel yang disimpan dan terhubung ke media elektronik, seperti aplikasi atau situs web. Database membuat penyimpanan dan pengelolaan data lebih efisien (Yudhistira, 2024)



Gambar 2.2 Sistem Manajemen *Database*

Pengertian lain tentang database menurut Oracle adalah sekumpulan informasi terstruktur yang tersusun rapi atau data yang tersimpan secara elektronik dalam sebuah sistem komputer. Biasanya, database dikelola oleh Sistem Manajemen Database (DBMS). Data, bersama dengan DBMS dan aplikasi terkaitnya, sering disebut sebagai sistem database. Basis data merujuk pada penyimpanan informasi secara terstruktur di dalam komputer, yang dapat diakses melalui program komputer untuk mengambil informasi yang diperlukan. Perangkat lunak yang bertanggung jawab atas manajemen dan eksekusi query basis data disebut Sistem Manajemen Basis Data (Database Management System, DBMS).



Gambar 2.3 Sistem Manajemen *Basis Data*



## 2. PHP (*Hypertext Preprocessor*)

*Hypertext Preprocessor*, merupakan bahasa scripting yang dirancang khusus untuk pengembangan web. Sebagai bahasa scripting server-side, *PHP* memerlukan web *server* untuk mengeksekusi kode-kodenya. *PHP* dapat diintegrasikan dengan *HTML*, *JavaScript*, *jQuery*, dan *Ajax*. (Riski Qisthiano, 2023).

*PHP* adalah *script* (bahasa pemrograman) yang berfungsi untuk merubah website statis menjadi lebih dinamis dan interaktif untuk *user* (pengguna website). Sebenarnya, penggunaan *HTML* dan *CSS* sudah cukup dalam pembuatan *website*.

Namun, yang dihasilkan adalah *website* statis, dimana konten website tersebut tidak berubah-ubah. Sekali dibuat dan online di internet, proses perubahan konten harus dilakukan secara manual dengan mengubah kode bahasa pemrogramannya. Penggunaan *PHP* membuat konten website menjadi lebih dinamis; konten *website* dapat diubah kapanpun sesuai kebutuhan melalui *CMS* (*Content Management System*).

Lebih lanjut, website dapat bersifat interaktif dengan menggunakan *PHP* karena mengimplementasikan fungsi *CRUD* (*Create, Read, Update and Delete*).

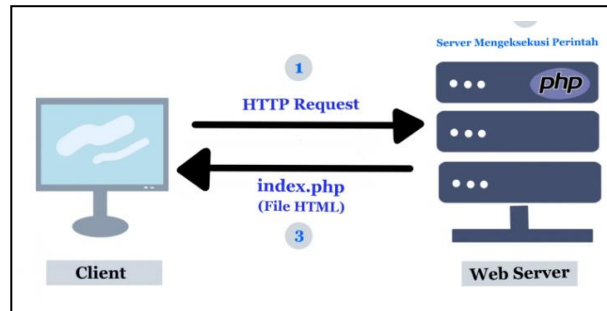
Pertama, *Create* merupakan operasi untuk membuat dan memasukkan data baru ke dalam *database website*. Contohnya adalah saat Anda melakukan *sign up* (registrasi baru) pada sebuah website.

Kemudian, *Read* merupakan operasi membaca dan mengambil data dari *database*. Contohnya adalah saat login (proses masuk) pada sebuah website. Website akan menjalankan operasi “*read*” dari dalam database untuk verifikasi atas akun yang akan Anda gunakan. Sebelum pada akhirnya menampilkan halaman milik anda di layar komputer.

Selanjutnya adalah *Update*, dimana Anda dapat melakukan proses edit data yang ada dalam *database*. Contohnya adalah saat Anda mengganti bio

profil dalam sosial media. *Website* akan menjalankan operasi pembaharuan ke dalam database. Respon yang didapatkan dari database adalah perubahan bio profil seperti yang telah Anda tambahkan.

Terakhir adalah *Delete*, dimana *website* akan menjalankan operasi penghapusan data yang ada dalam database. Contohnya saat Anda menghapus salah satu foto di media sosial Anda.



Gambar 2. 4 Cara Kerja PHP

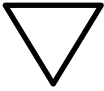

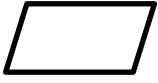

### 3. Flowchart

*Flowchart* (Diagram Alir) merupakan bagan yang mengarahkan alir di dalam prosedur sistem secara logika. *Flowchart* merupakan cara dalam menjelaskan tahap-tahap pemecahan masalah dengan merepresentasikan simbol-simbol tertentu yang mudah digunakan dan dipahami (Indra Sukma, 2022).

Simbol-simbol yang ada didalam *flowchart* dapat dilihat pada tabel berikut :

Tabel 2.2 Simbol-Simbol *Flowchart*

No.	Simbol	Nama	Fungsi
1.		Proses	Digunakan untuk menunjukkan pengolahan yang dilakukan komputer
2.		<i>Manual Operation</i>	Digunakan untuk menunjukkan pengolahan yang tidak dilakukan komputer
3.		<i>Decision</i>	Digunakan untuk memilih proses berdasarkan kondisi
5.		Terminal	Digunakan untuk memulai atau mengakhiri proses.

6.		<i>Offline Storage</i>	Digunakan untuk memberitahukan bahwa data akan disimpan ke suatu media tertentu
7.		Manual Input Simbol	Digunakan untuk input data secara manual dengan keyboard
8.		<i>Input/Output</i>	Digunakan untuk Input dan Output.
9.		Dokumen	Digunakan untuk data masukan dan keluaran dari dokumen

#### 4. Konsep *Unified Modelling Language (UML)*

*UML* merupakan suatu bahasa berbasis grafik yang digunakan untuk memvisualisasikan, menspesifikasikan, membangun, dan mendokumentasikan sistem pengembangan perangkat melalui gambaran grafis. *UML* adalah salah satu alat yang berguna dalam pemrograman berbasis objek, dan saat ini, *UML* dianggap akan menjadi standar bagi industri pengembangan sistem atau perangkat lunak yang mengadopsi pendekatan berorientasi objek (Harike, 2019).


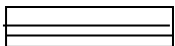

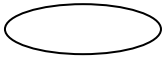
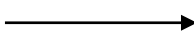
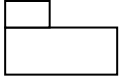
Ada beberapa macam diagram dalam *Unified Modeling Language (UML)*, yaitu:

##### a. *Use Case Diagram*

Diagram *Use Case* merupakan representasi abstrak dari interaksi antara sistem dan aktor. Diagram ini digunakan untuk mendeskripsikan bagaimana pengguna atau aktor berinteraksi dengan sistem melalui skenario penggunaan. *Diagram Use Case* sangat penting dalam menyusun kebutuhan sistem, berkomunikasi dengan klien, dan merancang kasus uji untuk fitur-fitur sistem. *Use case* adalah cara untuk menggambarkan bagaimana sistem akan dilihat oleh pengguna, sementara diagram use case membantu dalam komunikasi antara analis, pengguna, dan klien (Malik, Nirsal, et al., 2023).

Tabel 2.3 Simbol *Use Case Diagram*


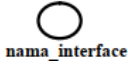
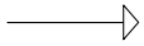




No	Simbol	Nama	Fungsi
----	--------	------	--------

1		<i>Actor</i>	Segala sesuatu yang berinteraksi dengan sistem aplikasi komputer.
2		<i>Class</i>	Pembentukan utama dari sistem berorientasi yang memiliki atribut dan operasional yang sama.
3		<i>Interface</i>	Kumpulan operasi tanpa implementasi dari suatu kelas.
4		<i>Use Case</i>	Menjelaskan yang dilakukan aktor dari sistem untuk mencapai tujuan tertentu.
5		<i>Interaction</i>	untuk menunjukkan baik alir pesan atau informasi antara objek maupun hubungan antar objek.
6		<i>Package</i>	Komentar atau wadah konseptual yang digunakan untuk mengelompokkan elemen-elemen dari sistem yang sedang dibangun, sehingga bisa dibuat model yang lebih sederhana.

b. *Class Diagram*

*Class* adalah dekripsi kelompok obyek-obyek dengan *property*, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya class diagram dapat memberikan pandangan global atas sebuah sistem. Hal tersebut tercermin dari class-class yang ada dan relasinya satu dengan yang lainnya. Didalam sebuah sistem biasanya memiliki beberapa class diagram. *Class diagram* sangat membantu dalam visualisasi struktur kelas dari suatu system (Harike, 2019).

Tabel 2.4 Simbol-Simbol *Class Diagram* (Indra Sukma, 2022)

No.	Simbol	Nama	Fungsi
1.		<i>Class</i>	Kelas pada struktur sistem
2.		Antarmuka / <i>Interface</i>	Sama dengan konsep interface dalam pemrograman berbasis objek.
3.		Generalisasi	Relasi antar kelas dengan makna generalisasi (umum khusus)
4.		Asosiasi Berarah	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain
5.		Asosiasi	Hubungan antar kelas dengan makna umum
6.		Kebergantungan	Hubungan antar kelas dengan makna kebergantungan antar kelas.
7.		Agregasi	Hubungan antar kelas dengan makna semua bagian.

c. *Component Software*

*Component software* merupakan bagian fisik dari sebuah sistem, karena menetap di komputer tidak berada di benak para analis. Komponen merupakan bagian implementasi software dari sebuah atau lebih class. Komponen dapat berupa *source code*, komponen biner, atau *executable component*. Sebuah komponen berisi informasi tentang *logic class* atau *class* yang diimplementasikan sehingga membuat pemetaan dari sebuah *logical view* ke komponen view. Sehingga komponen diagram dapat merepresentasikan dunia riil yaitu sebuah *component software* yang mengandung *component*, *interface* dan *relationship*.

d. *Deployment Diagram*

Menggambarkan tata letak sebuah system secara fisik, menampakkan bagian-bagian software yang berjalan pada bagian-bagian hardware, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Didalam sebuah *nodes*, *executeable component* dan *object* yang dialokasikan untuk dapat memperlihatkan sebuah unit perangkat lunak yang telah dieksekusi oleh sebuah node tertentu dan ketergantungan komponen.

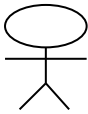
e. *State Diagram*


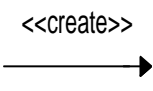
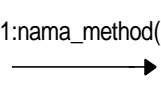
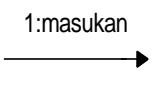
Menggambarkan semua *state* (kondisi). yang dimiliki oleh suatu object dari sebuah class dan keadaan yang menyebabkan state berubah. Kejadian ini dapat berupa *object* lain yang mengirim pesan. *State class* tidak digambarkan untuk semua class, hanya yang mempunyai sejumlah state yang terdefinisi dengan baik dan kondisi class berubah oleh state yang berbeda.

f. *Sequence Diagram*

*Sequence Diagram* digunakan untuk menggambarkan perilaku pada sebuah scenario. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antara object, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. *Sequence diagram* merupakan gambaran secara grafis sebuah skenario yang menunjukkan interaksi antar objek dalam sebuah urutan waktu, apa yang terjadi pertama kali dan apa yang terjadi berikutnya. *Sequence diagram* menggambarkan interaksi antar objek di dalam dan sekitar sistem (termasuk pengguna, tampilan, dan sebagainya), berupa pesan yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Tabel 2.5 Simbol *Sequncial Diagram*

No	Simbol	Nama	Fungsi
1		<i>Actor</i>	Segala sesuatu yang berinteraksi dengan sistem aplikasi komputer.

2		<i>Garis hidup</i>	Menyatakan kehidupan suatu objek
3		<i>Pesantipe create</i>	Objek yang lain, arah panah mengarah pada objek
4		<i>Pesan tipe call</i>	Menyatakan suatu objek memanggil operasi yang ada pada objek lain.
5		<i>Pesan tipe send</i>	Menyatakan bahwa suatu objek mengirimkan data/masukan ke objek lain.


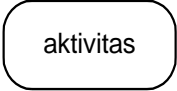
g. *Collaboration Diagram*



Menggambarkan kolaborasi dinamis seperti *sequence diagram*. Dalam menunjukkan proses pertukaran sebuah pesan, *collaboration diagram* menggambarkan *object* dan hubungannya (mengacu ke konteks). Jika penekannya pada waktu atau urutan gunakan *sequence diagram*, tapi jika penekanannya pada konteks gunakan *collaboration diagram*.

h. *Activity Diagram*

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi. *Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem (Sterpi & Validation, 2018).

Tabel 2.6 Simbol *Activity Diagram*

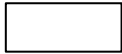
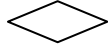
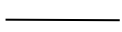

No	Simbol	Nama	Fungsi
1		Status Awal	Status awal aktifitas sistem sebuah diagram aktivitas memiliki sebuah status awal
2		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.

3		Percabangan	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4		Status akhir	Status akhir yang dilakukan sistem

i. *Entity Relationship Diagram*

*Entity Relationship Diagram* atau *ERD* adalah alat perancangan untuk memodelkan suatu basis data. Tujuan dibuatnya *Entity Relationship Diagram* adalah untuk pemodelan yang menunjukkan hubungan antar data yang mempunyai relasi, juga mendokumentasikan data yang ada dengan cara menjelaskan tiap masing-masing data dan relasinya. Simbol *ERD* ditunjukkan pada Tabel 2.7 berikut ini (Indra Sukma, 2022).

Tabel 2.7 Simbol-Simbol *Entity Relationship Diagram*

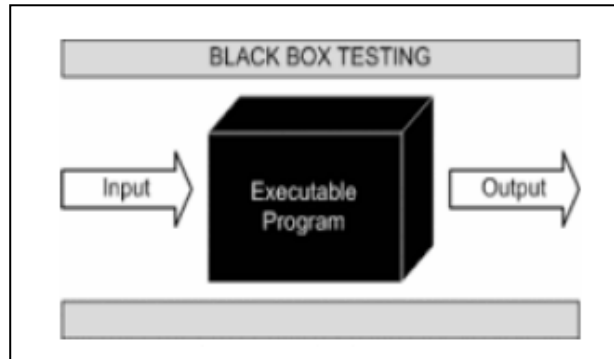
No	Simbol	Nama	Keterangan
1		<i>Entitas</i>	Suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
2		Relasi	Menunjukkan adanya hubungan diantara sejumlah entitas yang berbeda.
3		<i>Association</i>	Penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.
4		<i>Atribut</i>	Mendesripsikan karakter entitas (atribut yang berfungsi sebagai key yang diberi garis bawah).

*ng*

*Black box testing* adalah suatu teknik pengujian perangkat lunak di mana fokus utamanya adalah pada perilaku eksternal dari perangkat lunak tersebut, tanpa memerhatikan struktur internal atau detail implementasinya. Proses *black box* melibatkan pengujian program dengan memberikan berbagai



macam *input* pada antar muka pengguna atau *API* yang telah disediakan, tanpa memperhatikan detail bagaimana program tersebut diimplementasikan di dalamnya. (Simpony et al., 2022).



Gambar 2. 5 Black Box Testing

*Black box testing* adalah pengujian pada suatu aplikasi didasarkan pada detail aplikasi seperti fungsi-fungsi yang ada pada aplikasi, tampilan aplikasi, dan kesesuaian alur fungsi. Maksud dari *black box* ini yaitu berupa sistem atau perangkat lunak yang akan diuji. Seperti *iOS, Linux, Windows, Android*, atau situs *web* seperti *Yahoo, Google, Bing* dan lainnya. Tujuan utama dari pengujian ini adalah untuk memastikan bahwa program berfungsi sesuai dengan kebutuhan yang ditetapkan oleh perusahaan, tanpa harus memperhatikan bagaimana program tersebut dibuat atau diimplementasikan secara internal.

Pada suatu *black box testing* terdapat tiga jenis pengujian, berikut akan kami jelaskan :

a. *Functional testing*

Merupakan suatu proses pengujian pada fitur spesifik atau fungsi pada suatu software. Pengujian ini bisa dilakukan guna memastikan bahwa user tidak dapat masuk pada aplikasi yang diuji tanpa informasi sehingga program tersebut terjaga keamanannya. Umumnya functional testing hanya fokus pada pengujian aspek-aspek terpenting pada sebuah integrasi beberapa komponen utama dan *software*. Namun nyatanya black box ini dapat digunakan untuk menguji sistem secara keseluruhan.

b. *Non functional testing*

Memiliki fungsi yang tidak kalah dari *functional testing*, di mana *Non functional testing* merupakan suatu pengujian pada beberapa aspek

tambahan non fungsional. Bila *functional testing* menguji *software* menjalankan suatu tugas atau peringkat.

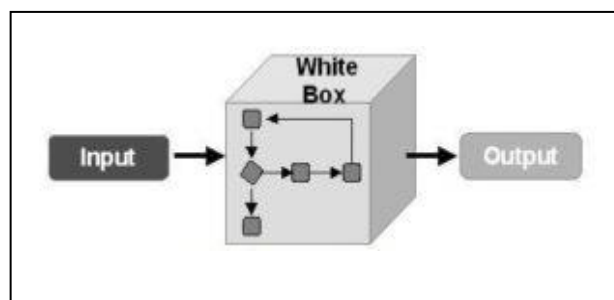
Maka *non functional testing* dilakukan untuk tahu bagaimana *software* tersebut melakukannya. Selain itu *non functional testing* memiliki fungsi untuk melihat apakah suatu *software* yang sudah didesain bisa dioperasikan secara optimal di perangkat dengan sistem operasi serta ukuran layar yang berbeda-beda.

c. *Regression testing*

Jenis ini merupakan pengujian yang dilakukan pada *software* untuk mengamati apakah terjadinya kemunduran atau regresi ketika *software* diperbaharui. Dalam *regression test* dilakukan pengecekan aspek *non fungsional* maupun *fungsional*.

## 6. *White Box Testing*

*White box testing* adalah metode pengujian yang berfokus pada analisis kode program dan mengharuskan penguji memiliki pengetahuan tentang struktur kode serta kemampuan untuk menulis kasus uji yang sesuai. Metode ini mencakup pengujian data *flow*, *control flow*, *basis path*, dan *loop*. *White box testing* juga dikenal dengan sebutan *clear box*, *glass box*, atau *open box*. Dalam penelitian ini, metode *basis path* dipilih untuk *white box testing* karena dapat menghasilkan *test case* dengan cakupan pengujian yang lebih menyeluruh dari pada teknik lainnya.



Gambar 2.6 *White Box Testing*

Teknik *basis path* melibatkan *flowgraph notation*, *Cyclomatic Complexity*, dan *independent path*. Ini adalah pendekatan yang fokus pada pengujian jalur eksekusi program. Untuk memastikan setiap bagian perangkat lunak diuji secara menyeluruh, penguji dapat mengembangkan skenario pengujian dengan memahami jalur-jalur tersebut (Y J Solissa et al., 2023).

Langkah-langkah untuk menerapkan pengujian *white box* dengan basis *path* meliputi :

1. Menggambarkan alur proses program,
2. Menentukan *Cyclomatic Complexity*,
3. Menentukan jalur dasar sesuai dengan *Complexity*, dan
4. Mendefinisikan kasus uji untuk setiap jalur dasar yang ditentukan.

*Cyclomatic Complexity* adalah sebuah *metric* perangkat lunak yang memberikan ukuran kuantitatif terhadap tingkat kompleksitas logis suatu program. *Metric* ini membantu penguji untuk menilai seberapa rumit atau kompleks logika yang terdapat dalam suatu program. Semakin tinggi nilai *Cyclomatic Complexity*, semakin kompleks dan sulit untuk melakukan pengujian terhadap kode tersebut. *Cyclomatic Complexity* juga dapat menentukan jumlah minimum *test case* yang diperlukan untuk menguji jalur-jalur independen dalam program. Dalam perhitungan *Cyclomatic Complexity*, terdapat tiga elemen utama : *node*, yang merupakan lingkaran yang mewakili statement dalam *flowgraph* secara berurutan; *edge*, yang merupakan panah yang menghubungkan node dan menggambarkan aliran kontrol program; dan *region*, yang merupakan area yang dibatasi oleh *edge* dan *node*

Metode pengujian pada *white box testing* ini seringkali dilakukan untuk :

1. Memberikan dan membuat suatu jaminan bahwa seluruh jalur-jalur yang independen hanya menggunakan modul minimal satu kali.
2. Keputusan yang sifatnya logis dapat digunakan di semua kondisi *true* (benar) atau *false* (salah).
3. Mengeksekusi seluruh perulangan yang ada ke pada batas nilai dan operasional di setiap situasi dan kondisi.
4. Syarat yang dilakukan dalam menjalankan strategi *white box testing*
5. Mendefinisikan tentang seluruh alur-alur logika yang ada.

Tabel 2.8 Tabel Hubungan *Cyclomatic Complexity* Dengan Resiko

Nilai CC	Tipe Prosedur	Tingkat Resiko
1-4	Prosedur sederhana	Rendah
5-10	Prosedur yang terstruktur dengan baik dan stabil	Rendah
11-20	Prosedur yang lebih kompleks	Menengah

21-50	Prosedur yang kompleks dan kritis	Tinggi
>50	Rentan kesalahan, sangat mengganggu dan prosedur tidak dapat diuji	Sangat tinggi

Pengguna metode *white box* dalam pengujian perangkat lunak harus memiliki pemahaman yang mendalam tentang kode sumber perangkat lunak. Dalam konteks penelitian ini, metode yang akan digunakan adalah *white box testing*. Teknik-teknik khusus yang digunakan dalam *white box testing* mencakup:

1. *Loop testing*, yang mengevaluasi validitas struktur perulangan dalam program.
2. *Data flow testing*, yang memeriksa pergerakan data dalam program.
3. *Control flow testing*, yang menggunakan aliran kontrol program sebagai model untuk menghasilkan kasus uji.
4. *Branch testing*, yang berfokus pada pengujian percabangan dalam program.
5. *Basis path testing*, yang menguji setiap pernyataan dalam kode setidaknya sekali untuk memastikan ketercukupan pengujian yang komprehensif (Sie et al., 2022).

### 2.2.7 Bahan Logistik Non Medis

Secara umum, logistik merujuk pada ilmu pengetahuan dan proses yang terkait dengan perencanaan, pengadaan, penyimpanan, distribusi, dan pemeliharaan material atau barang. Di konteks rumah sakit, logistik mencakup semua bahan yang diperlukan untuk operasional sehari-hari, seperti persediaan untuk perekam medik, dapur, farmasi, laboratorium, air, peralatan kantor, kebutuhan rumah tangga (seperti listrik, sabun, tisu, sapu, dan karbol), layanan laundry, serta persediaan makanan (Mokalu et al., 2019).

