



SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

**RANCANG BANGUN SISTEM *INVENTORY MANAGEMENT*
BENGKEL BERBASIS WEB MENGGUNAKAN *STACK MERN*:
STUDI KASUS DI BENGKEL AULIA MOTOR**

TUGAS AKHIR

MUHAMMAD ISMAIL

0110220170

PROGRAM STUDI TEKNIK INFORMATIKA

DEPOK

AGUSTUS 2024



**STT TERPADU
NURUL FIKRI**

SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

**RANCANG BANGUN SISTEM *INVENTORY MANAGEMENT*
BENGKEL BERBASIS WEB MENGGUNAKAN *STACK MERN*:
STUDI KASUS DI BENGKEL AULIA MOTOR**

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer

STT - NF
MUHAMMAD ISMAIL
0110220170

PROGRAM STUDI TEKNIK INFORMATIKA

DEPOK

AGUSTUS 2024

HALAMAN PERNYATAAN ORISINALITAS

Skripsi/Tugas Akhir ini adalah hasil karya penulis, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.



Nama : Muhammad Ismail
NIM : 0110220170

STT - NE Tempat, Depok 24 Juli 2024
Tanda Tangan



Muhammad Ismail

HALAMAN PENGESAHAN

Skripsi/Tugas Akhir ini diajukan oleh :

Nama : Muhammad Ismail

NIM : 0110220170

Program Studi : Teknik Informatika

Judul Skripsi : Rancang Bangun Sistem *Inventory Management* Bengkel Berbasis Web Menggunakan *Stack Mern*: Studi Kasus Di Bengkel Aulia Motor

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri

DEWAN PENGUJI

Pembimbing



(Salman El Farisi, S.Kom., M.Kom.)

Penguji



(Nasrul, S.Pd.I., S.Kom., M.Kom.)

Ditetapkan di : Depok

Tanggal : 24 Juli 2024

KATA PENGANTAR

Alhamdulillah, puji syukur kepada penulis dengan panjatkan ke hadirat Allah Subhanahu Wa Ta'ala yang atas berkat dan berupa rahmat-Nya, penulis dapat menyelesaikan skripsi/Tugas Akhir yang berjudul **“Rancang Bangun Sistem Inventory Management Bengkel Berbasis Web Menggunakan Stack Mern: Studi Kasus Di Bengkel Aulia Motor”**. Penulisan Tugas Akhir ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika di Sekolah Tinggi Teknologi Terpadu Nurul Fikri.

Tidak lupa, penulis juga menyadari bahwa tanpa bantuan dan dukungan dari berbagai pihak, mulai dari masa perkuliahan hingga penyusunan tugas akhir ini, penyelesaian tugas akhir ini akan sangat sulit. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Allah Subhanahu Wa Ta'ala.
2. Orang tua, semua anggota keluarga yang sudah memberikan dukungan moril atau materil dalam penyelesaian tugas akhir ini.
3. Bapak Dr. Lukman Rosyidi, S.T., M.M., M.T., selaku Ketua Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
4. Ibu Tifani Nabarian, S.Kom., M.T.I., sebagai Ketua Program Studi Teknik Informatika di Sekolah Tinggi Teknologi Terpadu Nurul Fikri
5. Bapak Nasrul, S.Pd.I., S.Kom., M.Kom., sebagai Dosen Pembimbing Akademik yang telah memberikan bimbingan kepada penulis selama berkuliah di Sekolah Tinggi Teknologi Terpadu Nurul Fikri, serta selaku Dosen Penguji Tugas Akhir penulis yang telah memberikan masukan berharga untuk penyempurnaan penelitian ini.
6. Bapak Salman El Farisi, S.Kom., M.Kom., selaku Dosen Pembimbing Tugas Akhir penulis yang telah membantu dengan bimbingannya dalam penyelesaian tugas akhir ini.
7. Para Dosen di lingkungan Sekolah Tinggi Teknologi Terpadu Nurul Fikri yang telah membimbing penulis dalam menuntut ilmu yang telah diberikan.

8. Kepada pemilik beserta karyawan dari Bengkel Aulia Motor yang telah meluangkan waktunya untuk memberikan data yang diperlukan bagi penulisan ilmiah ini.
9. Kepada Chika Aulia Zahwa, Ahmad Noval Fahmi, Gunawan, dan Muhammad Rayyan Azka Hudaya selaku teman perjuangan penulis dalam mengerjakan project tugas akhir bersama-sama, terimakasih banyak selalu memberikan semangat, motivasi, arahan, dan selalu mendengarkan keluhan kesah penulis.
10. Kepada semua pihak dan teman-teman yang tidak dapat disebutkan satu persatu yang secara langsung maupun tidak langsung telah membantu penulisan laporan tugas akhir ini.

Penulis sadar betul akan banyaknya kekurangan yang mungkin disebabkan oleh sebuah keterbatasan, kemampuan, dan pengetahuan penulis. Meskipun demikian penulis telah berusaha semaksimal mungkin untuk menyelesaikan penulisan ilmiah ini dan sebaik mungkin. Oleh karena itu, dengan rendah hati penulis menerima kritik dan saran dari pembaca untuk perbaikan dimasa mendatang.

Akhir kata, penulis berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga tugas akhir ini bisa bermanfaat bagi pengembangan ilmu pengetahuan.

Depok, 24 Juli 2024

STT - NF



Muhammad Ismail

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Sekolah Tinggi Teknologi Terpadu Nurul Fikri, saya yang bertanda tangan di bawah ini:

Nama : Muhammad Ismail

NIM : 0110220170

Program Studi : Teknik Informatika

Jenis karya : Skripsi / Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada STT-NF **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty - Free Right*)** atas karya ilmiah saya yang berjudul :

**Rancang Bangun Sistem *Inventory Management* Bengkel Berbasis Web
Menggunakan *Stack Mern*: Studi Kasus Di Bengkel Aulia Motor**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini STT-NF berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

STT - NF

Dibuat di : Depok

Pada tanggal : 24 Juli 2024

Yang Menyetakan



(Muhammad Ismail)

ABSTRAK

Nama : Muhammad Ismail
NIM : 0110220170
Program Studi : Teknik Informatika
Judul : Rancang Bangun Sistem *Inventory Management* Bengkel Berbasis Web Menggunakan *Stack Mern*: Studi Kasus Di Bengkel Aulia Motor

Indonesia memiliki jumlah pengguna sepeda motor yang besar, dengan penjualan mencapai 4.361.008 unit pada tahun 2020. Bengkel Aulia Motor, berdiri sejak 2008, menghadapi masalah dalam pencatatan transaksi dan stok suku cadang yang masih konvensional, menyebabkan ketidakakuratan data yang berdampak pada efisiensi dan perkembangan usaha. Penelitian ini bertujuan merancang dan membangun sistem *inventory management* berbasis web untuk Bengkel Aulia Motor menggunakan *Stack MERN* (MariaDB, ExpressJS, React JS, dan NodeJS). Sistem ini dirancang untuk pencatatan digital guna mengurangi kesalahan dan meningkatkan akurasi. Metode pengembangan yang digunakan adalah *Extreme Programming* (XP), yang cepat dan efisien dalam menghadapi perubahan persyaratan. Implementasi sistem berhasil meminimalisir kesalahan pencatatan dan meningkatkan efisiensi operasional, memungkinkan pencatatan data suku cadang secara *real-time* dan menghasilkan laporan keuangan yang lebih akurat. Pengujian dengan metode *User Acceptance Testing* (UAT) menunjukkan bahwa sistem memenuhi kebutuhan pengguna dan sesuai dengan spesifikasi yang ditetapkan. Penggunaan sistem *inventory management* berbasis web dengan *Stack MERN* di Bengkel Aulia Motor terbukti efektif dalam meningkatkan akurasi dan efisiensi pencatatan data, serta dapat menjadi model bagi bengkel lain dengan masalah serupa.

Kata kunci : Bengkel Aulia Motor, *Extreme Programming* (XP), Pencatatan data, *Inventory Management*, *Stack MERN*.

ABSTRACT

Name : Muhammad Ismail
NIM : 0110220170
Study Program : Informatics Engineering
Title : Design Of Web-Based Workshop Inventory Management System
Using Mern Stack: Case Study In Aulia Motor Workshop

Indonesia has a large number of motorcycle users, with sales reaching 4,361,008 units in 2020. Bengkel Aulia Motor, established in 2008, faces problems in recording transactions and spare parts stocks that are still conventional, causing data inaccuracies that have an impact on business efficiency and development. This research aims to design and build a web-based information system for Aulia Motor Workshop using Stack MERN (MariaDB, ExpressJS, React JS, and NodeJS). This system is designed for digital recording to reduce errors and increase accuracy. The development method used is Extreme Programming (XP), which is fast and efficient in dealing with changing requirements. The system implementation succeeded in minimizing recording errors and improving operational efficiency, allowing real-time recording of spare parts data and producing more accurate financial reports. Testing with the User Acceptance Testing (UAT) method shows that the system meets user needs and is in accordance with the specifications set. The use of a web-based information system with MERN Stack at Aulia Motor Workshop has proven effective in improving the accuracy and efficiency of data recording, and can be a model for other workshops with similar problems.

Key words : Aulia Motor Workshop, Data logging, Extreme Programming (XP), Inventory Management, MERN Stack.

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS.....	iii
HALAMAN PENGESAHAN.....	iv
KATA PENGANTAR	v
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS	vii
ABSTRAK.....	viii
<i>ABSTRACT</i>	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR	xiv
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan dan Manfaat Penelitian.....	3
1.4 Batasan Masalah.....	3
1.5 Sistematika Penulisan	3
BAB II KAJIAN LITERATUR.....	5
2.1 Tinjauan Pustaka.....	5
2.1.1 Bengkel Motor	5
2.1.2 <i>Inventory Management</i>	5
2.1.3 <i>Extreme Programming (XP)</i>	5
2.1.4 React JS	6
2.1.5 ExpressJS	7
2.1.6 NodeJS	7

2.1.7 MariaDB.....	8
2.1.8 <i>User Acceptance Testing (UAT)</i>	8
2.2 Penelitian Terkait	9
2.3 Posisi Penelitian	11
BAB III METODOLOGI PENELITIAN.....	13
3.1 Tahapan Penelitian.....	13
3.2 Rancangan Penelitian	15
3.2.1 Jenis Penelitian.....	15
3.2.2 Metode Analisis Data.....	16
3.2.3 Metode Pengumpulan Data.....	16
3.2.4 Metode Pengujian.....	17
3.2.5 Metode Implementasi dan Evaluasi	17
3.2.6 Lingkungan Pengembangan.....	18
3.2.6.1 Lokasi Penelitian.....	18
3.2.6.2 Alat dan Bahan.....	18
BAB IV IMPLEMENTASI DAN EVALUASI.....	19
4.1 Perencanaan (<i>Planning</i>)	19
4.1.1 <i>User Stories</i>	19
4.1.2 Kebutuhan Sistem	20
4.1.3 <i>Timeline</i> Pengembangan	22
4.1.4 Pemodelan Kebutuhan Sistem	23
4.1.4.1 <i>Use Case Diagram</i>	24
4.2 Perancangan (<i>Design</i>)	25
4.2.1 <i>Activity Diagram</i>	25
4.2.1.1 <i>Activity Diagram</i> Karyawan Mengelola Data Barang	25

4.2.1.2	<i>Activity Diagram</i> Pemilik Mengelola Laporan Keuangan.....	26
4.2.1.3	<i>Activity Diagram</i> Pemilik Mengelola <i>Users</i>	28
4.2.2	<i>Class Diagram</i>	29
4.2.2.1	Rancangan <i>Class Diagram</i>	29
4.2.3	Perancangan Antar Muka Sistem (<i>Low-Fi</i>).....	30
4.2.3.1	<i>Low-Fi</i> Tampilan Login.....	31
4.2.3.2	<i>Low-Fi</i> Tampilan <i>Dashboard</i>	31
4.2.3.3	<i>Low-Fi</i> Tampilan Kelola Barang	32
4.2.3.4	<i>Low-Fi</i> Tampilan Tambah Barang.....	32
4.2.3.5	<i>Low-Fi</i> Tampilan <i>Detail</i> dan <i>Edit</i> Barang	33
4.2.3.6	<i>Low-Fi</i> Tampilan Laporan Keuangan.....	33
4.2.3.7	<i>Low-Fi</i> Tampilan Tambah Laporan Keuangan.....	34
4.3	Pengkodean (<i>Coding</i>).....	34
4.3.1	Implementasi Algoritma Menampilkan Barang dan Cari Barang.....	34
4.3.2	Implementasi Algoritma Tambah Barang	36
4.3.3	Implementasi Algoritma <i>Edit</i> Barang	38
4.3.4	Implementasi Algoritma <i>Endpoint</i> Barang	40
4.3.5	Implementasi Algoritma <i>Endpoint Final</i>	40
4.3.6	Hasil Implementasi Antar Muka Sistem	41
4.3.6.1	Tampilan Halaman <i>Login</i>	42
4.3.6.2	Tampilan Halaman <i>Dashboard</i>	42
4.3.6.3	Tampilan Halaman Kelola Barang	43
4.3.6.4	Tampilan Halaman Tambah Barang.....	43
4.3.6.5	Tampilan <i>Detail</i> dan <i>Edit</i> Barang	44
4.3.6.6	Tampilan Laporan Keuangan.....	44

4.3.6.7	Tampilan Tambah Laporan Keuangan	45
4.4	Pengujian (<i>Testing</i>)	45
4.4.1	<i>User Acceptance Testing</i> (UAT).....	45
4.5	Evaluasi.....	49
4.5.1	Analisis Hasil Pengujian	49
4.5.2	Hasil Evaluasi Penelitian.....	50
BAB V KESIMPULAN DAN SARAN.....		52
5.1	Kesimpulan	52
5.2	Saran.....	52
DAFTAR PUSTAKA		53
LAMPIRAN A WAWANCARA.....		56
LAMPIRAN B PERMOHONAN PENGAMBILAN DATA.....		58



STT - NF

DAFTAR GAMBAR

Gambar 3.1 Tahapan Penelitian	13
Gambar 4.1 <i>Use Case Diagram</i> Bengkel Aulia Motor	24
Gambar 4.2 <i>Activity Diagram</i> Karyawan Mengelola Data Barang	25
Gambar 4.3 <i>Activity Diagram</i> Pemilik Mengelola Laporan Keuangan	26
Gambar 4.4 <i>Activity Diagram</i> Pemilik Mengelola <i>Users</i>	28
Gambar 4.5 Rancangan <i>Class Diagram</i>	29
Gambar 4.6 <i>Low-fi</i> Tampilan <i>Login</i>	31
Gambar 4.7 <i>Low-fi</i> Tampilan <i>Dashboard</i>	31
Gambar 4.8 <i>Low-fi</i> Tampilan Kelola Barang	32
Gambar 4.9 <i>Low-fi</i> Tampilan Tambah Barang	32
Gambar 4.10 <i>Low-fi</i> Tampilan Detail dan Edit Barang	33
Gambar 4.11 <i>Low-fi</i> Tampilan Laporan Keuangan	33
Gambar 4.12 <i>Low-fi</i> Tampilan Tambah Laporan Keuangan	34
Gambar 4.13 Tampilan Halaman <i>Login</i>	42
Gambar 4.14 Tampilan Halaman <i>Dashboard</i>	42
Gambar 4.15 Tampilan Halaman Kelola Barang	43
Gambar 4.16 Tampilan Halaman Tambah Barang	43
Gambar 4.17 Tampilan <i>Detail</i> dan <i>Edit</i> Barang	44
Gambar 4.18 Tampilan Laporan Keuangan	44
Gambar 4.19 Tampilan Tambah Laporan Keuangan	45

STT - NF

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait	9
Tabel 2.2 Posisi Penelitian	11
Tabel 3.1 Rencana Pengujian UAT.....	17
Tabel 4.1 <i>User Stories</i>	19
Tabel 4.2 Aturan Penomoran Kebutuhan Sistem.....	20
Tabel 4.3 Kebutuhan Fungsional Sistem	20
Tabel 4.4 <i>Timeline</i> pengembangan	22
Tabel 4.5 <i>Class Diagram</i>	30
Tabel 4.6 Implementasi Algoritma Menampilkan Barang dan Cari Barang (<i>BarangController.js</i>)	35
Tabel 4.7 Implementasi Algoritma Tambah Data Barang (<i>BarangController.js</i>).....	36
Tabel 4.8 Implementasi Algoritma <i>Edit</i> Barang (<i>BarangController.js</i>).....	38
Tabel 4.9 Implementasi Algoritma <i>Endpoint</i> Barang (<i>barangRoute.js</i>).....	40
Tabel 4.10 Implementasi Algoritma <i>Endpoint Final</i> (<i>index.js</i>).....	41
Tabel 4.11 <i>Test Case</i> Kelola Barang.....	46
Tabel 4.12 <i>Test Case</i> Kelola Produk.....	47
Tabel 4.13 <i>Test Case</i> Kelola Laporan Keuangan.....	47
Tabel 4.14 <i>Test Case</i> Kelola <i>Users</i>	48
Tabel 4.15 Evaluasi Analisis Hasil Pengujian UAT	49
Tabel 4.16 Evaluasi <i>timeline</i> pengembangan.....	50

STT - NF

BAB I

PENDAHULUAN

1.1 Latar belakang

Indonesia merupakan negara dengan populasi pengguna sepeda motor yang signifikan, banyak orang yang umumnya menggunakan sepeda motor sebagai alat transportasi sehari-hari. Berdasarkan informasi yang tercatat dalam laporan Asosiasi Industri Sepeda Motor Indonesia (AISI), terjadi penjualan sepeda motor di Indonesia sebanyak 4.361.008 unit pada tahun 2020[1]. Dari jumlah tersebut, penjualan di pasar domestik mencapai 3.600.616 unit, sementara sisanya, sebanyak 700.392 unit, diekspor[1]. Oleh karena itu, tidak mengherankan apabila Indonesia termasuk dalam daftar sepuluh negara dengan jumlah pengguna sepeda motor terbanyak[1]. Karena sepeda motor di Indonesia terus bertambah setiap tahun, maka usaha bengkel motor pun juga akan bermunculan sejalan dengan servis perawatan dan perbaikan pada sepeda motor. Pada salah satu tempat servis yaitu Bengkel Aulia Motor, yang sudah berdiri sejak tahun 2008 hingga sekarang, merupakan sebuah bengkel servis sepeda motor yang menyediakan jasa pelayanan berupa penjualan berbagai macam *sparepart* motor dan juga pelayanan servis berkala seperti ganti oli, tambal ban, dan lainnya.

Bengkel Aulia Motor yang sudah berdiri hampir 16 tahun ini memiliki beberapa permasalahan yang umum terjadi pada beberapa bengkel motor yang masih mencatat secara konvensional, seperti pelaksanaan transaksi pencatatan penjualan jasa atau suku cadang yang masih dilakukan dengan cara konvensional. Proses rekapitulasi bulanan juga masih memerlukan perhitungan *manual* setiap minggu atau bulan. Pendataan konvensional untuk produk suku cadang dapat menyebabkan ketidakkonsistenan dalam data yang tercatat. Ketidakkonsistenan ini dapat berdampak pada ketidakakuratan data dan ketidakvalidan data di dalam usaha. Dampak negatif dari ketidakakuratan data ini dapat mempengaruhi perkembangan Bengkel Aulia Motor secara keseluruhan karena akan mempengaruhi kinerja dan efisiensi terhadap pekerjaan.

