

BAB III METODOLOGI PENELITIAN

3.1 Tempat dan Waktu

Skripsi berjudul “Penerapan Algoritma Multilevel Feedback Queue Pada Layanan Pemesanan di Restoran Ayam Penyet Pemuda Semarang” dilaksanakan dengan studi literatur dan studi pustaka yang dilakukan pada bulan Mei 2024 sampai dengan bulan Juli 2024.

3.2 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan pada penelitian ini diantaranya adalah sebagai berikut:

a. Studi Pustaka

Studi pustaka dalam penelitian "Penerapan Algoritma Multilevel Feedback Queue pada Layanan Pemesanan di Restoran Ayam Penyet Pemuda Semarang" memiliki peran penting dalam memahami dasar teoretis algoritma MLFQ yang akan diterapkan dengan menelusuri penelitian terkait seperti jurnal ataupun website yang berkaitan dengan algoritma MLFQ.

b. Studi Literatur

Pendekatan ini dilakukan dengan mengumpulkan referensi dari berbagai jurnal, buku, dan hasil penelitian lain yang relevan dengan sistem serta dari sumber-sumber yang mendukung terkait objek penelitian yang dipilih. Studi literatur ini bertujuan sebagai acuan untuk memperkaya referensi dalam penelitian, khususnya dalam penerapan algoritma MLFQ.

3.3 Alat dan Bahan

3.3.1 Alat

Dalam era digital yang terus berkembang, adopsi Teknologi Informasi dan Komunikasi (TIK) menjadi krusial untuk meningkatkan efisiensi dan kualitas pelayanan dalam berbagai industri, termasuk industri kuliner. Penerapan algoritma

Multilevel Feedback Queue (MLFQ) dalam layanan pesanan berbasis Android diharapkan dapat membantu restoran ini meningkatkan kinerja operasionalnya.

Untuk mendukung implementasi sistem ini, diperlukan berbagai perangkat keras (*hardware*) dan perangkat lunak (*software*) yang memadai. Kebutuhan perangkat lunak dan perangkat keras yang digunakan dalam pembuatan aplikasi adalah sebagai berikut:

a. Kebutuhan Perangkat Lunak (*Software*)

Kebutuhan perangkat lunak yang digunakan untuk membuat aplikasi Berbasis Android adalah sebagai berikut:

1. Perangkat lunak sistem operasi yang digunakan di PC (*Personal Computer*) atau laptop adalah *Microsoft Windows* 10 dengan 64 bit
2. Perangkat lunak untuk sistem operasi Android minimal versi 10
3. Perangkat lunak yang digunakan untuk membuat program aplikasi adalah *Android Studio*
4. Perangkat lunak yang digunakan untuk pembuatan *database* adalah *Xampp*

b. Kebutuhan Perangkat Keras (*Hardware*)

Kebutuhan perangkat keras yang digunakan untuk membuat aplikasi Berbasis Android sebagai berikut:

1. Spesifikasi PC :
 - *Processor Intel Core i5*
 - *Ram 12 Gb.*
 - *Harddisk 500 Gb.*
 - *Keyboard dan Mouse*
2. Spesifikasi *Smartphone* :
 - *Prosesor Qualcomm Snapdragon 720.*
 - *Ram 8 Gb.*
 - *Storage 500 Mb*

3.3.2 Bahan

Data terkait jenis menu yang tersedia seperti ayam goreng, nasi, minuman, dan paket-paket tertentu, diambil sebagai bahan utama. Setiap jenis menu dapat memiliki kategori berdasarkan tingkat kesulitan atau waktu penyajian, yang nantinya akan menjadi dasar dalam pengaturan prioritas di sistem antrian MLFQ.