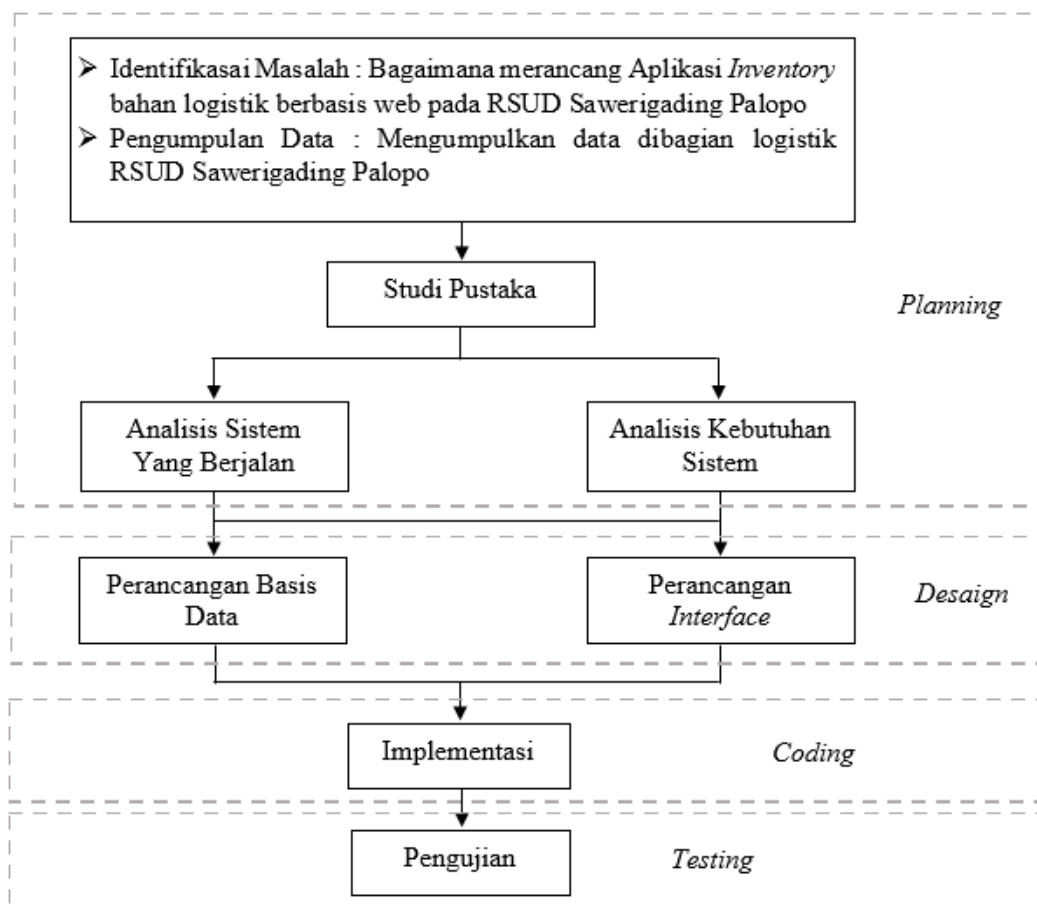
Di lingkungan rumah sakit, manajemen logistik dapat dijelaskan sebagai proses strategis dalam pengelolaan pengadaan, penyimpanan, distribusi, dan pemantauan persediaan bahan yang diperlukan untuk produksi layanan kesehatan. Ini merupakan bagian penting dalam mendukung pelayanan medis di rumah sakit, terutama dalam aspek administrasi dan operasional yang berkaitan dengan kepentingan pasien dan staf rumah sakit. Di lingkungan rumah sakit, manajemen logistik menjadi kunci dalam mengelola persediaan bahan yang diperlukan untuk

produksi layanan kesehatan, sehingga mendukung pelayanan medis secara efisien dan efektif. (Junus & Ambo, 2020).

### 2.2.8 Kerangka Berfikir Penelitian

Tahapan yang peneliti gunakan untuk merancang sistem *inventory* bahan logistik pada RSUD Daerah Sawerigading Palopo adalah pengumpulan data-data yang diperlukan dan pengembangan sistem *inventory* itu sendiri.

Dalam mengumpulkan data digunakan metode-metode berupa studi pustaka dan studi lapangan. Dalam penelitian ini juga menggunakan studi literatur untuk mempelajari bagaimana cara merancang aplikasi *inventory*. Untuk pengembangan penelitian digunakan metode pengembangan sistem *Extreme Programming (XP)* dengan pendekatan berorientasi objek dan bahasa pemodelan yang digunakan adalah *Unified Modeling Language (UML)*. Berikut ini gambaran dari kerangka penelitian yang dilakukan.



Gambar 2.7 Kerangka Penelitian

**BAB III**  
**ANALISIS DAN PERANCANGAN SISTEM**

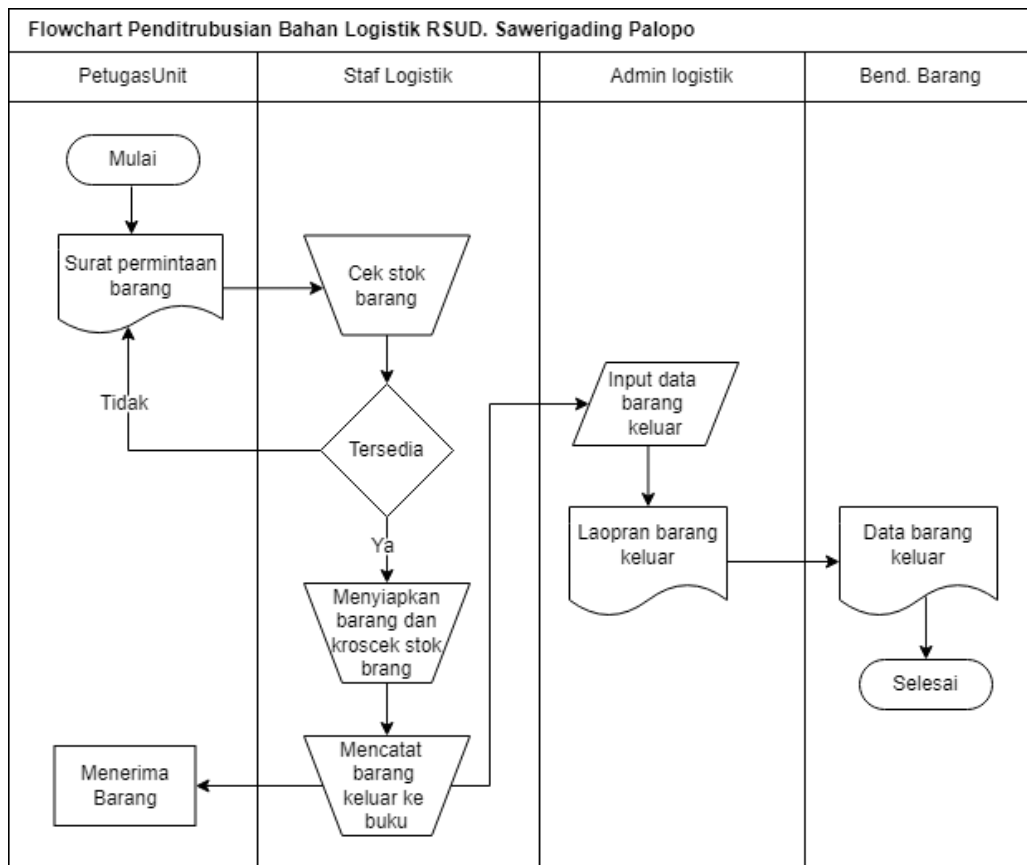
**3.1 Analisis Sistem**

**3.1.1 Analisis yang sedang berjalan**

Setelah dilakukan pengamatan dan analisis sistem yang berjalan pada sistem inventory di Rumah Sakit Umum Daerah Sawerigading Palopo diketahui bahwa sistem pencatatan barang masuk, barang keluar dan laporan persediaan barang masih dicatat di media pembukuan dan diinput ke program Microsoft Excel. Analisis prosedur yang berjalan bertujuan untuk mengetahui lebih jelas aktivitas-aktivitas yang berlangsung saat admin logistik dalam mengelola dan melaporkan data barang masuk, barang keluar, dan menyajikan laporan persediaan barang. Berikut ini adalah prosedur yang berjalan saat ini di RSUD. Sawerigading Palopo yang digambarkan dengan menggunakan pemodelan Diagram Alir (*flowchart*).

1. Flowchart Sistem Proses Barang Masuk

Adapun flowchart sistem proses barang masuk yang sedang berjalan di Rumah Sakit Umum Daerah Sawerigading Palopo dapat digambarkan 3.1.

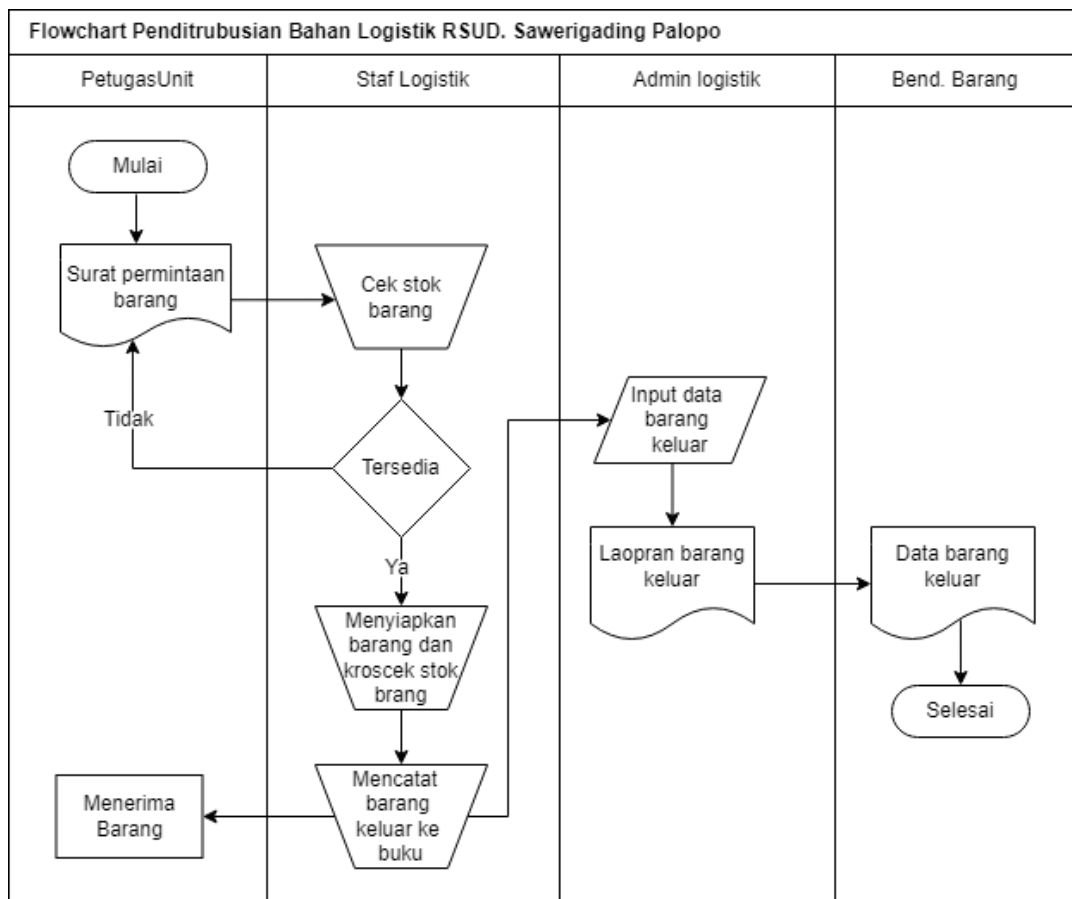


Gambar 3.1 *Flowchart* Sistem Barang Masuk

Proses pendataan barang masuk masih dilakukan secara manual karena pencatatan data di staf logistik saat barang diterima dari supplier hanya dicatat di pembukuan dan surat pengantar barang yang sudah di cek dengan barang diberikan ke admin logistik untuk diinput kedalam program Microsoft Excel dan diarsipkan pada map ordner. Setelah itu admin logistik akan mencetak laporan barang masuk untuk diserahkan ke Kabag. Logistik.

## 2. Flowchart Pendistribusian Bahan Logistik

Adapun *flowchart* sistem proses pendistribusian barang (bahan logistik) yang sedang berjalan di Rumah Sakit Umum Daerah Sawerigading Palopo dapat digambarkan sebagai berikut :



Gambar 3.2 *Flowchart* Sistem Pendistribusian Bahan Logistik

unit meminta barang, staf logistik harus mengecek stok barang secara langsung ke gudang dan jika barang ada maka staf logistik akan mengeluarkan barang dan mencatat pendistribusian barang dipembukuan. Setelah itu meninformasikan ke admin logistik untuk menginput barang yang keluar kedalam program Microsoft Excel dan admin logistik akan mencetak laporan barang keluar untuk diserahkan ke Kabag. Logistik.



### 3.1.2 Kebutuhan Sistem Informasi Yang Diusulkan

Berdasarkan hasil riset di Rumah Sakit Umum Daerah Sawerigading Palopo, terdapat kebutuhan di bagian logistik yang memerlukan aplikasi inventaris berbasis web untuk menangani berbagai permasalahan yang ada. Aplikasi berbasis web ini adalah sebuah *platform* yang menyimpan dan menampilkan dokumen multimedia (seperti teks, gambar, suara, animasi, dan video) menggunakan protokol *HTTP (Hypertext Transfer Protocol)* dan dapat diakses melalui perangkat lunak yang disebut browser.

Website dikelola oleh tiga user yaitu petugas unit dan admin bidang logistik dan bendahara barang :

#### 1. Kebutuhan Pengguna

Menjelaskan mengenai kebutuhan pengguna yang terdapat di website. Berikut analisis kebutuhan bagi pengguna :

##### a. Kebutuhan Petugas Unit

- 1) Dapat melakukan registrasi dan login sebagai petugas unit.
- 2) Dapat menginput data petugas unit.
- 3) Dapat melihat profile petugas unit pada halaman registrasi.
- 4) Dapat melakukan penginputan surat pengajuan bahan logistik.
- 5) Dapat mencetak surat pengajuan logistik.
- 6) Petugas unit dapat menerima informasi diterima / ditolak surat pengajuan barang logistik yang diajukan pada status pengajuan logistik.
- 7) Petugas unit dapat melihat status pengajuan.
- 8) Dapat melakukan logout.

##### b. Kebutuhan Admin Logistik

- 1) Dapat melakukan login.
- 2) Dapat mengelola data user.
- 3) Dapat melihat profile admin bidang logistik.
- 4) Dapat melihat dan mengelola surat pengajuan logistik.
- 5) Dapat mencetak surat pengajuan logistik.
- 6) Dapat mengelola data barang.
- 7) Dapat melihat stok barang yang tersedia.
- 8) Dapat mengubah, menghapus dan menambah barang.

- 9) Dapat mencetak laporan barang masuk, persediaan barang dan barang keluar.
  - 10) Dapat melakukan logout.
- c. Kebutuhan Bendahara Barang
- 1) Dapat melakukan login.
  - 2) Dapat melihat profile bendahara barang.
  - 3) Dapat Mencetak barang masuk, persediaan barang dan barang keluar.
  - 4) Dapat melakukan logout.

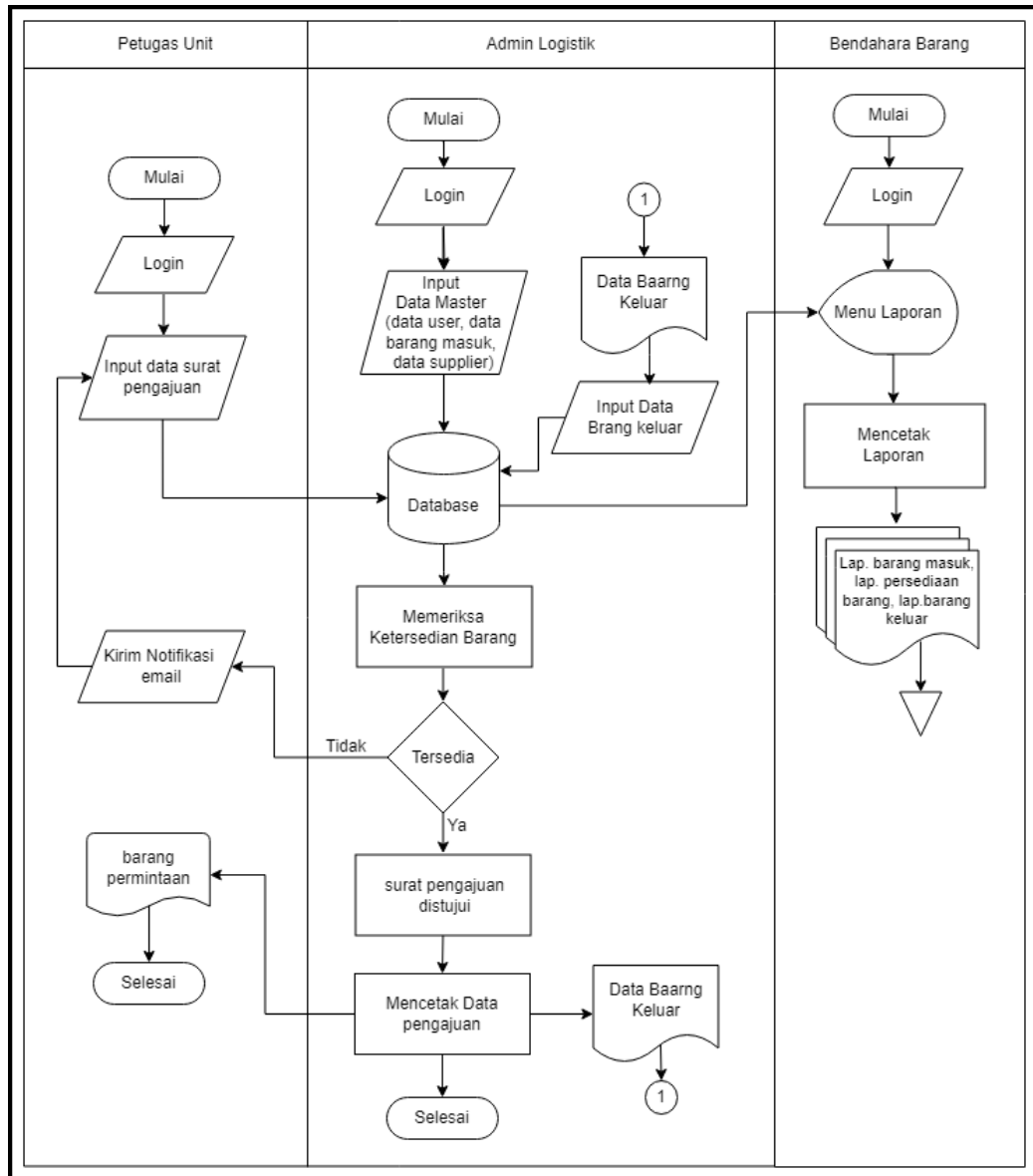
## **2. Kebutuhan Sistem**

- a. Kebutuhan Sistem Petugas Unit
- 1) Petugas unit harus melakukan registrasi jika belum terdaftar dalam web dengan mengisi form registrasi. Kemudian jika sudah, petugas unit dapat login dengan menggunakan nama pengguna dan kata sandi yang terdaftar pada saat registrasi.
  - 2) Dapat menginput data surat pengajuan logistik.
  - 3) Dapat unduh dan print pdf surat pengajuan logistik.
  - 4) Adanya notifikasi untuk menginformasikan bahwa surat pengajuan logistik diterima atau ditolak pada halaman status pengajuan logistik.
  - 5) Pemohon dapat logout dari web.
- b. Kebutuhan Sistem Admin Logistik
- 1) Admin bidang logistik dapat melakukan login untuk dapat mengakses web, dengan menggunakan nama pengguna dan kata sandi.
  - 2) Dapat mengelola data petugas unit, data supplier, data barang dengan fitur tambah, hapus dan edit.
  - 3) Dapat melihat, mengelola data dan cetak surat pengajuan logistik.
  - 4) Dapat menentukan status pengajuan.
  - 5) Dapat mengelola menginput, hapus, dan ubah data barang masuk dan data barang keluar.
  - 6) Dapat menampilkan stok barang yang tersedia.
  - 7) Mencetak laporan barang masuk, laporan persediaan dan laporan barang keluar.
  - 8) Admin bidang logistik dapat logout setelah menggunakan web.
- c. Kebutuhan Sistem Bendahara Barang

- 1) Bendahara barang dapat melakukan login untuk dapat mengakses web, dengan menggunakan nama pengguna dan kata sandi.
- 2) Mencetak laporan barang masuk, laporan persediaan dan laporan barang keluar.
- 3) Bendahara barang dapat logout setelah menggunakan web.

### **3. Flowchart Analisis Sistem Yang Diusulkan**

Berdasarkan hasil dari proses observasi pada Rumah Sakit Umum Daerah Sawerigading Kota Palopo. Terdapat kebutuhan pada bagian logistik, serta dibutuhkan aplikasi berbasis website agar bisa menangani permasalahan yang ada. Analisis sistem yang diusulkan dapat dilihat pada gambar dibawah ini.



Gambar 3.3 Flowchart Analisis sistem yang diusulkan

### 3.2 Deskripsi Data

Selain data proses, pada penelitian ini juga menggunakan data masukan (*input*) dan data keluaran (*output*).

#### 3.2.1 Data Input

Data input merupakan data yang digunakan sebagai masukan pada sistem yang dibangun. Data tersebut disimpan dalam basis data. Data input yang digunakan antara lain:

##### 1. Data Pengguna

Data pengguna terdiri dari *field Id user*, nama, *username*, email, no telepon, *role*, password, dan *isactive*.

2. Data Barang / Bahan Logistik

Data bahan logistik terdiri dari *field Id* barang, Nama barang, Stock, *Id* Satuan dan *Id* Jenis.

3. Data Barang Masuk

Data barang masuk terdiri dari *field Id* barang masuk, *Id user*, *Id* barang, jumlah dan tanggal masuk.

4. Data Barang Keluar

Data barang keluar terdiri dari *field Id* barang keluar, *Id user*, *Id* barang, jumlah dan tanggal keluar.

5. Data Pengajuan Logistik

Data pengajuan logistik terdiri dari *field Id* pengajuan, *Id user*, tanggal pengajuan, nama barang, jumlah, *Id* satuan, *Id* jenis dan keterangan keperluan.

6. Data Satuan dan Data Jenis

Data satuan dan data jenis terdiri dari *Id* barang, nama satuan dan nama jenis.

### 3.2.2 Data Output

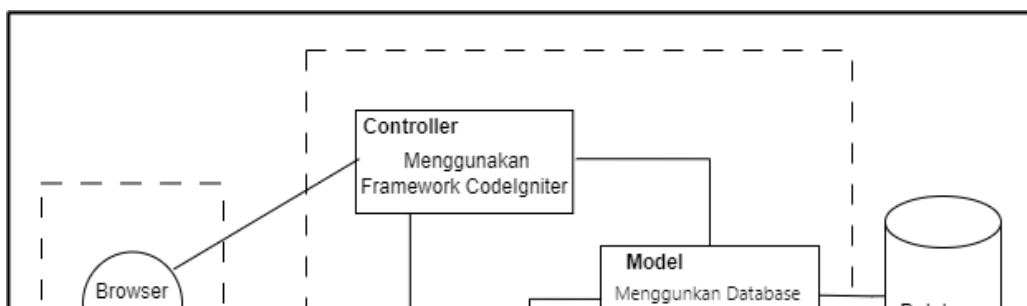
Data output adalah data keluaran hasil dari proses yang dilakukan oleh sistem. Data keluaran dari sistem ini antara lain:

1. Laporan barang masuk
2. Laporan persediaan barang
3. Laporan barang keluar.

### 3.3 Arsitektur Sistem

Arsitektur sistem menentukan bagaimana komponen-komponen tersebut diorganisasikan dan bagaimana mereka saling berinteraksi untuk memastikan kinerja, skalabilitas, keandalan, dan keamanan sistem secara keseluruhan. Arsitektur sistem dalam penelitian ini terdiri dari 3 bagian yaitu *Client Side*, *Server Side* dan *Database*.