Maka dari itu, untuk mengatasi permasalahan tersebut diperlukan penerapan sistem yang dapat mencatat secara digital, memungkinkan semua proses terkait pendataan yang sebelumnya masih menggunakan cara konvensional atau melalui kertas dapat dilakukan secara digital dan memonitor melalui *website*. Hal ini bertujuan untuk mengurangi kemungkinan ketidakakuratan atau kesalahan dalam mencatat data barang yang masuk dan keluar, serta meningkatkan akurasi pada catatan pengeluaran dan pemasukan. Sistem ini nantinya akan dibuat berbasis *website* yang akan memanfaatkan *stack* MERN, yang dimana akan menggunakan MariaDB sebagai basis datanya, *framework* React JS untuk bagian *front-end* dan juga untuk bagian *back-end* akan menggunakan *framework* ExpressJS untuk mengelola data-data dan menggunakan API-nya pada *front-end*. Dalam pembuatan *website* ini akan menggunakan metode *Extreme Programming* (XP), yang dimana metode *Extreme Programming* (XP) ini dapat dianggap cepat, efisien, dan juga dirancang untuk ketika menghadapi perubahan persyaratan yang cepat[2].

Oleh karena itu, berdasarkan permasalahan yang sudah diuraikan sebelumnya penulis ingin menerapkan dan mengimplementasikan pembuatan *website* untuk membantu mempermudah mengelola data barang dan laporan keuangan pada Bengkel Aulia Motor. Penelitian ini akan fokus pada **Rancang Bangun Sistem Inventory Management Bengkel Berbasis Web Menggunakan Stack Mern: Studi Kasus Di Bengkel Aulia Motor.**

1.2 Rumusan Masalah

Berdasarkan pada penjelasan yang telah disampaikan pada latar belakang sebelumnya, maka dari itu permasalahan akan dirumuskan sebagai berikut:

1. Bagaimana rancang bangun sistem *inventory management* bengkel Aulia Motor menggunakan *Stack* MERN?
2. Apakah sistem *inventory management* yang dikembangkan berjalan dengan baik dan sesuai dengan kebutuhan pengguna?

1.3 Tujuan dan Manfaat Penelitian

Berdasarkan rumusan masalah yang telah diuraikan sebelumnya, terdapat tujuan sebagai berikut :

1. Merancang dan membangun sistem *inventory management* bengkel Aulia Motor dengan *Stack MERN*.
2. Menguji sistem *inventory management* yang sudah dibangun, apakah sesuai dengan kebutuhan pengguna.

Adapun manfaat dari penelitian ini adalah :

1. Meningkatkan efisiensi waktu dan tenaga dengan meminimalisir kesalahan dalam pencatatan data suku cadang, pemasukan, dan pengeluaran.
2. Menggantikan proses *manual* dengan sistem otomatis yang menghasilkan laporan pemasukan dan pengeluaran, supaya memberikan kemudahan dalam pemantauan dan analisis data keuangan dengan laporan yang dihasilkan.

1.4 Batasan Masalah

Batasan masalah pada penelitian tugas akhir ini adalah sebagai berikut:

1. Penelitian ini akan difokuskan pada permasalahan terkait pencatatan data suku cadang, pemasukan, dan pengeluaran.
2. Penelitian yang dikembangkan ini hanya melakukan pencatatan melalui *website*, tidak melibatkan teknologi atau alat pemindaian sebagai bagian integral dari proses pencatatan.
3. Penelitian ini hanya berfokus pada pengelolaan data barang dan laporan keuangan dalam lingkup bengkel, tidak ada interaksi dengan pelanggan atau *customer*.

1.5 Sistematika Penulisan

Penyusunan tugas akhir mengikuti struktur penulisan sebagai berikut :

1. BAB I PENDAHULUAN

Pada bagian ini merinci latar belakang yang menjadi dasar tugas akhir, merumuskan pertanyaan masalah muncul dari konteks latar belakang,

menjelaskan tujuan dan manfaat yang diharapkan, serta menetapkan batasan masalah agar tetap berfokus pada lingkup tugas akhir.

2. BAB II LANDASAN TEORI

Pada bagian ini membahas tentang teori-teori dan penjelasan terkait dengan topik tugas akhir, serta merinci penelitian terkait yang telah dilakukan sebelumnya sebagai referensi.

3. BAB III METODOLOGI PENELITIAN

Pada bagian ini membahas terkait jalannya tugas akhir, mencakup metode penelitian, langkah-langkah yang digunakan selama proses penelitian untuk mencapai tujuan yang telah ditetapkan.

4. BAB IV IMPLEMENTASI DAN EVALUASI

Pada bagian ini membahas hasil dari implementasi dan evaluasi sistem *website* yang telah dikembangkan untuk Bengkel Aulia Motor.

5. BAB V KESIMPULAN DAN SARAN

Bab ini memuat kesimpulan dari seluruh pelaksanaan tugas akhir, serta memberikan saran-saran untuk penelitian atau pengembangan lebih lanjut, khususnya dengan topik penelitian terkait.



STT - NF

BAB II

KAJIAN LITERATUR

2.1 Tinjauan Pustaka

2.1.1 Bengkel Motor

Bengkel adalah fasilitas di mana seorang mekanik melaksanakan pekerjaan perbaikan dan perawatan kendaraan. Bengkel umum kendaraan bermotor berfungsi untuk memperbaiki, membetulkan, dan merawat kendaraan agar tetap memenuhi standar teknis dan layak jalan. Hal ini sesuai dengan peraturan yang mengatur bahwa setiap kendaraan bermotor harus memenuhi persyaratan teknis dan kelayakan. Bengkel juga dapat dianggap sebagai usaha kecil dan menengah yang menyediakan jasa perbaikan, baik untuk sepeda motor maupun mobil. Usaha bengkel sepeda motor bertujuan untuk memperbaiki sepeda motor sehingga dapat berfungsi dengan baik sesuai dengan keinginan pemiliknya atau sesuai dengan kondisi aslinya[3].

2.1.2 *Inventory Management*

Inventory management adalah pendekatan sistematis untuk mengelola, menyimpan, dan menjual barang, baik bahan mentah maupun barang jadi. Dalam *inventory management*, perusahaan harus memastikan memiliki stok yang cukup untuk memenuhi permintaan pelanggan. Namun, jika persediaan terlalu besar, maka uang perusahaan akan terikat dalam bentuk barang dan tidak dapat segera digunakan, yang dapat menyebabkan kerugian. Sebaliknya, jika persediaan terlalu sedikit, pelanggan mungkin merasa kecewa karena keterbatasan produk, yang dapat menyebabkan kehilangan penjualan dan kerugian bagi perusahaan. *Inventory management* sangat penting karena fungsinya mencakup untuk mengantisipasi kekurangan stok, menyesuaikan pesanan persediaan dengan kebutuhan, dan memastikan ketersediaan barang yang diperlukan di pasaran[4].

2.1.3 *Extreme Programming (XP)*

Pendekatan pada metodologi pengembangan dalam perangkat lunak yang berbasis *agile* atau bisa disebut juga dengan *Software Development Life Cycle*

(SDLC)[5]. *Extreme Programming* (XP) sendiri merupakan proses pengembangan perangkat lunak yang terutama mengadopsi pendekatan dengan orientasi objek. Tujuan utama dari metode ini adalah membentuk tim dengan skala kecil hingga sedang, yang cocok untuk menghadapi situasi di mana persyaratan mungkin tidak jelas atau berubah dengan cepat. Model ini mengikuti pendekatan yang sistematis dan berurutan, dimulai dari tahap perencanaan (*planning*), desain (*design*), pemrograman (*coding*), hingga pengujian (*testing*)[6]. Berikut adalah penjelasan tentang tahap dari perancangan yang sudah disebutkan sebelumnya:

1. *Planning* (Perencanaan)

Tahap awal dalam pengembangan sistem dimulai dengan proses perencanaan, yang melibatkan berbagai aktivitas mulai dari mengenali masalah, menganalisis kebutuhan, hingga menetapkan jadwal waktu untuk perancangan sistem.

2. *Design* (Perancangan)

Pada fase perancangan, terdapat aktivitas yang melibatkan pemodelan sistem, arsitektur, dan *database*. Pemetaan *Unified Modeling Language* (UML) digunakan untuk memodelkan sistem dan arsitektur, sedangkan *Diagram Entity Relationship* (ERD) digunakan untuk pemodelan *database*.

3. *Coding* (Pengkodean)

Pada tahap pengkodean, pemodelan diterapkan dalam antarmuka pengguna menggunakan *framework* React JS untuk sisi klien (*front-end*) dan ExpressJS untuk sisi *server* (*back-end*), dengan penggunaan bahasa pemrograman JavaScript. Dalam implementasinya, manajemen basis data dilakukan menggunakan MariaDB sebagai penyimpanan data.

4. *Testing* (Pengujian)

Pengujian dilakukan untuk mendeteksi kemungkinan potensi kesalahan selama pengoperasian *website*, serta untuk memastikan bahwa sistem yang telah dikembangkan sesuai dengan kebutuhan ekspektasi pengguna[7].

2.1.4 React JS

React JS ialah sebuah *framework* atau *library* menggunakan JavaScript, digunakan untuk membuat antarmuka *user-interface* yang dinamis dan interaktif. Dikembangkan oleh Facebook, React JS sangat populer dalam lingkup

pengembangan *website*. Fokus utama React JS adalah pada komponen, yang memungkinkan pengembang untuk memecah tampilan menjadi bagian yang lebih kecil dan mudah dikelola. Pendekatan tersebut tentu membuat proses pengembangan, pemeliharaan, dan optimalisasi menjadi lebih efisien. React JS memakai pendekatan "*Virtual DOM*" yang memungkinkan perubahan *user-interface* diterapkan secara efisien tanpa perlu *refresh* atau menggambar ulang seluruh halaman, meningkatkan responsivitas dan kecepatan aplikasi. Selain itu, React JS juga dapat diintegrasikan dengan berbagai alat dan perpustakaan lain, menjadikannya pilihan yang fleksibel dalam pengembangan[8].

2.1.5 ExpressJS

ExpressJS merupakan salah satu kerangka kerja web yang dibangun di atas NodeJS dan memiliki fitur yang andal untuk aplikasi web dan *mobile*. ExpressJS juga merupakan *framework* yang *di-develop* menggunakan bahasa pemrograman JavaScript. Dengan ExpressJS, pengembang dapat dengan mudah membangun dan menjalankan *server* web dengan kode yang minimal, sehingga mempercepat proses perancangan situs web[9].

Beberapa kelebihan dari ExpressJS meliputi: mendukung pembuatan *middleware*, mendukung berbagai metode HTTP seperti kirim (*POST*), tangkap(*GET*), *replace* (*PUT*), hapus (*DELETE*) dan lainnya. Dalam ExpressJS sudah terdapat *template engine jade* secara *default*, mengelola *file* statis seperti CSS dan JavaScript, serta memberikan fleksibilitas yang tinggi dalam penyesuaian[10].

2.1.6 NodeJS

NodeJS sebuah lingkungan *runtime* JavaScript yang *open-source* dan lintas *platform*. Ini merupakan *platform* yang terkenal untuk berbagai jenis proyek. Berbasis pada mesin *JavaScript V8*, NodeJS memungkinkan eksekusi kode JavaScript pada lingkup luar peramban. Dirancang khusus untuk membangun aplikasi yang dapat disesuaikan, sehingga sangat cocok digunakan dalam pengembangan aplikasi jaringan *real-time* seperti *server* web, *server game*, *server* obrolan, dan sejenisnya[11].

Selain itu, NodeJS mempunyai *library* atau pustaka *server* HTTP bawaan yang memungkinkannya untuk *server* web berjalan tanpa perlu menggunakan perangkat lunak *server* web, seperti Apache, Nginx, atau Wamp[12].

2.1.7 MariaDB

MariaDB adalah sistem manajemen basis data yang dikembangkan sebagai lanjutan mandiri dari MySQL, dengan tujuan untuk mempertahankan kompatibilitas tinggi dengan MySQL dan mendukung API serta perintah-perintahnya. Sebagai *fork* dari MySQL, MariaDB menggantikan mesin penyimpanan InnoDB dengan XtraDB dan dikembangkan oleh beberapa pengembang asli MySQL. Pengembangan ini dilakukan untuk menjaga agar sistem tidak diakuisisi oleh Oracle Corporation, yang merupakan pesaingnya. MariaDB dirancang untuk menyediakan berbagai jenis data, memudahkan pengguna dalam mengelola aplikasi basis data, dan mampu menangani skala besar dengan berbagai pilihan data yang tersedia[13][14].

2.1.8 User Acceptance Testing (UAT)

Sebuah proses pengujian yang dikerjakan oleh seorang *end-user* secara langsung dengan melakukan interaksi terhadap sistem atau *website*, untuk memverifikasi apakah fitur yang ada sudah berfungsi sesuai dengan kebutuhan dan fungsinya. *User Acceptance Testing* ini adalah proses formal yang di mana seorang *end-user* menggunakan pendekatan metode *black-box* untuk mengevaluasi sistem berdasarkan spesifikasi yang telah ditentukan, yang hasilnya dapat dijadikan bukti bahwa sistem mampu memberikan bantuan kepada para pengguna. Tujuan dari *User Acceptance Testing* dalam pengembangan perangkat lunak adalah untuk memastikan bahwa sistem memenuhi kebutuhan sebenarnya dari pengguna, bukan hanya mengikuti spesifikasi teknis[15].

2.2 Penelitian Terkait

Penelitian terkait merupakan analisis terhadap kajian-kajian sebelumnya yang berkaitan dengan tema yang sama yang sedang diselidiki oleh peneliti. Berikut ini adalah ringkasan tabel yang mencatat studi-studi yang diacu oleh peneliti:

Tabel 2.1 Penelitian Terkait

No	Nama dan Tahun	Judul	Topik	Tools	Metode	Hasil
1.	Sarah Asiti (2023)	Penerapan Metode <i>Extreme Programming</i> Pada Rancang Bangun Website <i>Company Profile</i>	<i>Company Profile</i> Bengkel	HTML, CSS, Javascript, PHP	<i>Extreme Programming (XP)</i>	Website <i>Company Profile</i> Bengkel Ahass
2.	Faris Mas'ud, Nuryuliani (2024)	Perancangan Aplikasi Pembelajaran Bahasa Asing Yang Interaktif Menggunakan Metode Mern	Pembelajaran Bahasa Asing Interaktif	MERN Stack (MongoDB, Express JS, React JS, NodeJS)	Waterfall	Aplikasi Obrolan Online Berbasis Website
3.	Ahmad Khainur Nadhif (2024)	Rancang Bangun Sistem Informasi Asistensi Berbasis Website Menggunakan Teknologi MERN Stack	Sistem Informasi Asistensi	MERN Stack (MongoDB, Express JS, React JS, NodeJS)	Rapid Application Development (RAD)	Website Sistem Informasi Asistensi
4.	Moch. Akbar Maulana, Haryoko, Banu Santoso, Lukman (2022)	Penerapan Teknologi Stack MERN Pada Aplikasi <i>Service Manajemen</i> Bengkel Berbasis Web	<i>Service Management</i>	MongoDB Database, Framework Express JS, Framework React JS, NodeJS	Web Development Life Cycle (WDLC)	Website <i>Service Management</i>

Pada bagian ini adalah penjelasan ataupun deskripsi dari penelitian terdahulu yang terkait dengan penulis yang sudah dipilih untuk dijadikan referensi :

1. Pada penelitian yang dilakukan oleh Sarah Asiti. yang berjudul “Penerapan Metode *Extreme Programming* Pada Rancang Bangun *Website Company Profile*”. Penelitian ini berisi tentang implementasi metode dari *Extreme Programming (XP)* yang membahas tentang bagaimana cara mempermudah seorang *admin* bengkel dalam mengelola sebuah data *service* dan data *sparepart*, serta memberikan kemudahan bagi pelanggan dalam melihat proses *service* dan daftar antrian *service*. Alasan penulis menggunakan referensi penelitian ini adalah karena terdapat keterkaitan pembahasan tentang bengkel motor serta metode yang akan digunakan oleh penulis, yaitu menggunakan metode *Extreme Programming (XP)*. Dimana referensi tersebut akan menjadi salah satu acuan penulis untuk mengetahui bagaimana cara mengembangkan *website* menggunakan metode tersebut[2].
2. Pada penelitian yang dilakukan oleh Faris Mas’ud dan Nuryuliani yang berjudul “Perancangan Aplikasi Pembelajaran Bahasa Asing Yang Interaktif Menggunakan Metode Mern”. Penelitian ini berisi tentang pembuatan aplikasi “Obrolan *Online* Berbasis *Website*”, yang dimana aplikasi ini telah berhasil dirancang dengan fitur *chatting real-time* untuk meningkatkan kemampuan bahasa menggunakan *MERN stack*. Dari penelitian tersebut, terdapat keterkaitan untuk bisa dijadikan referensi yang akan menjadi salah satu acuan penulis untuk mengetahui bagaimana cara implementasi *MERN stack* pada studi kasus penulis untuk mengembangkan sebuah aplikasi *website*[10].
3. Pada penelitian yang dilakukan oleh Ahmad Khainur Nadhif yang berjudul “Rancang Bangun Sistem Informasi Asistensi Berbasis *Website* Menggunakan Teknologi *MERN Stack*”. Penelitian ini berisi tentang penerapan teknologi *MERN stack* pada pengembangan sistem informasi asistensi berbasis *website*, yang menyangkut sistem presensi, rekapitulasi otomatis, dan penjadwalan terintegrasi. Salah satu alasan penulis

menggunakan referensi penelitian ini adalah karena terdapat salah satu ketertarikan, yaitu menggunakan teknologi *MERN stack* untuk mengembangkan atau membuat sebuah *website*[16].

4. Pada penelitian yang dilakukan oleh Moch. Akbar Maulana, Haryoko, Banu Santoso dan Lukman yang berjudul “Penerapan Teknologi *Stack* MERN Pada Aplikasi *Service* Manajemen Bengkel Berbasis Web”. Penelitian ini membahas tentang cara untuk mengatasi beberapa masalah yang terjadi di bengkel, seperti pencatatan transaksi *manual*, perhitungan bulanan yang memakan waktu, dan inkonsistensi data. Dari penelitian tersebut, terdapat beberapa keterkaitan seperti contohnya tentang *inventory* pada bengkel, lalu penerapan teknologi menggunakan *MERN stack*, yang dimana dari kedua keterkaitan tersebut bisa menjadi referensi bagi penulis untuk penelitian ini[1].

