
Software Requirements Specification

for

Implementation of the Moving Average Method in the Raw Material Forecasting System at Fadhi Coffee Kuningan

Version 1.0 approved

Prepared by

Muhammad Denaz

3 Maret, 2025

Table of Contents

1. Pendahuluan.....	1
1.1 Tujuan.....	1
1.2 Lingkup Sistem.....	1
1.3 Definisi, Akronim dan Singkatan	1
1.4 Refrensi.....	2
1.5 Gambaran Umum	2
2. Deskripsi Keseluruhan	3
2.1 Perspektif Produk	3
2.2 Fungsi Produk.....	3
2.3 Penggolongan Karakterik Pengguna	3
2.4 Batasan	3
2.5 Lingkungan Operasi	3
2.6 Asumsi dan Ketergantungan.....	4
2.7 Dokumentasi Pengguna	4
3. Kebutuhan Fungsional & Non-Fungsional.....	5
3.1 Kebutuhan Fungsional.....	5
3.2 Kebutuhan Non-Fungsional.....	6
3.3 Hardware Interface	7
4. Functional Requirement.....	8
4.1 Use Case Diagram	9
4.2 Activity Diagram	14
4.3 Sequence Diagram.....	22
4.4 Class Diagram	33
5. Non Functional Requirements	35

1. Pendahuluan

1.1 Tujuan

Dokumen ini disusun untuk memberikan deskripsi lengkap terkait spesifikasi perangkat lunak sistem peramalan bahan baku pada Fadhi Coffee and Eatery. Sistem ini dirancang untuk membantu proses pengambilan keputusan dalam pengadaan bahan baku dengan menggunakan metode *Simple Moving Average* (SMA). Dokumen ini ditunjukkan untuk:

- Tim pengembang sistem (*developer*, UI/UX, QA)
- Pemilik bisnis (*owner*)
- *Stakeholder* internal (admin operasional)
- Tim pengujian (*tester*)

1.2 Lingkup Sistem

Sistem ini adalah aplikasi berbasis web yang digunakan untuk meramalkan kebutuhan bahan baku harian berdasarkan data penjualan historis. Lingkup utamanya mencakup:

- Manajemen data penjualan, produk, dan bahan baku.
- Peramalan permintaan produk menggunakan metode SMA.
- Penyajian laporan hasil peramalan.
- Penyajian laporan hasil penjualan.
- Pengaturan user dan hak akses.

Sistem ini akan dibangun menggunakan PHP sebagai backend, MySQL untuk database, serta HTML, CSS, dan JavaScript untuk tampilan antarmuka.

1.3 Definisi, Akronim dan Singkatan

- | | |
|---------------|---|
| - SRS | : Software Requirements Specification |
| - SMA | : Simple Moving Average |
| - Forecasting | : Proses peramalan permintaan berdasarkan data historis |
| - Admin | : Pengguna sistem yang memiliki hak akses penuh |
| - Owner | : Pemilik Usaha |
| - PHP | : Hypertext Preprocessor |

- MySQL : My Structured Query Language
- CSS : Cascading Style Sheet
- HTML : Hypertext Markup Language

1.4 Refrensi

- Data analisis kebutuhan Fadhi Coffee & Eatery (internal)
- IEEE Std 830-1998 – Recommended Practice for Software Requirements Specifications
- Studi dokumentasi: metode Moving Average dalam sistem peramalan
- Template Atlassian SRS (<https://www.atlassian.com/software/confluence/templates/software-requirements-specification>)

1.5 Gambaran Umum

Dokumen ini disusun dalam struktur sebagai berikut:

- **Bab 1** menjelaskan tujuan, ruang lingkup, dan definisi penting dari sistem.
- **Bab 2** menggambarkan deskripsi umum sistem secara menyeluruh.
- **Bab 3** memuat kebutuhan fungsional dan non-fungsional.
- **Bab 4** mendeskripsikan antarmuka dan integrasi.
- **Bab 5** menjelaskan batasan teknis, asumsi, dan dependensi sistem.

2. Deskripsi Keseluruhan

2.1 Perspektif Produk

Sistem ini merupakan aplikasi baru yang dibangun dari awal dan tidak merupakan bagian dari sistem lain yang telah ada. Sistem berbasis web ini akan beroperasi dalam lingkungan client-server dan hanya dapat diakses oleh pengguna internal melalui browser.

Sistem ini akan mendukung dua jenis pengguna:

- Admin: pengguna operasional yang bertugas memasukkan dan mengelola data.
- Owner: pengguna pemilik usaha yang dapat melihat hasil peramalan dan mencetak laporan.

2.2 Fungsi Produk

Fungsi utama dari sistem ini meliputi:

- Autentikasi pengguna (login/logout)
- Pengelolaan data produk (tambah, edit, hapus)
- Pengelolaan bahan baku (tambah, edit, hapus)
- Pengelolaan penjualan (tambah, edit, hapus)
- Proses peramalan menggunakan metode *Simple Moving Average* (SMA)
- Penyajian hasil peramalan dalam bentuk tabel dan grafik
- Pencetakan laporan hasil peramalan
- Pengelolaan user dan hak akses (tambah, edit, hapus, oleh Admin)

2.3 Penggolongan Karakterik Pengguna

Sistem digunakan oleh dua tipe pengguna:

- Admin: memiliki hak akses penuh terhadap sistem, bertanggung jawab mengelola data dan menjalankan peramalan.
- Owner: dapat mengakses hasil peramalan dan mencetak laporan. Tidak memiliki akses pengelolaan data.

Tidak diperlukan latar belakang teknis untuk menggunakan sistem, namun pemahaman terhadap alur bisnis sangat disarankan.

2.4 Batasan

- Metode Peramalan: Hanya metode Moving Average yang akan diimplementasikan.
- Periode Peramalan: Periode peramalan ditetapkan 7 hari.
- Platform: Aplikasi akan berbasis web.
- Teknologi Menggunakan: PHP, JavaScript, HTML, CSS, dan MySQL.

2.5 Lingkungan Operasi

Sistem akan berjalan di lingkungan web browser dengan spesifikasi berikut:

- Sistem Operasi: Windows/Linux
- Web Browser: Google Chrome, Mozilla Firefox (versi terbaru)
- Server: PHP (versi 7.x atau lebih baru), MySQL
- Client: Komputer desktop dengan koneksi internet dan browser modern

2.6 Asumsi dan Ketergantungan

Asumsi dan ketergantungan dalam pengembangan sistem ini adalah:

- Data penjualan historis tersedia dan dapat diimpor ke sistem.
- Sistem hanya diakses oleh user yang memiliki akun.
- Tidak ada training khusus bagi user yang akan menggunakan, karena aplikasi tidak rumit dan tidak membutuhkan banyak operasi dalam penggunaannya
- Proses peramalan dan data yang dianalisis dalam sistem ini secara spesifik berfokus pada prediksi untuk periode tujuh hari ke depan
- Efektivitas peramalan bergantung pada stabilitas permintaan barang dari waktu ke waktu, karena saat itulah metode Moving Average paling efektif.

Peramalan bahan baku di Fadhi Coffee and Eatery. Sistem ini bertujuan membantu pengambilan keputusan dalam pengadaan bahan baku berdasarkan data penjualan historis dengan metode Simple Moving Average (SMA).

2.7 Dokumentasi Pengguna

<Daftar komponen dokumentasi pengguna (seperti user manual, on-line help, dan tutorial) yang akan disampaikan bersama dengan perangkat lunak yang akan dikirim>

3. Kebutuhan Fungsional & Non-Fungsional

3.1 Kebutuhan Fungsional

3.1.1 Autentifikasi Pengguna

- FR-1.1 Login: Sistem harus mengizinkan pengguna terdaftar (Admin dan Pemilik) untuk masuk dengan memberikan nama pengguna dan kata sandi yang valid.
 - Jika kredensial benar, sistem harus mengarahkan ke dasbor utama berdasarkan peran pengguna.
 - Jika kredensial salah, sistem harus menampilkan pesan kesalahan "User belum terdaftar" atau kembali ke halaman login.
- FR-1.2 Logout: Sistem harus mengizinkan pengguna yang masuk untuk keluar dari aplikasi dengan aman.

3.1.2 Pengelolaan Data (Peran Admin)

- FR-2.1 Pengelolaan Data Bahan Baku: Admin harus dapat menambah, mengedit, dan menghapus data produk dalam sistem.
- FR-2.2 Pengelolaan Data Produk: Admin harus dapat menambah, pengaitan produk dengan komponen bahan baku, mengedit, dan menghapus data produk dalam sistem.
- FR-2.3 Input Data Penjualan: Admin harus dapat menginput data penjualan ke dalam sistem.
- FR-2.4 Impor Data Penjualan: Admin harus dapat mengimpor data penjualan berupa file csv ke dalam sistem.
- FR-2.5 Pengelolaan Data Pengguna: Admin harus dapat menambah, mengedit, dan menghapus akun pengguna.

3.1.3 Fungsionalitas Peramalan

- FR-3.1 Lakukan Peramalan: Sistem harus mengizinkan peran Admin dan Owner untuk memulai dan melakukan peramalan penjualan untuk produk yang dipilih menggunakan metode Moving Average.
 - Sistem harus menampilkan hasil peramalan setelah proses selesai.
- FR-3.2 Tampilan Hasil Prediksi: Sistem harus menyajikan hasil prediksi penjualan produk berdasarkan metode Simple Moving Average dalam periode 7 hari, serta menampilkan estimasi kekurangan bahan baku yang diperlukan untuk memenuhi permintaan tersebut secara terstruktur dalam format tabel dan/atau grafik.

3.1.4 Fungsionalitas Pelaporan

- FR-4.1 Lihat Laporan: Sistem harus mengizinkan peran Admin dan Pemilik untuk melihat laporan penjualan untuk periode yang dipilih.
- Cetak Laporan: Sistem harus mengizinkan peran Admin dan Pemilik untuk mencetak laporan peramalan yang dihasilkan.

3.2 Kebutuhan Non-Fungsional

3.2.1 Kinerja

- Sistem harus memproses dan menampilkan hasil peramalan secara efisien.
- Sistem harus menangani operasi pengelolaan data (tambah, edit, hapus) dengan penundaan minimal.