1. *Client Side (Frontend)* merupakan jenis website yang memproses data dan menampilkan halaman *web* melalui *browser* pengguna (*Client*) seperti antarmuka pengguna (*User Interface*) terdiri dari :
  - a. Desain antarmuka menggunakan HTML, CSS dan *JavaScript*.
  - b. Fitur utama : Dashboard, Data User, Data Supplier, Master barang, Transaksi, Laporan dan pencarian.
2. *Server Side (Backend)* merupakan suatu istilah yang merujuk pada pemrosesan dan pengelolaan data yang terjadi di sisi server dalam sebuah aplikasi *web* atau sistem komputer. Dalam penelitian ini menggunakan *Framework CodeIgniter* yang terdiri dari :
  - a. *Codeigniter* digunakan untuk mengelola logika bisnis, basis data, dan kontrol aliran aplikasi.
  - b. Dengan menggunakan konsep model-view-controller (MVC), CodeIgniter memisahkan logika aplikasi menjadi tiga komponen utama: model, view, dan controller. Ini memungkinkan pengembang untuk memisahkan antara tampilan (view) yang menampilkan informasi kepada pengguna, logika bisnis (model) yang mengelola data dan operasi terkait, dan kontroler (controller) yang mengatur aliran aplikasi dan menanggapi permintaan pengguna.
  - c. Fitur Utama : Manajemen pengguna, manajemen logistik, manajemen transaksi barang masuk dan barang keluar dan manajemen supplier, serta laporan.
3. *Database* merupakan data yang disimpan dan dikelola sedemikian rupa sehingga dapat diakses, diperbaharui, dan dikelola dengan mudah. *Database* yang digunakan dalam penelitian ini yaitu menggunakan *MySQL* yang terdiri dari:
  - a. *MySQL* digunakan untuk menyimpan data.
  - b. Struktur *database* relasional dengan tabel utama : *Users* , data barang, jenis barang, satuan barang, transaksi barang masuk, transaksi barang keluar dan *Authors*. ORM (*Object-Relational Mapping*) *Eloquent* dari *CodeIgniter* digunakan untuk interaksi dengan *database*.



Gambar 3.4 *Diagram Arsitekur Sistem*

### **3.4 Rancangan Model Sistem**

#### **3.4.1 Perancangan UML (*Unified Modeling Language*)**

Tahap berikutnya setelah meninjau sistem yang ada dan memahami secara mendalam proses yang dianalisis adalah merancang sistem baru. Pada tahap perancangan ini, metode yang digunakan melibatkan *UML (Unified Modeling Language)* yang mencakup beberapa diagram, seperti *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*. Diagram-diagram ini memungkinkan penggambaran operasional sistem dalam bentuk visual yang terstruktur.

##### *1. Use Case Diagram*

Pada tahapan ini menggambarkan siapa saja actor yang terlibat di dalam sistem dan apa yang bisa dilakukan oleh actor di dalam sistem. Adapun actor yang terlibat dalam sistem yang akan dibangun adalah admin logistik, petugas unit dan bendahara barang. Adapaun gambar dari *use case* bisa dilihat pada gambar 3.4 di bawah ini:





3.	Transaksi Barang Masuk	Suatu proses dimana admin menginputkan data barang yang ada di RSUD. Sawerigading Palopo ke dalam sistem.
4.	Transaksi Pengajuan Logistik	Suatu proses yang dimana admin mengelola surat pengajuan bahan logistik dan petugas unit menginputkan surat pengajuan logistik dan dapat melihat status pengajuan .
5.	Transaksi Barang Keluar	Suatu proses dimana admin menginputkan data barang yang keluar.
6.	Laporan	Suatu proses dimana admin melakukan proses pengolahan laporan data barang masuk, laporan persediaan/ <i>stock</i> dan data barang keluar. Suatu proses dimana bendahara barang dapat melihat dan mencetak laporan data barang masuk, laporan persediaan/ <i>stock</i> dan data barang keluar.
7.	Data <i>User</i>	Suatu proses dimana admin logistik dapat menambahkan user baru
8.	Registrasi	Suatu proses dimana petugas unit dapat melakukan registrasi untuk memperoleh akun.

b. Skenario *Use Case*

Sekenario jalannya masing-masing use case yang sudah didefinisikan adalah sebagai berikut :

- Skenario Use Case *Login*

Nama *Use Case* : *Login*

Aktor : Admin logistik, petugas unit, dan bendahara barang

Tujuan : Untuk dapat mengakses web inventory bahan logistik

Tabel 3.2 Skenario *Use Case Login*

Admin, Petugas Unit, Bend. Barang	Sistem
Membuka web <i>inventory</i> barang	Menampilkan halaman <i>login</i>
Mengisi <i>username</i> dan <i>password</i>	Memvalidasi data <i>login</i> , jika data benar maka akan menampilkan halaman utama, jika data salah maka pengguna akan di minta <i>login</i> kembali.

- Skenario *Use Case* Data Barang

Nama *Use Case* : Data Barang

Aktor : Admin

Tujuan : *Input* satuan barang, jenis barang, data barang, dan melihat stok barang yang tersedia.

Tabel 3.3 Skenario *Use Case* Data Barang

Admin	Sistem
Klik menu data utama	Menampilkan <i>button</i> data barang, jenis barang, dan satuan barang
Menampilkan <i>form</i> satuan	Menampilkan <i>form</i> satuan barang
Mengisi nama satuan barang	Menampilkan <i>alert</i> data satuan barang berhasil ditambah
Klik jenis barang	Menampilkan <i>form</i> jenis barang
Mengisi data jenis barang	Menampilkan <i>alert</i> data jenis barang berhasil ditambah
Klik data barang	Menampilkan form data barang
Mengisi data- data barang	Menampilkan <i>alert</i> data barang berhasil ditambah dan menampilkan data barang yang tersedia.

- Skenario *Use Case* Transaksi Barang Masuk

Nama *Use Case* : Transaksi Barang Masuk

Aktor : Admin

Tujuan : *Input* data barang masuk.

Tabel 3.4 Sekenario *Use Case* Transaksi Barang Masuk

Admin	Sistem
Klik menu transaksi	Menampilkan <i>button</i> barang masuk, pengajuan logistik, dan barang keluar
Klik barang masuk	Menampilkan <i>form</i> barang masuk
Mengisi data barang masuk	Menampilkan <i>alert</i> data barang berhasil ditambah.

- Sekenario *Use Case* Transaksi Barang Keluar  
 Nama *Use Case* : Transaksi Barang Keluar  
 Aktor : Admin  
 Tujuan : *Input* data barang keluar.

Tabel 3.5 Sekenario *Use Case* Transaksi Barang Keluar

Admin	Sistem
Klik menu transaksi	Menampilkan <i>button</i> barang masuk, pengajuan logistik, dan barang keluar
Klik barang keluar	Menampilkan <i>form</i> barang keluar
Mengisi data barang keluar	Jika stok mencukupi akan menampilkan <i>alert</i> data barang keluar berhasil ditambah, jika stok tidak mencukupi akan menampilkan <i>alert</i> gagal stok tidak cukup

- Sekenario *Use Case* Transaksi Pengajuan Logistik  
 Nama *Use Case* : Transaksi Pengajuan Logistik  
 Aktor : Petugas unit  
 Tujuan : *Input* data pengajuan logistik.

Tabel 3. 6 Sekenario *Use Case* Transaksi Pengajuan Logistik

Petugas Unit	Sistem
Klik menu transaksi	Menampilkan <i>button</i> barang masuk, pengajuan logistik, dan barang keluar
Petugas unit klik pengajuan logistik	Menampilkan <i>form</i> pengajuan logistic
Mengisi data-data surat pengajuan	Jika surat pengajuan valid akan menampilkan <i>alert</i> surat pengajuan diterima, jika surat invalid akan menampilkan <i>alert</i> surat pengajuan gagal diterima.

- *Skenario Use Case Transaksi Pengajuan Logistik*  
 Nama *Use Case* : Transaksi Pengajuan Logistik  
 Aktor : Admin  
 Tujuan : *Input* data pengajuan logistik.

Tabel 3.7 *Skenario Use Case Transaksi Pengajuan Logistik*

Admin	Sistem
Klik menu transaksi	Menampilkan <i>button</i> barang masuk, pengajuan logistik, dan barang keluar
Petugas unit klik pengajuan logistik	Menampilkan <i>form</i> persetujuan surat pengajuan logistik
Mengoreksi surat pengajuan logistik	Jika surat pengajuan valid akan menampilkan <i>alert</i> surat pengajuan diterima, jika surat invalid akan menampilkan <i>alert</i> surat pengajuan gagal diterima.

- *Skenario Use Case Laporan*  
 Nama *Use Case* : Laporan  
 Aktor : Admin  
 Tujuan : Hasil laporan barang masuk, stok barang dan barang keluar.

Tabel 3.8 *Skenario Use Case Laporan*

Admin	Sistem
Klik menu laporan	Menampilkan <i>button</i> laporan masuk, laporan stok barang dan laporan keluar
Klik laporan barang masuk	Menampilkan form tanggal, bulan dan tahun
Mengisi nama tanggal, bulan dan tahun	Menampilkan laporan barang masuk sesuai dengan tanggal, bulan dan tahun yang di isi.
Klik laporan stok masuk	Menampilkan form tanggal, bulan dan tahun
Mengisi nama tanggal, bulan dan tahun	Menampilkan laporan stok barang sesuai dengan tanggal, bulan dan tahun yang di isi.
Klik laporan barang keluar	Menampilkan form tanggal, bulan dan tahun

Mengisi nama tanggal, bulan dan tahun	Menampilkan laporan barang keluar sesuai dengan tanggal, bulan dan tahun yang di isi.
---------------------------------------	---

- Skenario *Use Case View* Laporan

Nama *Use Case* : Laporan

Aktor : Bendahara Barang

Tujuan : Hasil laporan barang masuk, stok barang dan barang keluar.

Tabel 3.9 Skenario *Use Case View* Laporan

Admin	Sistem
Klik menu laporan	Menampilkan <i>button</i> laporan masuk, laporan stok barang dan laporan keluar
Klik laporan barang masuk	Menampilkan form tanggal, bulan dan tahun
Mengisi nama tanggal, bulan dan tahun	Menampilkan laporan barang masuk sesuai dengan tanggal, bulan dan tahun yang di isi.
Klik laporan stok masuk	Menampilkan form tanggal, bulan dan tahun
Mengisi nama tanggal, bulan dan tahun	Menampilkan laporan stok barang sesuai dengan tanggal, bulan dan tahun yang di isi.
Klik laporan barang keluar	Menampilkan form tanggal, bulan dan tahun
Mengisi nama tanggal, bulan dan tahun	Menampilkan laporan barang keluar sesuai dengan tanggal, bulan dan tahun yang di isi.

- Skenario *Use Case* Manajemen *User*

Nama *Use Case* : Manajemen *User*

Aktor : Admin

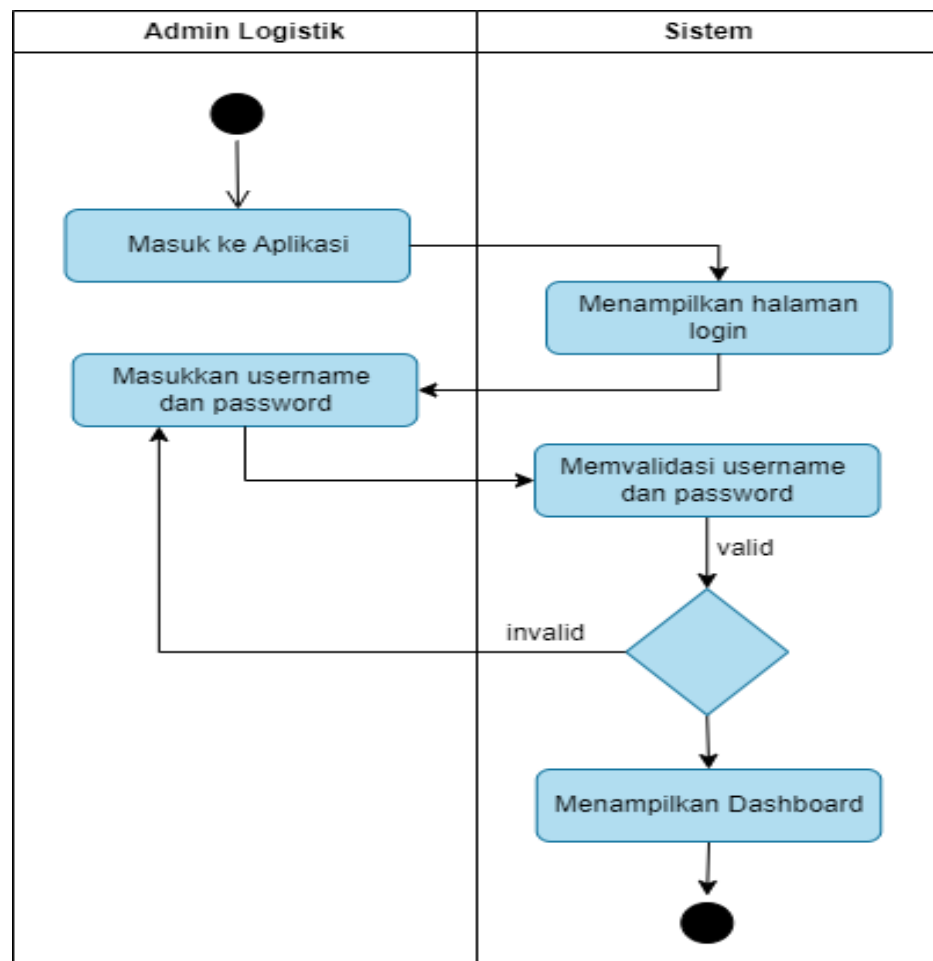
Tujuan : Untuk menambahkan user baru

Tabel 3.10 Skenario *Use Case* Manajemen *User*

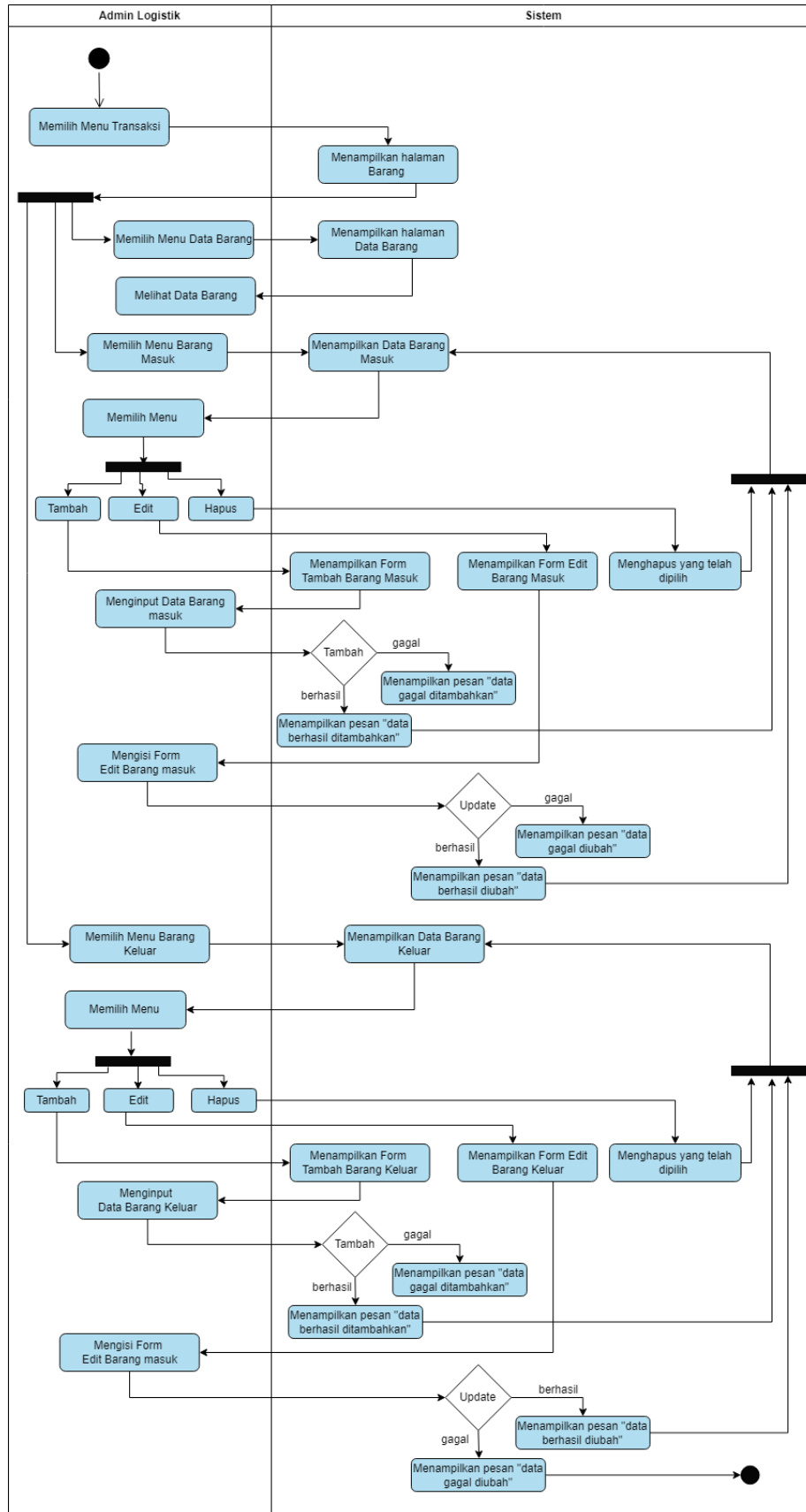
Admin	Sistem
Klik menu manajemen <i>user</i>	Menampilkan <i>form</i> manajemen <i>user</i>
Mengisi data <i>user</i>	Menampilkan alert <i>user</i> berhasil ditambah

## 2. Activity Diagram

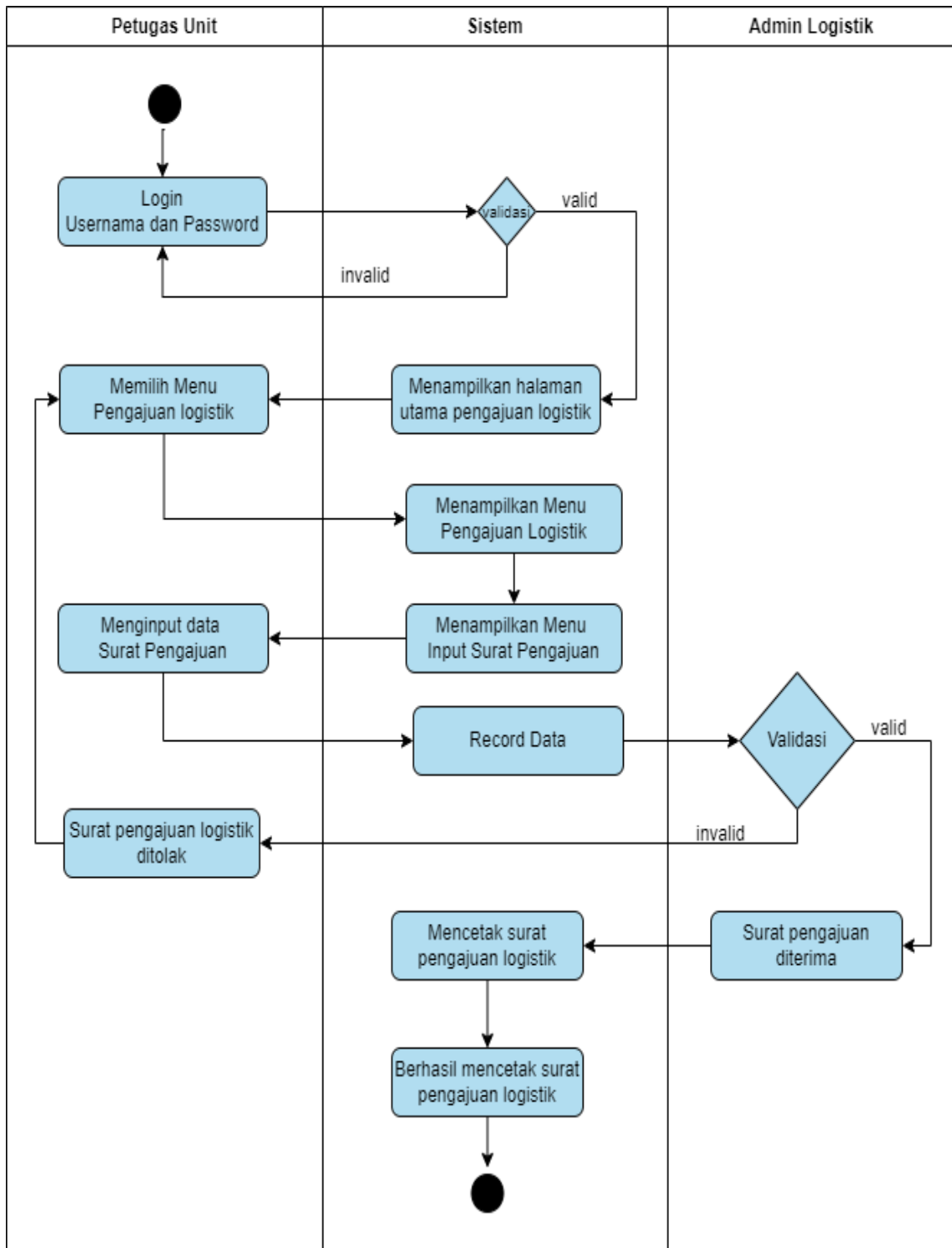
*Activity diagram* diagram diusulkan menampilkan petugas unit, admin logistik, bendahara barang harus melakukan login terlebih dahulu dengan username dan password yang telah dibuat. Sistem akan melakukan validasi username dan password yang dimasukkan. Apabila username dan password yang dimasukkan valid pengguna langsung diarahkan ke halaman utama sistem, namun apabila username dan password tidak valid akan kembali kehalaman login. Berikut adalah *activity diagram* dari aplikasi *inventory* barang pada Rumah Sakit Umum Daerah Palopo.