2.3 Posisi Penelitian

Tabel 2.2 Posisi Penelitian

No	Penelitian	<i>Inventory Management</i>	<i>MERN Stack</i>	<i>Extreme Programming (XP)</i>	<i>User Acceptance Test (UAT)</i>
1.	Sarah Asiti. (2023)	☑		☑	
2.	Faris Mas’ud, Nuryuliani. (2024)		☑		
3.	Ahmad Khainur Nadhif. (2024)		☑		
4.	Moch. Akbar Maulana, Haryoko, Banu Santoso, Lukman. (2022)	☑	☑		
5.	Muhammad Ismail, 2024	☑	☑	☑	☑

Dengan merujuk pada studi literatur yang disajikan pada tabel 2.2, penulis melakukan perbandingan posisi penelitian dengan penulis sebagai berikut:

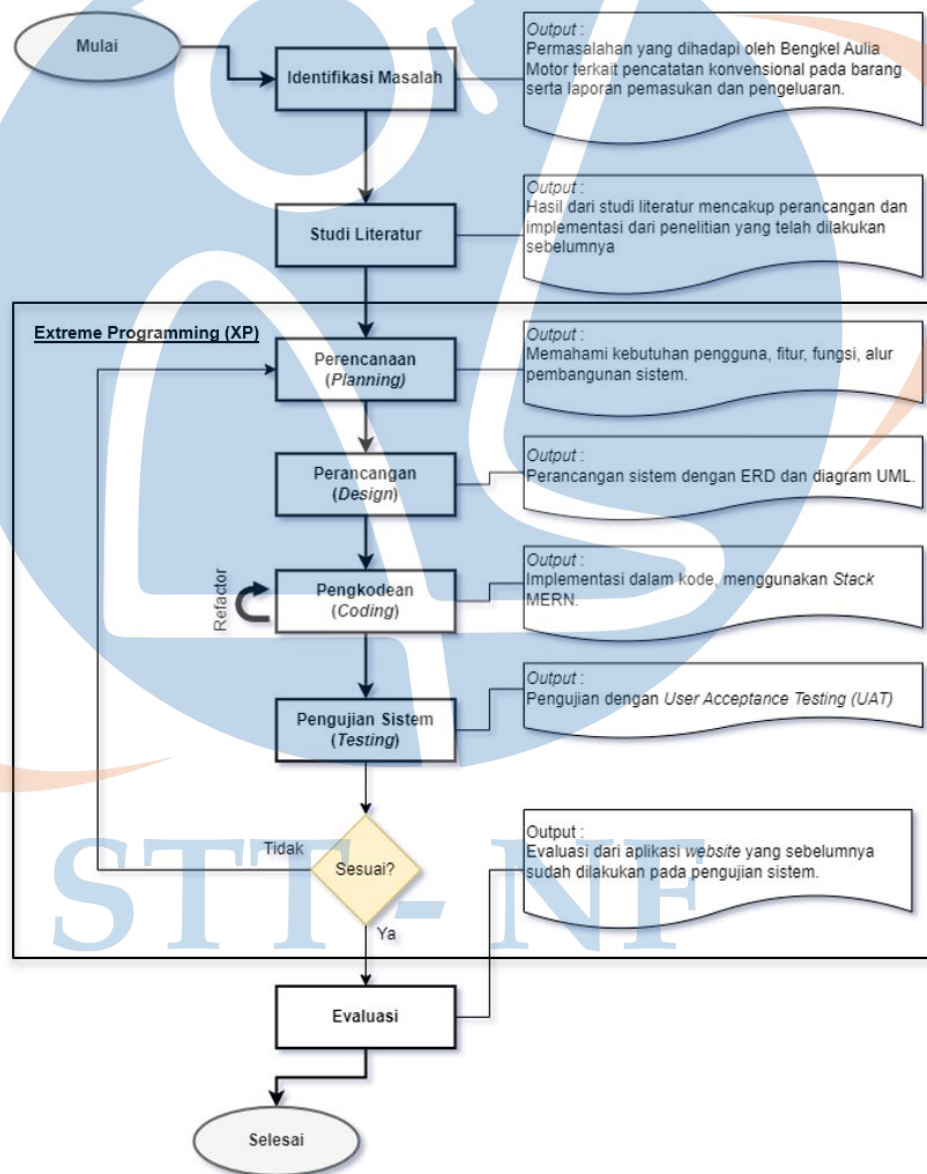
1. Penulis menggunakan metode *extreme programming* karena dari literatur sebelumnya menunjukkan bahwa metode ini dapat menghasilkan solusi inovatif dalam menangani suatu permasalahan, dan juga fokus utama *extreme programming* adalah meningkatkan kualitas *software* dan kemampuan tim untuk merespon akan perubahan.
2. Implementasi *stack* MERN dalam penelitian ini menggunakan metode *extreme programming*, berbeda dengan beberapa penelitian-penelitian sebelumnya yang menggunakan metode lain.
3. Pada penelitian ini penulis menggunakan *stack* MERN (MariaDB, ExpressJS, React JS, NodeJS), yang dimana penulis menggunakan MariaDB sebagai *database*-nya. Perbedaan dari penelitian sebelumnya yang menggunakan *stack* MERN juga adalah penelitian lain menggunakan MongoDB sebagai tempat menyimpan data (*database*).
4. Selain itu penulis menggunakan *User Acceptance Testing* (UAT) untuk melakukan pengujian pada sistem.

STT - NF

BAB III METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Pada tahap penelitian ini, alur yang akan dilalui dalam penelitian ini melibatkan serangkaian tahapan-tahapan, yang digambarkan pada Gambar 3.1:



Gambar 3.1 Tahapan Penelitian

Berikut penjelasan tahapan-tahapan penelitian pada Gambar 3.1:

1. Identifikasi Masalah

Pada tahapan ini melakukan identifikasi masalah yang terdapat pada Bengkel Aulia Motor. Tahapan ini merupakan tahapan awal yang diperlukan untuk mengidentifikasi masalah yang dihadapi. Hasil yang didapat dari tahapan ini adalah tentang permasalahan yang dihadapi oleh bengkel aulia motor terkait pencatatan pada barang, serta laporan pemasukan dan pengeluaran yang masih menggunakan metode konvensional.

2. Studi Literatur

Studi Literatur, tujuan mendukung temuan atau pengamatan yang telah dilakukan dengan mencari, mengumpulkan, membaca artikel, jurnal, tugas akhir, serta *website* yang relevan dengan topik penulis. Hasil yang didapat mencakup perancangan dan penerapan dari penelitian yang telah dilakukan sebelumnya.

3. Perencanaan (*Planning*)

Tahap perencanaan dimulai dari memahami terlebih dahulu kebutuhan pengguna dari masalah yang sudah diidentifikasi sebelumnya, dilanjut dengan pembuatan fitur dan fungsi yang akan menyelesaikan permasalahan tersebut, terakhir perencanaan terkait alur pembangunan sistemnya. Hasil dari tahap ini adalah memahami kebutuhan pengguna, membuat fitur, fungsi, dan alur pembangunan sistem.

4. Perancangan (*Design*)

Fokus utama pada tahapan ini adalah perancangan *diagram UML* untuk menggambarkan struktur dan interaksi antar komponen sistem secara *visual*. Hasil yang didapat dari tahapan ini adalah membuat sebuah perancangan sistem dengan *diagram UML*.

5. Pengkodean (*Coding*)

Pada tahap bagian penulisan kode *program*, yang dimana keluaran atau *output* pada tahapan ini adalah menggunakan pendekatan MERN *stack* yang terdiri dari MariaDB sebagai *database*, ExpressJS sebagai *back-end*, React JS sebagai bagian *front-end*, dan NodeJS sebagai lingkungan *runtime*

untuk menjalankan *server*. Selain itu penulis mengimplementasikan metode *Extreme Programming* yang dimana dengan pendekatan tersebut diharapkan dapat memfasilitasi pengembangan perangkat lunak yang efisien, cepat dan responsif terhadap perubahan.

6. Pengujian Sistem (*Testing*)

Tahapan ini adalah tahapan pengujian yang melibatkan *end-user*, atau karyawan dari Bengkel Aulia Motor untuk menguji seberapa kesesuaian aplikasi yang telah dibuat. Dengan dilakukannya pengujian menggunakan metode *User Acceptance Testing* (UAT) untuk memvalidasi apakah setiap fungsi yang ada telah berjalan sesuai dengan kebutuhan dan spesifikasinya. Keluaran yang terdapat pada tahapan ini adalah melakukan pengujian dengan *User Acceptance Test* (UAT).

7. Evaluasi

Pada tahapan ini adalah fase evaluasi pada sistem berdasarkan dari hasil pengujian sebelumnya yang menggunakan metode *User Acceptance Test* (UAT). Dimana keluaran atau *output* pada tahapan ini adalah sebuah kesimpulan dari hasil evaluasi yang sudah dilakukan pengujian sistem, seperti telah memenuhi standar dan harapan yang diinginkan atau belum memenuhi standar.

3.2 Rancangan Penelitian

3.2.1 Jenis Penelitian

Jenis penelitian yang digunakan pada penelitian ini yaitu menerapkan metode *Research and Development* (R&D), adalah suatu pendekatan untuk menciptakan produk baru atau menyempurnakan produk yang sudah ada, serta menguji keefektifan produk[17]. Hasil akhir dari penelitian ini adalah pengembangan sebuah *platform website* yang dapat membantu untuk meminimalisir kesalahan pada pengelolaan data suku cadang secara terstruktur, serta mempermudah untuk menghasilkan sebuah laporan pemasukan dan pengeluaran untuk bengkel aulia motor.

3.2.2 Metode Analisis Data

Pada studi ini, peneliti menerapkan metode analisis data kualitatif. Metode ini digunakan pada saat mengolah data wawancara yang diambil pada tahap pengumpulan kebutuhan, sehingga yang dapat dianalisis adalah hasil wawancara tersebut untuk mendapat pemahaman yang mendalam tentang kebutuhan yang ada. Metode penelitian kualitatif adalah suatu pendekatan yang berfokus pada pemahaman sifat hubungan, aktivitas, situasi, atau materi. Tujuan utama penelitian kualitatif adalah untuk menjelaskan fakta yang dapat dipahami untuk memfasilitasi perumusan hipotesis baru. Metode pengumpulan data yang umum dalam penelitian kualitatif meliputi observasi, wawancara, dan analisis[18]. Dengan pendekatan tersebut, memungkinkan peneliti untuk menyelidiki seluk-beluk keadaan saat ini dan memperoleh pemahaman yang lebih mendalam untuk solusi yang sesuai.

3.2.3 Metode Pengumpulan Data

Untuk mendukung pengumpulan data pada tugas akhir, penulis menggunakan teknik pengumpulan data seperti tinjauan studi literatur dan melakukan wawancara. Berikut penjelasan metode yang diterapkan untuk pengumpulan data sebagai berikut :

1. Studi Literatur

Sebuah proses pengumpulan data sekunder yang dimulai dengan mencari dan menganalisis berbagai referensi literatur seperti artikel, buku, jurnal ilmiah, serta penelitian sebelumnya yang telah dipublikasikan[12]. Pengumpulan data yang diperoleh dari berbagai sumber tersebut dapat membantu untuk memecahkan masalah yang ada dalam penelitian.

2. Wawancara

Wawancara merupakan interaksi langsung antara yang ingin diwawancarai dan yang bertugas sebagai responden yang bertujuan untuk menggali dan mencari informasi dari yang diharapkan, dengan tujuan mendapatkan data tentang responden tanpa bias sebanyak mungkin dan dengan efisien[12]. Dengan melakukan wawancara, penulis akan mengajukan pertanyaan langsung kepada pihak terkait untuk memperoleh informasi yang mendukung pengembangan *website* untuk bengkel aulia motor.

3.2.4 Metode Pengujian

Penelitian ini menggunakan pendekatan *User Acceptance Testing* (UAT) sebagai metode pengujian untuk mengevaluasi sejauh mana *website* yang telah dibuat sudah memenuhi kebutuhan. Pengujian *User Acceptance Testing* (UAT) ini akan dilakukan terhadap 2 pengguna yaitu kepada pemilik bengkel aulia motor, dan karyawan dari bengkel aulia motor. Berikut ini terlampir skenario pengujian pada tabel 3.1 rencana pengujian yang akan dilakukan oleh pengguna setelah melakukan uji coba *website*.

Tabel 3.1 Rencana Pengujian UAT

No.	Fitur	Deskripsi
1.	<i>Login</i>	Pengguna bisa <i>login</i> menggunakan akun yang sudah dibuat.
2.	Kelola Barang	Pengguna bisa akses halaman kelola barang, menampilkan data barang, mencari, tambah, ubah dan hapus.
3.	Kelola Produk	Pengguna bisa akses halaman kelola produk, menambahkan data, ubah, dan hapus data produk.
4.	Kelola Laporan Keuangan	Pengguna dengan <i>role</i> pemilik dapat akses halaman kelola laporan keuangan serta dapat menambahkan, <i>filter</i> , dan menghapus data.
5.	Kelola <i>Users</i>	Pengguna dengan <i>role</i> pemilik dapat akses halaman <i>kelola users</i> serta dapat menambahkan <i>user</i> baru baik karyawan maupun admin, ubah data, <i>filter</i> , dan hapus.

3.2.5 Metode Implementasi dan Evaluasi

Proses implementasi *website* bengkel aulia motor dengan menggunakan metode *Extreme Programming* (XP) ini terdiri dalam beberapa tahapan, yang antara lain adalah persiapan lingkungan pengembangan, perancangan desain sistem, pengembangan *website*, *testing*, dan *launching*. Setelah melalui tahap *testing* dan *launching*, *website* akan menjalani evaluasi melalui *User Acceptance Testing* (UAT) yang dimana akan terdapat beberapa fitur yang akan diuji coba untuk memastikan pencapaian tujuan yang sudah ditetapkan seperti berhasil atau tidaknya fitur tersebut. Dengan mengikuti tahapan-tahapan tersebut, proses pengembangan *website* dapat dilakukan dengan efektif dan pada akhirnya akan memberikan manfaat kepada bengkel aulia motor.

3.2.6 Lingkungan Pengembangan

3.2.6.1 Lokasi Penelitian

Pada saat melakukan penelitian, penulis melakukan penelitian secara *online* dikediaman penulis yang beralamat di Jalan Kp. Sawah Desa Jabon Mekar, Kec. Parung, Kab. Bogor, Jawa Barat 16330.

3.2.6.2 Alat dan Bahan

Penelitian yang dilakukan untuk membangun suatu *website* membutuhkan sebuah alat dan bahan sebagai sarana dalam pengembangan. Adapun bahan dan alat yang digunakan adalah sebagai berikut :

1. *PC Desktop*
 - a. Processor : AMD Ryzen 5 4500, 4,35 GHz
 - b. RAM : 16 GB
 - c. *SystemOS* : Windows 11 Pro 64-bit (10.0.22621 Build 22621).

2. XAMPP (v.3.30)

XAMPP adalah kumpulan dari berbagai *software open-source* yang digunakan untuk membantu *programmer* dalam menggunakan web *server*, *database*, dan juga *phpmyadmin*.

3. Google Chrome

Sebuah web *browser* yang digunakan untuk melihat hasil aplikasi yang sudah dibuat, serta web *browser* digunakan untuk melakukan *debugging* aplikasi web yang sedang dikembangkan.

4. MERN *Stack*

MERN *Stack* adalah sebuah teknologi *stack* yang sudah mencakup *database* dan *framework* yang berisi MariaDB, ExpressJS, React JS dan NodeJS yang digunakan untuk pengembangan aplikasi *website*.

5. Postman

Postman adalah sebuah *tools software* untuk melakukan pengecekan API yang telah dibuat pada *back-end*, sebelum nantinya API tersebut akan digunakan di bagian *front-end*.

BAB IV IMPLEMENTASI DAN EVALUASI

4.1 Perencanaan (*Planning*)

Dalam fase perencanaan ini, peneliti akan berupaya untuk mengidentifikasi dan memprioritaskan kebutuhan sistem yang akan dibangun. Analisis ini difokuskan pada kebutuhan fungsional sistem. Proses ini melibatkan identifikasi kebutuhan pengguna yang diperoleh dari *user story*.

4.1.1 *User Stories*

Pada bagian *user stories* menjelaskan peran-peran atau aktor yang terlibat dengan sistem berdasarkan *user stories*, yang nantinya akan dimasukkan ke dalam *use case diagram*. Berikut adalah peran-peran beserta karakteristik dan tindakan terhadap sistem akan dibahas berdasarkan hasil *user stories* menggunakan yang diperoleh dari metode wawancara (Lampiran A.1). Karyawan, yaitu bertindak sebagai pengguna biasa, serta pemilik yang merupakan owner dari bengkel aulia motor, akan disajikan dalam Tabel 4.1.

Tabel 4.1 *User Stories*

No	Aktor	<i>User Story</i>
1.	Karyawan	<ol style="list-style-type: none">1. Sebagai karyawan, saya ingin melakukan pengelolaan data barang atau <i>sparepart</i> dengan kemampuan menambahkan, mengubah, dan menghapus barang, sehingga saya dapat bisa meminimalisir kesalahan pencatatan dan memudahkan pendataan barang.2. Sebagai karyawan, saya ingin mengelola produk dengan kemampuan untuk menambahkan, mengubah, dan menghapus pada produk, sehingga pada saat memasukan data barang akan menjadi lebih mudah dan terdata rapih.
2.	Pemilik	<ol style="list-style-type: none">1. Sebagai pemilik, saya ingin mengawasi dan mengatur pengelolaan laporan keuangan tanpa campur tangan pengguna lain, sehingga karyawan tidak dapat mengakses kelola laporan keuangan sebagai pemilik2. Sebagai pemilik, saya ingin mengelola data atau akun karyawan dengan kemampuan menambah, mengubah, dan menghapus, sehingga ketika ada karyawan baru atau akun yang bermasalah bisa segera dikelola.3. Sebagai pemilik, saya ingin mengelola data barang atau <i>sparepart</i> dengan kemampuan menambah, mengubah, dan menghapus, sehingga saya bisa memantau data

		barang yang dimasukkan oleh karyawan dan melakukan perubahan jika terjadi kesalahan.
		4. Sebagai seorang pemilik, saya ingin mengelola data produk dengan kemampuan menambah, mengubah, dan menghapus, sehingga saya bisa memantau data produk yang dimasukkan oleh karyawan dan melakukan perubahan jika terjadi kesalahan.

4.1.2 Kebutuhan Sistem

Analisis kebutuhan sistem melibatkan pemeriksaan menyeluruh terhadap kebutuhan fungsional sistem. Kebutuhan fungsional ini mencakup berbagai fungsi atau proses yang harus disediakan oleh sistem berdasarkan permintaan dari *stakeholder*, yang informasinya dikumpulkan dari *user story*. Tabel 4.2 akan menampilkan aturan penomoran kebutuhan sistem yang akan diintegrasikan, sementara Tabel 4.3 akan menyajikan kebutuhan fungsional sistem secara lebih rinci.