3.2.2 Kegunaan

- Sistem harus memiliki antarmuka web yang intuitif dan mudah digunakan.
- Proses login harus mudah.
- Laporan dan hasil peramalan harus disajikan dengan jelas dan komprehensif.

3.2.3 Keandalan

- Sistem harus menjaga integritas data selama semua operasi (input, edit, hapus, peramalan).
- Perhitungan peramalan harus secara konsisten akurat berdasarkan metode Moving Average.

3.2.4 Keamanan

- Sistem harus mengimplementasikan mekanisme login yang aman untuk melindungi akun pengguna.
- Peran pengguna (Admin dan Owner) harus memiliki kontrol akses yang sesuai.

3.2.5 Pemeliharaan

- Sistem harus dikembangkan menggunakan pendekatan terstruktur (Model Waterfall) untuk memfasilitasi pemeliharaan dan peningkatan di masa mendatang.

3.2.6 Persyaratan Perangkat Keras dan Perangkat Lunak

1. Perangkat Keras

Tabel 1. 1 Persyaratan Perangkat Keras

Spesifikasi Hardware	Spesifikasi
Prosesor	Intel(R) Core(TM) atau yang setara
Hard Drive	Minimum 250 GB
Ram	Minimum 2 GB

2. Perangkat Lunak

Tabel 1. 2 Persyaratan Perangkat Lunak

Komponen	Spesifikasi
Sistem Operasi	Microsoft Windows
Peramban Web	Google Chrome

3.3 Hardware Interface

Sistem peramalan ini berinteraksi dengan komponen perangkat keras melalui lapisan perangkat lunak perantara, yaitu sistem operasi dan peramban web. Sistem tidak memerlukan antarmuka fisik langsung ke perangkat keras khusus (misalnya sensor atau kontroler). Interaksi logis meliputi:

- Tipe Perangkat yang Didukung: Komputer desktop atau laptop standar yang menjalankan sistem operasi yang kompatibel.
- Interaksi Data dan Kontrol: Sistem memproses data melalui input pengguna (keyboard, mouse) dan menampilkan informasi melalui monitor. Operasi dasar seperti penyimpanan data ke hard disk dan penggunaan memori (RAM) dikelola oleh sistem operasi.
- Protokol Komunikasi: Interaksi utama adalah antara aplikasi web (PHP, JavaScript, HTML, CSS) dan server web (yang berjalan pada perangkat keras), serta antara server web dan basis data MySQL. Komunikasi ini menggunakan protokol jaringan standar yang ditangani oleh sistem operasi

4. Functional Requirement

Bagian ini menguraikan pengorganisasian persyaratan fungsional untuk produk, disajikan berdasarkan fitur sistem dan layanan utama yang disediakan oleh produk. Setiap persyaratan diidentifikasi secara unik dan dijelaskan untuk memberikan pemahaman yang jelas tentang fungsionalitas yang diharapkan dari sistem. Berikut adalah daftar kebutuhan fungsional perangkat lunak, lengkap dengan ID dan penjelasan yang diperlukan:

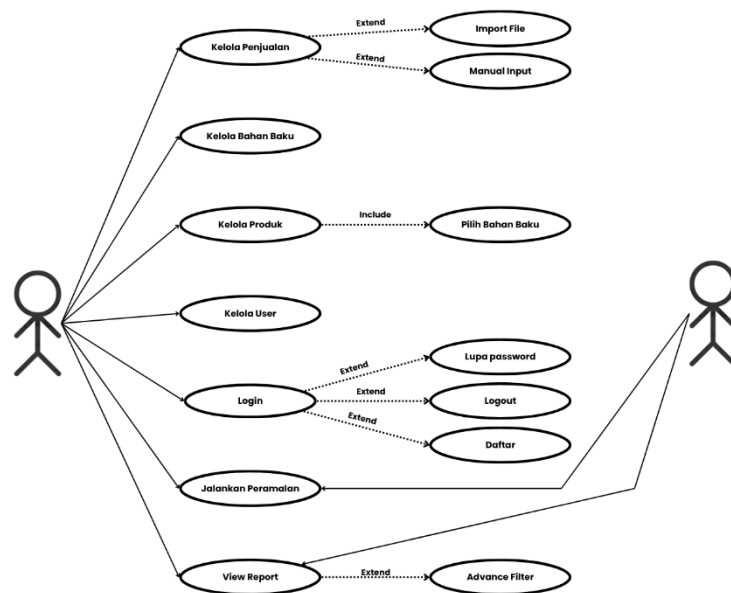
Tabel 1. 3 Function Requirement

ID	Kebutuhan Fungsional	Penjelasan
FR-1.1	Login Pengguna Sistem harus mengizinkan pengguna terdaftar untuk masuk.	Pengguna memasukkan username dan password. Jika valid, sistem mengarahkan ke dasbor sesuai peran. Jika tidak valid, sistem menampilkan pesan error dan tetap di halaman login.
FR-1.2	Logout Pengguna Sistem harus menyediakan fungsionalitas untuk keluar dari aplikasi.	Pengguna dapat mengakhiri sesi mereka dengan aman.
FR-2.1	Pengelolaan Data Produk Admin harus dapat mengelola (menambah, mengedit, menghapus) data produk.	Memungkinkan Admin untuk menjaga data produk tetap akurat dan mutakhir. Termasuk informasi seperti nama produk, harga, dan ID.
FR-2.2	Pengelolaan Data Bahan Baku Admin harus dapat mengelola (menambah, mengedit, menghapus) data bahan baku.	Memungkinkan Admin untuk menjaga data bahan baku tetap akurat dan mutakhir. Termasuk informasi seperti nama bahan baku, stok, dan harga.
FR-2.3	Input Data Penjualan Admin harus dapat mengimpor data penjualan historis ke dalam sistem.	Data penjualan historis diperlukan sebagai input untuk perhitungan peramalan. Format impor data harus jelas dan terstruktur.
FR-3.1	Pelaksanaan Peramalan Penjualan Sistem harus mengizinkan Admin dan Pemilik untuk memulai dan menjalankan proses peramalan penjualan menggunakan metode Moving Average.	Pengguna memilih produk atau kriteria lain untuk peramalan, dan sistem melakukan perhitungan peramalan berdasarkan data penjualan historis dan metode Moving Average (periode 7 hari).
FR-3.2	Tampilan Hasil Prediksi Sistem harus menampilkan hasil prediksi penjualan untuk produk.	Hasil peramalan (prediksi volume penjualan untuk periode 7 hari ke depan) harus disajikan dalam format yang jelas dan mudah dipahami, misalnya dalam tabel atau grafik.
FR-4.1	Lihat Laporan Penjualan dan Peramalan Sistem harus memungkinkan Admin dan Pemilik untuk melihat laporan.	Laporan harus mencakup ringkasan data penjualan historis dan hasil peramalan, dapat difilter berdasarkan periode waktu atau produk tertentu.

ID	Kebutuhan Fungsional	Penjelasan
FR-4.2	Cetak Laporan Sistem harus memungkinkan Admin dan Pemilik untuk mencetak laporan penjualan dan peramalan.	Laporan yang ditampilkan di layar harus dapat dicetak dalam format yang mudah dibaca.
FR-2.4	Pengelolaan Data Pengguna Admin harus dapat mengelola (menambah, mengedit, menghapus) akun pengguna.	Memungkinkan Admin untuk mengelola daftar pengguna sistem, termasuk peran (Admin atau Pemilik) dan kredensial dasar.

4.1 Use Case Diagram

Bagian ini akan menyajikan diagram kasus penggunaan yang menggambarkan interaksi antara aktor (pengguna) dan fungsionalitas utama sistem. Diagram ini secara visual merepresentasikan ruang lingkup fungsional sistem dari perspektif pengguna.



Gambar 4.1 Use case Diagram

4.1.1 Login

1. Deskripsi Use Case

Use case ini menjelaskan bagaimana pengguna (Admin atau Pemilik) dapat masuk ke dalam sistem untuk mengakses fungsionalitas yang sesuai dengan peran mereka.

2. Stimulus dan Respon

Tabel 4. 1 Tabel Stimulus & Respon - Login

<i>Action by user</i>	<i>Response from system</i>
1. Pengguna mengakses URL aplikasi web	
	2. Sistem menampilkan halaman login.
3. Pengguna memasukkan username dan password	
4. Pengguna mengklik tombol "Login".	
	5. Sistem memvalidasi kredensial pengguna (username dan password) terhadap data yang tersimpan di basis data.
	6. Jika kredensial valid, sistem mengarahkan pengguna ke halaman dashboard yang sesuai dengan perannya (misalnya, dashboard Admin atau dashboard Pemilik).
	7. Jika kredensial tidak valid, sistem menampilkan pesan kesalahan "User belum terdaftar" dan pengguna tetap berada di halaman login.

4.1.2 Kelola Penjualan

1. Deskripsi Use Case

Use case ini menjelaskan bagaimana Admin dapat mengimpor data penjualan historis ke dalam sistem, yang diperlukan untuk proses peramalan.

2. Stimulus dan Respon

Tabel 4. 2 Stimulus & Respon - Kelola Penjualan

<i>Action by user</i>	<i>Response from system</i>
1. 1. Admin navigasi ke menu "Penjualan" lalu "Import Data".	
	2. Sistem menampilkan halaman pengelolaan data penjualan dengan opsi impor.

3. Admin memilih file data penjualan (misalnya, CSV atau Excel) dari perangkat lokalnya.	
4. Pengguna mengklik tombol "import".	
	5. Sistem membaca dan memproses data dari file yang diunggah.
	6. Sistem memvalidasi format dan isi data yang diimpor.
	7. Jika data tidak valid, Sistem menampilkan pesan error
	8. Jika data valid, sistem menyimpan data penjualan ke database.
	9. Sistem memperbarui tampilan daftar data penjualan.

4.1.3 Kelola Bahan Baku

1. Deskripsi Use Case

Use case ini menjelaskan bagaimana aktori dapat menginput data bahan baku ke dalam sistem, yang diperlukan untuk proses pembuatan produk.