Gambar 3.6 Activity Diagram Login

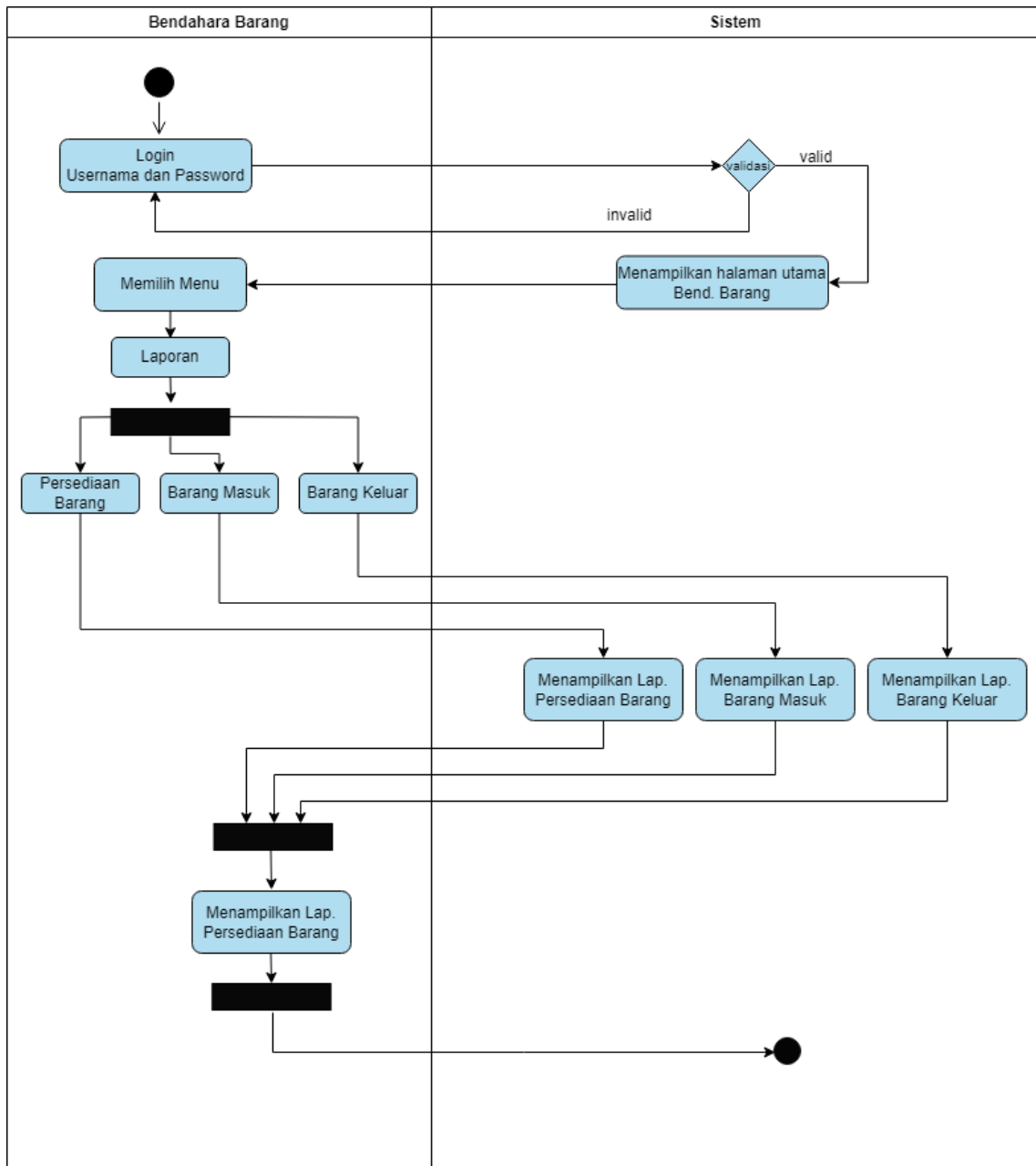


Gambar 3.7 Activity Diagram Transaksi Barang Masuk dan Keluar



Gambar 3.8 Activity Diagram Pengajuan Logistik

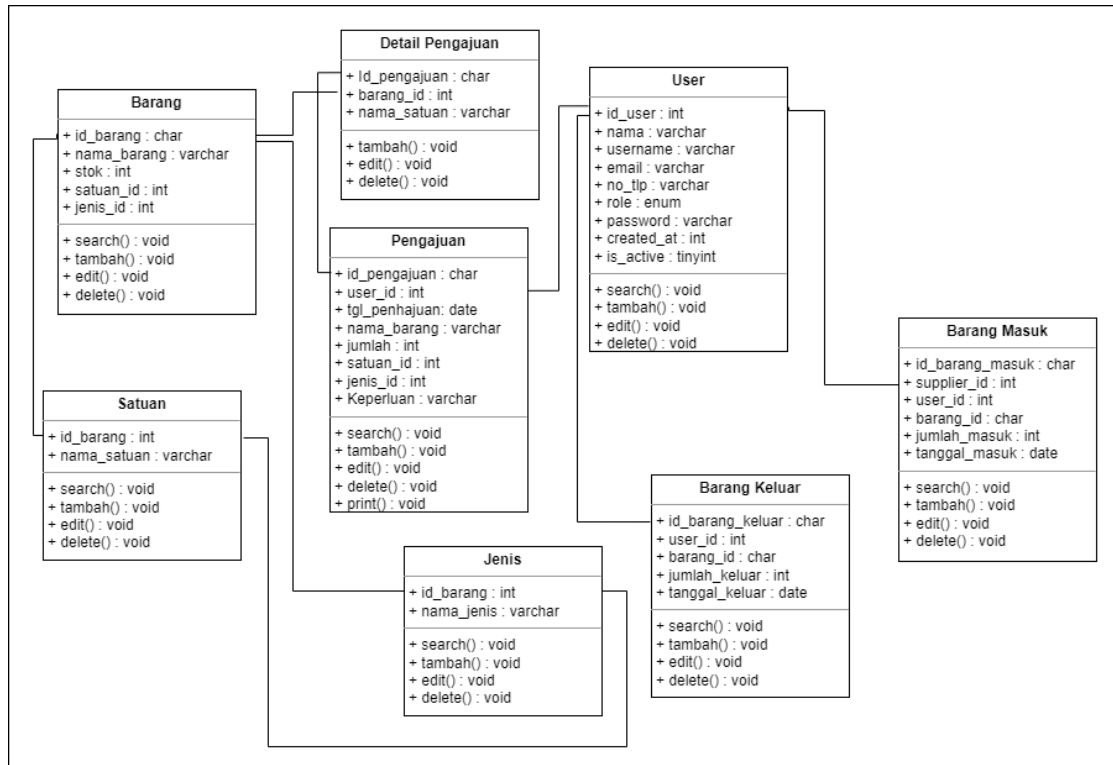




Gambar 3.9 Activity Diagram Bendahara Barang

### 3. Class Diagram

*Class Diagram* merupakan tampilan visual dari struktur program sistem pada tipe yang dibentuk. *Class diagram* menggambarkan struktur dari sebuah sistem dengan mendefinisikan kelas-kelas, atribut dan hubungan antar kelas yang akan dibuat untuk membangun sistem.



Gambar 3.10 *Class Diagram Inventory* barang logistik

### 3.4.2 Perancangan Database

Perancangan database merupakan rancangan tabel yang akan dibuat didalam database untuk memenuhi kebutuhan fungsi sistem yang diusulkan. Berikut ini perancangan tabel yang diusulkan :

#### 1. Tabel Pengguna

Untuk menyimpan data pengguna berupa *username* dan *password* yang digunakan untuk *login* kedalam sistem. Dimana masing-masing field mempunyai tipe dan keterangan tersendiri. Tabel data pengguna dapat dilihat pada tabel dibawah ini.

Tabel 3.11 Perancangan Tabel *User*

No	Field Name	Type Data	Size	Keterangan
1.	<i>Iduser</i>	<i>Integer</i>	11	<i>Primary Key</i>
2.	<i>Username</i>	<i>Varchar</i>	225	<i>Username</i>
3.	<i>Password</i>	<i>Varchar</i>	225	Kata sandi
4.	<i>Email</i>	<i>Varchar</i>	20	Alamat email
5.	<i>Level</i>	<i>Enum</i>	-	Admin atau user
6.	<i>Blokir</i>	<i>Enum</i>	-	<i>Yes / No</i>

## 2. Tabel Barang

Digunakan untuk menyimpan informasi data barang. Dimana masing-masing field mempunyai tipe dan keterangan tersendiri. Tabel data barang dapat dilihat pada tabel dibawah ini

Tabel 3.12 Perancangan Tabel Barang

No	Field Name	Type Data	Size	Keterangan
1.	IdBarang	<i>Char</i>	20	<i>Primary Key</i>
2.	Namabarang	<i>Varchar</i>	20	Nama barang
3.	Stock	<i>Integer</i>	11	Nomor telepon
4.	IdSatuan	<i>Integer</i>	11	Id satuan
5.	IdJenis	<i>Integer</i>	11	Id jenis barang
6.	<i>Created_by</i>	<i>Integer</i>	11	

## 3. Tabel Barang Masuk

Digunakan untuk menyimpan informasi data barang masuk. Dimana masing-masing field mempunyai tipe dan keterangan tersendiri. Tabel data barang masuk dapat dilihat pada tabel dibawah ini.

Tabel 3.13 Perancangan Tabel Barang Masuk

No	Field Name	Type Data	Size	Keterangan
1.	IdBarangMasuk	<i>Char</i>	11	<i>Primary Key</i>
4.	IdBarang	<i>Char</i>	11	Id barang
5.	JumlahMasuk	<i>Integer</i>	11	Jumlah barang
6.	TanggalMasuk	<i>Date</i>	-	Tanggal masuk barang

## 4. Tabel Barang Keluar

Digunakan untuk menyimpan informasi data barang keluar. Dimana masing-masing field mempunyai tipe dan keterangan tersendiri. Tabel data barang keluar dapat dilihat pada tabel dibawah ini.

Tabel 3.14 Perancangan Tabel Barang Keluar

No	Field Name	Type Data	Size	Keterangan
1.	IdBarangKeluar	<i>Char</i>	15	<i>Primary Key</i>
2.	IdBarang	<i>Char</i>	15	Id barang
3.	JumlahKeluar	<i>Integer</i>	5	Jumlah barang
4.	TanggalKeluar	<i>Date</i>	-	Tanggal keluar barang
5.	Status	<i>Integer</i>	11	Status

5. Tabel Pengajuan Logistik

Digunakan untuk menyimpan informasi pengajuan bahan logistik. Dimana masing-masing field mempunyai tipe dan keterangan tersendiri. Tabel data pengajuan logistik dapat dilihat pada tabel dibawah ini.

Tabel 3.15 Perancangan Tabel Pengajuan Logistik

No	Field Name	Type Data	Size	Keterangan
1.	IdPengajuan	<i>Char</i>	15	<i>Primary Key</i>
2.	IdUser	<i>Integer</i>	15	Id user
3.	TglPengajuan	<i>Date</i>	-	Tanggal pengajuan logistik
4.	NamaBarang	<i>Varchar</i>	50	Nama barang
5.	Jumlah	<i>Integer</i>	5	Jumlah barang
6.	IdSatuan	<i>Integer</i>	8	Id satuan
7.	IdJenis	<i>Integer</i>	15	Id jenis barang
8.	Keperluan	<i>Varchar</i>	50	Keterangan

6. Tabel Satuan

Digunakan untuk menyimpan informasi data satuan barang. Dimana masing-masing field mempunyai tipe dan keterangan tersendiri. Tabel data satuan barang dapat dilihat pada tabel dibawah ini.

Tabel 3.16 Perancangan Tabel Satuan Barang

No	Field Name	Type Data	Size	Keterangan
1.	IdBarang	<i>Char</i>	15	Id barang

2.	NamaSatuan	<i>Varchar</i>	15	Nama satuan barang
3.	<i>Created_by</i>	<i>Integer</i>	11	-

#### 7. Tabel Jenis

Digunakan untuk menyimpan informasi data jenis barang. Dimana masing-masing field mempunyai tipe dan keterangan tersendiri. Tabel data jenis barang dapat dilihat pada tabel dibawah ini.

Tabel 3.17 Perancangan Tabel Jenis Barang

No	Field Name	Type Data	Size	Keterangan
1.	IdBarang	<i>Char</i>	15	Id barang
2.	NamaJenis	<i>Varchar</i>	15	Nama jenis barang
3.	<i>Created_by</i>	<i>Integer</i>	11	-

### 3.4.3 Perancangan User Interface (Antarmuka Pengguna)

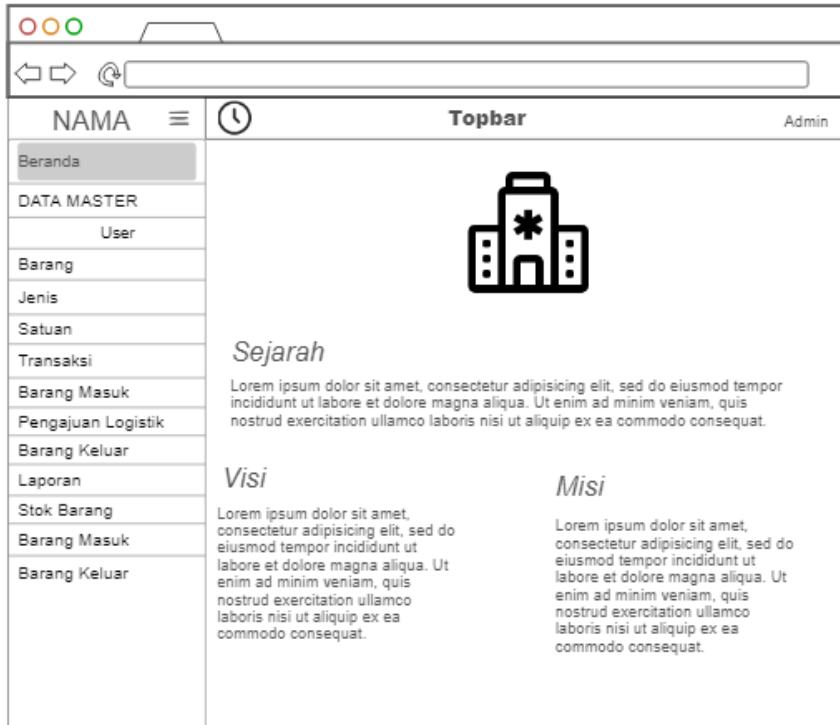
Rancangan *user interface* program merupakan desain untuk membuat rancangan tampilan program yang akan dibuat sebagai acuan dalam pembuatan aplikasi. Berikut ini rancangan *user interface* program yang diusulkan :

#### 1. *Interface* Halaman *Login*

The image shows a web browser window with the address bar displaying 'http://login'. The main content area contains a login form titled 'IVENTORY BAHAN LOGISTIK' with the subtitle 'RUMAH SAKIT UMUM DAERAH SAWERIGADING PALOPO'. The form has two input fields: 'Email' with a mail icon and the text 'username', and 'Password' with a lock icon and masked characters '\*\*\*\*\*'. Below these fields is a 'Login' button.

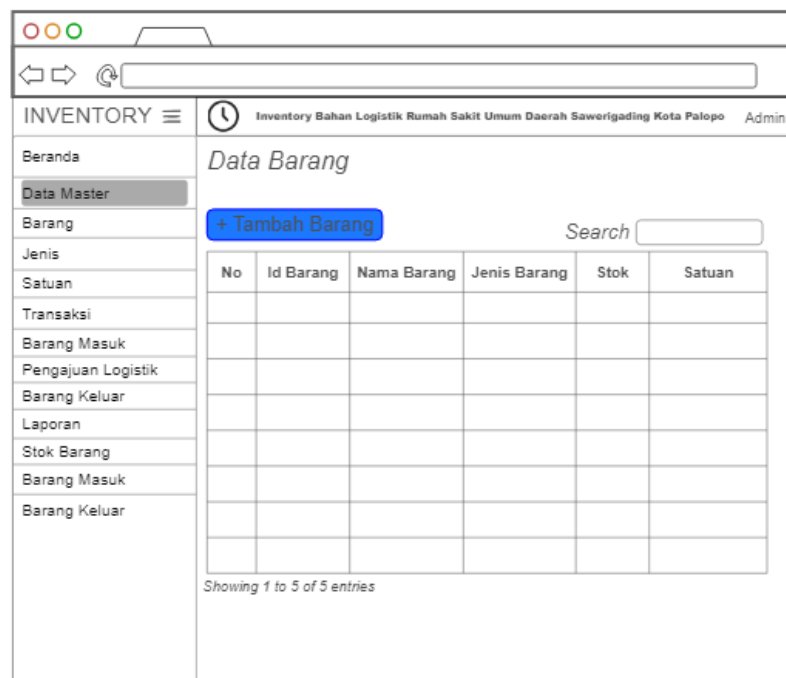
Gambar 3. 11 *Interface* Halaman *Login*

## 2. Interface Halaman Dashboard



Gambar 3.12 Interface Halaman Dashboard

## 3. Interface Halaman Data Barang



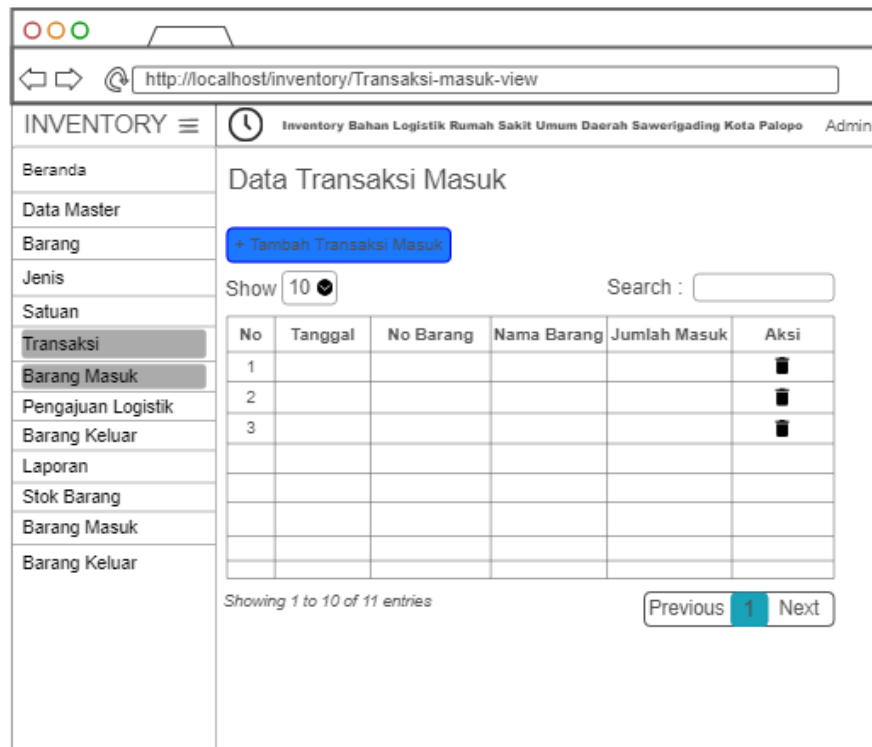
Gambar 3. 13 Interface Halaman Data Barang

4. *Interface* Halaman Data Jenis Barang



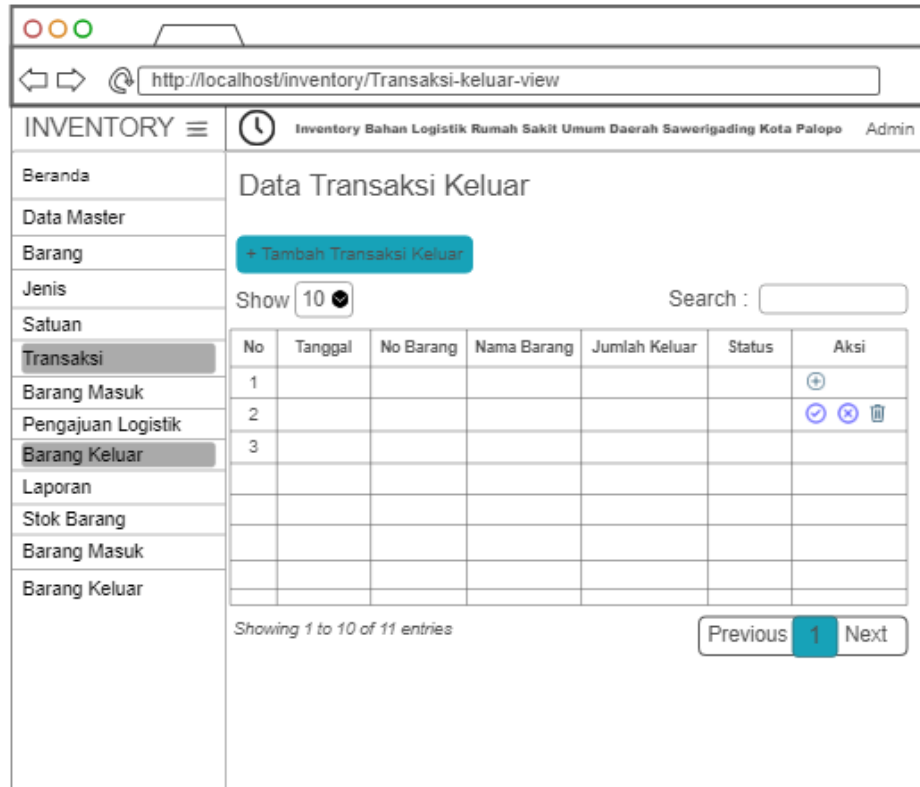
Gambar 3.14 *Interface* Halaman Data Jenis Barang

5. *Interface* Halaman Transaksi Barang Masuk



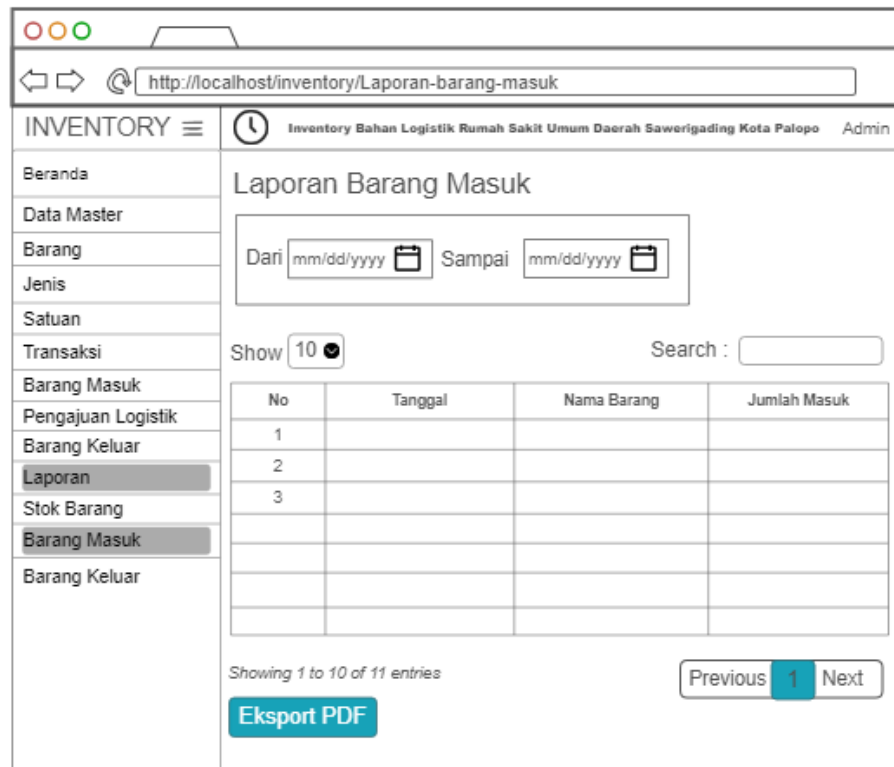
Gambar 3. 15 *Interface* Halaman Transaksi Barang Masuk

6. *Interface* Halaman Transaksi Barang Keluar



Gambar 3.16 *Interface* Halaman Transaksi Barang Keluar

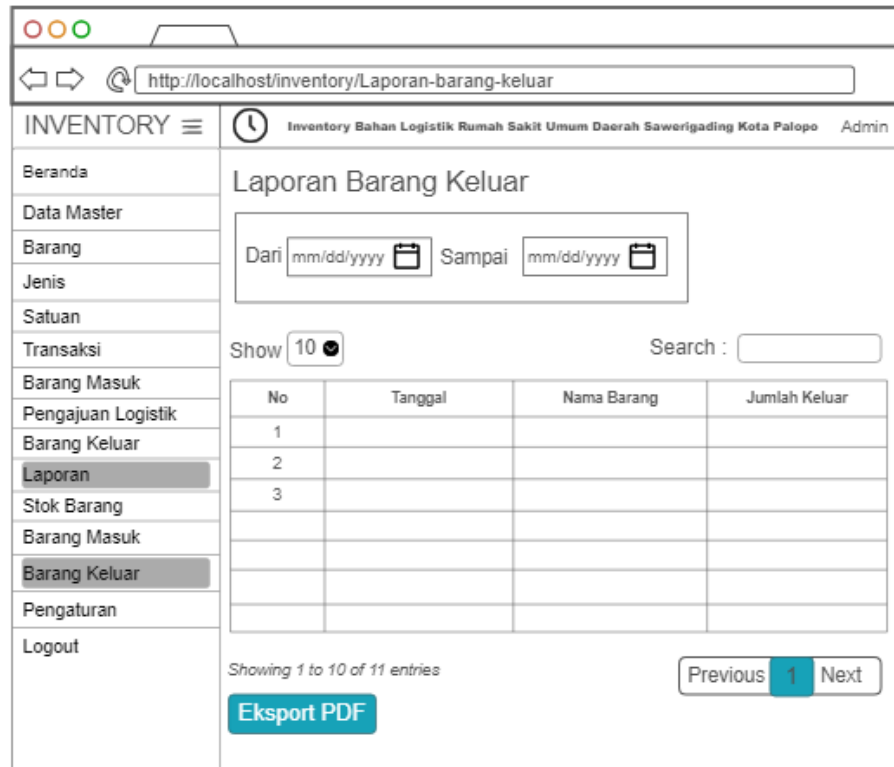
7. *Interface* Halaman Laporan Barang Masuk



Gambar 3.17 *Interface* Halaman Laporan Barang Masuk



## 8. Interface Halaman Laporan Barang Keluar



Gambar 3.18 Interface Halaman Laporan Barang Keluar

### 3.5 Rancangan Pengujian Sistem

Tahapan pengujian sistem untuk mengetahui kesalahan apa saja yang timbul saat aplikasi sedang berjalan serta mengetahui apakah sistem yang dibangun sudah sesuai dengan kebutuhan pengguna. Metode pengujian yang digunakan pada tahapan ini adalah metode *whitebox testing* dan *blackbox testing*, dimana pengujian yang dilakukan terhadap form beberapa masukkan apakah sudah berjalan sesuai dengan fungsinya masing – masing.