Tabel 4.2 Aturan Penomoran Kebutuhan Sistem

Kode	Keterangan
BAM-	Bengkel Aulia Motor
-(x)	Nomor urutan kode kebutuhan

Tabel 4.3 Kebutuhan Fungsional Sistem

No	Kode	Nama Fitur	User	Deskripsi & Spesifikasi
1.	BAM-01	<i>Login</i>	Karyawan & Pemilik	<ol style="list-style-type: none"> <i>User</i> melakukan <i>login</i> pada sistem. Sistem menampilkan formulir <i>login</i> yang terdiri dari bidang <i>email</i>, bidang <i>password</i>, dan tombol untuk <i>login</i>. Masuk halaman <i>dashboard</i> dengan sesuai dengan <i>role</i> pada akun yang digunakan ketika <i>login</i>.
2.	BAM-02	Kelola Barang	Karyawan & Pemilik	<ol style="list-style-type: none"> <i>User</i> memiliki kemampuan untuk menambahkan, mengubah, dan menghapus data barang. Sistem akan menampilkan <i>form</i> dengan tipe <i>pop-up</i> atau konfirmasi dari fitur

				kelola barang yang akan dipilih oleh <i>User</i> untuk pengelolaan data barang.
3.	BAM-03	Cari Barang	Karyawan & Pemilik	<ol style="list-style-type: none"> 1. <i>User</i> memiliki kemampuan untuk mencari barang berdasarkan kriteria pencarian. 2. Sistem akan menampilkan barang yang telah dimasukkan <i>user</i> pada kolom pencarian.
4.	BAM-04	Kelola Merk	Karyawan & Pemilik	<ol style="list-style-type: none"> 1. <i>User</i> dapat menambah, mengubah, dan menghapus data merk. 2. Sistem akan menampilkan <i>form</i> dengan tipe <i>pop-up</i> atau konfirmasi dari fitur kelola merk.
5.	BAM-05	Kelola Kategori	Karyawan & Pemilik	<ol style="list-style-type: none"> 1. <i>User</i> dapat menambah, mengubah, dan menghapus data kategori. 2. Sistem akan menampilkan <i>form</i> dengan tipe <i>pop-up</i> atau konfirmasi dari fitur kelola kategori.
6.	BAM-06	Kelola Ukuran	Karyawan & Pemilik	<ol style="list-style-type: none"> 1. <i>User</i> dapat menambah, mengubah, dan menghapus data ukuran. 2. Sistem akan menampilkan <i>form</i> dengan tipe <i>pop-up</i> atau konfirmasi dari fitur kelola ukuran.
7.	BAM-07	Cari Produk	Karyawan & Pemilik	<ol style="list-style-type: none"> 1. <i>User</i> dapat melakukan pencarian produk. 2. Sistem akan menampilkan produk yang telah dimasukkan <i>user</i> pada kolom pencarian.
8.	BAM-08	Kelola Karyawan (<i>users</i>)	Pemilik	<ol style="list-style-type: none"> 1. Pemilik dapat menambah, mengubah, dan menghapus data pengguna. 2. Sistem akan menampilkan <i>form</i> dengan tipe <i>pop-up</i> atau konfirmasi, dan pemilihan akun yang ingin dibuat untuk <i>role</i> karyawan atau pemilik.

9.	BAM-09	Kelola Laporan Keuangan	Pemilik	<p>3. Pemilik dapat melihat laporan keuangan dan dapat melakukan penambahan dan penghapusan data laporan keuangan pada bagian pemasukan atau pengeluaran.</p> <p>4. Sistem akan menampilkan <i>form</i> dengan tipe <i>pop-up</i> atau konfirmasi dari fitur kelola laporan keuangan yang dipilih oleh pemilik.</p>
10.	BAM-10	<i>Logout</i>	Karyawan & Pemilik	<p>1. <i>User</i> keluar dari sistem.</p> <p>2. Sistem akan menampilkan <i>pop-up</i> konfirmasi sebelum <i>logout</i>.</p>

4.1.3 Timeline Pengembangan

Berikut ini adalah Tabel 4.4 *Timeline* Pengembangan pada *website inventory management* bengkel aulia motor:

Tabel 4.4 *Timeline* pengembangan

<i>Sprint</i>	<i>Tanggal</i>	<i>Modul</i>	<i>Task</i>	<i>Point</i>	<i>Velocity</i>
	04 Maret 2024	<i>Sprint Planning</i>	<i>List Backlog</i>		
0	01 - 12 April 2024	Persiapan	Analisis kebutuhan sistem	5	10
			Perancangan sistem	5	
1	15 - 25 April 2024	<i>Authentication User</i>	<i>Register (API)</i>	5	10
			<i>Login dan Logout</i>	5	
2	28 April - 01 Mei 2024	Kelola Merk	<i>View & search data merk</i>	5	20
			Tambah merk	5	
			<i>Edit merk</i>	5	
			Hapus merk	5	
3	02 - 04 Mei 2024	Kelola Kategori	<i>View & search data kategori</i>	5	20
			Tambah kategori	5	
			<i>Edit kategori</i>	5	
			Hapus kategori	5	
4	05 - 07 Mei 2024	Kelola Ukuran	<i>View data ukuran</i>	5	20
			Tambah ukuran	5	
			<i>Edit ukuran</i>	5	
			Hapus ukuran	5	
5	10 - 28 Mei 2024	Kelola Barang	<i>View & search data barang</i>	5	34
			Tambah barang	9	

			<i>Edit barang</i>	8	
			<i>Hapus barang</i>	5	
			<i>Export pdf</i>	7	
6	01 - 09 Juni 2024	Laporan Keuangan	<i>View data laporan keuangan</i>	5	29
			<i>Tambah laporan keuangan</i>	5	
			<i>Hapus laporan keuangan</i>	5	
			<i>Export pdf</i>	7	
			<i>Filter laporan keuangan</i>	7	
7	10 - 14 Juni 2024	Kelola <i>User</i>	<i>View & search data user</i>	5	20
			<i>Tambah user</i>	5	
			<i>Edit user</i>	5	
			<i>Hapus user</i>	5	

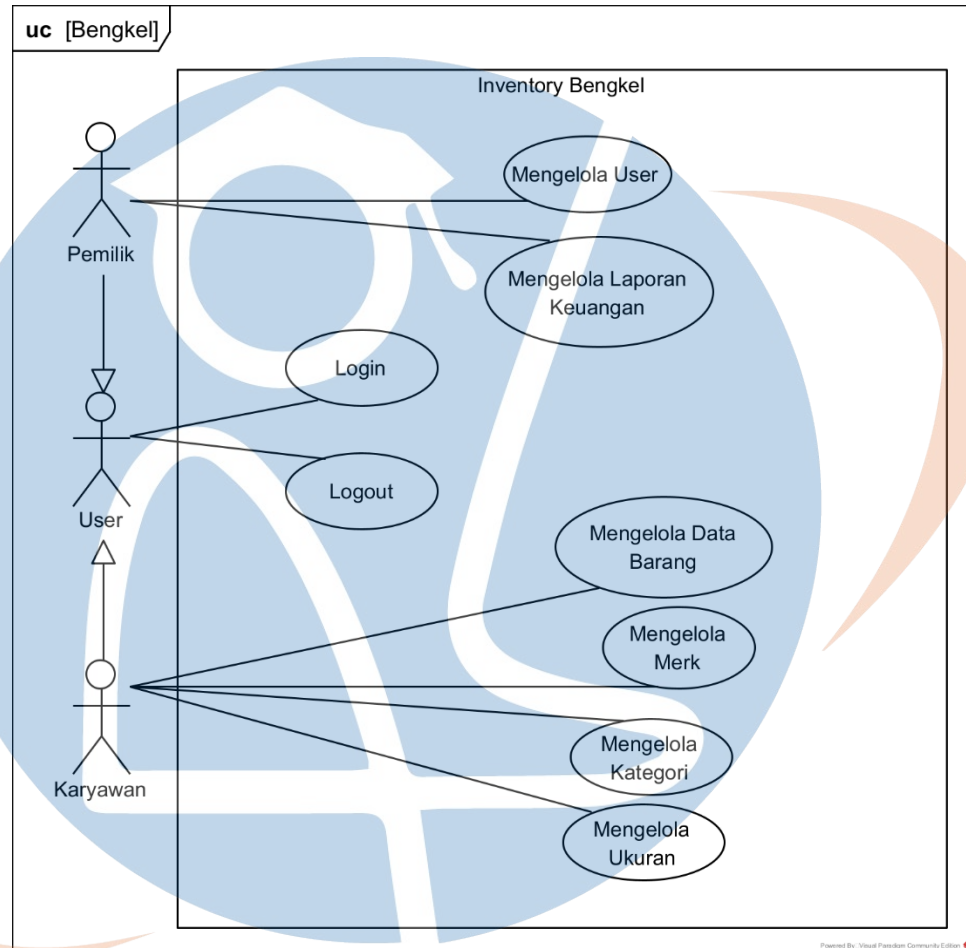
4.1.4 Pemodelan Kebutuhan Sistem

Tahap pemodelan kebutuhan sistem ini akan dilakukan sebuah pemodelan kebutuhan sistem untuk mempermudah langkah perancangan sistem selanjutnya. Peneliti menggunakan *Unified Modeling Language* (UML) untuk membuat pemodelan *diagram* dengan fokus pada *use case diagram*. Pemodelan ini akan dirancang berdasarkan tabel fungsionalitas sistem yang tercantum pada Tabel 4.3.

STT - NF

4.1.4.1 Use Case Diagram

Berdasarkan analisis dari hasil *user story* yang diperoleh dari wawancara pada Tabel 4.3, terdapat berbagai macam kebutuhan ataupun fitur yang telah disimpulkan. Maka dibuatlah perancangan *use case diagram* aplikasi dari *website* sebagai berikut.



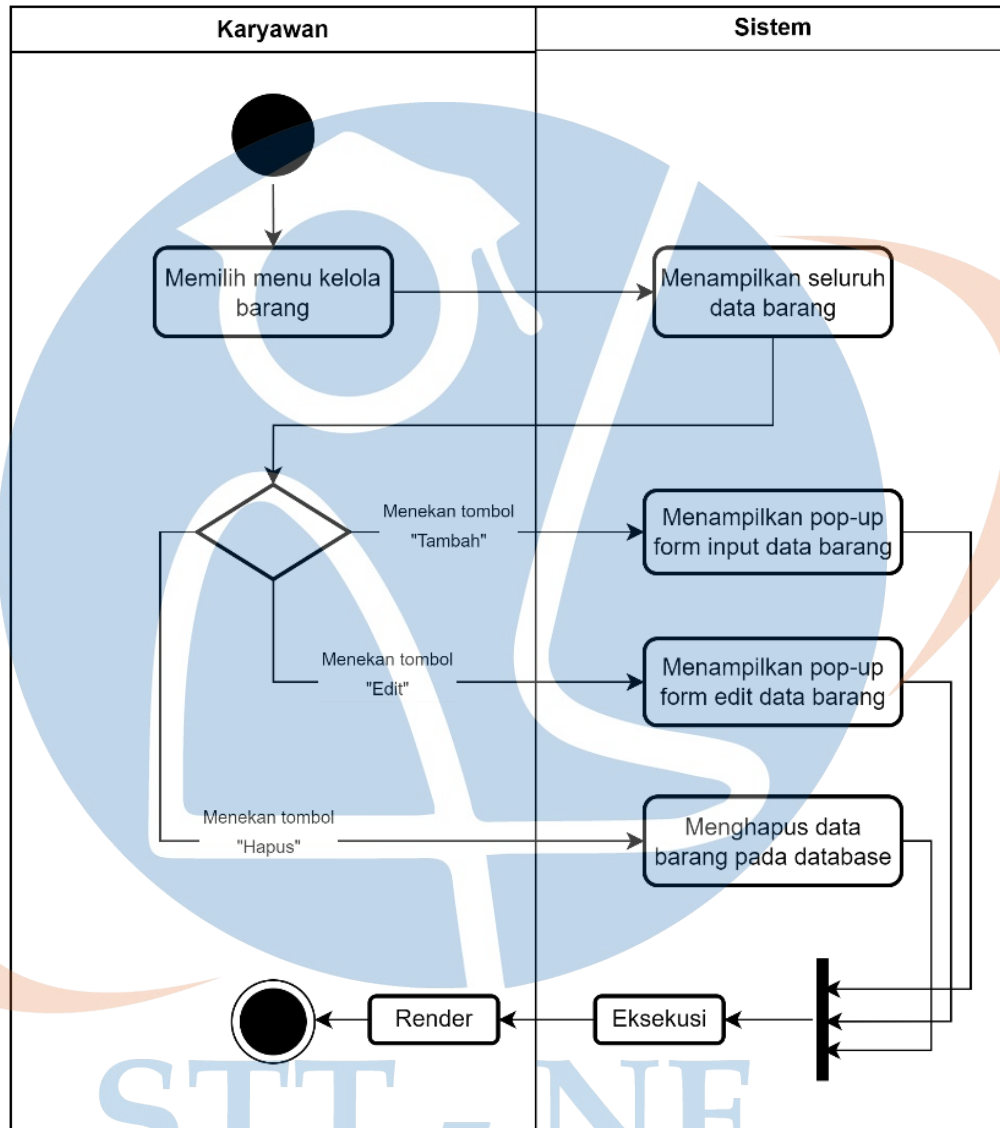
Gambar 4.1 Use Case Diagram Bengkel Aulia Motor

Dalam skenario perencanaan sistem *inventory management* bengkel aulia motor ini terdapat dua aktor utama, Pemilik dan Karyawan. Karyawan berperan sebagai pengguna yang bertanggung jawab mengelola data barang, merk, kategori, dan ukuran. Pemilik memiliki wewenang tambahan untuk mengelola seluruh aspek data bengkel, termasuk mengelola data *user* atau data karyawan, mengelola laporan keuangan baik pemasukan maupun pengeluaran, serta melakukan hal yang sama seperti karyawan yaitu mengelola data barang, merk, kategori dan ukuran. Semua fitur yang dilakukan oleh *user* karyawan dan pemilik diharuskan untuk *login* terlebih dahulu.

4.2 Perancangan (*Design*)

4.2.1 *Activity Diagram*

4.2.1.1 *Activity Diagram* Karyawan Mengelola Data Barang

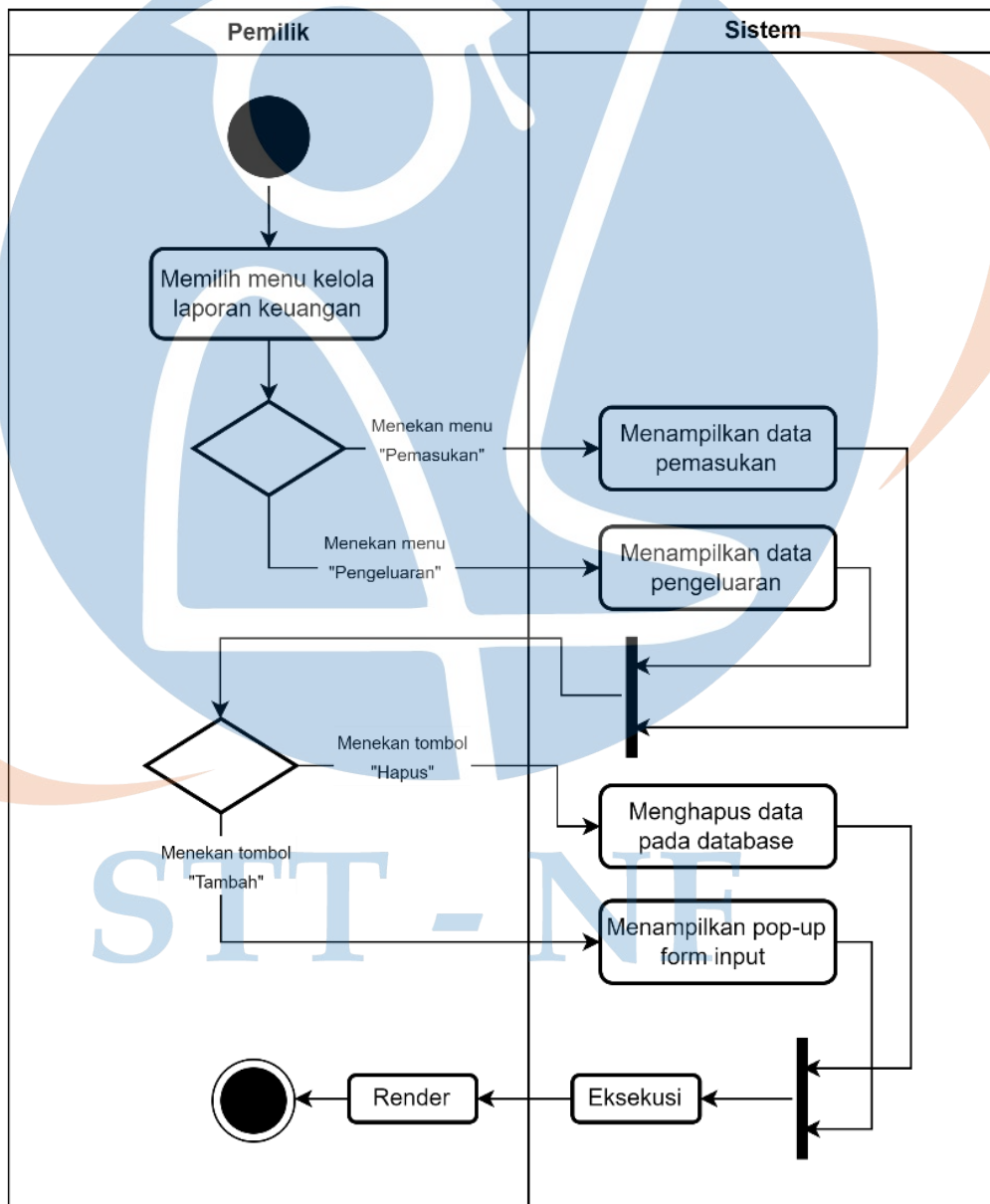


Gambar 4.2 *Activity Diagram* Karyawan Mengelola Data Barang

Dari rancangan *Activity diagram* yang terdapat pada Gambar 4.2 menjelaskan tentang alur proses untuk karyawan yang melakukan pengelolaan data barang. Karyawan memulai proses dengan cara memilih menu kelola barang. Sistem secara langsung akan menampilkan seluruh data barang pada halaman kelola barang. Dalam halaman kelola barang terdapat tombol tambah, *edit*, dan hapus, yang dimana ketika di klik salah satu tombol tersebut akan menjalankan

sesuai dengan tombol yang di klik. Pada tombol tambah data barang akan menampilkan *pop-up form* untuk *input* data barang, pada tombol *edit* data barang juga sama akan menampilkan *pop-up form* untuk *edit* data barang sesuai dengan data barang yang dipilih, pada tombol hapus data barang akan muncul *pop-up* untuk konfirmasi sebelum menghapus. Selanjutnya melanjutkan ke tahap untuk proses eksekusi program dan akan *me-render* halaman jika terdapat perubahan.