3.4 Algoritma *Multilevel Feedback Queue* (MLFQ)

Algoritma *Multilevel Feedback Queue* (MLFQ) adalah metode penjadwalan yang digunakan untuk mengelola proses atau tugas dengan berbagai prioritas yang dapat berubah secara dinamis. MLFQ berfungsi dengan membagi proses ke dalam beberapa antrian berdasarkan prioritas dan waktu eksekusi. Penerapan Algoritma *Multilevel Feedback Queue* (MLFQ) pada studi kasus di restoran adalah:

a. Pesanan Baru Masuk

Semua pesanan baru masuk ke *queue* 1. Misalnya, pesanan minuman atau lalapan yang cepat dibuat.

b. Pengolahan di *Queue* 1

Pesanan di *queue* 1 diproses selama maksimal 5 menit. Jika selesai dalam waktu ini, pesanan diselesaikan dan diantar ke pelanggan. Jika pesanan memerlukan lebih dari 5 menit, pesanan dipindahkan ke *queue* 2.

c. Pengolahan di *Queue* 2

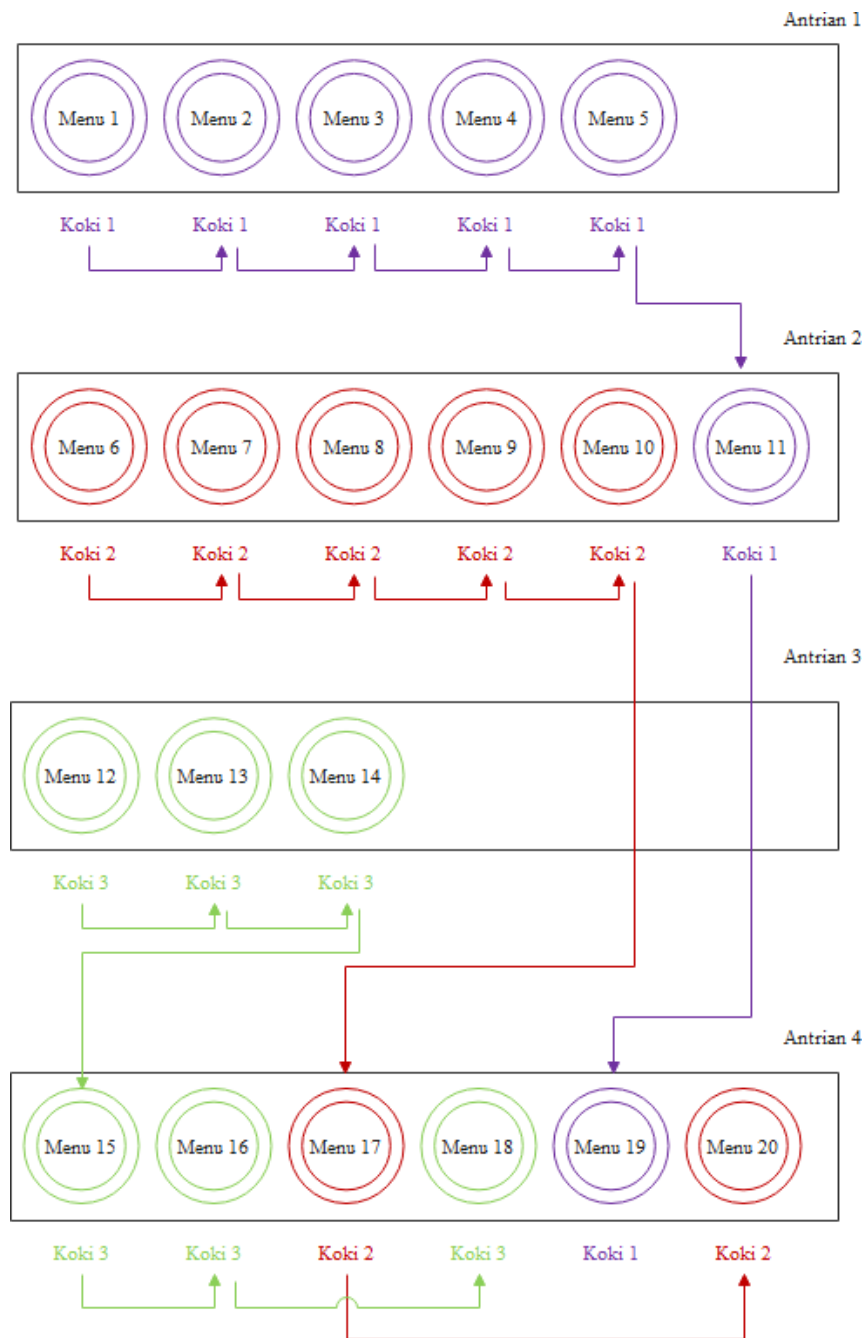
Pesanan di *queue* 2 diproses selama maksimal 10 menit seperti pengolahan gorengan. Jika selesai dalam waktu ini, pesanan diselesaikan dan diantar ke pelanggan. Jika pesanan masih memerlukan lebih dari 10 menit, pesanan dipindahkan ke *queue* 3.