#### 1. Black Box Testing

Pengujian menggunakan *Black Box* digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak yang dirancang. Berikut ini adalah tabel rencana pengujian *black box*.

Tabel 3.18 Tabel Perancangan *Black Box Testing*

No	Komponen yang diuji	Butir Uji	Jenis Pengujian
1	Fungsi menu data barang	Menambah data barang baru	<i>Black Box</i>
		Mengedit data barang	<i>Black Box</i>
		Menghapus data barang	<i>Black Box</i>
2	Fungsi Menu Jenis	Menambah jenis barang	<i>Black Box</i>

	Barang	baru	
		Mengedit jenis barang	<i>Black Box</i>
		Menghapus jenis barang	<i>Black Box</i>
3	Fungsi Menu Satuan	Menambah data satuan	<i>Black Box</i>
		Mengedit data satuan	<i>Black Box</i>
		Menghapus data satuan	<i>Black Box</i>
4	Fungsi Menu Transaksi Barang Masuk	Melakukan transaksi barang keluar	<i>Black Box</i>
		Menambah data barang masuk	<i>Black Box</i>
		Menghapus data barang masuk	<i>Black Box</i>
5	Fungsi Menu Transaksi Barang Masuk	Melakukan transaksi barang keluar	<i>Black Box</i>
		Menambah data barang keluar	<i>Black Box</i>
		Menghapus data barang keluar	<i>Black Box</i>
6	Fungsi Menu Laporan Stock Barang Barang Masuk Barang Keluar	Menampilkan data Laporan Stok Barang	<i>Black Box</i>
		Melakukan <i>Eksport</i> Laporan Barang Masuk	<i>Black Box</i>
		Melakukan <i>Eksport</i> Laporan Barang Keluar	<i>Black Box</i>

## 2. *White Box Testing*

Pengujian *White box* pada penelitian ini menggunakan Metode *Cyclomatic Complexity* (CC) yang digunakan untuk menghitung tingkat kompleksitas logika program. Dalam pengujian sistem, bagan alir program (flowchart) yang telah didesain sebelumnya dipetakan ke dalam bentuk bagan alir kontrol (*flowgraph*). Ini mempermudah penentuan jumlah region, *Cyclomatic Complexity* (CC), dan Independent Path. Jika jumlah region, *Cyclomatic Complexity* (CC), dan Independent Path sama, maka sistem dianggap benar. Namun, jika ada perbedaan, berarti sistem masih memiliki kesalahan, baik dari segi logika maupun aspek lainnya.

*Cyclomatic Complexity* (CC) dapat dihitung dengan menggunakan rumus :

$$V(G) = E - N + 2$$

Dimana :

E = jumlah edge pada *flowgraph*

N = Jumlah node pada *flowgraph*

## **BAB IV IMPLEMENTASI**

### **4.1 Implementasi**

Setelah memahami konsep dan merancang solusi, implementasi menjadi langkah krusial untuk mewujudkan ide menjadi kenyataan. Dalam bab ini, kami akan mengeksplorasi bagaimana konsep-konsep yang telah dirancang dan direncanakan sebelumnya diwujudkan dalam bentuk nyata. Kami akan menyelami proses detail yang terlibat dalam mentransformasikan ide menjadi kode, meliputi pemilihan alat dan teknologi yang sesuai, pengembangan aplikasi secara bertahap, serta pengujian untuk memastikan kualitas dan kinerja yang optimal.

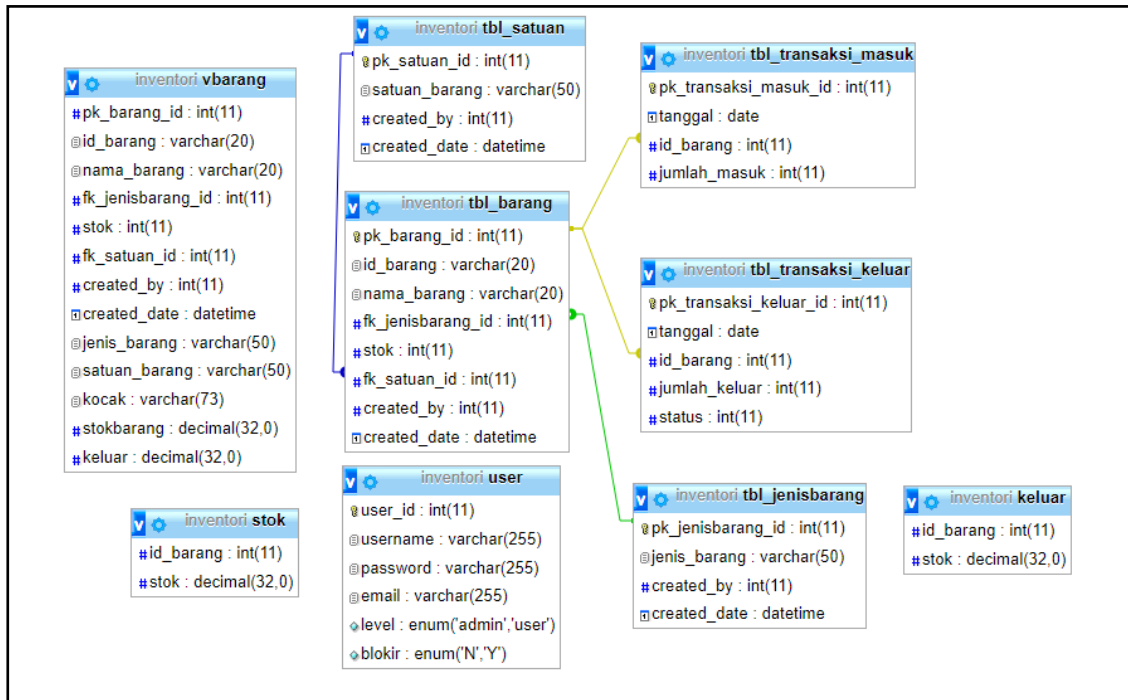
#### **4.1.1 Implementasi Sistem**

Implementasi sistem merupakan tahap akhir dari pengembangan sistem, tahap ini merupakan tindakan nyata dari tahap perencanaan dan perancangan sistem. Langkah-langkah implementasi akan berjalan baik jika perencanaan dan perancangan sistem memang terencana secara rapi. Implementasi sistem ini berupa desain dari halaman yang dibuat menggunakan bahasa pemrograman PHP dan Database MySQL. Spesifikasi *hardware* dan *software* yang digunakan dalam proses implementasi sistem, sebagai berikut:

1. Spesifikasi *Hardware* :
  - a. *Processor Intel Core i3-6006 @ 2.0GHz*
  - b. *Hard Disk Drive 500 GB*
  - c. *RAM 4 GB*
2. Spesifikasi *Software* :
  - a. *Sistem Operasi Windows 10*
  - b. *Framework CodeIgniter 3*
  - c. *XAMPP versi 5.6.23*
  - d. *Visual Studio Code*
  - e. *Web Browser*

#### **4.1.2 Implementasi Database**

Database digunakan untuk menyimpan data yang ada pada aplikasi inventory yang terdiri dari sembilan database sebagai berikut.



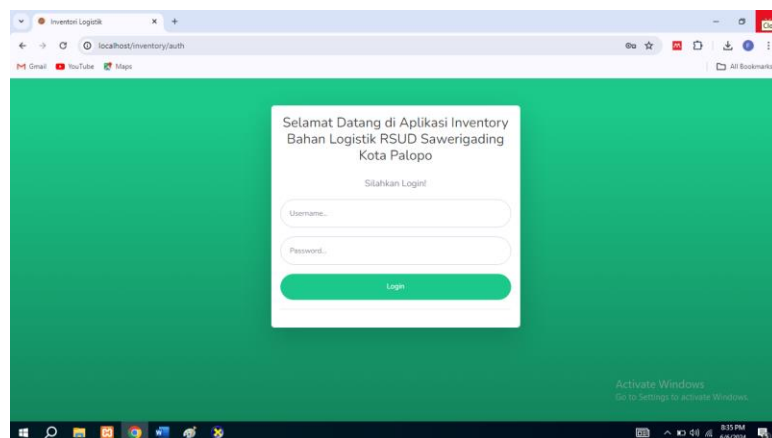
Gambar 4.1 Implementasi Database

### 4.1.3 Implementasi Program

Implementasi program merupakan tahapan dimana sistem yang telah dirancang kemudian diterjemahkan ke dalam kode-kode menggunakan bahasa pemrograman PHP dalam bentuk tampilan grafis yang siap dioperasikan oleh pengguna. Hasil implementasi antar muka dari perancangan aplikasi inventory bahan logistik berbasis web pada Rumah Sakit Umum Daerah Sawerigading Kota Palopo adalah sebagai berikut :

#### 1. Implementasi Halaman Login

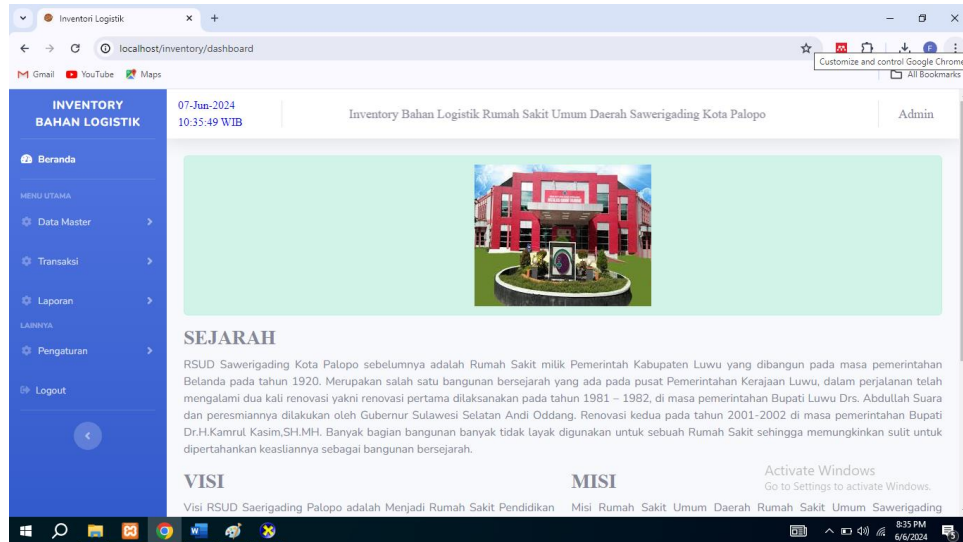
Halaman digunakan admin logistik, petugas unit dan bendahara barang untuk masuk kedalam sistem dengan mengisi username dan password.



Gambar 4.2 Tampilan Halaman Login

## 2. Implementasi Halaman Utama

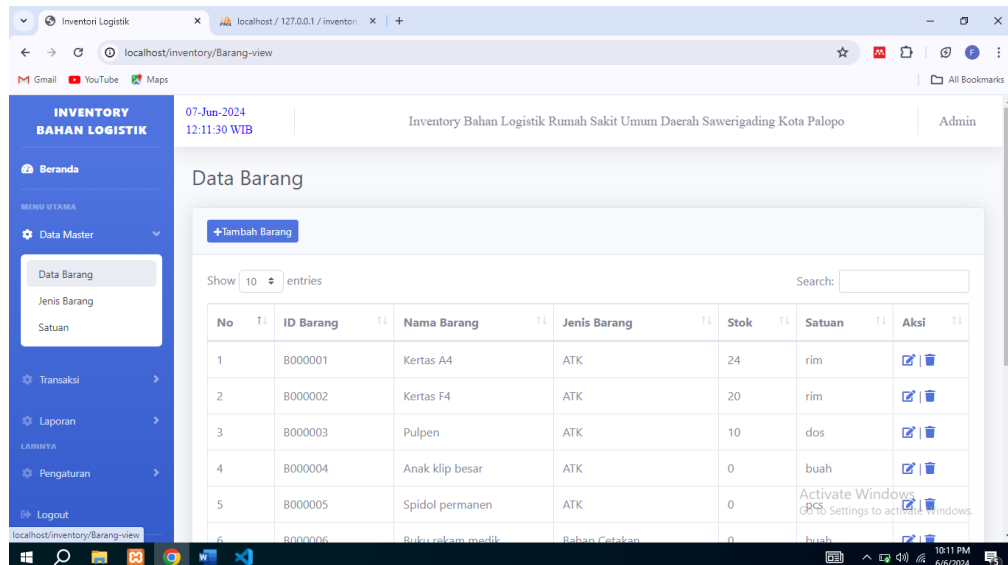
Setelah admin berhasil login, maka akan muncul halaman utama admin yang terdapat 3 menu halaman yaitu menu data master yang terdiri dari data barang, jenis barang, dan satuan. Menu transaksi yang terdiri dari barang masuk, dan barang keluar. Menu laporan terdiri dari laporan stok barang, barang masuk, dan laporan barang keluar.



Gambar 4.3 Tampilan Halaman Utama

## 3. Implementasi Halaman Data Barang

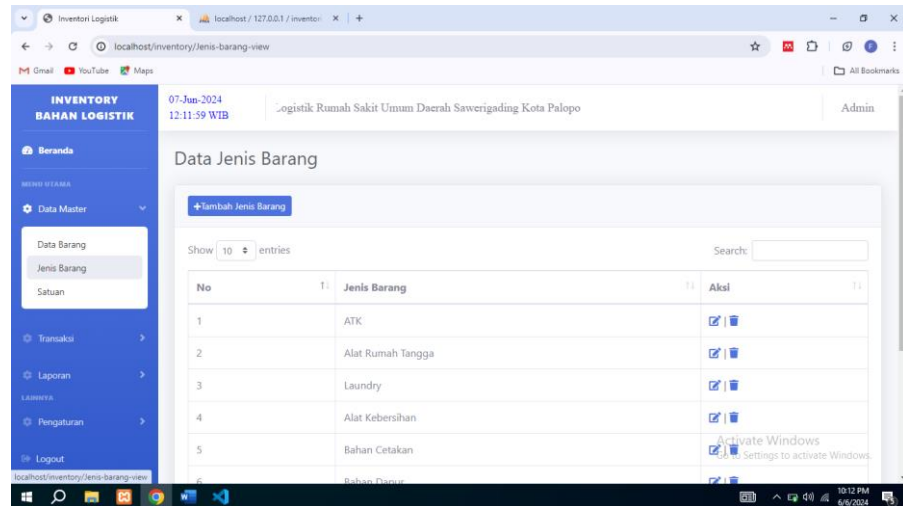
Halaman ini terdapat form data barang yang sudah terhubung dengan menu satuan dan jenis barang. Admin dapat menambahkan nama barang baru, mengedit, dan menghapus data barang tersebut.



Gambar 4.4 Tampilan Halaman Data Barang

#### 4. Implementasi Halaman Jenis Barang

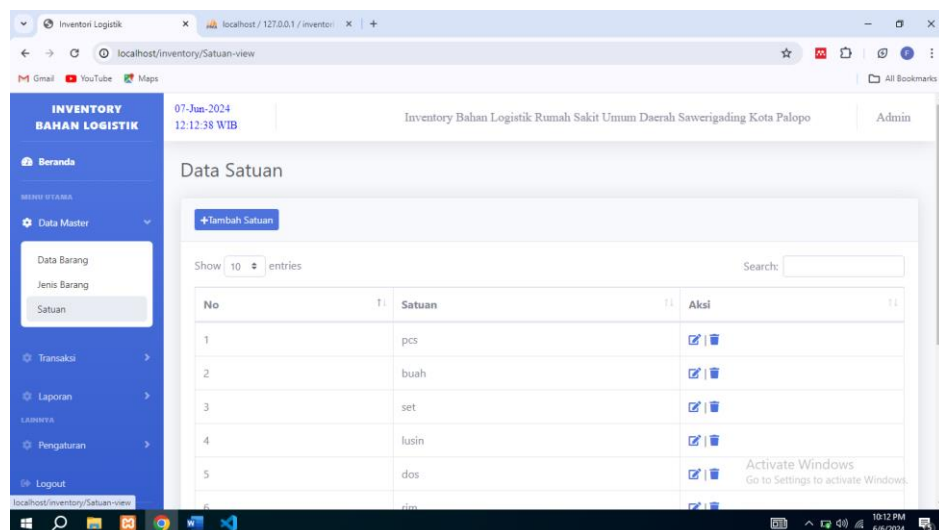
Halaman ini menampilkan form data jenis barang. Dalam halaman ini admin dapat menambahkan data kategori baru, mengedit, dan menghapus jenis barang tersebut.



Gambar 4.5 Tampilan Halaman Jenis Barang

#### 5. Implementasi Halaman Satuan Barang

Halaman ini menampilkan form data satuan barang yang berisi nama satuan barang yaitu buah, pcs, rim, refill dan lain-lain. Dalam halaman ini admin dapat menambahkan data satuan baru, mengedit dan menghapus data satuan tersebut.

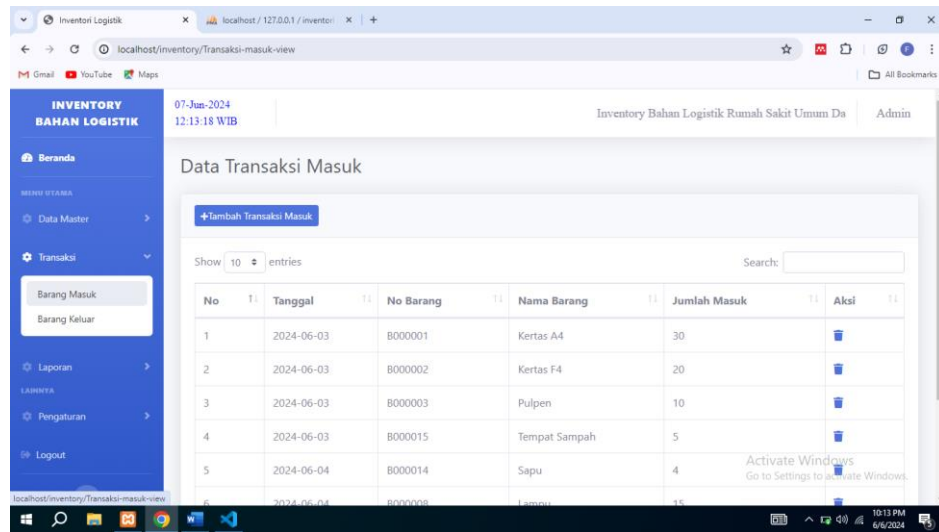


Gambar 4.6 Tampilan Halaman Satuan Barang

#### 6. Implementasi Halaman Transaksi Barang Masuk

Halaman ini terdapat form data transaksi barang masuk. Admin dapat menginputkan data barang masuk seperti nama barang, jumlah barang yang

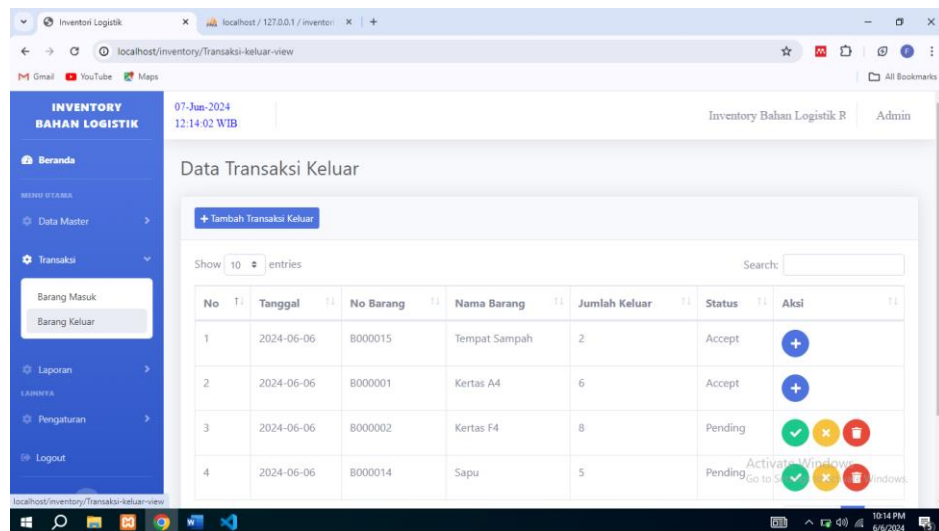
masuk, dan tanggal. Admin dapat mencari data barang masuk dengan menginputkan tanggal masuk dan nama barang.



Gambar 4.7 Tampilan Halaman Transaksi Barang Masuk

## 7. Implementasi Halaman Transaksi Barang Keluar

Halaman ini terdapat from data transaksi barang keluar. Admin dapat menginputkan data barang seperti nama barang, dan tanggal, jumlah, dan tanggal barang keluar. Admin dapat mencari data barang keluar dengan menginputkan tanggal, dan nama barang.