4.2.1.2 Activity Diagram Pemilik Mengelola Laporan Keuangan



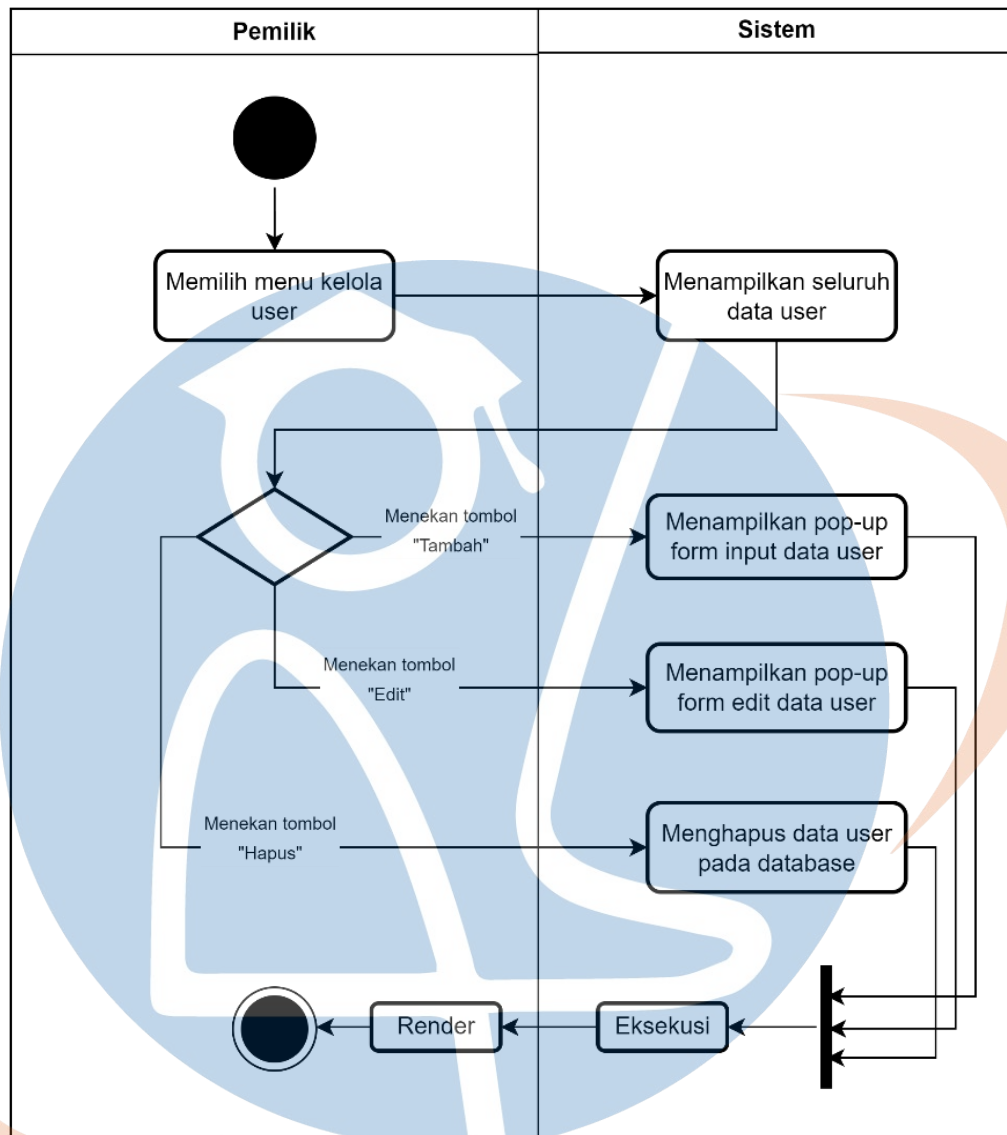
Gambar 4.3 Activity Diagram Pemilik Mengelola Laporan Keuangan

Dari rancangan *Activity diagram* yang terdapat pada Gambar 4.3 menjelaskan tentang alur proses untuk pemilik yang melakukan pengelolaan laporan keuangan. Pemilik memulai proses dengan cara memilih menu kelola laporan keuangan, yang didalamnya terdapat menu pemasukan dan pengeluaran. Jika pemilik klik menu pemasukan maka akan menampilkan seluruh data pemasukan, sementara jika pemilik klik menu pengeluaran maka akan menampilkan juga seluruh data pengeluaran. Selanjutnya pada halaman pengeluaran maupun halaman pemasukan, terdapat tombol tambah dan hapus. Jika klik tombol hapus maka sistem akan memproses untuk penghapusan data pada database, dan jika klik tombol tambah maka akan menampilkan *pop-up form* untuk *input* data. Selanjutnya melanjutkan ke tahap untuk proses eksekusi program dan akan *me-render* halaman jika terdapat perubahan.



STT - NF

4.2.1.3 Activity Diagram Pemilik Mengelola Users



Gambar 4.4 Activity Diagram Pemilik Mengelola Users

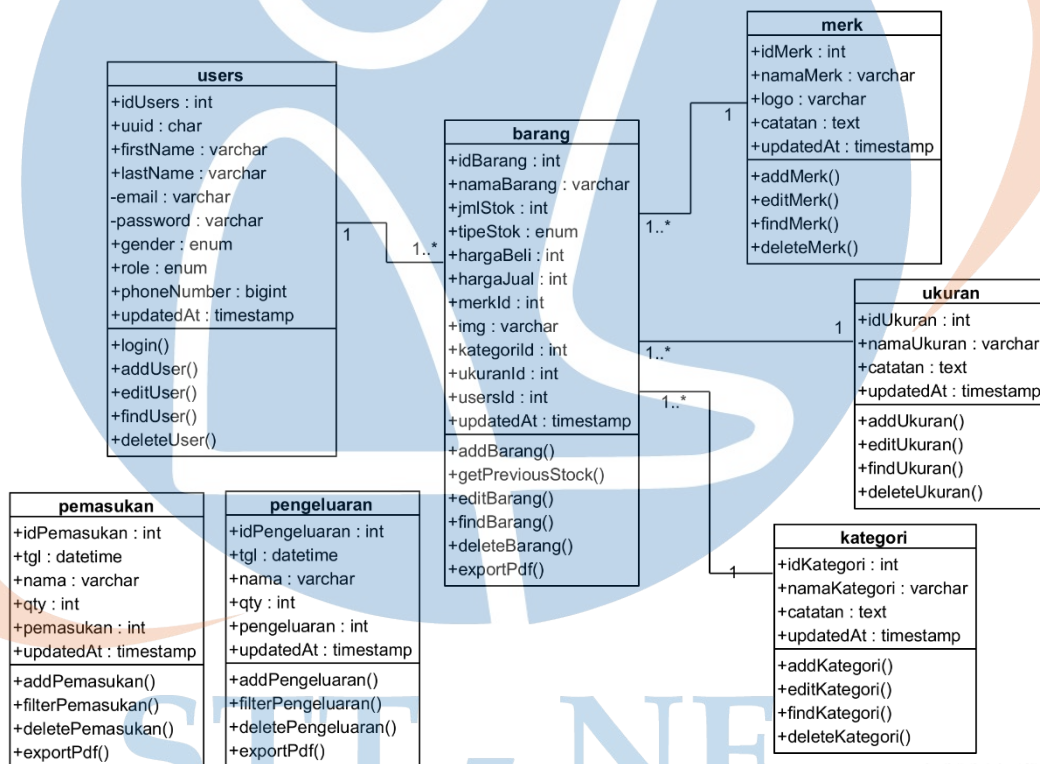
Dari rancangan *Activity diagram* yang terdapat pada Gambar 4.4 menjelaskan tentang alur proses untuk pemilik yang melakukan pengelolaan akun *users*. Pemilik memulai proses dengan cara memilih menu kelola *users*. Sistem secara langsung akan menampilkan seluruh data *user* pada halaman kelola *users*. Dalam halaman kelola *users* terdapat tombol tambah, *edit*, dan hapus, yang dimana ketika di klik salah satu tombol tersebut akan menjalankan sesuai dengan tombol yang di klik. Pada tombol tambah data *user* akan menampilkan *pop-up form* untuk *input* data *user* dan dalam *form* tersebut terdapat pemilihan *role* ketika ingin membuat *user* baru, pada tombol *edit* data *user* juga sama akan menampilkan *pop-*

up form untuk edit data user sesuai dengan data user yang dipilih, pada tombol hapus data user akan muncul *pop-up* untuk konfirmasi sebelum menghapus. Selanjutnya melanjutkan ke tahap untuk proses eksekusi program dan akan *render* halaman jika terdapat perubahan.

4.2.2 Class Diagram

Sebuah *Class diagram* yakni adalah representasi dari satu atau banyak objek yang mencakup berbagai atribut yang akan digunakan pada objek tersebut, serta hubungan antara satu objek dengan objek lainnya. Pada implementasi penulis, disini akan menggunakan rancangan *class diagram*.

4.2.2.1 Rancangan Class Diagram



Gambar 4.5 Rancangan Class Diagram

Sebuah *class diagram* adalah representasi *visual* dari struktur sebuah sistem yang mencakup berbagai objek, atribut, dan relasi antara objek-objek tersebut. *Class diagram* digunakan untuk memodelkan komponen-komponen sistem secara rinci, sehingga memudahkan dalam memahami alur kerja dan interaksi antar komponen. Berikut adalah entitas dan atribut yang akan dijelaskan pada tabel 4.5 sebagai berikut:

Tabel 4.5 Class Diagram

Entitas	Atribut
<i>users</i>	<i>idUsers, uuid, firstName, lastName, email, password, gender, role, phoneNumber, updatedAt.</i>
barang	<i>idBarang, namaBarang, jmlStok, tipeStok, hargaBeli, hargaJual, merkId, img, kategoriId, ukuranId, usersId, updatedAt.</i>
merk	<i>idMerk, namaMerk, logo, catatan, updatedAt.</i>
kategori	<i>idKategori, namaKategori, catatan, updatedAt.</i>
ukuran	<i>idUkuran, namaUkuran, catatan, updatedAt.</i>
pemasukan	<i>idPemasukan, tgl, nama, qty, pemasukan, updatedAt.</i>
pengeluaran	<i>idPengeluaran, tgl, nama, qty, pengeluaran, updatedAt.</i>

Penjelasan untuk hubungan atau relasi antar entitas pada Tabel 4.5 adalah sebagai berikut:

1. Entitas *users* dan barang memiliki hubungan *one to many*, yaitu satu *user* bisa memasukkan satu atau banyak barang.
2. Entitas barang dan merk memiliki hubungan *one to many*, yaitu satu atau banyak barang hanya bisa menggunakan atau mempunyai satu merk.
3. Entitas barang dan kategori memiliki hubungan *one to many*, yaitu satu atau banyak barang hanya bisa memiliki satu kategori.
4. Entitas barang dan ukuran memiliki hubungan *one to many*, yaitu satu atau banyak barang hanya dapat mempunyai satu ukuran.
5. Entitas pemasukan tidak memiliki relasi atau hubungan dengan entitas lain.
6. Entitas pengeluaran tidak memiliki relasi atau hubungan dengan entitas lain.

4.2.3 Perancangan Antar Muka Sistem (*Low-Fi*)

Perancangan antarmuka sistem yang akan dibuat menggunakan *low-fidelity prototype*. *Prototype* ini akan menampilkan tampilan sistem yang akan dibuat.

4.2.3.1 *Low-Fi* Tampilan Login



Gambar 4.6 *Low-fi* Tampilan *Login*

Pada tampilan *low-fi* halaman *login*, menampilkan tampilan awal dari sistem yang baru di akses. Pengguna masuk menggunakan akun yang telah disediakan dengan cara menekan tombol *login* pada tampilan.

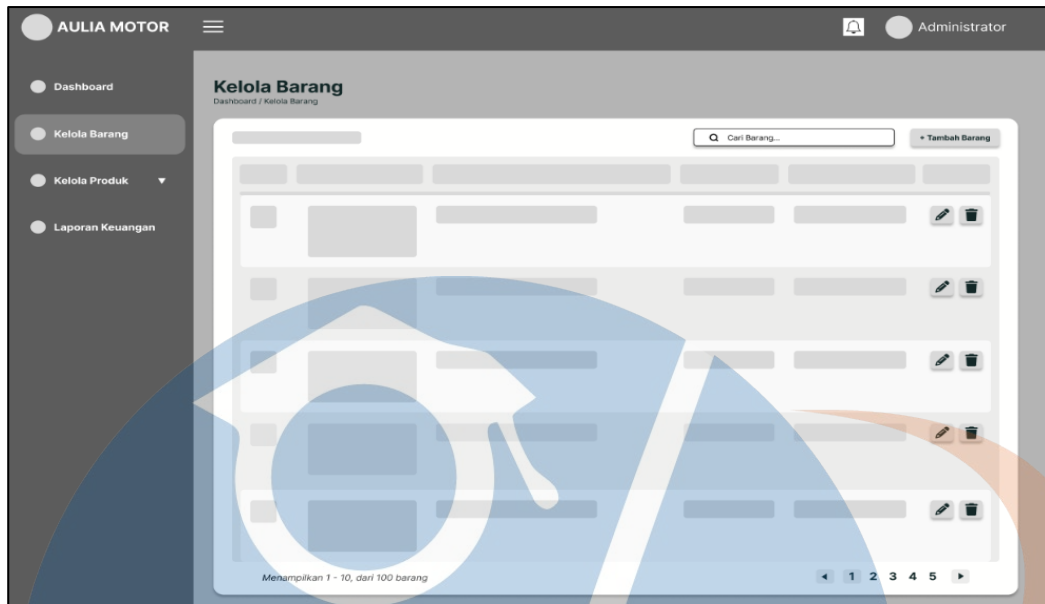
4.2.3.2 *Low-Fi* Tampilan *Dashboard*



Gambar 4.7 *Low-fi* Tampilan *Dashboard*

Pada tampilan *low-fi* halaman *dashboard*, *user* dapat melihat *status* atau statistik. Baik disajikan dalam grafik ataupun angka, dan juga terdapat beberapa data barang yang ditampilkan pada halaman *dashboard*.

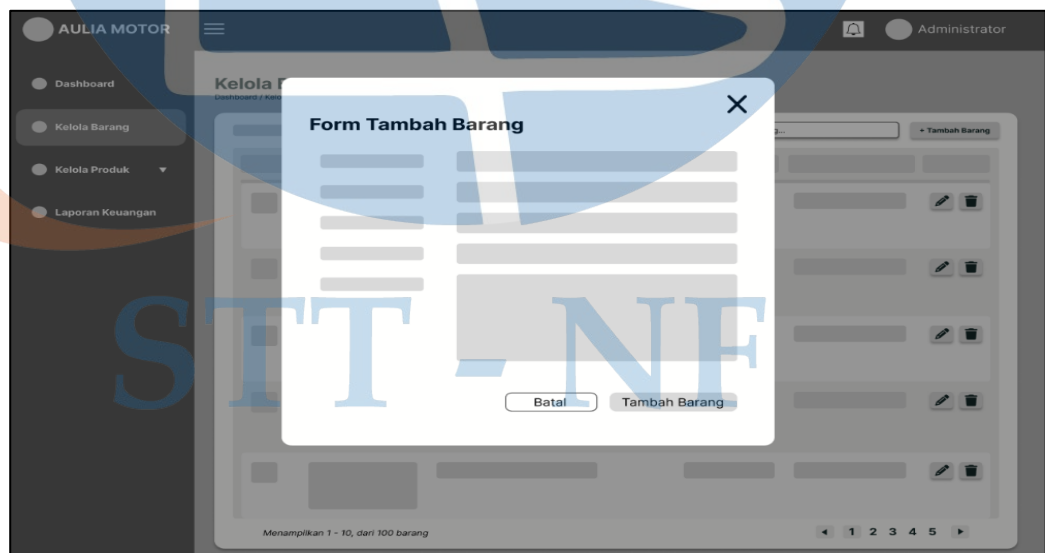
4.2.3.3 Low-Fi Tampilan Kelola Barang



Gambar 4.8 Low-fi Tampilan Kelola Barang

Pada tampilan *low-fi* halaman kelola barang, tampil ketika *user* memilih menu “Kelola Barang”. Pada halaman ini akan menampilkan seluruh data barang, termasuk menu tambah, pencarian, *edit* dan hapus.

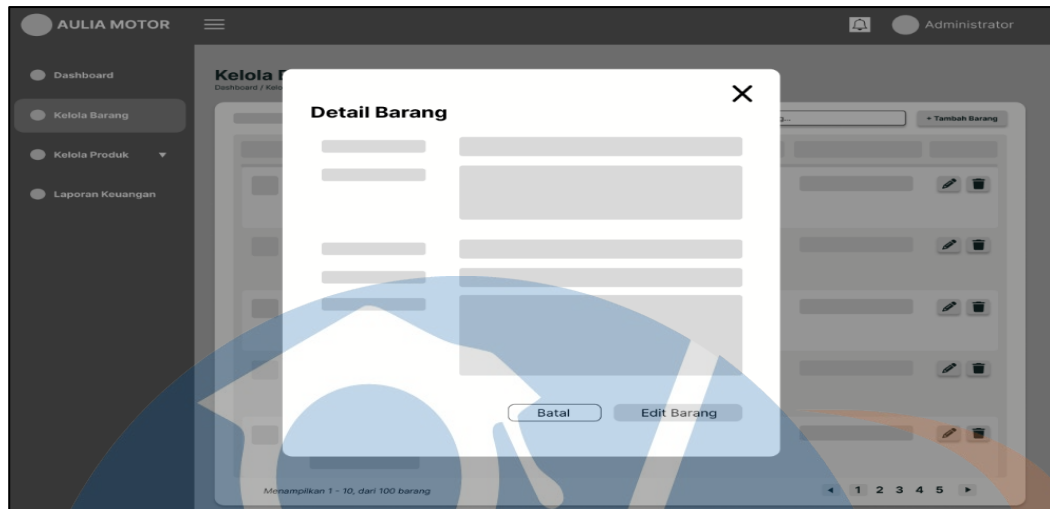
4.2.3.4 Low-Fi Tampilan Tambah Barang



Gambar 4.9 Low-fi Tampilan Tambah Barang

Pada tampilan *low-fi* halaman tambah barang, ketika *user* menekan tombol “Tambah Barang” maka akan muncul *pop-up form* untuk memasukkan data barang.

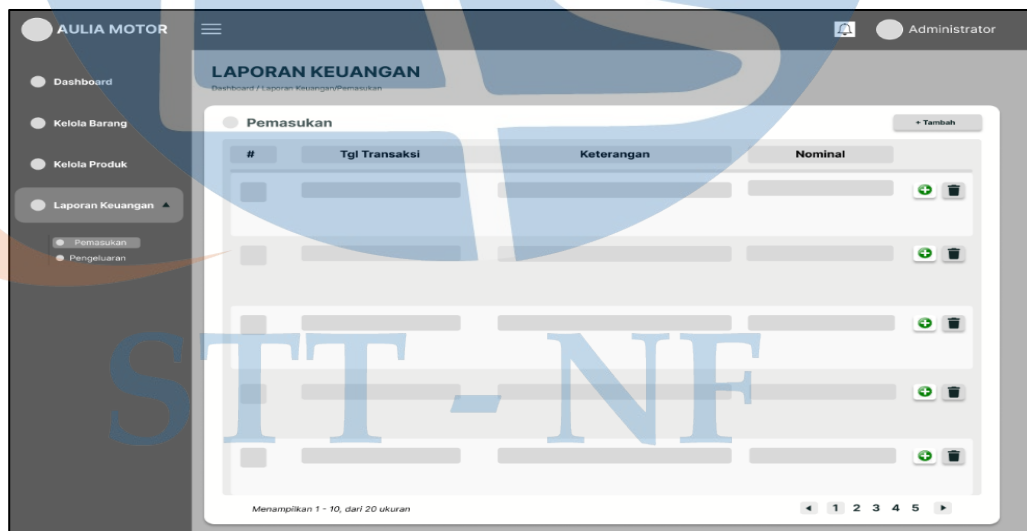
4.2.3.5 Low-Fi Tampilan *Detail* dan *Edit* Barang



Gambar 4.10 Low-fi Tampilan *Detail* dan *Edit* Barang

Pada tampilan *low-fi* halaman *detail* dan *edit* barang, ketika *user* menekan ikon pensil maka akan muncul *pop-up* berupa *form* yang menampilkan data barang sesuai dengan yang dipilih, dan juga menampilkan detail lain yang tidak ditampilkan pada halaman kelola barang.