d. Pengolahan di *Queue* 3:

Pesanan di *queue* 3 diproses selama maksimal 15 menit seperti pengolahan masakan. Jika selesai, pesanan diantar ke pelanggan. Jika pesanan ini masih belum selesai dalam waktu ini, prioritasnya dapat dinaikkan kembali ke *queue* 4 untuk diproses lebih lanjut.

e. Pengolahan di *Queue* 4:

Pesanan di *queue* 4 diproses selama maksimal 20 menit seperti pengolahan sayuran. Jika selesai, pesanan diantar ke pelanggan.

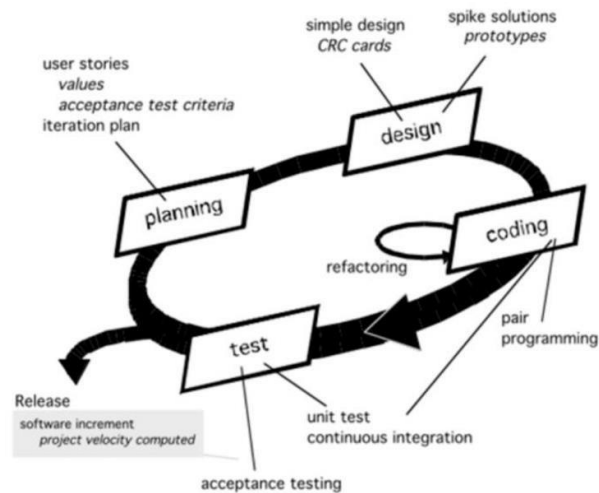


Gambar 3. 1 Contoh Penerapan Algoritma *Multilevel Feedback Queue* (MLFQ)

3.5 Metode Pengembangan Perangkat Lunak

Extreme Programming (XP) mengadvokasi pendekatan berorientasi objek dan cocok digunakan ketika terdapat ketidakpahaman terhadap kebutuhan atau terjadi perubahan kebutuhan dengan cepat (Sanjaya & Andry, 2019). Terdapat empat fase

yang harus dilalui dalam metode *Extreme Programming* (XP), yaitu perencanaan, perancangan, pengkodean, dan pengujian.



Gambar 3. 2 Pemodelan *Extreme Programming*

3.5.1 Perencanaan

Perencanaan penelitian adalah tahap awal yang sangat penting untuk memastikan penelitian berjalan sesuai dengan tujuan yang telah ditetapkan. Dalam konteks penelitian mengenai penerapan Algoritma *Multilevel Feedback Queue* (MLFQ) pada layanan pemesanan Restoran Ayam Penyet Pemuda Semarang, perencanaan ini mencakup dua aktivitas utama yaitu observasi dan studi pustaka.

a. Observasi

Tujuan observasi adalah memahami kondisi operasional saat ini di Restoran Ayam Penyet Pemuda Semarang, mengidentifikasi masalah yang dihadapi dalam layanan pesan, mengamati proses penerimaan pesanan, persiapan, dan penyajian makanan, serta melihat bagaimana pelayan dan koki berinteraksi dan mengelola pesanan.

b. Studi Pustaka

Tujuan studi pustaka untuk memahami konsep dan teori Algoritma *Multilevel Feedback Queue* (MLFQ) serta aplikasinya dalam berbagai konteks. Mengidentifikasi metode terbaik untuk menerapkan MLFQ dalam manajemen pesanan restoran. Mengakses jurnal ilmiah, buku, artikel konferensi, dan