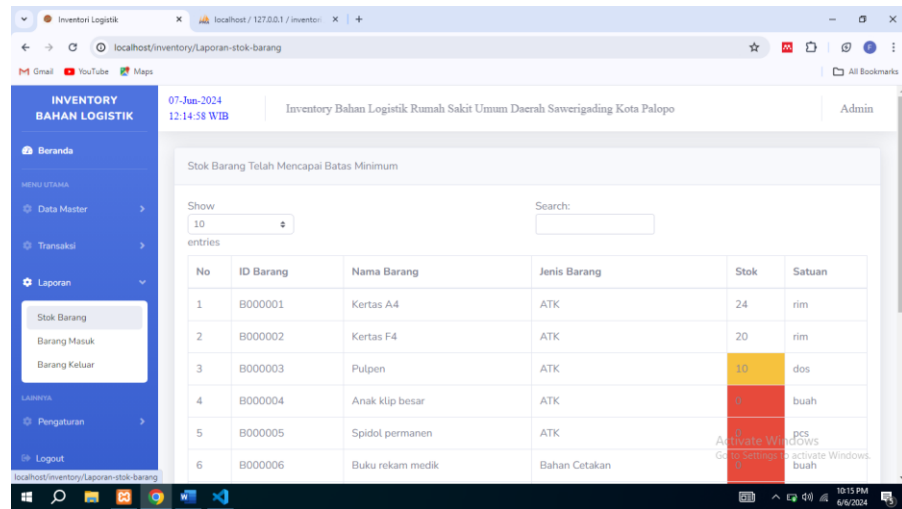


Gambar 4.8 Tampilan Halaman Transaksi Barang Keluar

## 8. Implementasi Halaman Laporan Stok Barang

Halaman ini bisa diakses oleh admin dan bendahara barang, terdapat data stok barang yang tersedia. Sistem ini memiliki jumlah minimum dalam persediaan stok yaitu 10 stok, jika jumlah salah satu stok barang menipis atau bahkan saat

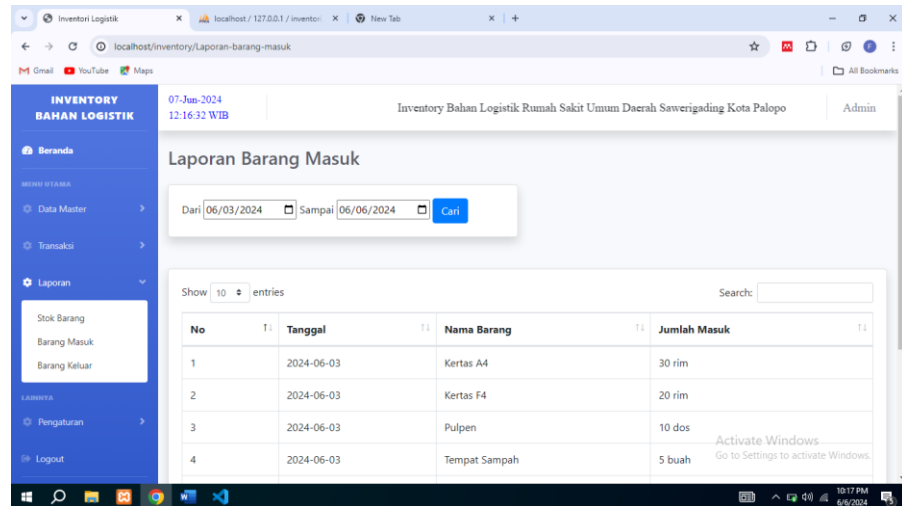
barang kosong, pada sistem data barang akan muncul pemberitahuan dengan menampilkan warna status barang atau warning jumlah stok yang tersisa.



Gambar 4.9 Tampilan Halaman Laporan Stok Barang

## 9. Implementasi Halaman Laporan Barang Masuk

Halaman ini bisa diakses oleh admin logistik dan bendahara barang, terdapat tools untuk memilih laporan sesuai dengan tanggal, bulan dan tahun yang diinputkan admin ataupun bendahara barang.

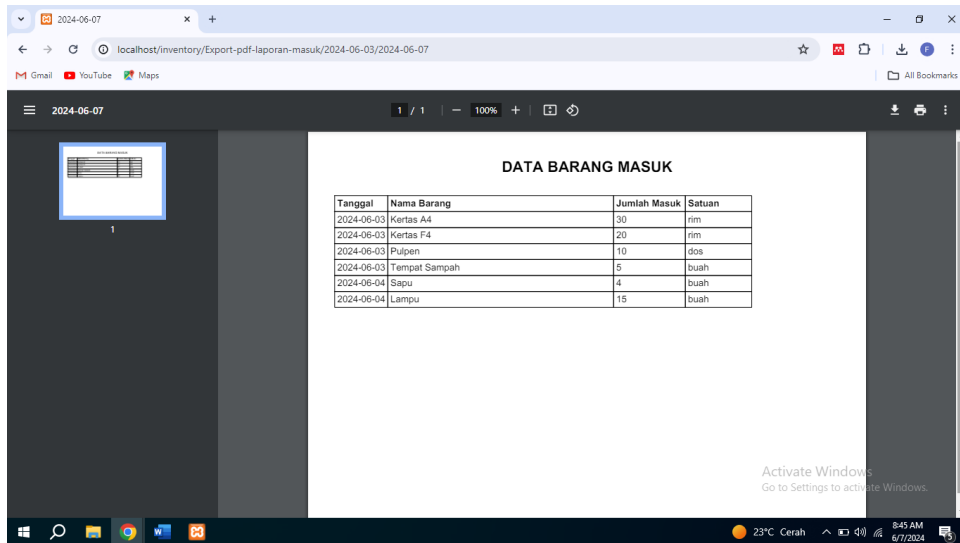


Gambar 4.10 Tampilan Halaman Laporan Barang Masuk

## 10. Implementasi Hasil Laporan Barang Masuk

Hasil laporan data barang masuk sesuai dengan nama tanggal, bulan dan tahun yang diinputkan admin atau bendahara barang. Menampilkan tanggal barang masuk, nama, satuan barang jumlah barang masuk.

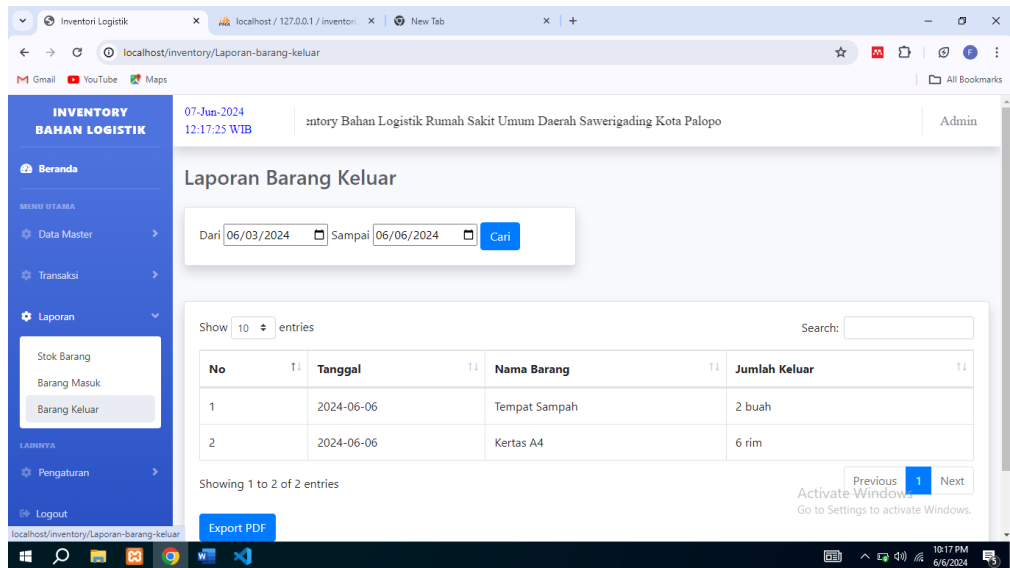




Gambar 4.11 Hasil Laporan Barang Masuk

## 11. Implementasi Halaman Laporan Barang Keluar

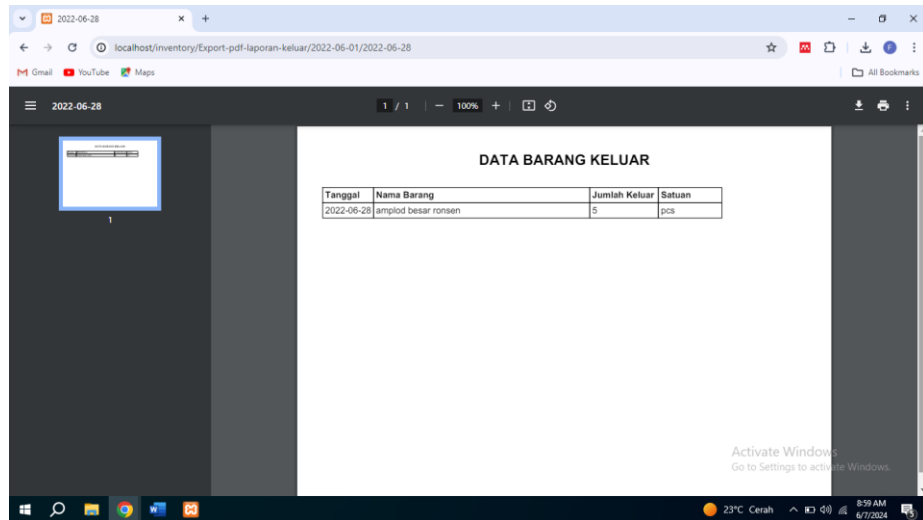
Halaman ini bisa diakses oleh admin logistic dan bendahara barang, terdapat tools untuk memilih laporan sesuai dengan tanggal, bulan dan tahun yang diinputkan admin logistic ataupun bendahara barang.



Gambar 4.12 Tampilan Hasil Laporan Barang Keluar

## 12. Implementasi Hasil Laporan Barang Keluar

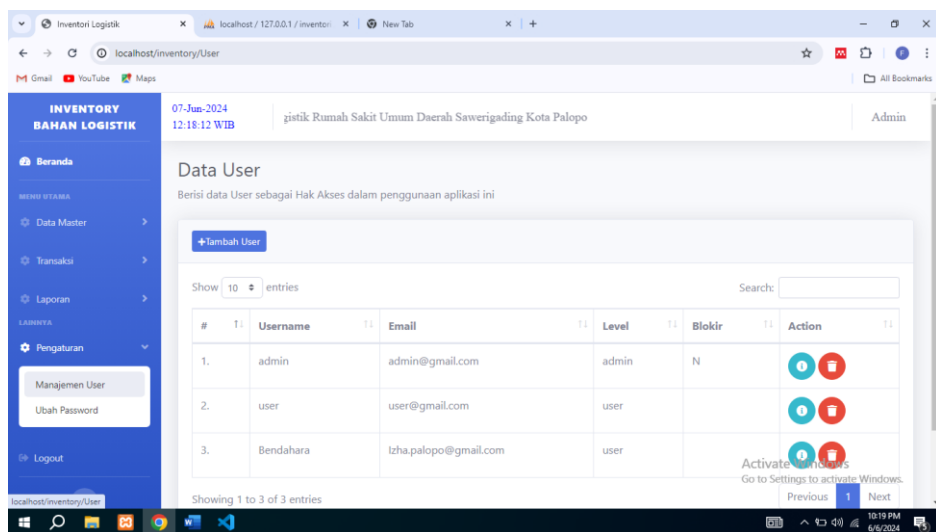
Hasil laporan data barang keluar sesuai dengan nama tanggal, bulan dan tahun yang diinputkan admin atau bendahara barang. Menampilkan tanggal barang keluar, nama, satuan barang jumlah barang keluar.



Gambar 4.13 Tampilan Hasil Laporan Barang Keluar

### 13. Implementasi Halaman Manajemen User

Halaman ini terdapat form manajemen user, dimana admin dapat menambahkan data user baru, mengedit, dan menghapus data tersebut.



Gambar 4.14 Tampilan Halaman Manajemen User

## 4.2 Hasil Pengujian

### 4.2.1 Hasil Pengujian Black Box

Berikut ini merupakan hasil dari pengujian *black box* ditunjukkan oleh tabel berikut:

Tabel 4.1 Tabel Pengujian *Black Box*

No	Kasus Uji	Langkah Uji	Hasil yang diharapkan	Hasil Aktual	Keterangan
1	Menu	Run	Menampilkan <i>form</i>	Tampilan	Berhasil

	<i>Login</i>	program, muncul <i>form</i> <i>login</i>	<i>login</i>	halaman utama	
2	Menu data barang	Klik menu data barang	Menampilkan <i>form</i> data barang	Tampilan data barang	Berhasil
3	Menu jenis barang	Klik menu jenis barang	Menampilkan <i>form</i> jenis barang	Tampilan data jenis barang	Berhasil
4	Menu satuan	Klik menu satuan	Menampilkan <i>form</i> satuan	Tampilan data satuan	Berhasil
5	Menu transaksi barang masuk	Klik Menu transaksi barang masuk	Menampilkan <i>form</i> data transaksi barang masuk	Tampilan data transaksi barang masuk	Berhasil
6	Menu transaksi barang keluar	Klik Menu transaksi barang keluar	Menampilkan <i>form</i> data transaksi barang keluar	Tampilan data transaksi barang keluar	Berhasil
7	Menu laporan stok barang	Klik Menu laporan stok barang	Menampilkan <i>form</i> laporan stok barang	Tampilan data laporan stok barang	Berhasil
8	Menu laporan barang masuk	Klik Menu laporan barang masuk	Menampilkan <i>form</i> laporan barang masuk	Tampilan eksport laporan barang masuk	Berhasil
9	Menu laporan barang masuk	Klik Menu laporan barang masuk	Menampilkan <i>form</i> laporan barang masuk	Tampilan eksport laporan barang masuk	Berhasil

#### 4.2.2 Hasil Pengujian White Box

White box testing untuk menguji perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan, dan keluaran yang sesuai dengan spesifikasi kebutuhan. Hasil pengujian white box dari

perancangan aplikasi inventory bahan logistik berbasis web pada Rumah Sakit Umum Daerah Sawerigading Kota Palopo adalah sebagai berikut :

## 1. Pengujian *White Box Router Form Login*

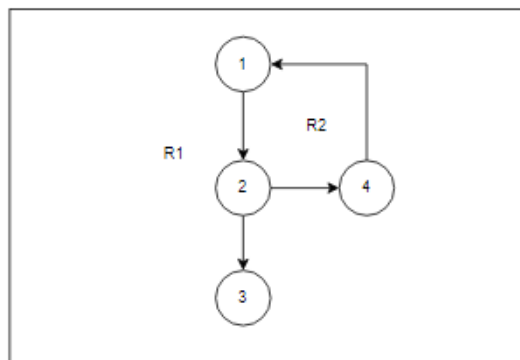
### a. *Source Code Login*

```
public function proses_login()
{
    $this->form_validation->set_rules('username',
'username','required', [
    'required' => 'Username wajib diisi!']);
    $this->form_validation->set_rules('password',
'password','required', [
    'required' => 'Password wajib diisi!']);

    if ($this->form_validation->run()== FALSE){
        $this->load->view('templates/head/dashboard');
        $this->load->view('master/login');
    }else{
        $username = $this->input->post('username');
        $password = $this->input->post('password');
```

Gambar 4.15 *Source Code Form Login*

Kemudian untuk tahap selanjutnya akan dibuat *flow graph* berdasarkan kode program *route form login* seperti pada gambar di bawah ini.



Gambar 4.17 *Flowgraph Form Login*

Berdasarkan flow graph pada Gambar diatas diketahui bahwa jumlah edge (E) = 4 yang merupakan garis yang menghubungkan node, jumlah node (N) = 4 yang merupakan lingkaran yang menggambarkan suatu aktivitas, jumlah predicate (P) = 1 yang merupakan node bercabang, dan jumlah region (R) = 2 yang menandakan suatu area dalam flow graph, yang dapat dilihat dengan simbol R1 hingga R2 pada Gambar diatas,

sehingga jika dimasukkan ke dalam rumus perhitungan cyclomatic complexity dan juga bantuan graph matrix pada Tabel 4.2 dan Tabel 4.3. maka akan menghasilkan sebagai berikut.

Tabel 4.2 *Cyclomatic Complexity Form Login*

$V(G) = E - N + 2$	$V(G) = P + 1$	$V(G) = R$
$V(G) = 4 - 4 + 2$	$V(G) = 1 + 1$	$V(G) = 2$
$V(G) = 2$	$V(G) = 2$	$V(G) = 2$

Tabel 4.3 *Graph Matrix Form Login*

Note	1	2	3	4	Hasil
1		1			0
2			1	1	1
3					0
4	1				0
Jumlah					2

Berdasarkan dari hasil perhitungan *cyclomatic complexity* pada tabel 4.1 dan *graph matrix* pada tabel 4.2, didapatkan jumlah hasil *independent path* untuk *form login* yaitu 2 dengan jalur *independent path* sebagai berikut:

- Jalur 1 : 1-2-3 (Skenario *login* berhasil)
- Jalur 2 : 1-2-4-1 (Skenario *login* gagal)

Berdasarkan hasil pengujian yang dilakukan, diketahui bahwa route form *login* dengan jumlah CC 2 memiliki tingkat resiko yang rendah dan prosedur yang sederhana. Setelah menentukan jalur *independent path* maka Langkah selanjutnya yaitu membuat *test case* dan melaksanakan pengujian. Hasil pengujian dapat dilihat pada Tabel 4.3.

Tabel 4.4 Tabel *Test Case Form Login*

No	Jalur	Keterangan	Hasil yang diharapkan	Hasil Sebenarnya	Sesuai/Tidak
1.	1-2-3	Memasukkan username dan password yang valid	Memuat dashboard dengan sukses	Berhasil menampilkan <i>dashboard</i>	Sesuai
2.	1-2-4	Validasi username dan password	Menampilkan pesan Username atau Password Salah”	Berhasil menampilkan pesan	Sesuai

## 2. Pengujian White Box Router Form Tambah Barang

### a. Source Code Router Form Tambah Barang

```
function add_view()
{
    $data['jenisbarang'] = $this->Mbarang->
    >get_master_toobject("tbl_jenisbarang","pk_jenisbarang_id",
    "jenis_barang","jenis_barang","Select","","");
    $data['satuan'] = $this->Mbarang->
    >get_master_toobject("tbl_satuan","pk_satuan_id","satuan_
    _barang","satuan_barang","Select","","");

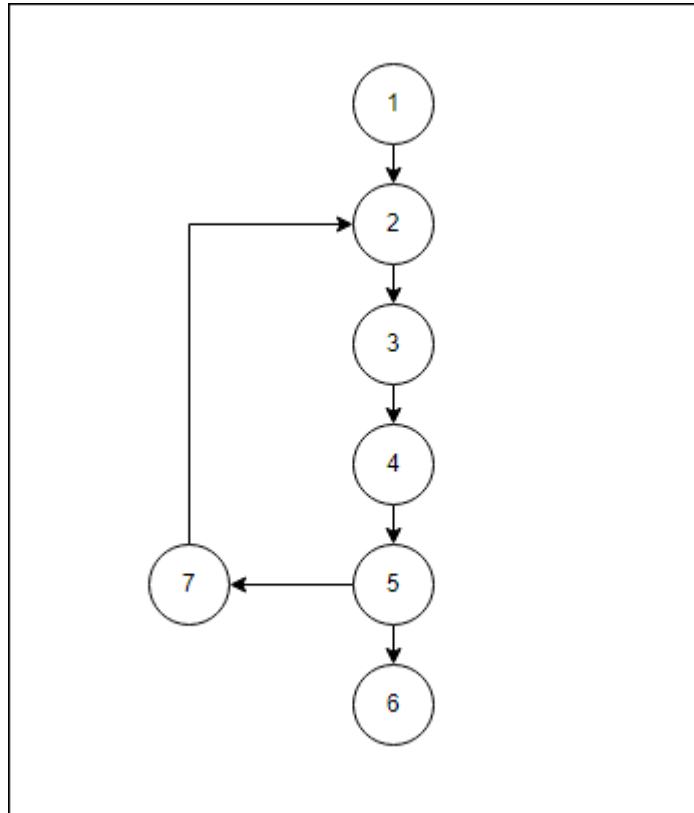
    $this->load->view('templates/head/tabel');
    $this->load->view('templates/sidebar');
    $this->load->view('templates/topbar');
    $this->load->view('master/barang/vaddbarang',
    $data);
    $this->load->view('templates/footer/tabel');
}
function save_data()
{ //data diri
    $barangno = $this->Mbarang->generatenobarang();
    // print_r($barangno);
    $nama = $this->input->post('nama_barang');
    $jenis = $this->input->post('jenis_barang');
    $satuan = $this->input->post('satuan');
    $barang = "B".$barangno[0]['barangid'];

    $data = array(
        'id_barang' => $barang,
        'nama_barang' => $nama,
        'fk_jenisbarang_id' => $jenis,
        'fk_satuan_id' => $satuan
    );

    $this->Mbarang->input_data($data, 'tbl_barang');
    $this->session->set_flashdata('message','<div
    class="alert alert-warning alert-dismissible"
    role="alert">
        <button type="button" class="close" data-
    dismiss="alert" aria-label="Close"><span aria-
    hidden="true">&times;</span></button>
        Data Berhasil Ditambahkan
    </div>');
    redirect('Barang-view');
}
```

Gambar 4.18 Source Code Router Form Tambah Barang

Kemudian untuk tahap selanjutnya akan dibuat flow graph berdasarkan kode program route form login seperti pada gambar di bawah ini.



Gambar 4.19 Flowgraph Form Tambah Data Barang

Berdasarkan flow graph pada Gambar diatas diketahui bahwa jumlah edge (E) = 7 yang merupakan garis yang menghubungkan node, jumlah node (N) = 7 yang merupakan lingkaran yang menggambarkan suatu aktivitas, jumlah predicate (P) = 1 yang merupakan node bercabang, dan jumlah region (R) = 2 yang menandakan suatu area dalam flow graph, yang dapat dilihat dengan simbol R1 hingga R2 pada Gambar diatas, sehingga jika dimasukkan ke dalam rumus perhitungan cyclomatic complexity dan juga bantuan graph matrix pada Tabel 4.5 dan Tabel 4.6. maka akan menghasilkan sebagai berikut.

Tabel 4. 5 Tabel Cyclomatic Complexity Router Form Tambah Barang

$V(G) = E - N + 2$	$V(G) = P + 1$	$V(G) = R$
$V(G) = 7 - 7 + 2$	$V(G) = 1 + 1$	$V(G) = 2$
$V(G) = 2$	$V(G) = 2$	$V(G) = 2$

Tabel 4.6 *Graph Matrix Router Form Tambah Barang*

Note	1	2	3	4	5	6	7	Hasil
1		1						0
2			1					0
3				1				0
4					1			0
5						1	1	1
6								0
7								0
Jumlah								2

Berdasarkan dari hasil perhitungan *cyclomatic complexity* pada tabel 4.5 dan *graph matrix* pada tabel 4.6, didapatkan jumlah hasil *independent path* untuk *form login* yaitu 2 dengan jalur *independent path* sebagai berikut:

- Jalur 1 : 1-2-3-4-5-6 (Tambah barang berhasil)
- Jalur 2 : 1-2-3-4-5-7-2 (Tambah barang gagal)

Berdasarkan hasil pengujian yang dilakukan, diketahui bahwa route form *login* dengan jumlah CC 2 memiliki tingkat resiko yang rendah dan prosedur yang sederhana. Setelah menentukan jalur *independent path* maka Langkah selanjutnya yaitu membuat *test case* dan melaksanakan pengujian. Hasil pengujian dapat dilihat pada Tabel 4.7.