4.2.3.6 Low-Fi Tampilan Laporan Keuangan



Gambar 4.11 Low-fi Tampilan Laporan Keuangan

Pada tampilan *low-fi* halaman laporan keuangan, ketika *user* memilih menu “Laporan Keuangan” akan muncul *sub-menu* “Pemasukan” dan “Pengeluaran”. Halaman ini menampilkan data-data tentang pemasukan atau pengeluaran pada masing-masing *sub-menu*.

4.2.3.7 *Low-Fi* Tampilan Tambah Laporan Keuangan



Gambar 4.12 *Low-fi* Tampilan Tambah Laporan Keuangan

Pada tampilan *low-fi* halaman laporan keuangan, ketika *user* melakukan interaksi dengan tombol tambah, maka akan menampilkan *pop-up form* untuk tambah data pemasukan atau pengeluaran secara *manual*.

4.3 Pengkodean (*Coding*)

Setelah melakukan tahap perancangan yang menghasilkan gambaran dari fungsional sistem. Pada tahap pengkodean ini akan diimplementasikan menggunakan *node.js* yang sudah terintegrasi dengan *express* pada bagian *back-end* antara lain algoritma menampilkan barang dan cari barang, tambah barang, *edit* barang, *endpoint* barang, dan *endpoint final routes*. Pada bagian *front-end* akan menggunakan *framework react js*, antara lain akan menampilkan tampilan halaman dari yang sudah dirancang sebelumnya pada sub-bab 4.2.3 Perancangan Antar Muka Sistem (*Low-Fi*).

4.3.1 Implementasi Algoritma Menampilkan Barang dan Cari Barang

Implementasi algoritma untuk menampilkan data barang dan hasil dari cari barang menggunakan *query*. Akan menghasilkan *source code* yang dapat dilihat pada tabel dibawah ini.

Tabel 4.6 Implementasi Algoritma Menampilkan Barang dan Cari Barang (*BarangController.js*)

```
1. const query = require("../database");
2.
3. export const getBarang = async (req, res) => {
4.   try {
5.     const data = await query(
6.       `SELECT barang.id AS id_barang, n_barang, jml_stok,
7.         tipe_stok, h_beli, h_jual, merk.id AS id_merk,
8.         n_merk, img, kategori.id AS id_kategori, n_kategori,
9.         ukuran.id AS id_ukuran, n_ukuran, f_name, l_name,
10.        barang.updated_at AS waktu
11.        FROM barang
12.        INNER JOIN kategori ON kategori.id =
13.        barang.kategori_id
14.        INNER JOIN merk ON merk.id = barang.merk_id
15.        INNER JOIN ukuran ON ukuran.id = barang.ukuran_id
16.        INNER JOIN users ON users.id = barang.users_id
17.        ORDER BY waktu DESC`
18.    );
19.
20.    const { q } = req.query;
21.    const keys = ["n_barang", "n_merk"];
22.    const search = (data) => {
23.      return data.filter((item) =>
24.        keys.some((key) =>
25.          item[key].toLowerCase().includes(q)
26.        )
27.      );
28.
29.      return res.status(200).json({
30.        success: true,
31.        message: "Menampilkan seluruh Data Barang",
32.        data: data,
33.        qq: search(data),
34.      });
35.    } catch (error) {
36.      return res.status(400).json({
37.        success: false,
38.        message: "Data Barang tidak ditemukan / Gagal",
39.      });
40.    }
41.  };
42. }
```

Pada Tabel 4.6 menampilkan potongan kode dari fungsi *method getBarang* untuk menampilkan seluruh data barang dan menampilkan hasil dari pencarian barang. Pada baris 5 hingga 18 adalah proses menampilkan data barang menggunakan *query*. Pada baris 20 hingga baris 27 adalah proses menampilkan pencarian barang berdasarkan *query* yang diterima, kata kunci pencarian barang bisa menuju pada nama barang atau nama merk. Sementara pada baris 29 hingga baris 34 merupakan data yang akan di *return* dalam bentuk *json* ketika kondisi

terpenuhi, untuk data barang akan disimpan pada properti 'data', dan untuk pencarian barang akan disimpan pada properti 'qq'.

4.3.2 Implementasi Algoritma Tambah Barang

Implementasi algoritma untuk menambahkan data barang menggunakan *query*, *path*, dan *fs*. Akan menghasilkan *source code* yang dapat dilihat pada tabel dibawah ini.

Tabel 4.7 Implementasi Algoritma Tambah Data Barang (*BarangController.js*)

```
1.  const query = require("../database");
2.  const path = require("path");
3.  const fs = require("fs");
4.
5.  export const createBarang = async (req, res) => {
6.    if (req.files === null) {
7.      try {
8.        const {
9.          n_barang, jml_stok, tipe_stok, h_beli, h_jual,
10.         merk_id, img, kategori_id, ukuran_id, users_id,
11.        } = req.body;
12.
13.        // validation
14.        // ...
15.        // end of validation
16.
17.        const result = await query(
18.          `
19.            insert into barang(n_barang, jml_stok, tipe_stok,
20.              h_beli, h_jual, merk_id, img, kategori_id,
21.              ukuran_id, users_id) values(?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
22.          `,
23.          [
24.            n_barang, jml_stok, tipe_stok, h_beli,
25.            h_jual, merk_id, img, kategori_id, ukuran_id,
26.            users_id,
27.          ]
28.        );
29.
30.        const date = new Date();
31.        const formattedDate =
32.          date.toISOString().split("T")[0];
33.        const barangId = result.insertId;
34.        const pengeluaran = jml_stok * h_beli;
35.        // Tambahkan pengeluaran ke laporan keuangan
36.        await query(
37.          `INSERT INTO pengeluaran (tgl, nama, qty,
38.            pengeluaran) VALUES (?, ?, ?, ?)`,
39.          [formattedDate, n_barang, jml_stok, pengeluaran]
40.        );
41.
42.        return res.status(200).json({
43.          success: true,
44.          message: "Barang berhasil ditambahkan!",
45.          data: { id: barangId, ...req.body },
```

```

46.     });
47.   } catch (error) {
48.     console.error(error.message);
49.     return res.status(400).json({
50.       success: false,
51.       message: "Data Barang Gagal ditambahkan",
52.     });
53.   }
54. } else {
55.   const file = req.files.img;
56.   const fileSize = file.data.length;
57.   const ext = path.extname(file.name);
58.   const fileName = Date.now() + "_" + file.md5 + ext;
59.   const url =
60.   `${req.protocol}://${req.get("host")}/images/${fileName}`;
61.   // const url = `/images/${fileName}`;
62.   const allowedType = [".png", ".jpg", ".jpeg"];
63.
64.   // check file
65.   // ...
66.   // end of check file
67.
68.   const {
69.     n_barang, jml_stok, tipe_stok, h_beli, h_jual,
70.     merk_id, img, kategori_id, ukuran_id, users_id,
71.   } = req.body;
72.
73.   file.mv(`./public/images/${fileName}`, async (err) =>
74.   {
75.     try {
76.       const result = await query(
77.         `insert into barang(n_barang, jml_stok,
78.           tipe_stok, h_beli, h_jual, merk_id, img,
79.           kategori_id, ukuran_id, users_id)
80.         values(?, ?, ?, ?, ?, ?, ?, ?, ?, ?)`,
81.         [ n_barang, jml_stok, tipe_stok, h_beli, h_jual,
82.           merk_id, img, kategori_id, ukuran_id, users_id,
83.         ]
84.       );
85.
86.       const date = new Date();
87.       const formattedDate =
88.         date.toISOString().split("T")[0];
89.       const barangId = result.insertId;
90.       const pengeluaran = jml_stok * h_beli;
91.       // Tambahkan pengeluaran ke laporan keuangan
92.       await query(
93.         `INSERT INTO pengeluaran (tgl, nama, qty,
94.           pengeluaran) VALUES (?, ?, ?, ?)`,
95.         [formattedDate, n_barang, jml_stok, pengeluaran]
96.       );
97.
98.       return res.status(200).json({
99.         success: true,
100.        message: "Barang berhasil ditambahkan!",
101.        data: { id: barangId, ...req.body },
102.      });
103.    } catch (error) {
104.      console.error(error.message);

```

105.	return res.status(400).json({
106.	success: false,
107.	message: "Data Barang Gagal ditambahkan",
108.	});
109.	}
110.	});
111.	}
112.	};

Pada Tabel 4.7 menampilkan potongan kode dari fungsi *createBarang* untuk menambahkan data barang. Pada baris 8 hingga baris 11 adalah parameter yang dibutuhkan untuk memasukkan data pada *database*. Pada baris 17 hingga baris 40 adalah proses menambahkan data kedalam tabel barang dan juga ke tabel pengeluaran. Pada baris 42 hingga baris 46, jika kondisi logika *if* pada baris 6 terpenuhi maka akan memberikan *return* data dalam bentuk *json*.

Jika kondisi *if* pada baris 6 tidak terpenuhi maka akan masuk pada logika *else*, yang pada bagian baris 55 hingga baris 66 adalah proses pendefinisian nama *file* gambar dan validasi gambar. Pada baris 73 hingga baris 110 adalah proses penyimpanan gambar pada direktori *'/images'* yang ada dalam *folder public*, dan penambahan data pada tabel barang dan tabel pengeluaran menggunakan *query*. Sementara pada baris 98 hingga baris 102 merupakan data yang akan di *return* dalam bentuk *json* ketika kondisi terpenuhi.

4.3.3 Implementasi Algoritma *Edit Barang*

Implementasi algoritma untuk *edit* data barang menggunakan *query*, *path*, dan *fs*. Akan menghasilkan *source code* yang dapat dilihat pada tabel dibawah ini.

Tabel 4.8 Implementasi Algoritma *Edit Barang* (*BarangController.js*)

2.	const query = require("../database");
3.	const path = require("path");
4.	const fs = require("fs");
5.	
6.	export const updateBarang = async (req, res) => {
7.	const { id } = req.params;
8.	
9.	const barang = await query(`SELECT * FROM barang
10.	WHERE id = ?`, [id]);
11.	
12.	let fileName = "";
13.	if (req.files === null) {
14.	fileName = barang[0].img;
15.	} else {
16.	const file = req.files.img;
17.	const fileSize = file.data.length;
18.	const ext = path.extname(file.name);
19.	fileName = Date.now() + "_" + file.md5 + ext;

```

20.     const allowedType = [".png", ".jpeg", ".jpg"];
21.
22.     // check file
23.     // ...
24.     // end of check file
25.
26.     const theImg = barang[0].img;
27.     const split = theImg.split("images/")[1];
28.     const filePath = `./public/images/${split}`;
29.     if (split !== undefined) {
30.         fs.unlinkSync(filePath);
31.     }
32.
33.     file.mv(`./public/images/${fileName}`, (err) => {
34.         if (err) {
35.             return res.status(500).json({ success: false,
36.                 message: err.message });
37.         }
38.     });
39. }
40.
41.     const url =
42. ` ${req.protocol}://${req.get("host")}/images/${fileName}`;
43.
44.     try {
45.         const {
46.             n_barang, jml_stok, tipe_stok, h_beli, h_jual,
47.             merk_id, img = url, kategori_id, ukuran_id, users_id,
48.         } = req.body;
49.
50.         // validation
51.         // ...
52.         // end of validation
53.
54.         const data = await query(
55.             `UPDATE barang SET n_barang = ?, jml_stok = ?,
56.                 tipe_stok = ?, h_beli = ?, h_jual = ?, merk_id = ?,
57.                 img = ?, kategori_id = ?, ukuran_id = ?, users_id=?
58.                 WHERE id = ?`,
59.             [
60.                 n_barang, jml_stok, tipe_stok, h_beli, h_jual,
61.                 merk_id, img, kategori_id, ukuran_id, users_id, id,
62.             ]
63.         );
64.
65.         return res.status(200).json({
66.             success: true,
67.             message: "Data Barang berhasil diupdate!",
68.             data: { id, ...req.body },
69.         });
70.     } catch (err) {
71.         console.error(err.message);
72.         return res.status(500).json({
73.             success: false,
74.             message: err.message,
75.         });
76.     }
77. };

```

Pada Tabel 4.8 menampilkan potongan kode dari fungsi *editBarang* untuk *edit* data barang yang ingin di *edit*. Pada baris 15 hingga baris 39 adalah logika ketika barang yang di *edit* sebelumnya memiliki foto atau tidak. Sementara pada baris 65 hingga baris 69 merupakan data yang akan di *return* dalam bentuk *json* ketika kondisi terpenuhi.

4.3.4 Implementasi Algoritma *Endpoint* Barang

Implementasi algoritma untuk *handle endpoint* pada barang menggunakan *express*, *routes* dan *file BarangController*. Akan menghasilkan *source code* yang dapat dilihat pada tabel dibawah ini.

Tabel 4.9 Implementasi Algoritma *Endpoint* Barang (*barangRoute.js*)

```
1. const express = require("express");
2. const routes = express();
3. const {
4.   getBarang,
5.   findBarangById,
6.   createBarang,
7.   updateBarang,
8.   deleteBarang,
9. } = require("../controller/BarangController");
10.
11. routes.get("/barang", getBarang);
12. routes.get("/barang/:id", findBarangById);
13. routes.post("/barang", createBarang);
14. routes.put("/barang/:id", updateBarang);
15. routes.delete("/barang/:id", deleteBarang);
16.
17. module.exports = routes;
```

Pada Tabel 4.9 menampilkan potongan kode untuk menangani *routes* bagian barang. Untuk menampilkan seluruh data barang menggunakan *endpoint* *‘/barang’*, untuk pencarian barang dengan *id* menggunakan *endpoint* *‘/barang:id’*, untuk menambahkan barang menggunakan *endpoint* *‘/barang’* dengan metode *post*, untuk *edit* barang menggunakan *endpoint* *‘/barang’* dengan metode *put*, dan juga hapus barang menggunakan *endpoint* *‘/barang:id’* dengan menggunakan metode *delete*. Selanjutnya *routes* tersebut akan di *export* menggunakan *module.exports*.

4.3.5 Implementasi Algoritma *Endpoint* Final

Implementasi algoritma untuk *handle endpoint users*, barang, kategori, merk, ukuran, pengeluaran, dan *auth* menggunakan *express* dan *app*, serta *file-file routes*.

Tabel 4.10 Implementasi Algoritma *Endpoint Final (index.js)*

1.	<code>const express = require("express");</code>
2.	<code>const app = express();</code>
3.	
4.	<code>const users = require("../UserRoute");</code>
5.	<code>const barang = require("../BarangRoute");</code>
6.	<code>const kategori = require("../KategoriRoute");</code>
7.	<code>const merk = require("../MerkRoute");</code>
8.	<code>const ukuran = require("../UkuranRoute");</code>
9.	<code>const pengeluaran = require("../PengeluaranRoute");</code>
10.	<code>const auth = require("../auth/AuthRoute");</code>
11.	
12.	<code>const apiUrl = "/api/v1";</code>
13.	
14.	<code>app.use(apiUrl, users);</code>
15.	<code>app.use(apiUrl, barang);</code>
16.	<code>app.use(apiUrl, kategori);</code>
17.	<code>app.use(apiUrl, merk);</code>
18.	<code>app.use(apiUrl, ukuran);</code>
19.	<code>app.use(apiUrl, pengeluaran);</code>
20.	<code>app.use(apiUrl, auth);</code>
21.	
22.	<code>module.exports = app;</code>

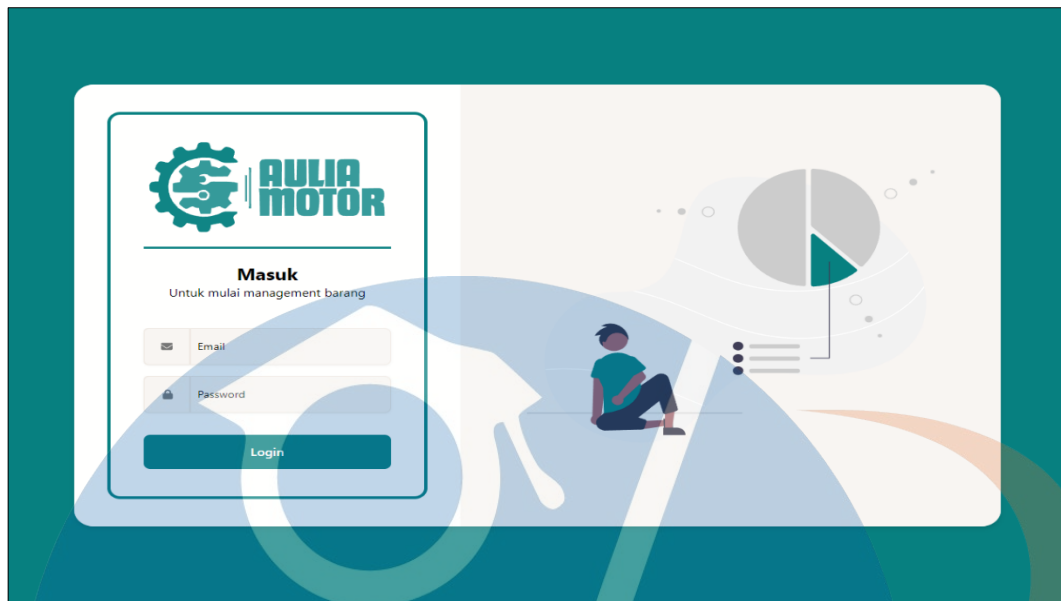
Pada Tabel 4.10 menampilkan potongan kode untuk menangani *routes* pada bagian *users*, *barang*, *kategori*, *merk*, *ukuran*, *pengeluaran* dan juga *auth*. Pada *file index.js* ini, hasil *routes* yang telah didefinisikan sebelumnya hanya mendapat penambahan *url* berupa *'/api/v1'* dengan tujuan untuk memberi tahu jika ini adalah *url api*. Selanjutnya *app* yang sudah didefinisikan tersebut akan di *export* pada *module.exports* yang akan digunakan pada *index.js* pada direktori *root*.

4.3.6 Hasil Implementasi Antar Muka Sistem

Hasil implementasi antar muka sistem adalah hasil dari implementasi perancangan antar muka sistem yang telah dibuat pada *sub bab* 4.2.3 Perancangan Antar Muka Sistem (*Low-Fi*).