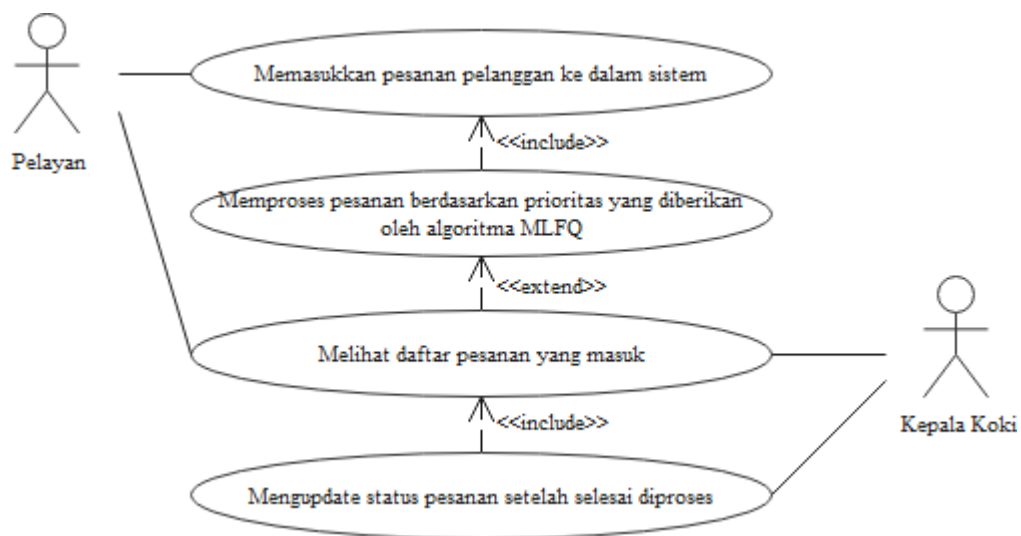
publikasi lainnya yang berkaitan dengan MLFQ dan manajemen operasional restoran.

3.5.2 Perancangan

Penelitian ini bertujuan untuk merancang dan mengimplementasikan aplikasi berbasis Android yang menggunakan Algoritma *Multilevel Feedback Queue* (MLFQ) untuk mengelola layanan antrian pesanan di Restoran Ayam Penyet Pemuda Semarang. Perancangan aplikasi ini mencakup pembuatan *use case diagram* untuk menggambarkan interaksi antara aktor dan sistem, serta pembuatan antarmuka pengguna yang intuitif untuk pelayan dan kepala koki.

3.5.2.1 Use Case Diagram

Use case diagram adalah alat penting dalam perancangan sistem untuk menggambarkan interaksi antara pengguna (aktor) dan sistem. Gambar 3.4 berikut merupakan deskripsi dari *use case diagram* untuk aplikasi ini.



Gambar 3. 3 Perancangan *Use Case Diagram* Pada Sistem yang Diusulkan

Agar implementasi sistem ini dapat berjalan dengan baik, diperlukan pemahaman yang jelas mengenai cara kerja dari berbagai skenario yang ada dalam sistem manajemen pesanan. Berikut ini adalah detail *use case* untuk empat skenario penting: memasukkan pesanan pelanggan ke dalam sistem, memproses pesanan

berdasarkan prioritas yang diberikan oleh algoritma MLFQ, melihat daftar pesanan yang masuk, dan mengupdate status pesanan setelah selesai diproses. Setiap *use case* akan dijelaskan secara rinci untuk memastikan bahwa seluruh aktor yang terlibat, termasuk pelayan, koki, dan sistem, dapat beroperasi sesuai dengan peran dan tanggung jawab masing-masing.

a. *Use Case*: Memasukkan Pesanan Pelanggan ke Dalam Sistem

Penjelasan detail mengenai *use case*: Memasukkan Pesanan Pelanggan ke Dalam Sistem terlihat pada tabel 3.1.