Tabel 4.7 Tabel *Test Case Router Form Tambah Barang*

No	Jalur	Keterangan	Hasil yang diharapkan	Hasil Sebenarnya	Sesuai/Tidak
1.	1-2-3-4-5-6	Input data barang	Data berhasil disimpan dan menampilkan pesan “data berhasil ditambahkan”	Berhasil Menambahkan data barang Berhasil menampilkan pesan	Sesuai
2.	1-2-3-4-5-7-2	Validasi data barang	Menampilkan pesan “data berhasil ditambahkan”	Berhasil menampilkan pesan	Sesuai



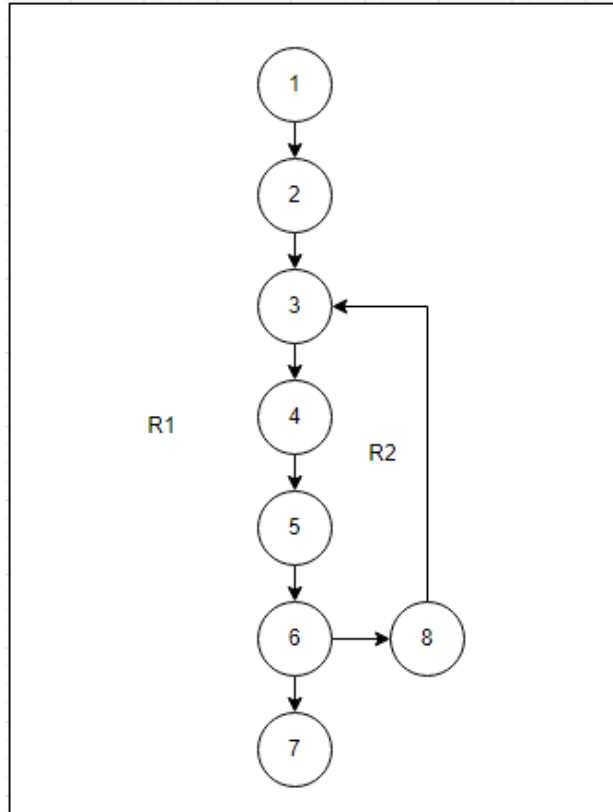
### 3. Pengujian *White Box Router Form* Tambah Transaksi Masuk

#### a. *Source Code Router Form* Tambah Transaksi Masuk

```
function add_view()
{
    $data['barang'] = $this->MTransaksi-
>get_master_toobject("vbarang","pk_barang_id","kocak","n
ama_barang","Select","","");
    $this->load->view('templates/head/tabel');
    $this->load->view('templates/sidebar');
    $this->load->view('templates/topbar');
    $this->load-
>view('master/transaksi_masuk/vaddtransaksimasuk',
$data);
    $this->load->view('templates/footer/tabel');
}
function save_data()
{ //data diri
  // print_r($barangno);
  $tanggal = $this->input->post('tanggal');
  $idbarang = $this->input->post('id_barang');
  $jumlahmasuk = $this->input->post('jumlah_masuk');
  $data = array(
      'tanggal' => $tanggal,
      'id_barang' => $idbarang,
      'jumlah_masuk' => $jumlahmasuk
  );
  $this->MTransaksi->input_data($data,
'tbl_transaksi_masuk');
  $this->session->set_flashdata('message','<div
class="alert alert-warning alert-dismissible"
role="alert">
      <button type="button" class="close" data-
dismiss="alert" aria-label="Close"><span aria-
hidden="true">&times;</span></button>
      Data Berhasil Ditambahkan
    </div>');
  redirect('Transaksi-masuk-view');
}
```

Gambar 3.19 *Source Code Router Form* Tambah Transaksi Masuk

Kemudian untuk tahap selanjutnya akan dibuat flow graph berdasarkan kode program route form login seperti pada gambar di bawah ini.



Gambar 3.20 *Flowgraph Router Form Tambah Transaksi Masuk*

Berdasarkan flow graph pada Gambar diatas diketahui bahwa jumlah edge (E) = 8 yang merupakan garis yang menghubungkan node, jumlah node (N) = 8 yang merupakan lingkaran yang menggambarkan suatu aktivitas, jumlah predicate (P) = 1 yang merupakan node bercabang, dan jumlah region (R) = 2 yang menandakan suatu area dalam flow graph, yang dapat dilihat dengan simbol R1 hingga R2 pada Gambar diatas, sehingga jika dimasukkan ke dalam rumus perhitungan cyclomatic complexity dan juga bantuan graph matrix pada Tabel 4.5 dan Tabel 4.6. maka akan menghasilkan sebagai berikut.

Tabel 4.8 *Cyclomatic Complexity Router Form Tambah Transaksi Masuk*

$V(G) = E - N + 2$	$V(G) = P + 1$	$V(G) = R$
$V(G) = 7 - 7 + 2$	$V(G) = 1 + 1$	$V(G) = 2$
$V(G) = 2$	$V(G) = 2$	$V(G) = 2$

Tabel 4.9 *Graph Matrix Router Form* Tambah Transaksi Masuk

Note	1	2	3	4	5	6	7	8	Hasil
1		1							0
2			1						0
3				1					0
4					1				0
5						1			0
6							1	1	1
7									0
8									0
Jumlah									2

Berdasarkan dari hasil perhitungan *cyclomatic complexity* pada tabel 4.8 dan *graph matrix* pada tabel 4.9, didapatkan jumlah hasil *independent path* untuk *form login* yaitu 2 dengan jalur *independent path* sebagai berikut:

- Jalur 1 : 1-2-3-4-5-6-7 (Tambah transaksi berhasil)
- Jalur 2 : 1-2-3-4-5-6-8-3 (Tambah transaksi gagal)

Berdasarkan hasil pengujian yang dilakukan, diketahui bahwa route form *login* dengan jumlah CC 2 memiliki tingkat resiko yang rendah dan prosedur yang sederhana.

## **BAB V PENUTUP**

### **5.1 Kesimpulan**

Berdasarkan hasil penelitian, analisis, dan pembahasan pada bab-bab sebelumnya yang telah dilakukan tentang "Perancangan Aplikasi Inventory Bahan Logistik Rumah Sakit Umum Daerah Sawerigading Palopo Berbasis Web Menggunakan Metode Extreme Programming" maka dapat di ambil kesimpulan sebagai berikut :

1. Penerapan metode *Extreme Programming* ini menghasilkan sistem *inventory* barang untuk Rumah Sakit Umum Daerah Sawerigading Palopo yang memberikan dampak positif karena, sistem ini dapat mempermudah admin dalam mengelola data *inventory* barang / bahan logistik, dan memonitoring data khususnya data *inventory* barang pada Bagian Logistik Rumah Sakit Umum Daerah Sawerigading Palopo karena sistem ini saling terintegrasi antara data barang masuk, data stok, dan data barang keluar. Sehingga admin dapat dengan mudah mengelola data barang masuk, data stok, data barang keluar dengan hanya menginputkannya langsung ke dalam sistem. Hal ini dapat mengurangi kesalahan dalam mengelola data seperti tidak adanya data yang redundansi, tidak ada data stok kosong karena terdapat pemberitahuan minimum stok yang tersedia, tidak ada barang yang menumpuk, dan waktu menjadi lebih efisien.
2. Sistem ini dapat mempermudah bendahara barang dalam memonitoring data laporan barang yang tersedia. Tidak perlu lagi menunggu admin menyetorkan laporan, pimpinan dapat melihat laporan data stok, barang masuk dan barang keluar di dalam sistem sehingga laporan yang dihasilkan dapat digunakan oleh bendahara barang sebagai referensi pengambilan keputusan.

### **5.2 Saran**

Berdasarkan kesimpulan yang telah diuraikan, saran untuk pengembangan dan implementasi aplikasi *inventory* berbasis web di RSUD Sawerigading Palopo adalah Saran untuk penelitian ini adalah :

1. untuk memperhatikan secara cermat integrasi antara kebutuhan praktis rumah sakit dan fitur yang disediakan oleh aplikasi web yang dirancang. Penting untuk melibatkan pemangku kepentingan dari RSUD secara aktif selama pengembangan aplikasi, serta melakukan uji coba yang menyeluruh untuk

memastikan bahwa aplikasi dapat berfungsi dengan baik dalam lingkungan operasional sehari-hari. Selain itu, penting juga untuk memberikan pelatihan yang memadai kepada staf RSUD dalam penggunaan aplikasi agar mereka dapat memanfaatkannya secara optimal untuk pengelolaan inventory yang efektif dan efisien.

2. Desain website yang diharapkan dapat terus dikembangkan atau diperbaharui sesuai dengan kebutuhan yang ada.

## DAFTAR PUSTAKA

- Amdi Rizal, M., Ahmad, I., Aftirah, N., & Lestari, W. (2022). Aplikasi Inventory Persediaan Barang Berbasis Web Menggunakan Metode Extreme Programming (Studi Kasus : Esha 2 Cell). *Jl. ZA. Pagar Alam*, 3(2), 2774–5384.
- Firdaus, T. (2021). Perancangan Aplikasi Inventory Barang Pada Puskesmas Balida Berbasis Web. *jurnal Proceeding SENDIU*, ISBN: 978-979-3649-72-6, 978–979.
- Hanadhito Riswantoro. (2019). Perancangan Prosedur Pengeluaran Kas Pada Mini Market Syar'e Mart. *Perancangan Sistem Informasi Terpadu Pemerintah Daerah Kabupaten Paser*, 53(9), 1689–1699.
- Harike, M. H. (2019). Aplikasi E-voting Pemilihan OSIS pada SMA NEGERI 11 Luwu. *Seminar Nasional Teknologi Informasi dan Komputer*, 175–183.
- Ii, B. A. B., & Pustaka, T. (2002). *BAB II Tinjauan Pustaka BAB II TINJAUAN PUSTAKA 2.1*. 1–64.
- Indra Sukma. (2022). *BAB II Tinjauan Pustaka BAB II TINJAUAN PUSTAKA 2.1*. 1–64. *Gastronomía ecuatoriana y turismo local.*, 1(69), 5–24.
- Junus, D., & Ambo, Z. (2020). Profil Perencanaan Logistik Non Medik Rumah Sakit Haji Makassar Overview of Non Medical Logistic Planning of the Haji Hospital, Makassar. *Jurnal Penelitian Kesehatan Pelamonia Indonesia*, 03(2009), 63–67.
- Kesuma, C., Handayani, V. R., Damayanti, O., Kampus, T. K., Banyumas, K., Bina, U., & Informatika, S. (2022). Sistem Informasi Inventory Alat Kesehatan Menggunakan Metode Fefo. *Journal Speed-Sentra Penelitian Engineering dan Edukasi*, 14(1), 63–68.
- Limbong, T., & Sianipar, J. F. (2022). Sistem Informasi Inventory Aset dan Logistik Rumah Sakit Bina Kasih Berbasis Website dengan Menggunakan Metode Rapid Application Development (RAD). *KAKIFIKOM (Kumpulan Artikel Karya ...*, 04(02), 63–79.  
<http://www.ejournal.ust.ac.id/index.php/KAKIFIKOM/article/view/2306%0Ahttp://www.ejournal.ust.ac.id/index.php/KAKIFIKOM/article/view/2306/2009>
- Malik, A., Hasanuddin, M., & Kurniawan, F. (2023). Analisis dan Perancangan Sistem Informasi Desa Berbasis Web Menggunakan OpenSID dengan Metode Prototyping. *Dewantara Journal of Technology*, 4(2), 6–11.
- Malik, A., Nirsal, N., Bantun, S., & Sari, J. Y. (2023). Optimalisasi Rute Pengiriman

- Untuk E-Commerce: Aplikasi Kurir Berbasis Web Menggunakan Algoritma Simple Hill Climbing. *SemanTIK: Teknik Informasi*, 9(2), 157. <https://doi.org/10.55679/semantik.v9i2.45346>
- Mawardi, C. (2020). *RSUD Sawerigading Palopo*. <https://makassar.tribunnews.com>. <https://rsudswg.palopokota.go.id/blog/page/sejarah>
- Mokalu, G., Maramis, F. R. R. M., & Tucunan, A. A. T. T. (2019). Sistem Penyimpanan Dan Pendistribusian Logistik Non Medis Di Rumah Sakit Jiwa Prof. Dr. V.L. Ratumbuysang Provinsi Sulawesi Utara. *Sistem Penyimpanan Dan Pendistribusian Logistik Non Medis Di Rumah Sakit Jiwa Prof. Dr. V.L. Ratumbuysang Provinsi Sulawesi Utara*, 8(7), 85–92.
- Pengenalan perancangan basis data*. (2023). Sistem Informasi Desain Garfis. <http://bj.imwi.ac.id/2019/01/tahapan-perancangan-basis-data.html>
- Prasetio, N. (2021). Sistem Informasi Penyewaan Kendaraan Berbasis Web (Studi Kasus Chandra Trans Bali). *Jurnal Ilmiah Methonomi*, 3(2), 28–29.
- PT Cloud Hosting, I. (2019). Mengenal Apa itu Framework CodeIgniter \_ IDCloudHost. In *ID Cloud House*. <https://idcloudhost.com/panduan/mengenal-apa-itu-framework-codeigniter/>
- Ridwan, M., Sinaga, T. H., & Elsera, M. (2022). Penerapan Framework Codeigniter Dalam Perancangan Aplikasi Manajemen Iuran Perumahan Griya Mandiri. *Djtechno: Jurnal Teknologi Informasi*, 3(1), 49–58. <https://doi.org/10.46576/djtechno.v3i1.2196>
- Riski Qisthiano, M. (2023). Perancangan Sistem Informasi Inventaris pada CV Cemerlang Komputer dengan Metode Extreme Programming. *Dinamika Informatika : Jurnal Ilmiah Teknologi Informasi*, 15(1), 1–10.
- Sakit, R., Daerah, U., & Palopo, S. K. (2018). *Rumah Sakit Umum Daerah (RSUD) Sawerigading Kota Palopo*. 1–119.
- Septiani, N. A., & Habibie, F. Y. (2022). Penggunaan Metode Extreme Programming Pada Perancangan Sistem Informasi Pelayanan Publik. *Jurnal Sistem Komputer dan Informatika (JSON)*, 3(3), 341. <https://doi.org/10.30865/json.v3i3.3931>
- Sie, J. B. L., Izmy Alwiah Musdar, & Syamsul Bahri. (2022). Pengujian White Box Testing Terhadap Website Room Menggunakan Teknik Basis Path. *KHARISMA Tech*, 17(2), 45–57. <https://doi.org/10.55645/kharismatech.v17i2.235>
- Simpony, B. K., Rizaldy, S. I. P., Suleman, & Widodo, P. (2022). *SISTEM INFORMASI*

- LOGISTIK MENGGUNAKAN METODE PROTOTYPE. 10(2), 90–98.*
- Sterpi, S., & Validation, K. F. C. (2018). *DAFTAR PUSTAKA Agarwal, A., Xie, B., Vovsha, I., Rambow, O. dan Passonneau, R., 2011. Sentiment Analysis of Twitter Data, In.*
- Suryaningrum, F. (2022). *Mengenal PHP – Sejarah, Istilah, Fungsi, & Cara Kerjanya.* Askara Data Digital. <https://aksaradata.id/blog/php-adalah/>
- Syafnidawaty. (2020). *White Box Testing.* Universitas Raharja. <https://raharja.ac.id/2020/10/19/white-box-testing/>
- Widodo, B. K. S. S. I. P. R. S. P. (2022). *SISTEM INFORMASI LOGISTIK MENGGUNAKAN METODE PROTOTYPE. 10(2), 90–98.*
- Y J Solissa, F Putra, A N Putri, & S R C Nursari. (2023). Pengujian White Box Berbasis Path pada *Form* Daftar Jobstreet.co.id. *KONSTELASI: Konvergensi Teknologi dan Sistem Informasi* , 3(2), 353–362.
- Yudhistira. (2024). *Mengenal Apa itu Database serta Fungsi dan Jenisnya.* Blog Bhinneka. <https://www.bhinneka.com/blog/database-adalah/#>



## LAMPIRAN

### 1. Login.Model ke PHP

```
<?php

class Login_model extends CI_Model{

    public function cek_login($user, $pass)
    {
        $this->db->where('username', $user);
        $this->db->where('password', $pass);
        $query = $this->db->get('user');
        return $query;
    }

    public function getLoginData($user, $pass){
        $u = $user;
        $p = md5($pass);

        $query_cekLogin = $this->db->get_where('user',
array('username'=> $u, 'password' => $p));

        if (count($query_cekLogin->result()) > 0) {
            foreach ($query_cekLogin->result() as $qck) {
                foreach ($query_cekLogin->result() as $ck) {
                    $sess_data ['logged_in'] = TRUE;
                    $sess_data ['username'] = $ck->username;
                    $sess_data ['password'] = $ck->password;
                    $sess_data ['level'] = $ck->level;
                    $sess_data ['user_id'] = $ck->user_id;
                    $this->session->set_userdata($sess_data);
                }
                redirect('dashboard');
            }
        }else{
            $this->session->set_flashdata('Pesan','<div
class="alert alert-danger alert-dismissible fade show"
role="alert"> Username atau Password Salah! <button
type="button" class="close" data-dismiss="alert" aria-
label="Close" <span aria-hidden="true">&times;</span>
</button> </div>');
            redirect('auth/login');
        }
    }else{
        $this->session->set_flashdata('Pesan','<div class="alert
alert-danger alert-dismissible fade show" role="alert"> Username atau
Password Salah! <button type="button" class="close" data-
```

```

dismiss="alert" aria-label="Close" <span aria-
hidden="true">&times;</span> </button> </div>');
        redirect('auth/login');
    }
}
}
}

```

## 2. Barang.Model.php

```

<?php
/**
 *
 */
class Mbarang extends CI_Model
{
    function data_barang()
    {
        $this->db->select('*');
        $this->db->from('vbarang');
        $query = $this->db->get();
        return $query->result();
    }
    function
get_master_toobject($tablename,$value,$display,$orderby,$objtype,$default
tvalue,$condition,$columns='*',$encrypt=false) {
        $enc_value='';
        if ($condition==''){
            $query=$this->db->query("SELECT $columns FROM ". $tablename
            ." order by ". $orderby);
        }else{
            $query=$this->db->query("SELECT $columns FROM ". $tablename
            ." where ". $condition ." order by ". $orderby);
        }
        if($query->num_rows()>0)
        {
            $fetch_data=$query->result();
            $data = array();
            foreach($fetch_data as $row)
            {
                if ($encrypt==false){
                    $enc_value=$row->$value;
                }else{
                    $enc_value=encrypt($row->$value);
                }
                if ($objtype=="Select"){
                    if ($defaultvalue!="" && $row-
>$value==$defaultvalue){
                        $data[] = "<option value=". $enc_value ."

```

```

selected>".$row->$display."</option>";
        }else{
            $data[] = "<option value=". $enc_value
.">".$row->$display."</option>";
        }
    }
    }
    return $data;
}
else
{
    return FALSE;
}
}

function generatenobarang()
{
    $query = $this->db->query("SELECT generate_barang_no() as
barangid");
    return $query->result_array();
}
function getbarangbyid($id)
{
    $query = $this->db->query("SELECT * from tbl_barang where
pk_barang_id = $id");

    if ($query->num_rows() > 0) {
        return $query->result();
    } else {
        return FALSE;
    }
}
function input_data($data,$table){
    $this->db->insert($table,$data);
}
function update_data($where,$data,$table){
    $this->db->where($where);
    $this->db->update($table,$data);
}
function hapus_data($where,$table){
    $this->db->where($where);
    $this->db->delete($table);
}
}
}

```

### 3. Model – Jenis Barang.php

```

<?php
/**
 *
 */
class MJenisbarang extends CI_Model
{
    function data_jenisbarang()
    {
        $this->db->select('*');
        $this->db->from('tbl_jenisbarang');
        $query = $this->db->get();
        return $query->result();
    }
    function
get_master_toobject($tablename,$value,$display,$orderby,$objtype,$default
value,$condition,$columns='*',$encrypt=false) {
        $enc_value='';
        if ($condition==''){
            $query=$this->db->query("SELECT $columns FROM ". $tablename ."
order by ". $orderby);
        }else{
            $query=$this->db->query("SELECT $columns FROM ". $tablename ."
where ". $condition ." order by ". $orderby);
        }
        if($query->num_rows()>0)
        {
            $fetch_data=$query->result();
            $data = array();
            foreach($fetch_data as $row)
            {
                if ($encrypt==false){
                    $enc_value=$row->$value;
                }else{
                    $enc_value=encrypt($row->$value);
                }
                if ($objtype=="Select"){
                    if ($defaultvalue!="" && $row-
>$value==$defaultvalue){
                        $data[] = "<option value=". $enc_value ."
selected>".$row->$display."</option>";
                    }else{
                        $data[] = "<option value=". $enc_value .">".$row-
>$display."</option>";
                    }
                }
            }
        }
        return $data;
    }
}

```

```

        else
        {
            return FALSE;
        }
    }

    function generatenobarang()
    {
        $query = $this->db->query("SELECT generate_barang_no() as
barangid");
        return $query->result_array();
    }
function input_data($data,$table){
    $this->db->insert($table,$data);
}
function hapus_data($where,$table){
    $aksi = $this->db->where($where);
    $aksi = $this->db->delete($table);
}
function getjenisbarangbyid($id)
{
    $query = $this->db->query("SELECT * from tbl_jenisbarang where
pk_jenisbarang_id = $id");
    if ($query->num_rows() > 0) {
        return $query->result();
    } else {
        return FALSE;
    }
}
function update_data($where,$data,$table){
    $this->db->where($where);
    $this->db->update($table,$data);
}
}

```

#### 4. Model – Laporan.php

```

<?php
/**
 *
 */
class MLaporan extends CI_Model
{
    function graph()
    {
        $data = $this->db->query("SELECT tanggal,MONTH(tanggal) AS bular
SUM(jumlah_masuk) AS jumlah_masuk FROM tbl_transaksi_masuk GROUP BY

```

```

bulan");
    return $data->result();
}
function graph_keluar()
{
    $data = $this->db->query("SELECT tanggal,MONTH(tanggal) AS bulan,
SUM(jumlah_keluar) AS jumlah_keluar FROM tbl_transaksi_keluar where
status=1 GROUP BY bulan");
    return $data->result();
}
function show_barang()
{
    $barang = $this->db->query("SELECT a.*,
b.nama_barang,b.satuan_barang FROM tbl_transaksi_masuk a left join
vbarang b on a.id_barang = b.pk_barang_id");
    return $barang->result();
}
function show_barang_keluar()
{
    $barang = $this->db->query("SELECT a.*,
b.nama_barang,b.satuan_barang FROM tbl_transaksi_keluar a left join
vbarang b on a.id_barang = b.pk_barang_id where status =1");
    return $barang->result();
}
function data_barang($dari, $sampai)
{
    if($dari == '' && $sampai == ''){
        $barang = $this->db->query("SELECT a.*,
b.nama_barang,b.satuan_barang FROM tbl_transaksi_masuk a left join
vbarang b on a.id_barang = b.pk_barang_id");
    }else
    {
        $barang = $this->db->query("SELECT a.*,
b.nama_barang,b.satuan_barang FROM tbl_transaksi_masuk a left join
vbarang b on a.id_barang = b.pk_barang_id
WHERE tanggal >= '$dari' AND tanggal <= '$sampai' ");
    }
    return $barang->result();
}
function data_barang_keluar($dari, $sampai)
{
    if($dari == '' && $sampai == ''){
        $barang = $this->db->query("SELECT a.*,
b.nama_barang,b.satuan_barang FROM tbl_transaksi_keluar a left join
vbarang b on a.id_barang = b.pk_barang_id
WHERE STATUS=1");
    }else
    {