STT - NF

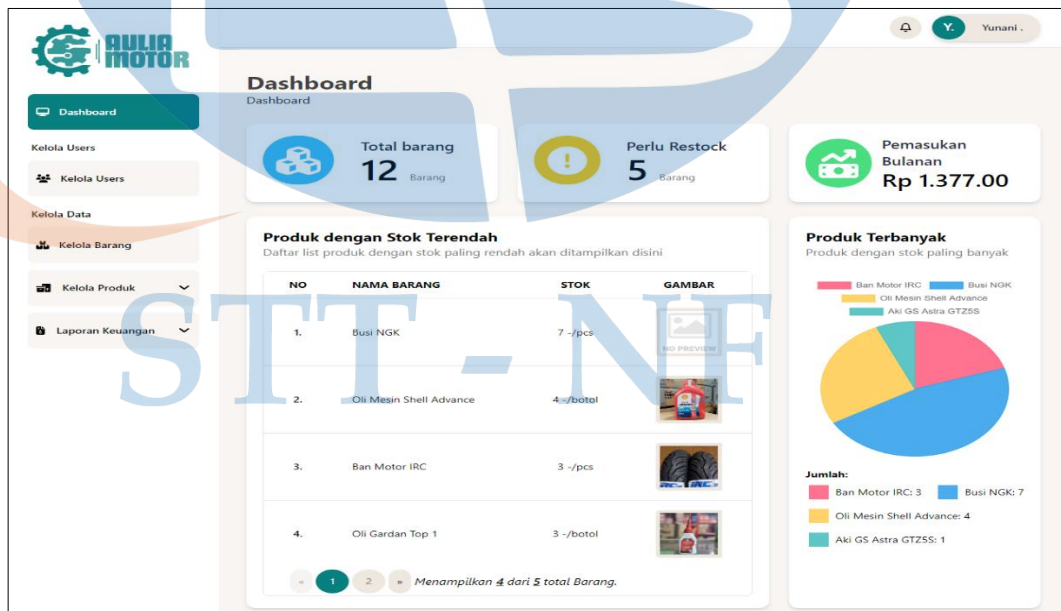
4.3.6.1 Tampilan Halaman *Login*



Gambar 4.13 Tampilan Halaman *Login*

Pada gambar 4.13 ditampilkan implementasi dari *login*, dimana pengguna yang sudah mempunyai akun dapat masuk ke dalam sistem. Jika belum mempunyai akun harus melakukan daftar akun terlebih dahulu melalui pemilik.

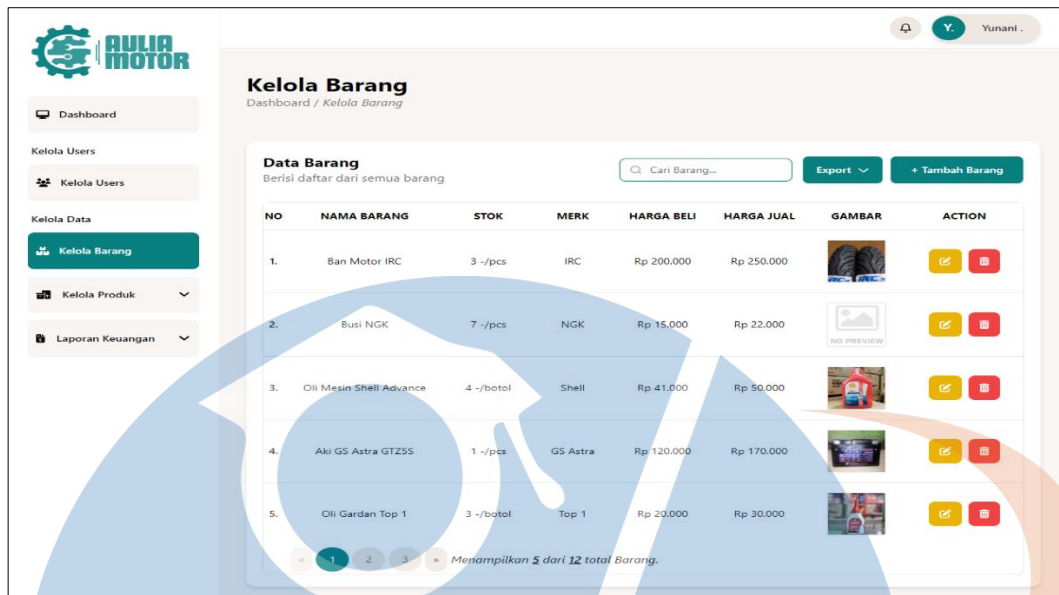
4.3.6.2 Tampilan Halaman *Dashboard*



Gambar 4.14 Tampilan Halaman *Dashboard*

Pada gambar 4.14 ditampilkan implementasi dari *dashboard*, dimana ketika berhasil *login* akan langsung diarahkan pada halaman *dashboard*.

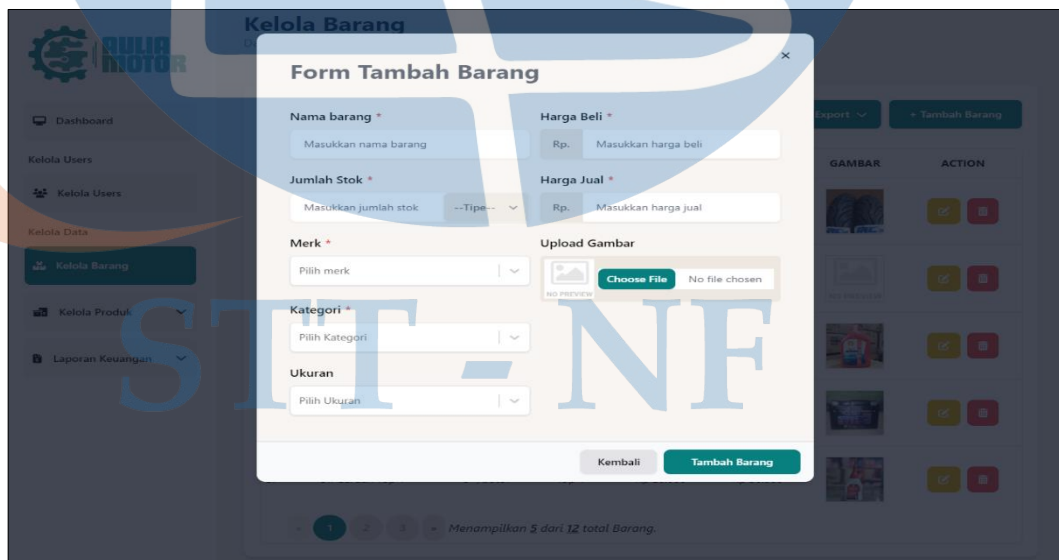
4.3.6.3 Tampilan Halaman Kelola Barang



Gambar 4.15 Tampilan Halaman Kelola Barang

Pada gambar 4.15 ditampilkan implementasi dari kelola barang, dimana pada halaman ini pengguna dapat melihat seluruh daftar barang, cari barang, *export* pdf, tambah barang, *edit* barang, serta hapus barang.

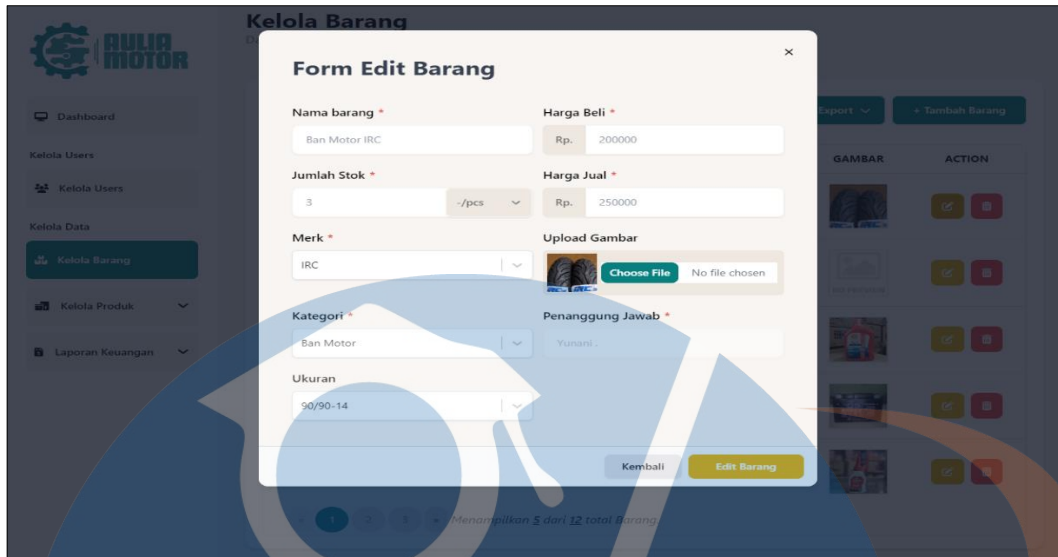
4.3.6.4 Tampilan Halaman Tambah Barang



Gambar 4.16 Tampilan Halaman Tambah Barang

Pada gambar 4.16 ditampilkan implementasi dari tambah barang, dimana *form* pengisian tersebut berupa *pop-up* dan berisi beberapa *input field* wajib yang harus dimasukkan.

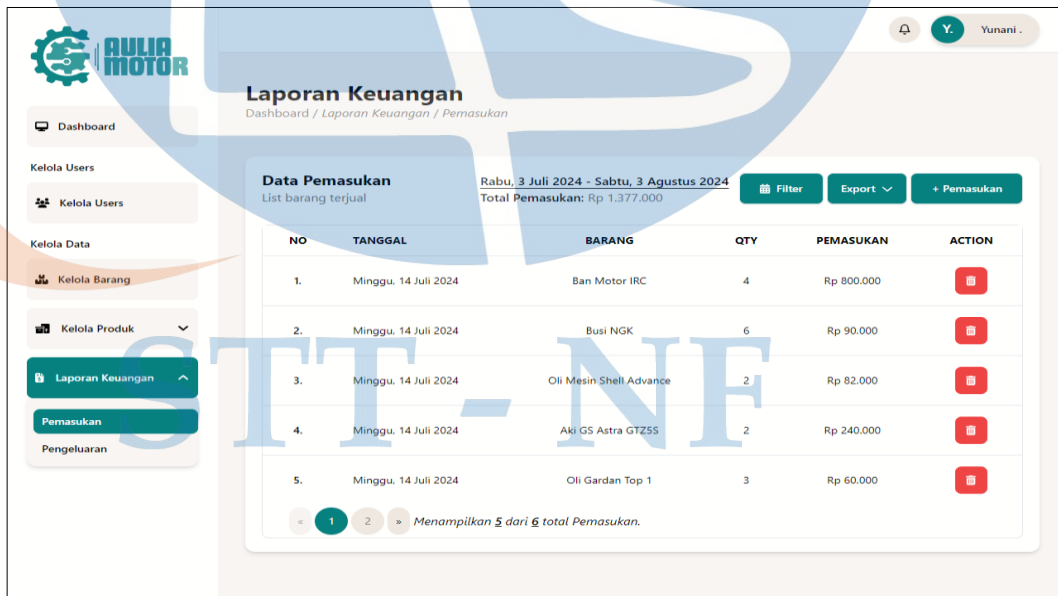
4.3.6.5 Tampilan *Detail* dan *Edit* Barang



Gambar 4.17 Tampilan *Detail* dan *Edit* Barang

Pada gambar 4.17 ditampilkan implementasi dari *detail* dan *edit* barang, dimana akan menampilkan *pop-up* yang sesuai dengan barang yang di pilih, serta menampilkan *preview* gambar barang.

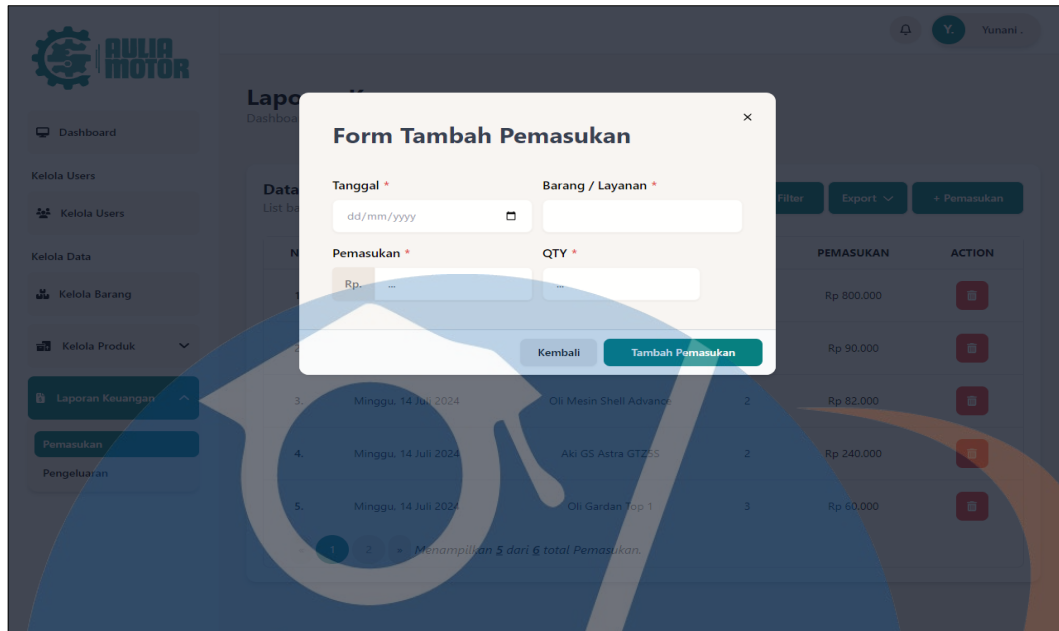
4.3.6.6 Tampilan Laporan Keuangan



Gambar 4.18 Tampilan Laporan Keuangan

Pada gambar 4.18 ditampilkan implementasi dari laporan keuangan bagian pemasukan, dimana pengguna dapat melakukan *filter* tanggal sesuai yang di inginkan, serta *export* data sebagai pdf.

4.3.6.7 Tampilan Tambah Laporan Keuangan



Gambar 4.19 Tampilan Tambah Laporan Keuangan

Pada gambar 4.19 ditampilkan implementasi dari tambah laporan keuangan, dimana menampilkan *pop-up* untuk menambahkan data pemasukan yang ingin di *input*. Opsi lain dari tambah barang *manual* adalah ketika terjadi perubahan pada stok barang, maka akan otomatis masuk pada laporan keuangan sehingga akan memudahkan pengguna.

4.4 Pengujian (*Testing*)

Sebelum sistem diimplementasikan atau langsung digunakan oleh pengguna, pada tahap pengujian ini terlebih dahulu akan melalui tahap pengujian dan analisis untuk memastikan bahwa sistem tidak menyimpang dari harapan *stackholder* dan tidak terjadi kesalahan dalam sistem yang perlu dilakukan ketika sistem diimplementasikan secara langsung. Pada tahap ini, pengujian sistem akan dilakukan dengan menggunakan teknik pengujian *User Acceptance Testing (UAT)*.

4.4.1 *User Acceptance Testing (UAT)*

Pengujian *user acceptance testing (UAT)* ini akan dilakukan dengan menggunakan *test case* pada fitur kelola laporan keuangan dan kelola *user*, yang akan dilakukan oleh pemilik. Sementara pengujian yang akan dilakukan oleh karyawan akan menguji pada fitur kelola barang dan kelola produk. Untuk mengetahui berhasil atau tidaknya program dijalankan, maka akan diberikan

beberapa *test case* yang akan dilakukan oleh pengguna. Beberapa *test case* akan dilakukan pada fitur kelola barang, kelola produk, kelola laporan keuangan dan kelola *user* akan dijelaskan pada Tabel 4.11 - 4.14.

Tabel 4.11 *Test Case* Kelola Barang

<p>Step No. 1 Deskripsi : Karyawan telah berhasil <i>login</i> dan memilih menu kelola barang. Hasil yang diharapkan : Menampilkan halaman kelola barang beserta datanya. Hasil Pengujian : Sesuai dengan hasil yang diharapkan.</p> <p>Step No. 2 Deskripsi : Setelah menampilkan halaman kelola barang. Mengisi data pada <i>form</i> tambah barang sebanyak 3 kali. a) Tambah barang pertama “tanpa menggunakan gambar”. b) Tambah barang kedua dan ketiga “dengan menggunakan gambar”. Hasil yang diharapkan : Data barang yang di <i>input</i> tampil pada halaman kelola barang. Hasil Pengujian : Sesuai dengan hasil yang diharapkan.</p> <p>Step No. 3 Deskripsi : Setelah data berhasil di <i>input</i>. <i>Edit</i> salah satu data barang dengan “mengganti data lain dan gambar”. Hasil yang diharapkan : Data barang yang di <i>edit</i> “berubah” dan menampilkan gambar yang di <i>input</i> sebelumnya. Hasil Pengujian : Sesuai dengan hasil yang diharapkan.</p> <p>Step No. 4 Deskripsi: Setelah berhasil <i>edit</i> data barang. Memasukkan <i>keyword</i> barang pada “kolom pencarian”. Hasil yang diharapkan : Data barang muncul sesuai <i>keyword</i> yang dimasukkan. Hasil Pengujian : Sesuai dengan hasil yang diharapkan.</p> <p>Step No. 5 Deskripsi : Mencoba “menghapus” salah satu data barang. Hasil yang diharapkan : Data berhasil dihapus. Hasil Pengujian : Sesuai dengan hasil yang diharapkan.</p> <p>STATUS : Tes Lulus</p>

Tabel 4.12 Test Case Kelola Produk

<p>Step No. 1 Deskripsi : Karyawan telah berhasil <i>login</i> dan memilih menu kelola produk, yang ketika di klik akan menampilkan <i>sub-menu</i>. a) Klik pada <i>sub-menu</i> merk produk. b) Klik pada <i>sub-menu</i> kategori. c) Klik pada <i>sub-menu</i> ukuran. Hasil yang diharapkan : Menampilkan data merk, kategori, dan ukuran yang sesuai pada halaman kelola produk. Hasil Pengujian : Sesuai dengan hasil yang diharapkan.</p> <p>Step No. 2 Deskripsi : Setelah menampilkan halaman terkait, tambahkan data dan memasukkan data <i>valid</i> pada <i>form</i>. a) Tambah data pada merk produk. b) Tambah data pada kategori. c) Tambah data pada ukuran. Hasil yang diharapkan : Data produk berhasil ditambahkan dan tampil pada halaman. Hasil Pengujian : Sesuai dengan hasil yang diharapkan.</p> <p>STATUS : Tes Lulus</p>

Tabel 4.13 Test Case Kelola Laporan Keuangan

<p>Step No. 1 Deskripsi : Pemilik telah berhasil <i>login</i> dan memilih menu laporan keuangan, yang ketika di klik akan menampilkan <i>sub-menu</i>. a) Klik pada <i>sub-menu</i> pemasukan. b) Klik pada <i>sub-menu</i> pengeluaran. Hasil yang diharapkan : Menampilkan data pemasukan dan pengeluaran yang sesuai pada halaman laporan keuangan. Hasil Pengujian : Sesuai dengan hasil yang diharapkan.</p> <p>Step No. 2 Deskripsi : Setelah menampilkan halaman terkait, tambahkan data dan memasukkan data pada <i>form</i>. a) Tambah data pada pemasukan b) Tambah data pada pengeluaran Hasil yang diharapkan : Data keuangan berhasil ditambahkan dan tampil pada halaman. Hasil Pengujian : Sesuai dengan hasil yang diharapkan.</p> <p>Step No. 3 Deskripsi : Setelah berhasil menambahkan data laporan keuangan, <i>filter</i> data laporan keuangan. a) <i>Filter</i> berdasarkan hari, bulan dan tahun pada pemasukan.</p>
--

b) *Filter* berdasarkan hari, bulan, dan tahun pada pengeluaran.
Hasil yang diharapkan : Data yang ditampilkan akan ter-*filter* sesuai dengan hari, bulan dan tahun yang dimasukkan.

Hasil Pengujian : Sesuai dengan hasil yang diharapkan.

STATUS : Tes Lulus

Tabel 4.14 *Test Case* Kelola Users

Step No. 1

Deskripsi : Pemilik telah berhasil *login* dan memilih menu kelola *users*.