2. Stimulus dan Respon

Tabel 4. 3 Stimulus & Respon - Kelola Bahan Baku

Action by user	Response from system
1. 1. Admin navigasi ke menu "Bahan Baku".	
	2. Sistem menampilkan halaman pengelolaan data bahan baku.
3. Admin memilih menginputkan 'nama', 'jumlah' dan 'satuan'	
4. Pengguna mengklik tombol "simpan".	
	5. Sistem membaca dan memproses data dari inputan user.

	6. Sistem memvalidasi data
	7. Jika data sudah ada, sistem akan menjumlahkan dengan data tersebut dan memprosesnya ke database
	8. Jika data belum ada, Sistem akan memproses ke database
	9. Sistem memperbarui tampilan daftar data bahan baku.

4.1.4 Kelola Produk

1. Deskripsi Use Case

Use case ini menjelaskan bagaimana aktor dapat melakukan input data produk ke dalam sistem, yang nantinya akan ditampilkan di daftar produk yang dijual.

2. Stimulus dan Respon

Tabel 4. 4 Stimulus & Respon - Kelola Produk

<i>Action by user</i>	<i>Response from system</i>
1. 1. Admin navigasi ke menu "produk".	
	2. Sistem menampilkan halaman pengelolaan data produk.
3. Admin melakukan input 'nama', lalu mengatikkannya dengan bahan baku dan menginputkan jumlah bahan baku yang dibutuhkan	
4. Pengguna mengklik tombol "simpan".	
	5. Sistem membaca dan memproses data dari inputan user.
	6. Sistem memvalidasi data
	7. Jika data tidak valid, sistem akan menampilkan pesan error.
	8. Jika data valid, Sistem akan memproses data ke database

	9. Sistem memperbarui tampilan daftar data produk.
--	--

4.1.5 Jalankan Peramalan

1. Deskripsi Use Case

Use case ini menjelaskan bagaimana aktor yaitu admin dan owner dapat menjalankan peramalan berdasarkan data produk dan data penjualan historis yang ada sistem.

2. Stimulus dan Respon

Tabel 4. 5 Stimulus & Respon - Jalankan Peramalan

<i>Action by user</i>	<i>Response from system</i>
1. User navigasi ke menu "Peramalan".	
	2. Sistem menampilkan halaman peramalan.
3. Aktor memilih produk yang akan diramal	
4. Aktor mengklik tombol "ramal".	
	5. Sistem memproses data penjualan dengan metode SMA dan menghitung hasil prediksi
	6. Sistem menampilkan hasil perhitungan peramalan

4.1.6 View Laporan

1. Deskripsi Use Case

Use case ini menjelaskan bagaimana aktor yaitu admin dan owner dapat melihat laporan berdasarkan data penjualan historis yang ada sistem.

2. Stimulus dan Respon

Tabel 4. 6 Stimulus & Respon View Laporan

<i>Action by user</i>	<i>Response from system</i>
1. User navigasi ke menu "laporan"	
	2. Sistem menampilkan halaman laporan.

3. Aktor memilih rentang waktu laporan untuk di tampilkan	
	4. Sistem menampilkan hasil laporan

4.1.7 Kelola User

Deskripsi Use Case

Use case ini menjelaskan bagaimana aktor yaitu admin dapat melakukan kelola user yang ada sistem.

2. Stimulus dan Respon

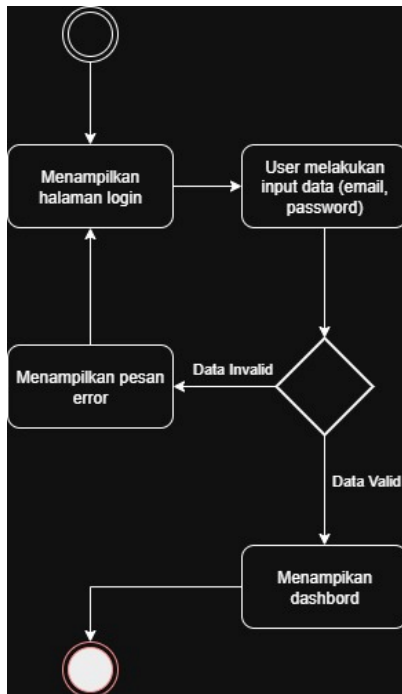
Tabel 4. 7 Stimulus & Respon - Kelola User

<i>Action by user</i>	<i>Response from system</i>
1. Aktor navigasi ke menu "User"	
	2. Sistem menampilkan halaman untuk mengelola data user.
3. Aktor melakukan input data untuk membuat user baru	
4. Aktor menekan tombol "tambah"	
	5. Sistem membaca dan memproses data dari inputan user.
	6. Sistem memvalidasi data
	7. Jika data tidak valid, sistem akan menampilkan pesan error.
	8. Jika data valid, Sistem akan memproses data ke database
	9. Sistem memperbarui tampilan daftar data user.

4.2 Activity Diagram

Bagian ini menyajikan diagram aktivitas yang menggambarkan alur kerja dan urutan tindakan untuk proses-proses kunci dalam sistem. Diagram ini membantu memahami bagaimana berbagai aktivitas dalam sistem saling terkait dan dijalankan. Setiap diagram akan dilengkapi dengan deskripsi singkat untuk memperjelas alurn

4.2.1 Activity Diagram Login

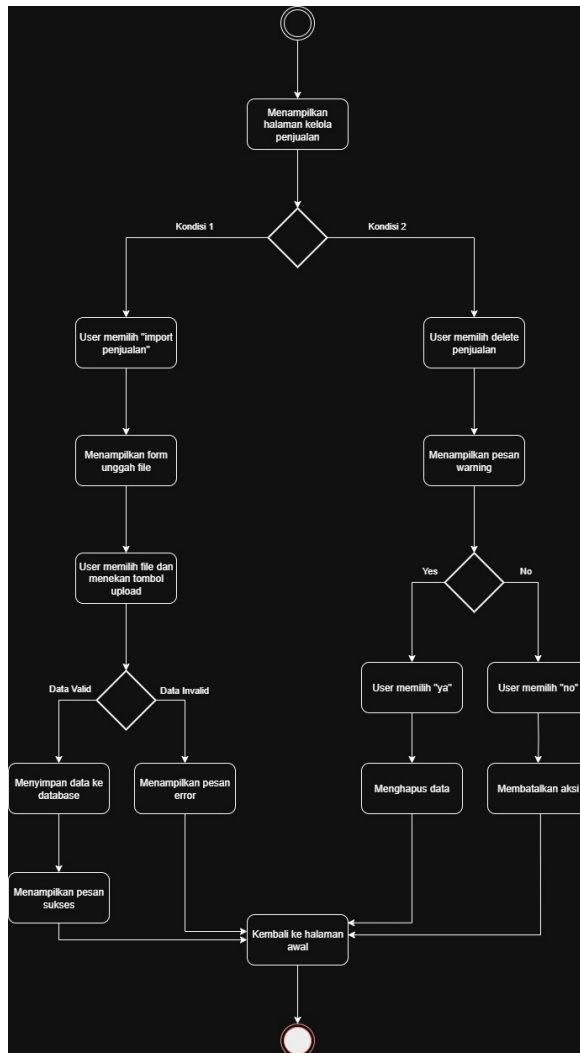


Gambar 4. 2 Activity Diagram Login

Berikut adalah alurnya:

1. Pengguna mengakses system dan memasuki halaman login
2. Pengguna melakukan input username dan password
3. Pengguna menekan tombol Login
4. Sistem memvalidasi data : Jika data valid maka akan diarahkan ke dashboard, dan Jika tidak valid maka akan diarahkan kembali ke halaman login
5. Proses selesai

4.2.2 Activity Diagram Kelola Penjualan

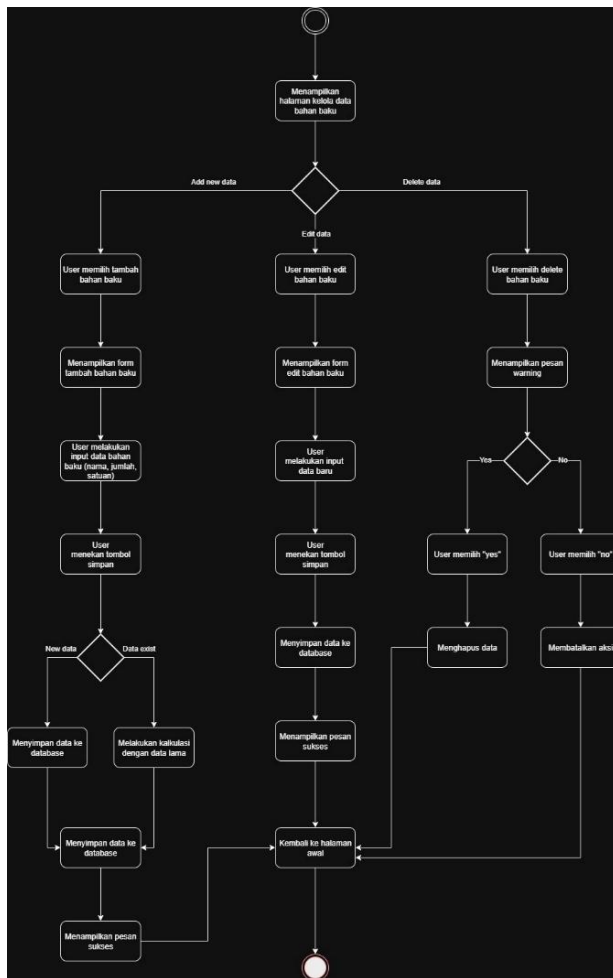


Gambar 4. 3 Activity Diagram Kelola Penjualan

Berikut adalah alurnya:

1. Admin memulai dengan mengakses fitur pengelolaan data penjualan di sistem.
2. Sistem menampilkan halaman yang memungkinkan Admin untuk melihat data penjualan yang ada atau memulai proses impor data baru.
3. Admin memilih opsi untuk "Import Data Penjualan".
4. Sistem meminta Admin untuk mengunggah file data penjualan (misalnya, dalam format CSV atau Excel).
5. Admin memilih dan mengunggah file yang berisi data penjualan historis.
6. Sistem melakukan validasi terhadap format dan isi data dalam file yang diunggah.
7. Sistem akan memvalidasi : jika data Valid, sistem akan menyimpan data penjualan ke dalam basis data dan kemudian menampilkan pesan konfirmasi keberhasilan impor. Daftar data penjualan akan diperbarui. dan Jika data Tidak Valid, sistem akan menampilkan pesan kesalahan yang mengindikasikan masalah validasi, dan mungkin meminta Admin untuk mengunggah ulang file yang benar.
8. Proses berakhir