Tabel 3. 1 Skenario *Use Case* Memasukkan Pesanan Pelanggan ke Dalam Sistem

Aktor	:	Pelayan
Deskripsi	:	Pelayan memasukkan pesanan pelanggan ke dalam sistem melalui aplikasi berbasis Android
Langkah-langkah	:	<p>Pelayan menerima pesanan dari pelanggan.</p> <p>Pelayan membuka aplikasi layanan pesanan di perangkat Android.</p> <p>Pelayan memilih nomor meja di aplikasi.</p> <p>Pelayan memasukkan detail pesanan (jenis makanan, jumlah, permintaan khusus atau deskripsi pesanan).</p> <p>Pelayan mengonfirmasi pesanan.</p> <p>Aplikasi menyimpan pesanan di <i>database</i> dan memberikan informasi bahwa pesanan telah berhasil dimasukkan.</p>
Prekondisi	:	<p>Pelayan telah <i>login</i> ke aplikasi.</p> <p>Sistem dalam kondisi siap menerima <i>input</i> pesanan.</p>
Postkondisi	:	<p>Pesanan pelanggan tersimpan di <i>database</i>.</p> <p>Informasi berhasil memasukkan pesanan ditampilkan.</p>

b. *Use Case*: Memproses Pesanan Berdasarkan Prioritas yang Diberikan oleh Algoritma MLFQ

Penjelasan detail mengenai *use case*: Memproses Pesanan Berdasarkan Prioritas yang Diberikan oleh Algoritma MLFQ terlihat pada tabel 3.2.

Tabel 3. 2 Skenario *Use Case* Memproses Pesanan Berdasarkan Prioritas yang Diberikan oleh Algoritma MLFQ

Aktor	:	Kepala Koki
Deskripsi	:	Sistem memproses pesanan berdasarkan antrian yang diberikan oleh algoritma <i>Multilevel Feedback Queue</i> (MLFQ)
Langkah-langkah	:	Sistem mengelompokkan pesanan ke dalam beberapa antrian. Algoritma MLFQ menentukan antrian pesanan. Koki membuka aplikasi dan mengakses daftar pesanan yang telah dikelompokkan. Koki memproses pesanan sesuai dengan urutan antrian. Koki menandai pesanan sebagai selesai di aplikasi.
Prekondisi	:	Pesanan telah dimasukkan ke dalam sistem. Algoritma MLFQ telah diimplementasikan dan aktif di sistem.
Postkondisi	:	Pesanan diproses sesuai antrian yang diberikan oleh algoritma MLFQ. Pesanan yang telah selesai diproses ditandai di sistem (status selesai).

c. *Use Case*: Melihat Daftar Pesanan yang Masuk

Penjelasan detail mengenai *use case*: Melihat Daftar Pesanan yang Masuk terlihat pada tabel 3.3.

Tabel 3. 3 Skenario *Use Case* Melihat Daftar Pesanan yang Masuk

Aktor	:	Kepala Koki dan Pelayan
Deskripsi	:	Kepala Koki atau pelayan melihat daftar pesanan yang masuk melalui aplikasi
Langkah-langkah	:	Kepala Koki atau pelayan membuka aplikasi layanan pemesanan di perangkat Android. Kepala Koki atau pelayan memilih menu “Pesanan” untuk melihat daftar pesanan. Sistem menampilkan daftar pesanan yang masuk, disusun berdasarkan antrian oleh algoritma MLFQ.
Prekondisi	:	Pesanan telah dimasukkan ke dalam sistem. Aplikasi dalam kondisi siap menampilkan daftar pesanan.
Postkondisi	:	Daftar pesanan yang masuk ditampilkan di aplikasi

d. *Use Case*: Mengupdate Status Pesanan Setelah Selesai Diproses

Penjelasan detil mengenai *use case*: Mengupdate Status Pesanan Setelah Selesai Diproses terlihat pada tabel 3.4.

Tabel 3. 4 Skenario *Use Case* Mengupdate Status Pesanan Setelah Selesai Diproses

Aktor	:	Kepala Koki
Deskripsi	:	Kepala Koki mengupdate status pesanan di aplikasi setelah selesai diproses.
Langkah-langkah	:	Kepala Koki membuka aplikasi layanan pemesanan di perangkat Android. Kepala Koki memilih pesanan yang telah selesai diproses. Kepala Koki mengupdate status pesanan menjadi selesai di aplikasi.

		Sistem memperbarui status pesanan di <i>database</i> dan memberikan notifikasi ke pelayan bahwa pesanan telah selesai.
Prekondisi	:	Pesanan telah dimasukkan ke dalam sistem. Kepala Koki telah memproses pesanan.
Postkondisi	:	Status pesanan diperbarui menjadi selesai di <i>database</i> . Informasi status pesanan selesai ditampilkan ke pelayan.