Hasil yang diharapkan : Menampilkan halaman kelola *users* beserta data-data *users*.

Hasil Pengujian : Sesuai dengan hasil yang diharapkan.

Step No. 2

Deskripsi : Setelah menampilkan halaman kelola *users*. Mengisi data pada *form* tambah *user*.

a) Tambah *user* dengan *role* karyawan.

b) Tambah *user* dengan *role* pemilik.

Hasil yang diharapkan : *User* baru berhasil dibuat dan tampil pada halaman kelola *users*.

Hasil Pengujian : Sesuai dengan hasil yang diharapkan.

Step No. 3

Deskripsi : Setelah penambahan *user* berhasil, selanjutnya *logout* dan mencoba *login* dengan akun yang sudah dibuat sebelumnya.

Hasil yang diharapkan : Bisa *login* sampai *dashboard* dengan *role* yang sesuai.

STATUS : Tes Lulus

STT - NF

4.5 Evaluasi

4.5.1 Analisis Hasil Pengujian

Dari hasil pengujian *User Acceptance Testing* (UAT) yang telah dilakukan dengan menggunakan *test case* pada fitur kelola laporan keuangan dan kelola *user* oleh *role* pemilik, dan pada fitur kelola barang serta kelola produk oleh *role* karyawan. dapat disimpulkan sebagai berikut :

Tabel 4.15 Evaluasi Analisis Hasil Pengujian UAT

No.	Fitur Pengujian	Hasil Pengujian	Status
1.	Kelola Barang	Semua langkah pengujian mulai dari menampilkan halaman kelola barang, menambahkan data barang (dengan dan tanpa gambar), <i>edit</i> data barang, menggunakan fitur pencarian, hingga menghapus data barang telah berhasil sesuai dengan hasil yang diharapkan.	Lulus
2.	Kelola Produk	Pengujian pada fitur kelola produk yang meliputi penambahan data merk produk, kategori, dan ukuran, serta menampilkan data yang sesuai pada halaman kelola produk telah berhasil dan sesuai dengan hasil yang diharapkan.	Lulus
3.	Kelola Laporan Keuangan	Pengujian pada fitur kelola laporan keuangan, mulai dari menampilkan data pemasukan dan pengeluaran, menambah data keuangan, hingga melakukan <i>filter</i> data laporan keuangan berdasarkan hari, bulan, dan tahun, semuanya berhasil sesuai dengan hasil yang diharapkan.	Lulus
4.	Kelola Users	Pengujian pada fitur kelola <i>users</i> juga menunjukkan hasil yang sesuai. Semua langkah, termasuk menampilkan halaman kelola <i>users</i> , menambahkan <i>user</i> dengan <i>role</i> berbeda, serta mencoba <i>login</i> dengan akun yang baru dibuat telah berhasil sesuai dengan hasil yang diharapkan.	Lulus

Dari hasil evaluasi analisis hasil pengujian tersebut, dapat disimpulkan bahwa seluruh fitur yang diuji dalam *website* ini telah berfungsi sesuai dengan spesifikasi yang diharapkan, dan berhasil melewati tahap pengujian *User Acceptance Testing* (UAT) dengan tingkat keberhasilan 100%. Dengan demikian, *website* ini siap digunakan dan diimplementasikan lebih lanjut pada bengkel aulia motor.

4.5.2 Hasil Evaluasi Penelitian

Penelitian ini bertujuan untuk merancang dan membangun sistem *inventory management* berbasis web untuk Bengkel Aulia Motor menggunakan *stack* MERN. Pengumpulan data penelitian ini menggunakan metode wawancara dengan mengajukan pertanyaan langsung kepada pihak terkait untuk memperoleh informasi. Pada tahapan desain, digunakan *diagram* UML, sedangkan implementasi menggunakan metode *extreme programming* (XP) dan menggunakan *stack* MERN (MariaDB, ExpressJS, React JS, NodeJS). Aktifitas *sprint extreme programming* (XP) diperlihatkan pada tabel 4.16 dibawah ini.

Tabel 4.16 Evaluasi *timeline* pengembangan

<i>Sprint</i>	Tanggal	Modul	Task	Point	Velocity
	04 Maret 2024	<i>Sprint Planning</i>	<i>List Backlog</i>		
0	01 - 12 April 2024	Persiapan	Analisis kebutuhan sistem	5	10
			Perancangan sistem	5	
1	15 - 25 April 2024	<i>Authentication User</i>	<i>Register (API)</i>	5	10
			<i>Login dan Logout</i>	5	
2	28 April - 01 Mei 2024	Kelola Merk	<i>View & search data merk</i>	5	20
			Tambah merk	5	
			<i>Edit merk</i>	5	
			Hapus merk	5	
3	02 - 04 Mei 2024	Kelola Kategori	<i>View & search data kategori</i>	5	20
			Tambah kategori	5	
			<i>Edit kategori</i>	5	
			Hapus kategori	5	
4	05 - 07 Mei 2024	Kelola Ukuran	<i>View data ukuran</i>	5	20
			Tambah ukuran	5	
			<i>Edit ukuran</i>	5	
			Hapus ukuran	5	
5	10 - 28 Mei 2024	Kelola Barang	<i>View & search data barang</i>	5	34
			Tambah barang	9	
			<i>Edit barang</i>	8	
			Hapus barang	5	
			<i>Export pdf</i>	7	
6	01 - 09 Juni 2024	Laporan Keuangan	<i>View data laporan keuangan</i>	5	29
			Tambah laporan keuangan	5	

			Hapus laporan keuangan	5	
			<i>Export pdf</i>	7	
			<i>Filter laporan keuangan</i>	7	
7	10 - 14 Juni 2024	Kelola <i>User</i>	<i>View & search data user</i>	5	20
			<i>Tambah user</i>	5	
			<i>Edit user</i>	5	
			<i>Hapus user</i>	5	
Total <i>velocity</i>					163
Rata-Rata					20,4

Berdasarkan hasil evaluasi yang tertera pada tabel 4.16, perancangan web ini mampu menyelesaikan dengan rata-rata 20,4 poin cerita (*story points*) per *sprint* dengan total *velocity* mencapai 163 poin. Setiap fitur diuji menggunakan metode *User Acceptance Testing* (UAT) dan hasilnya dikonfirmasi oleh pengguna, memastikan bahwa sistem memenuhi kebutuhan pengguna.

STT - NF

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil penelitian yang telah dilakukan, dapat disimpulkan hal-hal sebagai berikut:

1. Mengembangkan aplikasi sistem *inventory management* Bengkel Aulia Motor berbasis web dilakukan menggunakan teknologi *MERN Stack* yang mencakup MariaDB, ExpressJS, React JS, dan NodeJS dan metode *Extreme Programming* (XP) didapati pengembangan (*velocity*) rata-rata 20,4 untuk satu orang pengembang.
2. Evaluasi hasil pengujian yang dilakukan dengan *User Acceptance Testing* (UAT) menunjukkan bahwa sistem yang dibangun mampu memenuhi kebutuhan fungsional pengguna. Analisis hasil pengujian menunjukkan bahwa seluruh fitur yang diuji telah berfungsi dengan baik, dan tidak ada masalah signifikan yang ditemukan selama pengujian. Hal ini dibuktikan dengan tingkat keberhasilan pengujian yang mencapai 100% untuk setiap fitur utama, yang menunjukkan bahwa sistem memiliki performa yang baik dan dapat diandalkan oleh *user* karyawan dan pemilik.

5.2 Saran

Penulis menyadari bahwa tulisan ini masih memiliki kekurangan dan memerlukan pengembangan lebih lanjut dari apa yang telah dilakukan. Diharapkan agar penelitian serupa di masa mendatang dapat menjadi lebih baik. Adapun saran dari penelitian ini adalah sebagai berikut:

1. Pengembangan fitur tambahan untuk fitur notifikasi otomatis untuk mengingatkan pemilik dan karyawan tentang stok barang yang hampir habis. Fitur ini akan meningkatkan efisiensi operasional dan mencegah kekurangan stok.
2. Penggunaan *class diagram* dalam tahap perancangan sistem pada penelitian kurang efektif karena tidak diikuti dengan implementasi model yang tepat pada *framework* express.js. Sehingga hal ini menyebabkan beberapa manfaat potensial dari *class diagram* tidak dapat dimaksimalkan sepenuhnya.

DAFTAR PUSTAKA

- [1] M. A. Maulana, H. Haryoko, B. Santoso, and L. Lukman, "Penerapan Teknologi Stack MERN pada Aplikasi Service Manajemen Bengkel Berbasis Web," *J. Media Inform. Budidarma*, vol. 6, no. 3, p. 1536, 2022, doi: 10.30865/mib.v6i3.4147.
- [2] S. Astiti, "Penerapan Metode Extreme Programming Pada Rancang Bangun Website Company Profile," *RESOLUSI Rekayasa Tek. Inform. dan Inf.*, vol. 3, no. 3, pp. 247–257, 2023, [Online]. Available: <http://djournals.com/resolusi/article/view/685>
- [3] D. Marwan, "Sistem Informasi Manajemen Bengkel Berbasis Web," *Gastron. ecuatoriana y Tur. local.*, vol. 1, no. 69, pp. 5–24, 2018.
- [4] G. Ayu and K. Ita, "OPTIMALISASI INVENTORY MANAGEMENT SPAREPART DI BENGKEL MUNCUL JAYA MOTOR," vol. 3, pp. 480–485, 2024.
- [5] I. C. Christian, A. A. Supianto, and R. I. Rokhmawati, "Pengembangan Aplikasi Media Pembelajaran Berbasis Website Untuk Materi Laju Reaksi di Tingkat SMA (Studi Pada SMA Brawijaya Smart School Malang)," *J. Pengembangann Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 7, pp. 7051–7059, 2019.
- [6] Nur Fauziah, Irsan Jaelani, and Minarto, "Pembangunan Aplikasi Penggajian Dengan Sistem Skala Tunggal Berbasis Web Menggunakan Metode Extreme Programming," *J. Inform. Teknol. dan Sains*, vol. 5, no. 3, pp. 351–356, 2023, doi: 10.51401/jinteks.v5i3.3217.
- [7] M. Yovanka *et al.*, "PERANCANGAN SISTEM INFORMASI KULINER ' VOKASI LAPER ' BERBASIS WEB DENGAN METODE EXTREME PROGRAMMING," vol. 1, no. 3, pp. 1–16, 2024.
- [8] B. A. Permadi, "Bangun Aplikasi Edukasi Pernikahan Berbasis Web Menggunakan Framework Reactjs," *J. Teknol. Pint.*, vol. 3, no. 8, pp. 1–25, 2023, [Online]. Available: <http://teknologipintar.org/index.php/teknologipintar/article/view/477>
- [9] M. A. Ramdhan, A. Fauzi, A. S. Nugroho, M. A. Ramdhan, A. Fauzi, and

- A. S. Nugroho, "Implementasi Back-end Sistem Otomasi Pendataan Mahasiswa pada Food Truck Undip Menggunakan Framework Express.js," *J. Tek. Komput.*, vol. 2, no. 2, pp. 115–122, 2023, doi: 10.14710/jtk.v2i2.38527.
- [10] Faris Mas'ud and Nuryuliani, "Perancangan Aplikasi Pembelajaran Bahasa Asing Yang Interaktif Menggunakan Metode Mern," *J. Ilm. Tek.*, vol. 3, no. 1, pp. 96–104, 2024, doi: 10.56127/juit.v3i1.1197.
- [11] G. P. M. Putra, A. Tenriawaru, and Gunawan, "Rancang Bangun Virtual Assistant Chatbot Menggunakan Node.Js pada Layanan Sistem Informasi Akademik," *Pros. Semin. Nas. Pemanfaat. Sains Dan Teknol. Inf.*, vol. 1, no. 1, pp. 345–352, 2023.
- [12] T. Bratakusuma, I. U. Azmi, and S. Ayuningtiyas, "Pengembangan Back End Pada Aplikasi Alat Tulis Kantor Bank Indonesia Perwakilan Purwokerto Menggunakan Nodejs," *Semin. Nas. Inov. dan Pengemb. Teknol. Terap. Cilacap*, pp. 119–127, 2022, [Online]. Available: <https://ejournal.pnc.ac.id/index.php/senovtek>
- [13] S. Jurnal, "JURNAL SISTEM INFORMASI DAN TEKNOLOGI (S I N T E K)," vol. IV, no. 02, pp. 53–57.
- [14] T. Akhir, A. Zubaidi, F. Teknologi, I. Dan, and U. Semarang, "APLIKASI PENJUALAN DAN INVENTORY BERBASIS," 2024.
- [15] I. Wahyudi and F. Alameka, "Analisis Blackbox Testing Dan User Acceptance Testing Terhadap Sistem Informasi Solusimedsosku," *J. Teknosains Kodepena* /, vol. 04, no. 01, pp. 1–9, 2023.
- [16] A. K. Nadhif, "Rancang Bangun Sistem Informasi Asistensi Berbasis Website Menggunakan Teknologi MERN Stack Forum Asisten STMIK Amikom Surakarta merupakan suatu forum di bawah UPT Berdasarkan permasalahan tersebut maka penulis akan membuat sistem informasi pada membantu," *URANUS J. Ilm. Tek. Elektro, Sains dan Inform.*, vol. 2, no. 1, pp. 10–20, 2024.
- [17] Okpatrioka, "Research And Development (R & D) Penelitian yang Inovatif dalam Pendidikan," *J. Pendidikan, Bhs. dan Budaya*, vol. 1, no. 1, pp. 86–100, 2023.

- [18] M. R. Fadli, “Memahami desain metode penelitian kualitatif,” *Humanika*, vol. 21, no. 1, pp. 33–54, 2021, doi: 10.21831/hum.v21i1.38075.



STT - NF

LAMPIRAN A WAWANCARA

A.1 Kerangka Wawancara

Daftar pertanyaan wawancara :

1. Apa saja layanan utama yang disediakan oleh Bengkel Aulia Motor?
2. Apakah Bengkel Aulia Motor menggunakan sistem komputerisasi (menggunakan komputer/alat digital) ?
3. Apa saja tanggung jawab pekerja di Bengkel Aulia Motor ?
4. Bagaimana proses pencatatan dan manajemen data dilakukan di bengkel, seperti data sparepart motor ?
5. Data apa saja yang dimasukkan ke dalam laporan penjualan sparepart ?
6. Seberapa sering bengkel mengalami masalah terkait ketidakakuratan data atau pencatatan yang tidak konsisten ?
7. Sistem seperti apa yang dibutuhkan oleh Bengkel Aulia Motor untuk meningkatkan kinerja dan efisiensi ?

STT - NF


A.2 Hasil Wawancara

Judul : Wawancara Alur Kerja Pada Bengkel Aulia Motor
Narasumber : Ibu Yunani (Pemilik Bengkel)
Pewawancara : Muhammad Ismail
Lokasi : Bengkel Aulia Motor di Kp.Sawah Desa Jabon Mekar
Tanggal : 12 Juni 2024

Hasil Wawancara :

1. Apa saja layanan utama yang disediakan oleh Bengkel Aulia Motor?
Jawab: “Iya dibengkel ini kayak bengkel pinggir jalan aja, pompa, tambel ban, terus ganti oli, ganti yang rusak dibagian mesin tuh ada semua di dalem barangnya, ganti aki bisa, ganti lampu ada juga tuh didalem.”
2. Apakah Bengkel Aulia Motor menggunakan sistem komputerisasi (menggunakan komputer/alat digital) ?
Jawab: “Selama ini kita nyatet masih manual mas, paling kalo alat digital itu ya kalkulator buat bantu ngitung aja disamping meja tuh.”
3. Apa saja tanggung jawab pekerja di Bengkel Aulia Motor ?
Jawab: “Kalo saya sendiri (pemilik) biasanya nyetok barang aja yang udah mau abis, bisa tau dari catetann yang dia (karyawan) catet atau gak ngomong langsung ke saya, sama ngitung pendapatan hari ini gimana. Kalo si dimas yudi (karyawan) paling yang layanin aja tuh, kadang juga nyatet di buku.”
4. Bagaimana proses pencatatan dan manajemen data dilakukan di bengkel, seperti data sparepart motor ?
Jawab: “Pake buku itu aja, yang ada dimeja mas. Semua dicatet disitu.”
5. Data apa saja yang dimasukkan ke dalam laporan penjualan sparepart ?
Jawab: “Yang dicatet, nama barangnya, tanggal, merknya, terus harganya doank, sama biasanya kalo udah tutup dijumlahin tuh hari ini dapet berapa gitu, pengeluarannya berapa buat beli barang...”
6. Seberapa sering bengkel mengalami masalah terkait ketidakakuratan data atau pencatatan yang tidak konsisten ?
Jawab: “Sering gak sering sih mas yang ini, kadang kalo lagi teledor aja ato salah nulis, karena yang catet kan bukan saya aja gitu.”
7. Sistem seperti apa yang dibutuhkan oleh Bengkel Aulia Motor untuk meningkatkan kinerja dan efisiensi ?
Jawab: “Kalo ada sistem yang pake komputer itu pengennya mah mas saya bisa catet barang sama laporan keuangannya lebih gampang gitu, soalnya kadang kan kalo lagi rame nih mas kita catet dulu biasanya dibuku tuh barang apa yang dia beli biar restoknya tau, nah disitu tuh kadang kan diantara nulisnya buru-buru terus galengkap, pas diliat ini apa, nanya lagi ke yang catet, dan yang catet gatau siapa....”

LAMPIRAN B
PERMOHONAN PENGAMBILAN DATA



**SEKOLAH TINGGI TEKNOLOGI
TERPADU NURUL FIKRI**

STT - NF

Nomor : 006/S.Peng/BAAK/PRODI_TI/VI/2024
Perihal : Permohonan Pengambilan Data

Yth., Kepada Ibu Yunani
Bengkel Aulia Motor
Di Kp. Sawah Desa Jabon Mekar
Kec. Parung, Kab. Bogor.


Yang bertanda tangan di bawah ini, Kepala Program Studi **Teknik Informatika** Sekolah Tinggi Terpadu Nurul Fikri (STT-NF), ingin memberitahukan bahwa mahasiswa kami:

No	Nama Mahasiswa	NIM	Jurusan
1	Muhammad Ismail	0110220170	Teknik Informatika

membutuhkan data dalam rangka mengerjakan penelitian pada mata kuliah **Tugas Akhir** di tempat Ibu. Besar harapan kami Ibu bersedia memberikan bantuan serta petunjuk-petunjuk untuk keperluan tersebut di atas sehingga Program Mahasiswa tersebut berjalan dengan baik.

Atas perhatian dan kerjasamanya, kami mengucapkan terima kasih.


Depok, 11 Juni 2024
Kepala Program Studi Teknik Informatika
Sekolah Tinggi Teknologi Terpadu Nurul Fikri



Tifani Nabarian, S.Kom., M.T.I.
NIP: 2200890201

STT - NF

Sekolah Tinggi Teknologi Terpadu Nurul Fikri Kampus A, Jl. Situ Indah No. 116 Depok 16451 021 - 29842347
www.nurulfikri.ac.id info@nurulfikri.ac.id Kampus B1 & B2, Jl. Lenteng Agung Raya No. 20 - 21 Jakarta Selatan 12640 021 - 7841041

 Dipindai dengan CamScanner