4.2.3 Activity Diagram Kelola Bahan Baku

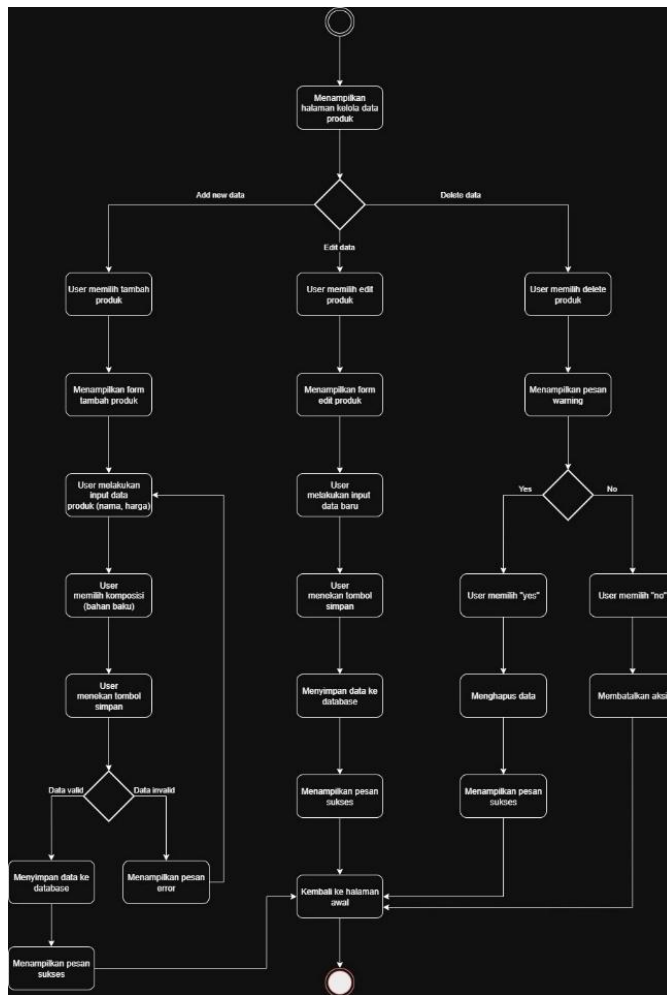


Gambar 4. 4 Activity Diagram Kelola Bahan Baku

Berikut adalah alurnya:

1. Admin memulai dengan memilih opsi pengelolaan bahan baku dari antarmuka sistem.
2. Sistem menampilkan daftar bahan baku yang terdaftar. Dari daftar tersebut, Admin dapat:
3. Menambah Bahan Baku Baru: Admin mengisi formulir dengan detail bahan baku baru, kemudian mengirimkannya. Sistem akan memvalidasi data yang dimasukkan, menyimpannya ke database jika valid, dan memperbarui daftar tampilan.
4. Mengedit Bahan Baku: Admin memilih bahan baku yang ingin diedit, melakukan perubahan pada informasi yang relevan, dan menyimpan modifikasi tersebut. Sistem akan memvalidasi input, memperbarui data, dan me-refresh daftar.
5. Menghapus Bahan Baku: Admin memilih bahan baku dari daftar dan mengkonfirmasi aksi penghapusan. Sistem akan menghapus data bahan baku dari database dan memperbarui tampilan daftar.
6. Setelah setiap operasi, sistem akan memberikan konfirmasi keberhasilan atau pesan kesalahan, serta memastikan daftar bahan baku yang ditampilkan selalu yang terbaru.
7. Proses berakhir

4.2.4 Activity Diagram Kelola Produk

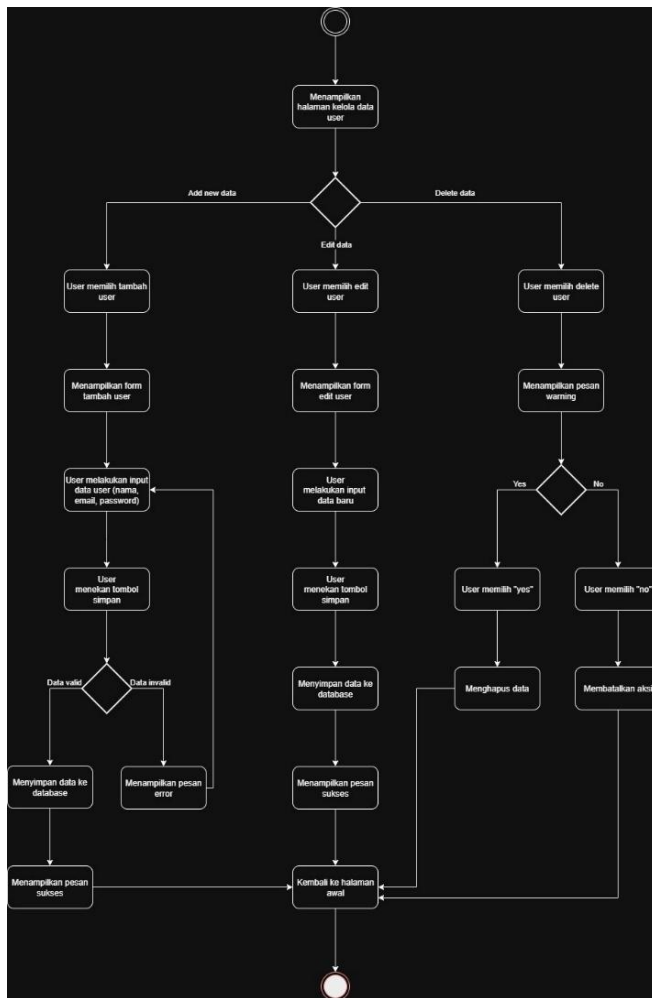


Gambar 4. 5 Activity Diagram Kelola Produk

Berikut adalah alurnya:

1. Admin memulai proses dengan mengakses menu pengelolaan produk.
2. Sistem menampilkan daftar produk yang ada. Admin memiliki opsi untuk:
3. Menambah Produk Baru: Admin mengisi formulir penambahan produk dan mengirimkannya. Sistem memvalidasi input, menyimpan data jika valid, dan memperbarui daftar.
4. Mengedit Produk: Admin memilih produk dari daftar, memodifikasi detailnya, dan menyimpan perubahan. Sistem memvalidasi input, memperbarui data jika valid, dan memperbarui daftar.
5. Menghapus Produk: Admin memilih produk dari daftar dan mengkonfirmasi penghapusan. Sistem menghapus data produk dari database dan memperbarui daftar.
6. Pada setiap operasi (tambah, edit, hapus), sistem memberikan feedback apakah operasi berhasil atau gagal, dan memperbarui tampilan daftar produk.
7. Proses berakhir

4.2.5 Activity Diagram Kelola User

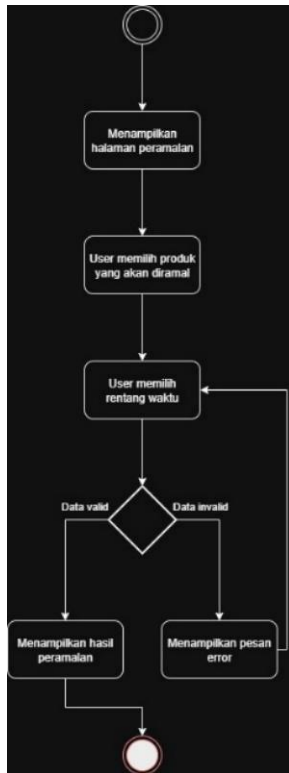


Gambar 4. 6 Activity Diagram Kelola User

Berikut adalah alurnya:

1. Admin memulai proses dengan mengakses modul pengelolaan pengguna.
2. Sistem menampilkan daftar akun pengguna yang terdaftar. Admin memiliki beberapa pilihan:
3. Menambah Pengguna Baru: Admin mengisi formulir pendaftaran pengguna baru (termasuk peran Admin/Pemilik), lalu mengirimkannya. Sistem memvalidasi input, menyimpan data pengguna ke database jika valid, dan memperbarui daftar.
4. Mengedit Detail Pengguna: Admin memilih pengguna dari daftar, memodifikasi informasi akunnya (misalnya, username, peran), dan menyimpan perubahan. Sistem memvalidasi input, memperbarui data pengguna, dan me-refresh daftar.
5. Menghapus Pengguna: Admin memilih akun pengguna yang ingin dihapus dan mengkonfirmasi penghapusan. Sistem menghapus data pengguna dari database dan memperbarui daftar.
6. Setelah setiap operasi pengelolaan (tambah, edit, hapus), sistem memberikan umpan balik (berhasil/gagal) dan memastikan daftar pengguna yang ditampilkan selalu akurat.
7. Proses selesai

4.2.6 Activity Diagram Jalankan Peramalan

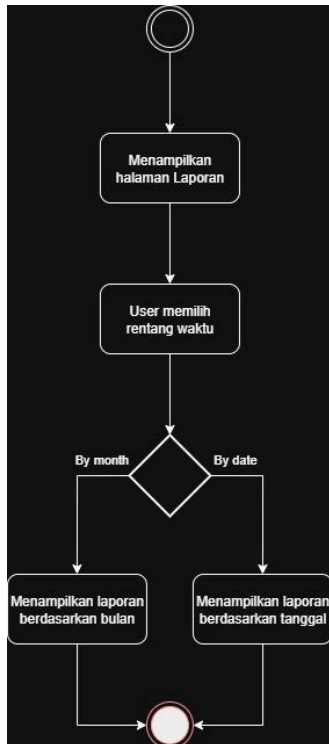


Gambar 4. 7 Activity Diagram Jalankan Peramalan

Berikut adalah alurnya:

1. Pengguna (Admin atau Pemilik) memulai proses dengan memilih menu "Peramalan".
2. Sistem menampilkan antarmuka peramalan, dengan opsi untuk memilih produk dan rentang waktu.
3. Pengguna memilih produk atau kriteria lain yang akan diramalkan
4. Pengguna mengklik tombol "ramalkan"
5. Sistem memproses permintaan:
6. Sistem mengambil data penjualan historis yang relevan dari database.
7. Sistem melakukan perhitungan peramalan menggunakan metode Moving Average (periode 7 hari).
8. Sistem menampilkan hasil peramalan kepada pengguna, biasanya dalam bentuk tabel atau grafik yang menunjukkan prediksi volume penjualan untuk periode mendatang.
9. Proses selesai

4.2.7 Activity Diagram View Laporan



Gambar 4. 8 Activity Diagram View Report

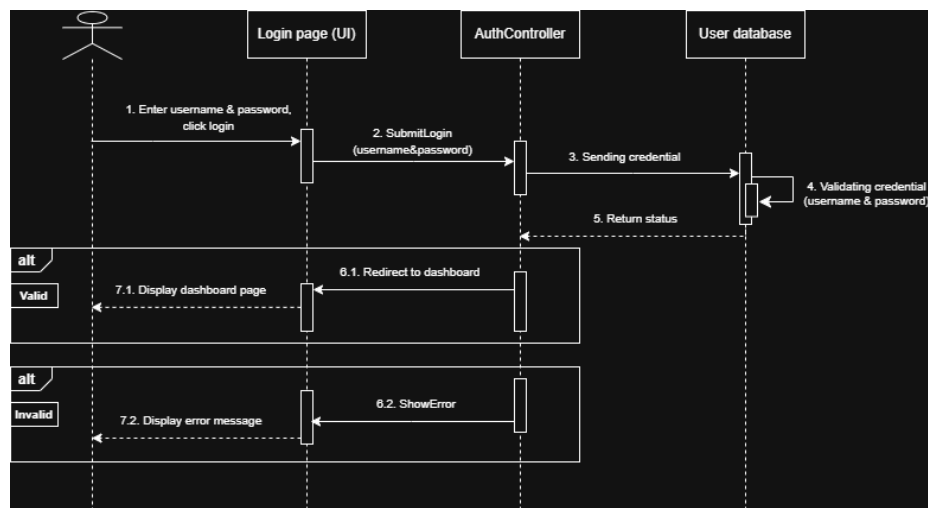
Berikut adalah alurnya:

1. Pengguna (Admin atau Pemilik) memulai proses dengan mengakses menu "Laporan".
2. Sistem menampilkan halaman view laporan.
3. Sistem meminta pengguna untuk memasukkan kriteria laporan (misalnya, rentang tanggal, produk spesifik).
4. Pengguna mengklik tombol "Tampilkan Laporan" atau "Generate Laporan".
5. Sistem memproses permintaan:
6. Sistem mengumpulkan data yang relevan dari database berdasarkan kriteria yang dipilih.
7. Sistem memformat data tersebut menjadi laporan yang terstruktur.
8. Sistem menampilkan laporan di layar.
9. Aktivitas pencetakan laporan selesai.

4.3 Sequence Diagram

Diagram Urutan berfokus pada interaksi dinamis antar objek dalam sistem dari waktu ke waktu, menunjukkan urutan pesan yang dipertukarkan antara objek-objek tersebut dalam skenario tertentu. Diagram ini menekankan pada dimensi waktu, di mana sumbu vertikal merepresentasikan waktu dan sumbu horizontal merepresentasikan berbagai objek atau aktor yang terlibat. Setiap pesan digambarkan sebagai panah antara *lifelines* objek, menunjukkan panggilan fungsi, pengiriman data, atau sinyal yang memicu respons. Diagram Urutan sangat efektif untuk memahami bagaimana suatu fungsionalitas sistem dieksekusi melalui kolaborasi antar objek secara spesifik.

4.3.1 Sequence Diagram Login



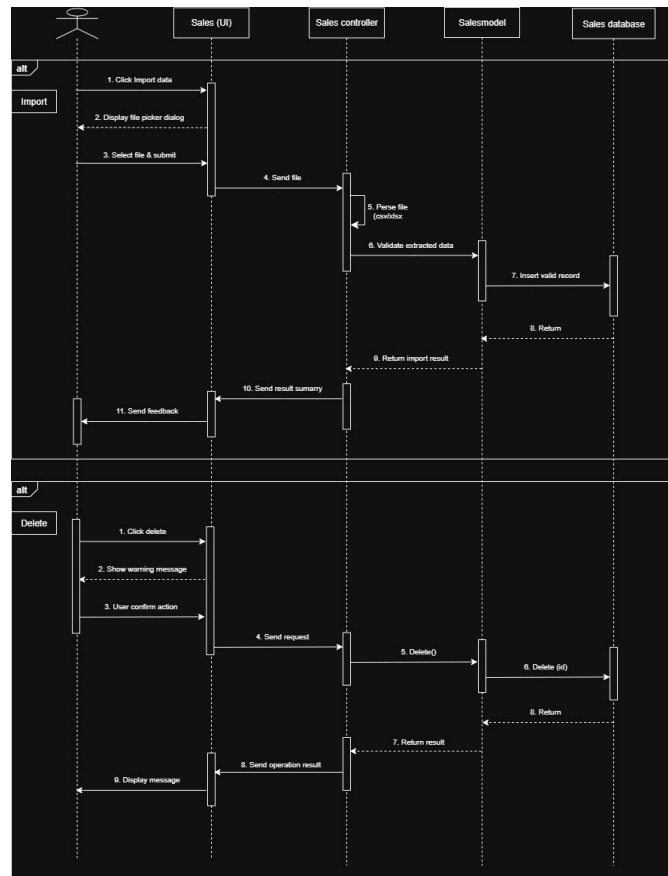
Gambar 4. 9 Sequence Diagram Login

Berikut alurnya :

1. Pengguna mengirimkan permintaan untuk mengakses halaman login.
2. Sistem (melalui server web) menampilkan halaman login kepada pengguna.
3. Pengguna memasukkan kredensial (username dan password) dan mengirimkan permintaan login.
4. Permintaan login diterima oleh Kontrol Login.
5. Kontrol Login berinteraksi dengan objek model Pengguna untuk memverifikasi kredensial.
6. Objek Pengguna melakukan kueri ke Basis Data untuk mencocokkan kredensial.
7. Basis Data mengembalikan hasil verifikasi (valid/tidak valid) ke objek Pengguna.
8. Objek Pengguna mengirimkan status validasi kembali ke Kontrol Login.

9. Percabangan Logika (oleh Kontrol Login):
10. Jika kredensial Valid: Kontrol Login akan mengirimkan pesan navigasi ke antarmuka untuk mengarahkan pengguna ke dashboard yang sesuai dengan perannya.
11. Jika kredensial Tidak Valid: Kontrol Login akan mengirimkan pesan error ke antarmuka.
12. Antarmuka menampilkan dashboard atau pesan kesalahan kepada pengguna.

4.3.2 Sequence Diagram Kelola Penjualan



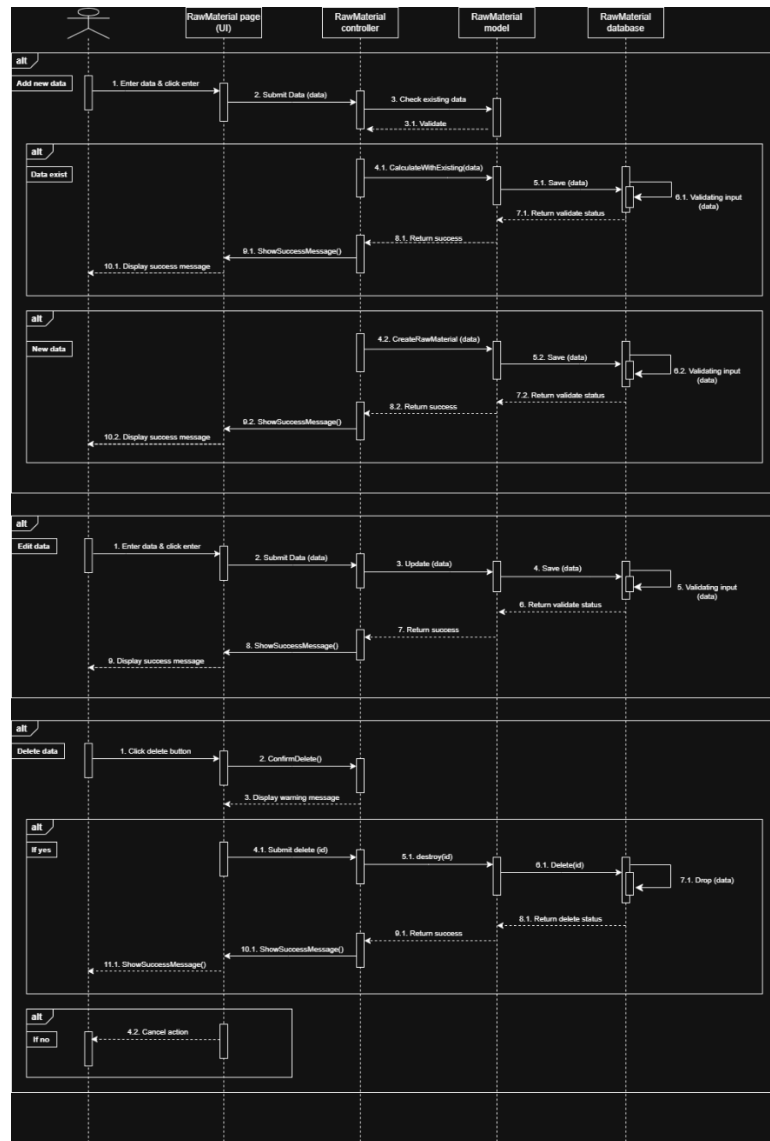
Gambar 4. 10 Sequence Diagram Kelola Penjualan

Berikut alurnya :

1. Admin memulai interaksi dengan mengirimkan permintaan ke sistem untuk mengakses halaman pengelolaan penjualan.
2. Antarmuka Pengelolaan Penjualan ditampilkan kepada Admin.
3. Admin memilih opsi untuk mengunggah atau mengimpor data penjualan, yang memicu permintaan ke kontrol data penjualan.
4. Kontrol Data Penjualan kemudian berinteraksi dengan model data penjualan (misalnya, objek Penjualan) untuk mempersiapkan penyimpanan data.
5. Proses validasi data dilakukan oleh kontrol atau model.
6. Jika data valid, data penjualan baru disimpan ke dalam Basis Data.
7. Basis Data mengirimkan konfirmasi keberhasilan penyimpanan kembali ke kontrol.
8. Kontrol meneruskan status berhasil kembali ke antarmuka.