3.5.2.2 Class Diagram

Dalam pengembangan sistem layanan pemesanan untuk Restoran Ayam Penyet Pemuda Semarang, penting untuk merancang struktur data yang tepat agar seluruh informasi dapat diorganisir dan diakses dengan efisien. Salah satu alat utama yang digunakan dalam perancangan perangkat lunak adalah *class diagram*. *Class diagram* memberikan representasi visual dari struktur sistem, menunjukkan berbagai kelas yang digunakan dalam sistem beserta atribut dan metode yang terkait, serta hubungan di antara kelas-kelas tersebut.

Untuk sistem layanan pemesanan ini, beberapa kelas utama telah diidentifikasi sebagai komponen kunci yang diperlukan untuk mencakup fungsionalitas utama. Kelas-kelas tersebut adalah Karyawan, Menu, Pesanan, dan Detil Pesanan. Setiap kelas memiliki peran khusus dalam sistem, yaitu:

a. Kelas Karyawan

Kelas ini merepresentasikan data mengenai karyawan restoran, termasuk pelayan dan kepala koki. Atribut dalam kelas ini mencakup informasi pribadi karyawan seperti ID karyawan, *karyawan*, *username*, *password*, dan *level*. Metode dalam kelas ini akan mencakup fungsi membaca, menambah, mengupdate, dan menghapus karyawan.

b. Kelas Menu

Kelas ini berfungsi untuk menyimpan informasi mengenai item menu yang tersedia di restoran. Atributnya meliputi ID menu, jenis, menu, deskripsi, harga,

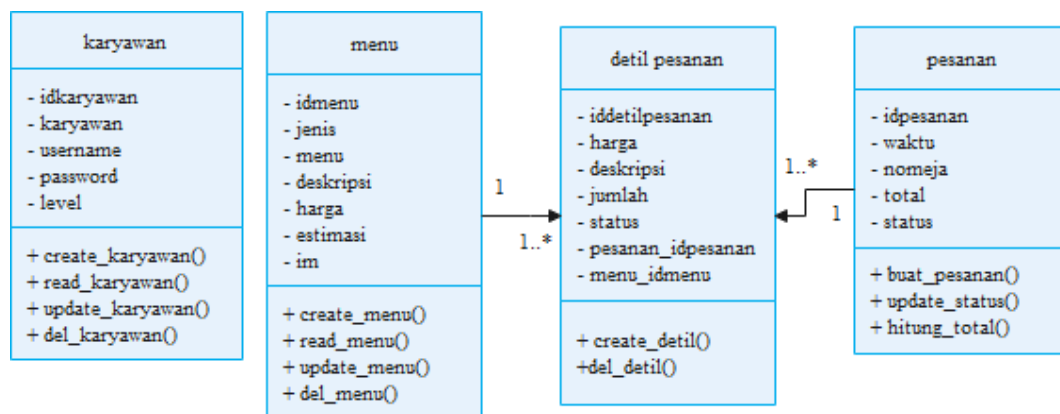
dan estimasi. Metode dalam kelas ini akan mencakup fungsi untuk membaca, menambah, mengupdate, dan menghapus item menu.

c. Kelas Pesanan

Kelas ini mencatat informasi tentang pesanan yang dibuat oleh pelanggan. Atributnya meliputi ID pesanan, waktu, nomeja, total, dan status. Metode dalam kelas ini mencakup fungsi untuk membuat pesanan baru, memperbarui status pesanan, dan menghitung total harga.

d. Kelas Detil Pesanan

Kelas ini berfungsi untuk menyimpan rincian setiap item yang dipesan dalam sebuah pesanan. Atributnya meliputi ID detil pesanan, harga, deskripsi, pesanan_idpesanan, dan menu_idmenu. Metode dalam kelas ini akan mencakup fungsi untuk menambah dan menghapus rincian pesanan.