```

```

        $barang = $this->db->query("SELECT a.*,
b.nama_barang,b.satuan_barang FROM tbl_transaksi_keluar a left join
vbarang b on a.id_barang = b.pk_barang_id
        WHERE tanggal >= '$dari' AND tanggal <= '$sampai' and status=1
");
    }
    return $barang->result();
}
}

```

## 5. Model- Satuan.php

```

<?php
/**
 *
 */
class MSatuan extends CI_Model
{
    function data_satuan()
    {
        $this->db->select('*');
        $this->db->from('tbl_satuan');
        $query = $this->db->get();
        return $query->result();
    }
    function
get_master_toobject($tablename,$value,$display,$orderby,$objtype
,$defaultvalue,$condition,$columns='*',$encrypt=false) {
        $enc_value='';
        if ($condition==''){
            $query=$this->db->query("SELECT $columns FROM ".
$tablename ." order by ". $orderby);
        }else{
            $query=$this->db->query("SELECT $columns FROM ".
$tablename ." where ". $condition ." order by ". $orderby);
        }
        if($query->num_rows()>0)
        {
            $fetch_data=$query->result();
            $data = array();
            foreach($fetch_data as $row)
            {
                if ($encrypt==false){
                    $enc_value=$row->$value;
                }else{
                    $enc_value=encrypt($row->$value);
                }
            }
            if ($objtype=="Select"){

```

```

        if ($defaultvalue!="" && $row-
>$value==$defaultvalue){
            $data[] = "<option value=". $enc_value
." selected>".$row->$display."</option>";
        }else{
            $data[] = "<option value=". $enc_value
.">".$row->$display."</option>";
        }
    }
    }
    return $data;
}
else
{
    return FALSE;
}
}

function generatenobarang()
{
    $query = $this->db->query("SELECT generate_barang_no()
as barangid");
    return $query->result_array();
}
function input_data($data,$table){
    $this->db->insert($table,$data);
}
function hapus_data($where,$table){
    $aksi = $this->db->where($where);
    $aksi = $this->db->delete($table);
}
function getsatuanbyid($id)
{
    $query = $this->db->query("SELECT * from tbl_satuan
where pk_satuan_id = $id");
    if ($query->num_rows() > 0) {
        return $query->result();
    } else {
        return FALSE;
    }
}
function update_data($where,$data,$table){
    $this->db->where($where);
    $this->db->update($table,$data);
}
}
}

```

## 6. Model - Transaksi.php



```

<?php
/**
 *
 */
class MTransaksi extends CI_Model
{
    function transaksi_masuk()
    {
        $query = $this->db->query("SELECT tt.*,
tb.nama_barang, tb.id_barang AS nomorbarang FROM
tbl_transaksi_masuk AS tt
LEFT JOIN tbl_barang tb
ON tt.id_barang = tb.pk_barang_id");
        return $query->result();
    }
    function transaksi_keluar()
    {
        $query = $this->db->query("SELECT tt.*,
tb.nama_barang, tb.id_barang AS nomorbarang FROM
tbl_transaksi_keluar AS tt
LEFT JOIN tbl_barang tb
ON tt.id_barang = tb.pk_barang_id");
        return $query->result();
    }
    function
get_master_toobject($tablename,$value,$display,$orderby,$objt
ype,$defaultvalue,$condition,$columns='',$encrypt=false) {
        $enc_value='';
        if ($condition==''){
            $query=$this->db->query("SELECT $columns FROM ".
$tablename ." order by ". $orderby);
        }else{
            $query=$this->db->query("SELECT $columns FROM ".
$tablename ." where ". $condition ." order by ". $orderby);
        }
        if($query->num_rows()>0)
        {
            $fetch_data=$query->result();
            $data = array();
            foreach($fetch_data as $row)
            {
                if ($encrypt==false){
                    $enc_value=$row->$value;
                }else{
                    $enc_value=encrypt($row->$value);
                }
                if ($objtype=="Select"){
                    if ($defaultvalue!="" && $row-

```

```

>$value==$defaultvalue){
    $data[] = "<option value=" .
$enc_value ." selected>".$row-
>$display."</option>";
    }else{
        $data[] = "<option value=" .
$enc_value .">".$row->$display."</option>";
    }
}
return $data;
}
else
{
    return FALSE;
}
}

function generatenobarang()
{
    $query = $this->db->query("SELECT
generate_barang_no() as barangid");
    return $query->result_array();
}
function input_data($data,$table){
    $this->db->insert($table,$data);
}
function hapus_data($where,$table){
    $aksi = $this->db->where($where);
    $aksi = $this->db->delete($table);
}
function accept_data($id)
{
    $accept = $this->db->query("
UPDATE tbl_transaksi_keluar
SET status = 1
WHERE pk_transaksi_keluar_id = $id");
    return $accept;
}
function reject_data($id)
{
    $reject = $this->db->query("
UPDATE tbl_transaksi_keluar
SET status = 2
WHERE pk_transaksi_keluar_id = $id");
    return $reject;
}
}

```

```
}
```

## 7. Model – User.php

```
<?php
defined('BASEPATH') OR exit('No direct script access
allowed');
class User_m extends CI_Model {
    public function get($user_id)
    {
        $this->db->from('user');
        if($user_id != null){
            $this->db->where('user_id', $user_id);
        }
        $query = $this->db->get();
        return $query;
    }
    public function total_jml_user()
    {
        $query = $this->db->get('user');
        if ($query->num_rows()>0)
        {
            return $query->num_rows();
        }
        else
        {
            return 0;
        }
    }
    public function insert($data){
        $this->db->insert('user',$data);
    }
    public function ambil_data($user_id)
    {
        $this->db->where('username', $user_id);
        return $this->db->get('user')->row();
    }
    public function ambil_data_id($user_id)
    {
        $this->db->where('user_id', $user_id);
        return $this->db->get('user')->row();
    }
    public function tampil_data()
    {
        return $this->db->get('user');
    }
    public function add($post)
    {
```

```

$params['name']      = $post['fullname'];
$params['username'] = $post['username'];
$params['password'] = sha1($post['password']);
$params['address']  = $post['address'] != "" ?
$post['address'] : null;
$params['level']    = $post['level'];
$this->db->insert('user', $params);
}
public function edit_data($where,$table){
    return $this->db->get_where($table, $where);
}
public function update_data($where,$data,$table){
    $this->db->where($where);
    $this->db->update($table,$data);
}
public function hapus_data($where,$table){
    $this->db->where($where);
    $this->db->delete($table);
}

function data_barang()
{
    $this->db->select('*');
    $this->db->from('vbarang');
    $query = $this->db->get();
    return $query->result();
}
function totalbarang()
{
    $query = $this->db->query('SELECT * FROM vbarang');
    return $query->num_rows();
}
function barangmasuk()
{
    $query = $this->db->query('SELECT * FROM
tbl_transaksi_masuk');
    return $query->num_rows();
}
function barangkeluar()
{
    $query = $this->db->query('SELECT * FROM
tbl_transaksi_keluar where status=1');
    return $query->num_rows();
}
function user()
{
    $query = $this->db->query("SELECT * FROM user where
blokir = 'N' ");
}

```

```

        return $query->num_rows();
    }
}

```

## 8. View – Tambah Barang.php

```

<head>
    <link href="https://cdn.jsdelivr.net/npm/select2@4.1.0-
beta.1/dist/css/select2.min.css" rel="stylesheet" />
    <script src="https://cdn.jsdelivr.net/npm/select2@4.1.0-
beta.1/dist/js/select2.min.js"></script>
</head>
<!-- Begin Page Content -->
<div class="container-fluid">
    <!-- Page Heading -->
    <h1 class="h3 mb-4 text-gray-800">Tambah Barang</h1>
    <div class="row">
        <div class="col-lg-6">
            <!-- Circle Buttons -->
            <div class="card shadow mb-12">
                <div class="card-body">
                    <form method="post"
action="<?=base_url('Barang-save')?>" autocomplete="off">
                        <div class="form-group col-lg-12">
                            <label>Nama Barang *</label>
                            <input type="text"
name="nama_barang" class="form-control" name="" id="">
                        </div>
                        <div class="form-group col-lg-12">
                            <label>Jenis Barang *</label>
                            <select name="jenis_barang"
class="form-control js-single" required autofocus>
                                <?php foreach($jenisbarang as
$row)
                                    {
                                        echo($row);
                                    }
                                ?>
                            </select>
                        </div>
                        <div class="form-group col-lg-12">
                            <label>Satuan *</label>
                            <select name="satuan" class="form-
control js-single" required autofocus>
                                <?php foreach($satuan as $row)
                                    {
                                        echo($row);
                                    }
                                ?>
                            </select>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>

```

```

        ?>
        </select>
    </div>
    <div class="form-group col-lg-6">
        <button type="submit" class="btn
btn-primary btn-flat">
            <i class="fa fa-pencil"></i>
Simpan</button>
            <button type="reset" class="btn
btn-info">Reset</button>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- /.container-fluid -->
</div>
<!-- End of Main Content -->
<script>
    $('<!-- .js-single').select2({
        placeholder: 'Select an option'
    });
</script>

```

## 9. View – Edit Barang.php

```

<head>
    <link href="https://cdn.jsdelivr.net/npm/select2@4.1.0-
beta.1/dist/css/select2.min.css" rel="stylesheet" />
    <script src="https://cdn.jsdelivr.net/npm/select2@4.1.0-
beta.1/dist/js/select2.min.js"></script>
</head>
<!-- Begin Page Content -->
<div class="container-fluid">
    <!-- Page Heading -->
    <h1 class="h3 mb-4 text-gray-800">Edit Barang</h1>
    <div class="row">
        <div class="col-lg-6">
            <!-- Circle Buttons -->
            <div class="card shadow mb-12">
                <div class="card-body">
                    <form method="post"
action="<?==base_url('Barang-update')?>" autocomplete="off":
                    <input type="hidden" name="barang_id"

```

```

value="<?php echo $barang->pk_barang_id ?>">
        <div class="form-group col-lg-12">
            <label>Nama Barang *</label>
            <input type="text"
name="nama_barang" class="form-control" name="" id=""
value="<?php echo $barang->nama_barang ?>";>
            </div>
            <div class="form-group col-lg-12">
                <label>Jenis Barang *</label>
                <select name="jenis_barang"
class="form-control js-single" required autofocus>
                    <?php foreach($jenisbarang as
$row)
                        {
                            echo($row);
                        }
                    ?>
                </select>
            </div>
            <div class="form-group col-lg-12">
                <label>Satuan *</label>
                <select name="satuan"
class="form-control js-single" required autofocus>
                    <?php foreach($satuan as $row)
                        {
                            echo($row);
                        }
                    ?>
                </select>
            </div>
            <div class="form-group col-lg-6">
                <button type="submit" class="btn
btn-primary btn-flat">
                    <i class="fa fa-pencil"></i>
                Simpan</button>
                <button type="reset" class="btn
btn-info">Reset</button>
            </div>
        </div>
    </div>
</div>
</div>
</div>
<!-- /.container-fluid -->
</div>

```

```

<!-- End of Main Content -->
<script>
    $(' .js-single').select2({
        placeholder: 'Select an option'
    });
</script>

```

## 10. View – Tambah Jenis Barang.php

```

<head>
    <link href="https://cdn.jsdelivr.net/npm/select2@4.1.0-
beta.1/dist/css/select2.min.css" rel="stylesheet" />
    <script src="https://cdn.jsdelivr.net/npm/select2@4.1.0-
beta.1/dist/js/select2.min.js"></script>
</head>
<!-- Begin Page Content -->
<div class="container-fluid">
    <!-- Page Heading -->
    <h1 class="h3 mb-4 text-gray-800">Tambah Jenis
Barang</h1>
    <div class="row">
        <div class="col-lg-6">
            <!-- Circle Buttons -->
            <div class="card shadow mb-12">
                <div class="card-body">
                    <form method="post"
action="<?=base_url('Jenis-barang-save')?>"
autocomplete="off">
                        <div class="form-group col-lg-12">
                            <label>Nama Jenis Barang
*</label>
                            <input type="text"
name="jenis_barang" class="form-control" name="" id="">
                            </div>
                            <div class="form-group col-lg-6">
                                <button type="submit" class="btn
btn-primary btn-flat">
                                    <i class="fa fa-pencil"></i>
Simpan</button>
                                <button type="reset" class="btn
btn-info">Reset</button>
                            </div>
                        </div>
                    </div>
                </div>
            </div>

```



```

        </div>
</div>
</div>
</div>
<!-- /.container-fluid -->
</div>
<!-- End of Main Content -->
<script>
    $('<code>.js-single</code>').select2({
        placeholder: 'Select an option'
    });
</script>

```

## 11. View – Edti Jenis Barang.php

```

<head>
    <link href="https://cdn.jsdelivrivr.net/npm/select2@4.1.0-
beta.1/dist/css/select2.min.css" rel="stylesheet" />
    <script src="https://cdn.jsdelivrivr.net/npm/select2@4.1.0-
beta.1/dist/js/select2.min.js"></script>
</head>
<!-- Begin Page Content -->
<div class="container-fluid">
    <!-- Page Heading -->
    <h1 class="h3 mb-4 text-gray-800">Edit Jenis Barang</h1>
    <div class="row">
        <div class="col-lg-6">
            <!-- Circle Buttons -->
            <div class="card shadow mb-12">
                <div class="card-body">
                    <form method="post"
action="<code>=base_url('Jenis-barang-update-save')&lt;/code&gt;?"
autocomplete="off"&gt;
                        &lt;input type="hidden"
name="jenis_barang_id" value="&lt;code&gt;<?=php echo $jenisbarang-
&gt;pk_jenisbarang_id ?&gt;&lt;/code&gt;"
                        &lt;div class="form-group col-lg-12"&gt;
                            &lt;label&gt;Nama Jenis Barang
*&lt;/label&gt;
                            &lt;input type="text"
name="jenis_barang" class="form-control" value="&lt;code&gt;<?=php echo
$jenisbarang-&gt;jenis_barang ?&gt;&lt;/code&gt;"
                            &lt;/div&gt;
                            &lt;div class="form-group col-lg-6"&gt;
                                &lt;button type="submit" class="
btn-primary btn-flat"&gt;
                                    &lt;i class="fa fa-pencil"&gt;&lt;/i&gt;
</pre

```

```

        Simpan</button>
                <button type="reset" class="btn
btn-info">Reset</button>
            </div>
        </div>
    </div>
</div>
</div>
</div>
<!-- /.container-fluid -->
</div>
<!-- End of Main Content -->
<script>
    $('<!-- .js-single').select2({
        placeholder: 'Select an option'
    });
</script>

```

## 12. View – Tambah Satuan Barang.php

```

<head>

    <link href="https://cdn.jsdelivr.net/npm/select2@4.1.0-
beta.1/dist/css/select2.min.css" rel="stylesheet" />
    <script src="https://cdn.jsdelivr.net/npm/select2@4.1.0-
beta.1/dist/js/select2.min.js"></script>
</head>
<!-- Begin Page Content -->
<div class="container-fluid">

    <!-- Page Heading -->
    <h1 class="h3 mb-4 text-gray-800">Tambah Satuan</h1>
    <div class="row">
        <div class="col-lg-6">
            <!-- Circle Buttons -->
            <div class="card shadow mb-12">
                <div class="card-body">
                    <form method="post"
action="<?=>base_url('Satuan-save')?>" autocomplete="off">
                        <div class="form-group col-lg-12">
                            <label>Satuan Barang *</label>
                            <input type="text" name="satuan"
class="form-control" name="" id="">
                        </div>
                        <div class="form-group col-lg-6">
                            <button type="submit" class="btn

```



```

class="form-control" value="<?php echo $satuan-
>satuan_barang;?>">
        </div>
        <div class="form-group col-lg-6">
            <button type="submit" class="btn
btn-primary btn-flat">
                <i class="fa fa-pencil"></i>
Simpan</button>
                <button type="reset" class="btn
btn-info">Reset</button>
            </div>
        </div>
    </div>
</div>
</div>
</div>
<!-- /.container-fluid -->
</div>
<!-- End of Main Content -->

```

#### 14. View – Transaksi Barang Masuk.php

```

<head>
    <link href="https://cdn.jsdelivr.net/npm/select2@4.1.0-
beta.1/dist/css/select2.min.css" rel="stylesheet" />
    <script src="https://cdn.jsdelivr.net/npm/select2@4.1.0-
beta.1/dist/js/select2.min.js"></script>
</head>
<!-- Begin Page Content -->
<div class="container-fluid">
    <!-- Page Heading -->
    <h1 class="h3 mb-4 text-gray-800">Tambah Transaksi
Masuk</h1>
    <div class="row">
        <div class="col-lg-6">
            <!-- Circle Buttons -->
            <div class="card shadow mb-12">
                <div class="card-body">
                    <form method="post"
action="<?=base_url('Transaksi-masuk-save')?>"
autocomplete="off">
                        <div class="form-group col-lg-12">
                            <label>Tanggal *</label>
                            <?php
                                $date = date("mm/dd/yyyy");

```

```

        ?>
        <input type="date" name="tanggal"
class="form-control" >
    </div>
    <div class="form-group col-lg-12">
        <label>Nama Barang *</label>
        <select name="id_barang"
class="form-control js-single" required autofocus>
            <?php foreach($barang as
$row)
                {
                    echo($row);
                }
            ?>
        </select>
    </div>
    <div class="form-group col-lg-12">
        <label>Jumlah Barang Masuk
*</label>
        <input type="number"
name="jumlah_masuk" class="form-control" >
    </div>
    <div class="form-group col-lg-6">
        <button type="submit" class="btn
btn-primary btn-flat">
            <i class="fa fa-pencil"></i>
Simpan</button>
        <button type="reset" class="btn
btn-info">Ulang</button>
    </div>
</div>
</div>
</div>
</div>
<!-- /.container-fluid -->
</div>
<!-- End of Main Content -->
<script>
    $('<code>.js-single</code>').select2({
        placeholder: 'Select an option'
    });
</script>

```

#### 15. View – Transaksi Barang Keluar.php

```
<head>
```

```

    <link href="https://cdn.jsdelivr.net/npm/select2@4.1.0-
beta.1/dist/css/select2.min.css" rel="stylesheet" />
    <script src="https://cdn.jsdelivr.net/npm/select2@4.1.0-
beta.1/dist/js/select2.min.js"></script>
</head>
<!-- Begin Page Content -->
<div class="container-fluid">
    <!-- Page Heading -->
    <h1 class="h3 mb-4 text-gray-800">Tambah Transaksi
Keluar</h1>
    <div class="row">
        <div class="col-lg-6">
            <!-- Circle Buttons -->
            <div class="card shadow mb-12">
                <div class="card-body">
                    <form method="post"
action="<?=base_url('Transaksi-keluar-save')?>"
autocomplete="off">
                        <div class="form-group col-lg-12">
                            <label>Tanggal *</label>
                            <?php
                                $date = date("mm/dd/yyyy");
                            ?>
                            <input type="date" name="tanggal"
class="form-control" >
                        </div>
                        <div class="form-group col-lg-12">
                            <label>Nama Barang *</label>
                            <select name="id_barang"
class="form-control js-single" required autofocus>
                                <?php foreach($barang as
$row)
                                    {
                                        echo($row);
                                    }
                                ?>
                            </select>
                        </div>
                        <div class="form-group col-lg-12">
                            <label>Jumlah Barang Keluar
*</label>
                            <input type="number"
name="jumlah_keluar" class="form-control" >
                        </div>
                        <div class="form-group col-lg-6">
                            <button type="submit" class="btn
btn-primary btn-flat">

```

```

                                <i class="fa fa-pencil"></i>
Simpan</button>
                                <button type="reset" class="btn
btn-info">Reset</button>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
<!-- /.container-fluid -->
</div>
<!-- End of Main Content -->

```

## 16. View – Tambah User.php

```

<!-- Begin Page Content -->
    <div class="container-fluid">

        <!-- Page Heading -->
        <h1 class="h3 mb-4 text-gray-800">Tambah User</h1>
        <div class="row">
            <div class="col-lg-6">
                <!-- Circle Buttons -->
                <div class="card shadow mb-4">
                    <div class="card-header py-3">
                        <h6 class="m-0 font-weight-bold text-
primary">Form Add User</h6>
                    </div>
                    <div class="card-body">
                        <form
action="<?=base_url()?>user/tambah_aksi" method="post">
                            <div class="form-group">
                                <label>Username</label>
                                <input type="text" name="username"
class="form-control" required>
                            </div>
                            <div class="form-group">
                                <label>Password</label>
                                <input type="text" name="password"
class="form-control" required>
                            </div>
                            <div class="form-group">
                                <label>Email</label>
                                <input type="email" name="email"
class="form-control" required>

```

```

        </div>
        <div class="form-group">
            <label>Level</label>
            <select name="level" class="form-
control" required>
                <option value="Belum Memilih"
>Pilih</option>
                <option value="ADMIN"
>Admin</option>
                <option value="USER"
>User</option>
            </select>
        </div>
        <div class="form-group">
            <label>Blokir</label>
            <select name="blokir" class="form-
control" required>
                <option value="Belum Memilih"
>Pilih</option>
                <option value="YES"
>Yes</option>
                <option value="NO" >No</option>
            </select>
        </div>
        <div>
            <button type="submit" class="btn
btn-primary btn-flat">
                <i class="fa fa-
pencil"></i> Simpan</button>
            <button type="reset" class="btn
btn-info">Reset</button>
        </div>
    </form>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- /.container-fluid -->
</div>
<!-- End of Main Content -->

```

## 17. View – Edit User.php

```
<!-- Begin Page Content -->
```



```

<div class="container-fluid">
  <!-- Page Heading -->
  <h1 class="h3 mb-4 text-gray-800">Edit User</h1>
  <div class="row">
    <div class="col-lg-6">
      <!-- Circle Buttons -->
      <div class="card shadow mb-4">
        <div class="card-header py-3">
          <h6 class="m-0 font-weight-bold text-
primary">Form Edit User</h6>
        </div>
        <div class="card-body">
          <?php foreach($user as $data) : ?>
            <form method="post" action="<?php echo
base_url('user/update_aksi'); ?>">
              <div class="form-group">
                <label>Username</label>
                <input class="form-control"
type="hidden" name="user_id" value="<?php echo $data->user_id
?>">
                <input type="text" name="username"
class="form-control" value="<?php echo $data->username ?>"
required>
              </div>
              <div class="form-group">
                <label>Password</label>
                <input type="text"
name="password" class="form-control" value="<?php echo $data-
>password ?>" required>
              </div>
              <div class="form-group">
                <label>Email</label>
                <input type="email" name="email"
class="form-control" value="<?php echo $data->email ?>"
required>
              </div>
              <div class="form-group">
                <label>Level</label>
                <select name="level" class="form-
control" required>
                  <option value="<?php echo $data-
>level ?>" ><?php echo $data->level ?></option>
                  <option value="ADMIN"
>Admin</option>
                  <option value="USER"
>User</option>
                </select>
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

        <div class="form-group">
            <label>Blokir</label>
            <select name="blokir" class="form-
control" required>
                <option value="<?php echo $data-
>blokir ?>" ><?php echo $data->blokir ?></option>
                <option value="Yes" >Yes</option>
                <option value="No" >No</option>
            </select>
        </div>
        <div>
            <button type="submit" class="btn
btn-primary btn-flat">
                <i class="fa fa-
pencil"></i> Edit</button>
            <button type="reset" class="btn
btn-info">Reset</button>
        </div>
    </form>
    <?php endforeach; ?>
</div>
</div>
</div>
</div>
</div>
<!-- /.container-fluid -->
</div>
<!-- End of Main Content -->

```