9. Antarmuka menampilkan pesan konfirmasi kepada Admin bahwa data penjualan telah berhasil diimpor/disimpan, dan memperbarui daftar data penjualan.

4.3.3 Sequence Diagram Kelola Bahan Baku



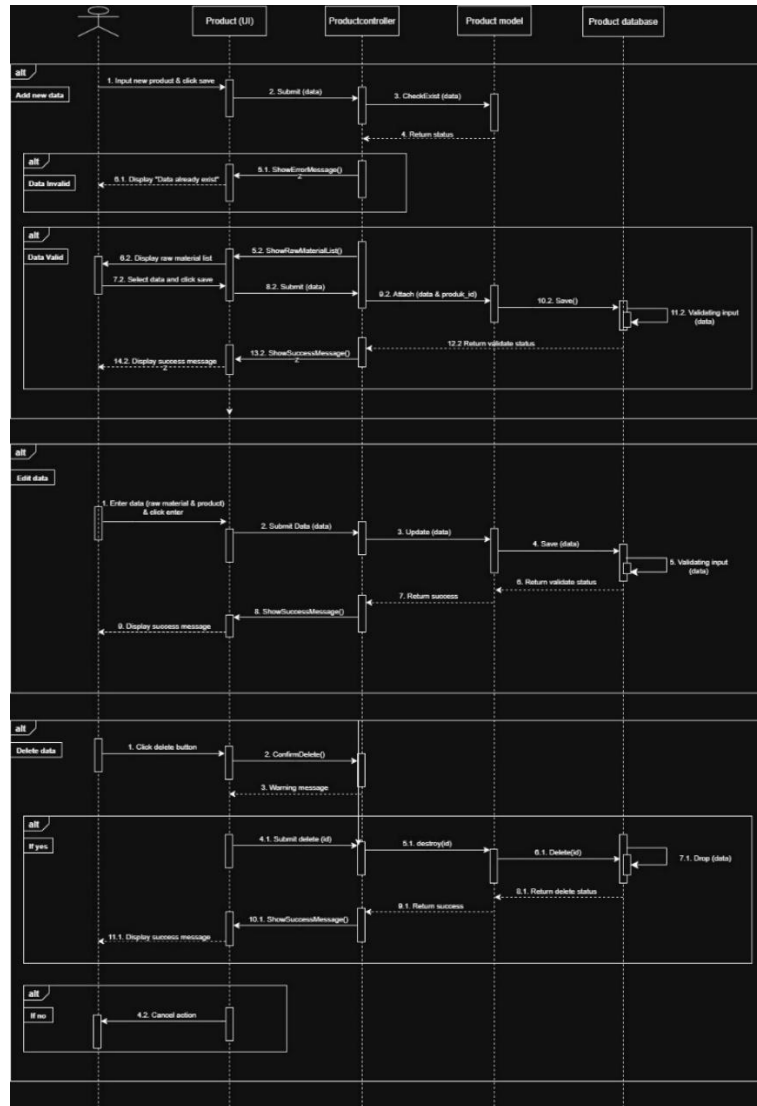
Gambar 4. 11 Sequence Diagram Kelola Bahan Baku

Berikut alurnya :

1. Admin memulai dengan mengirimkan permintaan ke sistem untuk mengakses halaman pengelolaan bahan baku.
2. Antarmuka Pengelolaan Bahan Baku ditampilkan kepada Admin.
3. Admin melakukan aksi pengelolaan (misalnya, mengisi form tambah/edit atau memilih bahan baku untuk dihapus) dan mengirimkan permintaan ke kontrol bahan baku.

4. Kontrol Bahan Baku menerima permintaan dan berinteraksi dengan objek model BahanBaku untuk memproses data.
5. Objek BahanBaku berkomunikasi dengan Basis Data untuk menyimpan, memperbarui, atau menghapus informasi bahan baku.
6. Basis Data mengirimkan konfirmasi operasi kembali ke objek BahanBaku.
7. Objek BahanBaku meneruskan status kembali ke Kontrol Bahan Baku.
8. Kontrol Bahan Baku mengirimkan respon kembali ke antarmuka.
9. Antarmuka menampilkan pesan konfirmasi (berhasil/gagal) dan memperbarui daftar bahan baku yang ditampilkan kepada Admin.

4.3.4 Sequence Diagram Kelola Produk

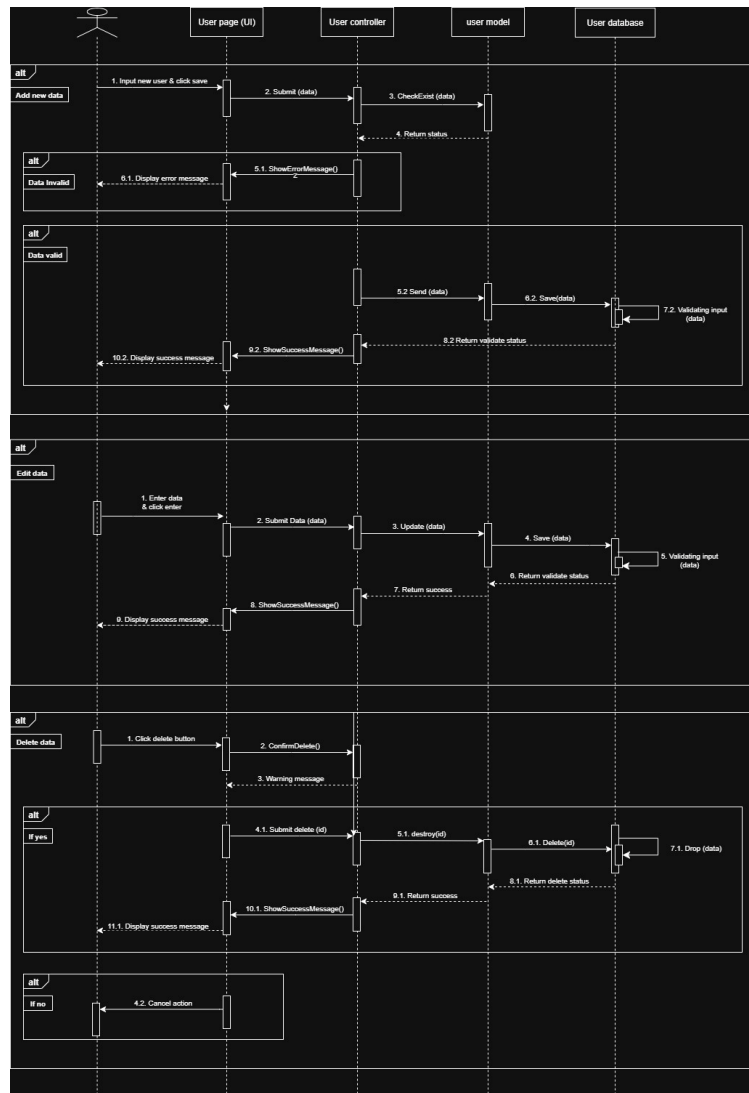


Gambar 4. 12 Sequence Diagram Kelola Produk

Berikut alurnya :

1. Admin mengirimkan permintaan ke sistem untuk mengakses modul pengelolaan produk.
2. Antarmuka Pengelolaan Produk ditampilkan kepada Admin.
3. Admin melakukan aksi pengelolaan (misalnya, mengisi formulir tambah/edit, atau memilih untuk menghapus), dan mengirimkan permintaan ke kontrol produk.
4. Kontrol Produk menerima permintaan dan berinteraksi dengan objek model Produk untuk memproses data.
5. Objek Produk berinteraksi dengan Basis Data untuk menyimpan, memperbarui, atau menghapus informasi produk.
6. Basis Data mengirimkan konfirmasi operasi kembali ke objek Produk.
7. Objek Produk meneruskan status kembali ke Kontrol Produk.
8. Kontrol Produk mengirimkan respon kembali ke antarmuka.
9. Antarmuka menampilkan pesan konfirmasi (berhasil/gagal) dan memperbarui daftar produk yang ditampilkan kepada Admin.

4.3.5 Sequence Diagram Kelola User



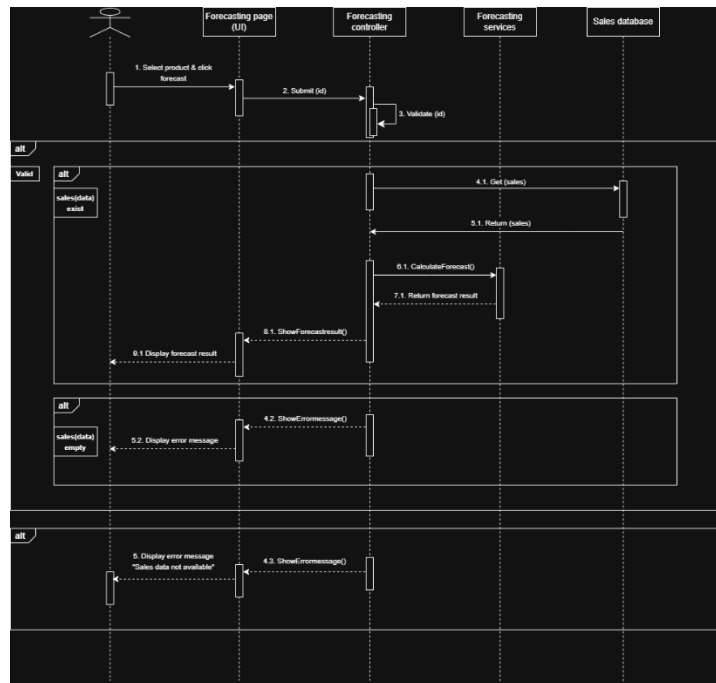
Gambar 4. 13 Sequence Diagram Kelola User

Berikut alurnya :

1. Admin memulai interaksi dengan sistem untuk mengakses halaman pengelolaan pengguna.
2. Antarmuka Pengelolaan Pengguna ditampilkan kepada Admin.
3. Admin melakukan aksi pengelolaan (misalnya, mengisi formulir untuk menambah pengguna baru, atau memilih pengguna untuk diedit/dihapus) dan mengirimkan permintaan ke kontrol pengguna.
4. Kontrol Pengguna menerima permintaan dan berinteraksi dengan objek model Pengguna untuk memproses detail akun.