Gambar 3. 4 Perancangan *Class Diagram* Sistem Diusulkan

3.5.2.3 Antarmuka Pengguna

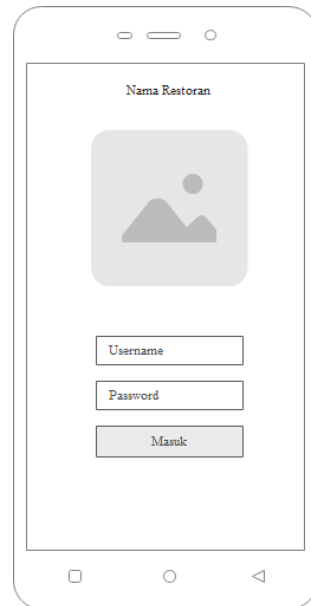
Antarmuka pengguna (*user interface*) adalah aspek penting dalam perancangan aplikasi, yang menentukan seberapa mudah dan intuitif aplikasi tersebut digunakan oleh pelayan dan kepala koki. Berikut adalah desain awal antarmuka pengguna yang akan dikembangkan:

a. Antarmuka Pelayan

Perancangan antarmuka pelayan pada penerapan *Algoritma Multilevel Feedback Queue* pada pelayanan pemesanan Restoran Ayam Penyet Pemuda Semarang adalah sebagai berikut:

1. Antarmuka *Login*

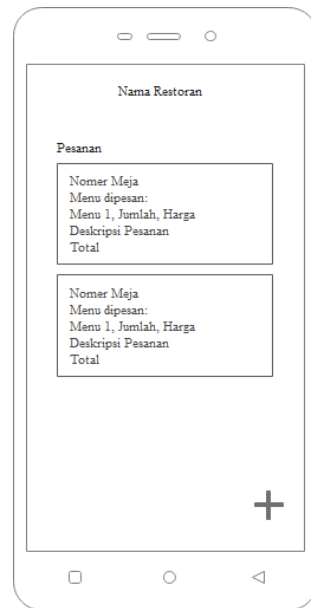
Perancangan antarmuka menu *login* pelayan pada aplikasi Penerapan Algoritma *Multilevel Feedback Queue* Pada Layanan Pemesanan Pada Restoran Ayam Penyet Pemuda Semarang terlihat pada gambar 3.6.



Gambar 3. 5 Perancangan Antarmuka *Login* Pelayan

2. Antarmuka Menu Utama

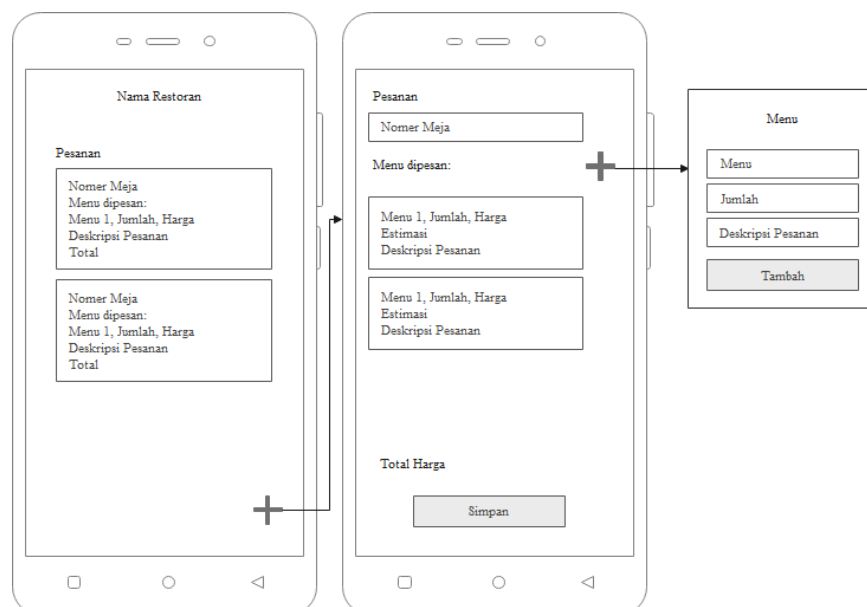
Perancangan antarmuka menu utama dengan akses pelayan pada aplikasi Penerapan Algoritma *Multilevel Feedback Queue* Pada Layanan Pemesanan Pada Restoran Ayam Penyet Pemuda Semarang terlihat pada gambar 3.7.