5. Objek Pengguna berkomunikasi dengan Basis Data untuk menyimpan, memperbarui, atau menghapus informasi akun pengguna.
6. Basis Data mengirimkan konfirmasi operasi kembali ke objek Pengguna.
7. Objek Pengguna meneruskan status kembali ke Kontrol Pengguna.
8. Kontrol Pengguna mengirimkan respon kembali ke antarmuka.
9. Antarmuka menampilkan pesan konfirmasi kepada Admin dan memperbarui daftar akun pengguna yang ditampilkan.

4.3.6 Sequence Diagram Jalankan Peramalan



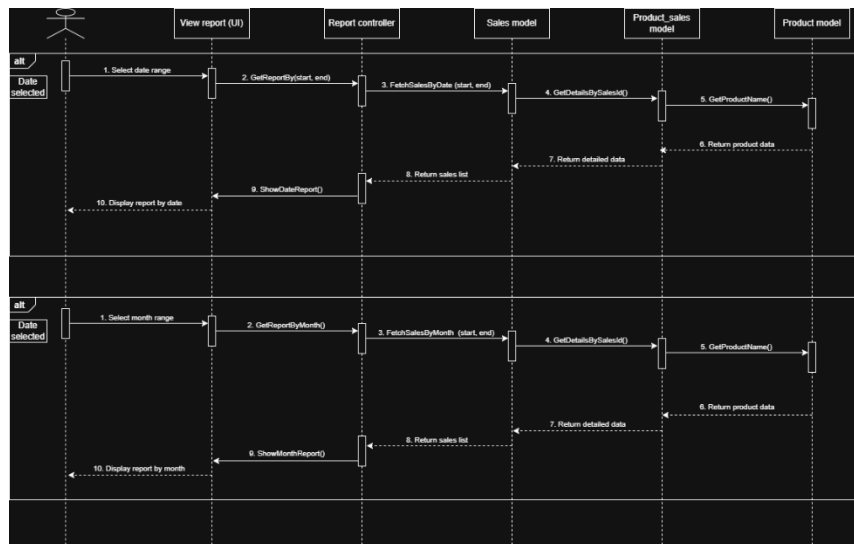
Gambar 4. 14 Jalankan Peramalan

Berikut alurnya :

1. Pengguna (Admin atau Pemilik) mengirimkan permintaan ke sistem untuk mengakses fitur peramalan.
2. Antarmuka Peramalan ditampilkan kepada pengguna.
3. Pengguna memilih parameter peramalan (misalnya, produk yang akan diramal) dan mengirimkan permintaan untuk melakukan peramalan.
4. Permintaan diterima oleh Kontrol Peramalan.
5. Kontrol Peramalan berinteraksi dengan objek model Produk untuk mendapatkan data terkait produk yang dipilih.

6. Kontrol Peramalan kemudian berinteraksi dengan objek model Penjualan untuk mengambil data penjualan historis yang diperlukan.
7. Data historis dikirim kembali ke Kontrol Peramalan.
8. Kontrol Peramalan memanggil modul atau kelas yang bertanggung jawab untuk perhitungan Moving Average.
9. Modul Peramalan melakukan perhitungan dan mengembalikan hasil prediksi ke Kontrol Peramalan.
10. Kontrol Peramalan mengirimkan hasil peramalan kembali ke antarmuka.
11. Antarmuka menampilkan hasil peramalan kepada pengguna (misalnya, dalam tabel atau grafik).

4.3.7 Sequence Diagram View Laporan



Gambar 4. 15 Sequence Diagram View Laporan

Berikut alurnya :

1. Pengguna (Admin atau Pemilik) mengirimkan permintaan ke sistem untuk mengakses fitur laporan.
2. Antarmuka Laporan ditampilkan kepada pengguna, mungkin dengan opsi untuk memilih jenis laporan atau kriteria filter.
3. Pengguna memilih jenis laporan yang diinginkan dan/atau memasukkan kriteria filter, lalu mengirimkan permintaan untuk melihat laporan.
4. Permintaan diterima oleh Kontrol Laporan.
5. Kontrol Laporan berinteraksi dengan objek model Penjualan dan/atau Peramalan untuk mengambil data yang diperlukan dari Basis Data.

6. Basis Data mengembalikan data yang relevan ke objek model.
7. Objek model meneruskan data ke Kontrol Laporan.
8. Kontrol Laporan memproses dan memformat data untuk laporan.
9. Kontrol Laporan mengirimkan data laporan yang sudah diformat ke antarmuka.
10. Antarmuka menampilkan laporan kepada pengguna.
11. (Opsional) Jika pengguna memilih untuk mencetak, antarmuka akan memicu fungsi cetak browser atau menghasilkan file yang dapat dicetak.

4.4 Class Diagram

Diagram Kelas adalah fondasi arsitektur statis sistem yang memvisualisasikan struktur kode dan data. Diagram ini menampilkan berbagai kelas dalam sistem sebagai entitas utama, lengkap dengan atribut (properti data) dan operasi (metode atau fungsi) yang dimiliki setiap kelas. Selain itu, Diagram Kelas juga menggambarkan berbagai jenis hubungan antar kelas, seperti asosiasi (bagaimana kelas berinteraksi), agregasi (hubungan "memiliki" yang lemah), komposisi (hubungan "memiliki" yang kuat), dan generalisasi (pewarisan), memberikan pandangan menyeluruh tentang bagaimana komponen-komponen sistem saling terorganisir dan berinteraksi di tingkat objek.



Gambar 4. 16 Class Diagram

Diagram Kelas ini kemungkinan besar akan menampilkan entitas inti dari sistem peramalan bahan baku Fadhi Coffee & Eatery, seperti:

1. Kelas Pengguna: Mewakili pengguna sistem (Admin, Pemilik) dengan atribut seperti `id_user`, `username`, `password`, dan `role`.

2. Kelas Produk: Mewakili produk-produk yang dijual oleh Fadhi Coffee & Eatery dengan atribut seperti id_produk, nama_produk, harga, dan mungkin stok_minimum.
3. Kelas BahanBaku: Mewakili bahan baku yang digunakan dalam pembuatan produk, dengan atribut seperti id_bahan_baku, nama_bahan_baku, stok, dan unit.
4. Kelas Penjualan: Mewakili catatan penjualan harian atau transaksi, dengan atribut seperti id_penjualan, tanggal_penjualan, id_produk, jumlah_terjual.
5. Kelas Peramalan / MovingAverage: Mewakili hasil peramalan atau konfigurasi peramalan, dengan atribut seperti id_peramalan, tanggal_prediksi, id_produk, hasil_prediksi, dan periode_MA.

Hubungan :

1. Pengguna dapat melakukan operasi pada Produk, BahanBaku, Penjualan, dan Peramalan.
2. Produk akan memiliki hubungan dengan BahanBaku (misalnya, melalui tabel relasi Produk_BahanBaku untuk resep).
3. Penjualan akan memiliki hubungan dengan Produk.
4. Peramalan akan terkait dengan Produk dan Penjualan (data historisnya).

5. System Testing

Pengujian adalah tahap terakhir yang digunakan ketika membangun sebuah sistem dengan tujuan agar mengetahui apakah sistem tersebut sudah berjalan dengan yang diharapkan.

5.1 Blackbox Testing

Pengujian black box ini berfokus pada spesifikasi fungsional dari sistem informasi. Selain itu, dengan pengujian ini dapat mendefinisikan kondisi input dan melakukan pengetesan pada fungsi-fungsi yang ada. Sehingga dapat diketahui apakah fungsi-fungsi tersebut sudah berjalan sesuai yang diharapkan atau belum.

1. Pengujian fungsi *login* dan *logout*

Tabel 4. 8 Blackbox Testing – Login & Logout

No	Fungsi	Skenario Pengujian	Hasil yang diharapkan	Hasil pengujian	Keterangan
1.	Login	a. Admin melakukan <i>login</i> dengan <i>name</i> dan <i>password</i> yang sesuai. b. Admin melakukan <i>login</i> dengan <i>name</i> dan <i>password</i> yang tidak sesuai.	a. Menampilkan notifikasi “Anda berhasil masuk” b. Menampilkan notifikasi “ <i>user</i> belum terdaftar”	a. Berhasil masuk ke sistem. penjualan dapat menggunakan menu yang disediakan disetiap penjualan yang terdaftar. b. Gagal <i>login</i> .	<i>Valid</i>
2.	Logout	Pengguna memilih menu <i>logout</i>	Pengguna keluar dari sesi dengan sukses.	Penjualan berhasil keluar dari sistem	<i>Valid</i>

2. Pengujian input data

Tabel 4. 9 Blackbox Testing - Input Data

No	Fungsi	Skenario Pengujian	Hasil yang diharapkan	Hasil pengujian	Keterangan
----	--------	--------------------	-----------------------	-----------------	------------

1.	<i>Input</i> Data Penjualan	Kasir memasukan data penjualan.	Sistem menerima data valid	a. Data penjualan yang <i>diinput</i> berhasil disimpan b. Sistem menampilkan data penjualan ke halaman data penjualan. c. Sistem menampilkan data penjualan yang sudah ada	<i>Valid</i>
2.	<i>Input</i> Data Bahan baku	Admin memasukkan data Bahan baku.	Sistem menerima input jika sesuai format	a. Data Bahan baku yang <i>diinput</i> berhasil disimpan b. Sistem menampilkan data Bahan baku ke halaman data Bahan baku. c. Sistem menampilkan data Bahan baku yang sudah ada	<i>Valid</i>
3.	<i>Input</i> Data Produk	Admin memasukkan data Produk	Sistem menerima data valid	a. Data Produk yang <i>diinput</i> berhasil disimpan b. Sistem menampilkan data Produk ke halaman data Produk. c. Sistem menampilkan data Produk yang sudah ada	<i>Valid</i>

3. Pengujian edit data

Tabel 4. 10 Blackbox Testing Edit Data

No	Fungsi	Skenarion Pengujian	Hasil yang diharapkan	Hasil pengujian	Keterangan
1.	Edit Data penjualan	Admin mengedit data Penjualan.	Sistem menyimpan perubahan dan menampilkan pesan sukses.	<ul style="list-style-type: none"> a. Data penjualan yang diedit berhasil disimpan b. Sistem menampilkan data penjualan ke halaman data penjualan. c. Sistem menampilkan data penjualan yang sudah ada 	<i>Valid</i>
2.	Edit Data Bahan baku	Admin mengedit data Bahan baku.	Sistem menampilkan pesan sukses.	<ul style="list-style-type: none"> a. Data Bahan baku yang diedit berhasil disimpan b. Sistem menampilkan data Bahan baku ke halaman data Bahan baku. c. Sistem menampilkan data Bahan baku yang sudah ada 	<i>Valid</i>

3.	Edit Data Produk	Admin mengedit data Produk	Sistem menyimpan perubahan dan menampilkan pesan sukses, "Data produk berhasil diperbarui."	a. Data Produk yang di edit berhasil disimpan b. Sistem menampilkan data Produk ke halaman data Produk. c. Sistem menampilkan data Produk yang sudah ada	<i>Valid</i>
----	------------------	----------------------------	---	--	--------------

4. Pengujian Hapus data

Tabel 4. 11 Blackbox Testing Hapus Data

No	Fungsi	Pengujian	Hasil yang diharapkan	Hasil pengujian	Keterangan
1.	Hapus Data Penjualan	Admin menghapus salah satu data penjualan.	Sistem menghapus data penjualan yang dipilih dan menampilkan pesan sukses.	a. Data penjualan berhasil dihapus b. Sistem tidak menampilkan data penjualan yang sudah dihapus ke halaman data penjualan.	<i>Valid</i>

2.	Hapus Data Bahan baku	Admin menghapus salah satu data Bahan baku.	a. Sistem menghapus Bahan baku dan menampilkan pesan sukses, seperti " Bahan baku berhasil dihapus." b. Bahan baku tersebut tidak muncul lagi dalam daftar Bahan baku.	a. Data Bahan baku berhasil dihapus b. Sistem tidak menampilkan data Bahan baku yang sudah dihapus ke halaman data Bahan baku.	<i>Valid</i>
3.	Hapus Data Produk	Admin menghapus salah satu data Produk	a. Sistem menghapus produk dan menampilkan pesan sukses, seperti "Produk berhasil dihapus." b. Produk tersebut tidak muncul lagi dalam daftar produk.	a. Data Produk berhasil dihapus b. Sistem tidak menampilkan data Produk yang sudah dihapus ke halaman data kategori produk.	<i>Valid</i>

5.2 Whitebox Testing

Pengujian white box merupakan proses evaluasi perangkat lunak yang menguji kemampuan desain dan kode program untuk memastikan bahwa mereka memproduksi fungsi, input, dan output yang sesuai dengan spesifikasi kebutuhan. Hal ini dilakukan dengan menganalisis logika dan struktur dari dalam kode program untuk memastikan bahwa semua aspeknya bekerja sesuai yang diharapkan.

Tabel 4. 12 Whitebox Testing

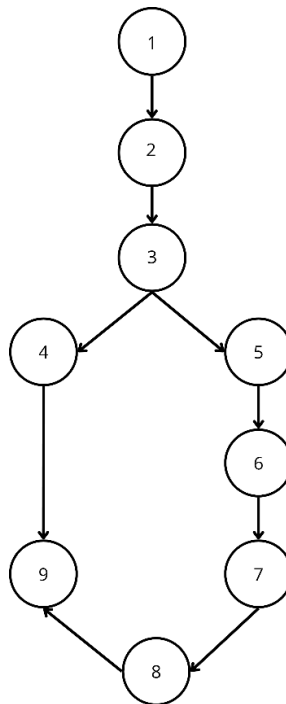
<i>Node</i>	<i>Script</i>
1	<pre>// Inisialisasi parameter \$maPeriod = 7; \$forecastDays = 7; \$produkId = 1; // ID produk yang akan di-test</pre>
2	<pre>// Ambil data penjualan historis \$penjualan = TransaksiProduk::select(DB::raw('DATE(transaksi.tanggal_transaksi) as tanggal'), DB::raw('SUM(jumlah) as total_terjual')) ->join('transaksi', 'transaksi_produk.transaksi_id', '=', 'transaksi.id') ->where('transaksi_produk.produk_id', \$produkId) ->groupBy('tanggal') ->orderBy('tanggal', 'ASC') ->get();</pre>
3	<pre>// Cek jika data penjualan kosong if (\$penjualan->isEmpty()) { return null; }</pre>

4	<pre>// Format data penjualan \$salesData = []; foreach (\$penjualan as \$item) { \$salesData[\$item->tanggal] = \$item->total_terjual; }</pre>
5	<pre>// Hitung moving average \$dates = array_keys(\$salesData); \$values = array_values(\$salesData); \$movingAverages = []; \$salesData[\$dateStr] = 0; // Default ke 0 \$startDate = strtotime('+1 day', \$startDate); }</pre>
6	<pre>// Loop untuk menghitung moving average for (\$i = 0; \$i < count(\$values); \$i++) { \$sum = 0; \$count = 0; for (\$j = \$i; \$j >= 0 && \$j >= \$i - (\$maPeriod - 1); \$j--) { \$sum += \$values[\$j]; \$count++; } }</pre>

	<pre> \$movingAverages[\$dates[\$i]] = \$count > 0 ? round(\$sum / \$count, 2) : 0; } </pre>
7	<pre> // Prediksi 7 hari ke depan \$lastValues = array_slice(\$values, -\$maPeriod); \$forecast = []; \$lastDate = end(\$dates); </pre>
8	<pre> // Generate forecast for (\$i = 1; \$i <= \$forecastDays; \$i++) { \$nextDate = date('Y-m-d', strtotime(\$lastDate . " +{\$i} days")); \$avg = round(array_sum(\$lastValues) / count(\$lastValues), 2); \$forecast[\$nextDate] = \$avg; array_shift(\$lastValues); \$lastValues[] = \$avg; } </pre>
9	<pre> // Return hasil return ['historical' => \$movingAverages, 'forecast' => \$forecast, </pre>

	<pre>'last_7_days' => array_slice(\$movingAverages, -7, 7, true)];</pre>
10	<pre>// Test calculateMaterialNeeds \$produk = Produk::find(\$produkId); \$forecastData = ['2023-01-01' => 5, '2023-01-02' => 6]; \$materialNeeds=\$this->calculateMaterialNeeds(\$produk, \$forecastData);</pre>
11	<pre>// Test storeDailyForecast \$this->storeDailyForecast(); \$storedData = MovingAverage::where('produk_id', \$produkId)- >count();</pre>

Berdasarkan Table di atas, maka peneliti dapat membuat flowgraph seperti berikut :



Gambar 4. 17 Flowgraph

Cyclomatic Complexity dari flowgraph di atas dapat diperoleh dengan cara menghitung daerah yang dibentuk oleh graph.

-Edge / Aksi (E) = 9

-Node / Lingkaran (N) = 9

-Predikat Node / Percabangan (P) = 1

Dengan perhitungan dari persamaan yang kita dapat kita lihat di atas sebagai berikut:

$$a. V(G) = E - N + 2P$$

$$= 9 - 9 + (2 \times 1)$$

$$= 0 + 2$$

$$= 2$$

Jadi, Cyclomatic Complexity untuk flowgraph ini adalah 2, yang berarti terdapat 2 jalur independen (2 path) yang wajib diuji.

b. Langkah / alurnya sebagai berikut :

o Path 1 = 1 → 2 → 3 → 4 → 9

o Path 2 = 1 → 2 → 3 → 5 → 6 → 7 → 8 → 9

6. Non Functional Requirements

Bagian ini menguraikan persyaratan non-fungsional yang harus dipenuhi oleh sistem. Persyaratan ini menjelaskan kualitas, batasan, dan atribut lain dari sistem yang tidak terkait langsung dengan fungsionalitas spesifik, tetapi penting untuk kinerja, keandalan, dan kegunaan secara keseluruhan.

Tabel 4. 13 Non-Functional Requirements

ID	Parameter	Kebutuhan
NFR-01	Availability	Sistem harus tersedia dan beroperasi 24 jam sehari, 7 hari seminggu, kecuali untuk waktu pemeliharaan terjadwal.
NFR-02	Reliability	Sistem harus dapat melakukan perhitungan peramalan secara konsisten dan akurat, dengan toleransi kegagalan data processing di bawah 0.1%.
NFR-03	Ergonomy	Antarmuka pengguna sistem harus intuitif dan mudah dinavigasi, memungkinkan pengguna untuk menyelesaikan tugas dengan efisien.
NFR-04	Portability	Sistem berbasis web ini harus dapat diakses melalui berbagai peramban web modern (Google Chrome, Firefox, Edge) pada sistem operasi Windows.
NFR-05	Memory	N/A (Tidak ada batasan memori internal kritis untuk aplikasi web ini)
NFR-06	Response time	Halaman aplikasi harus memuat dalam waktu kurang dari detik pada koneksi internet standar. Operasi peramalan harus menampilkan hasil dalam waktu maksimal 4 detik.
NFR-07	Safety	N/A (Sistem tidak secara langsung berinteraksi dengan kontrol fisik atau sistem yang memengaruhi keselamatan manusia)
NFR-08	Security	Sistem harus memiliki mekanisme autentikasi pengguna yang aman. Data pengguna dan data sensitif lainnya harus dilindungi dari akses tidak sah.
NFR-09	Others 1: Bahasa Komunikasi	Semua antarmuka pengguna, pesan kesalahan, dan laporan dalam sistem harus menggunakan Bahasa Indonesia.
NFR-10	Others 2: Branding	Setiap layar pada aplikasi web harus menampilkan logo "Fadhi Coffee & Eatery" di bagian atas atau sudut yang konsisten.