Gambar 3. 6 Perancangan Antarmuka Menu Utama Akses Pelayan

3. Antarmuka Pesanan

Perancangan antarmuka menu pesanan dengan akses pelayan pada aplikasi Penerapan Algoritma *Multilevel Feedback Queue* Pada Layanan Pemesanan Pada Restoran Ayam Penyet Pemuda Semarang terlihat pada gambar 3.8.



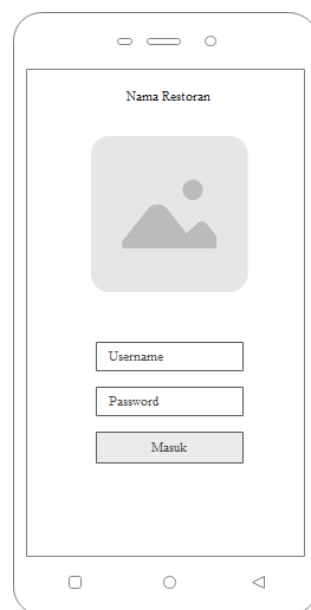
Gambar 3. 7 Perancangan Antarmuka Menu Pesanan Akses Pelayan

b. Antarmuka Kepala Koki

Perancangan antarmuka kepala koki pada penerapan *Algoritma Multilevel Feedback Queue* pada pelayanan Restoran Ayam Penyet Pemuda Semarang adalah sebagai berikut:

1. Antarmuka Login

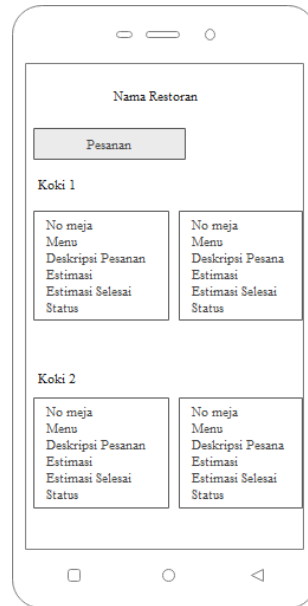
Perancangan antarmuka login kepala koki pada aplikasi Penerapan Algoritma *Multilevel Feedback Queue* Pada Layanan Pemesanan Pada Restoran Ayam Penyet Pemuda Semarang terlihat pada gambar 3.9.



Gambar 3. 8 Perancangan Antarmuka Menu *Login* Kepala Koki

2. Antarmuka Menu Utama

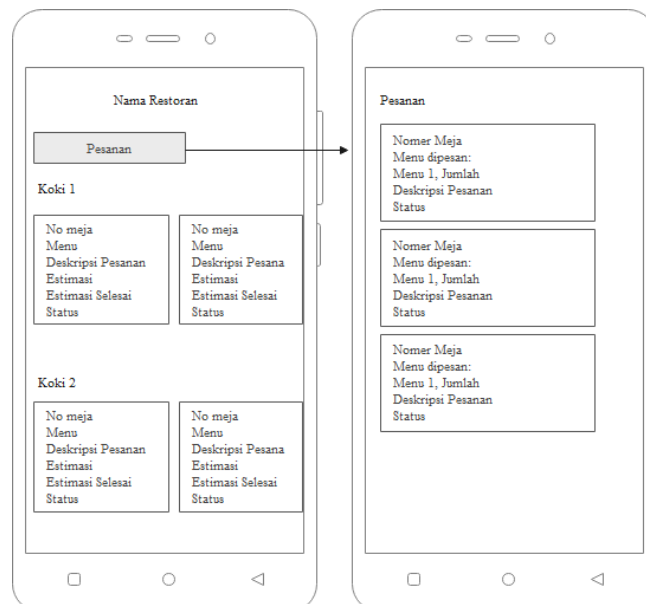
Perancangan antarmuka menu utama akses kepala koki pada aplikasi Penerapan Algoritma *Multilevel Feedback Queue* Pada Layanan Pemesanan Pada Restoran Ayam Penyet Pemuda Semarang terlihat pada gambar 3.10.



Gambar 3. 9 Perancangan Antarmuka Menu Utama Akses Kepala Koki

3. Antarmuka Daftar Pesanan

Perancangan antarmuka daftar pesanan akses kepala koki pada aplikasi Penerapan Algoritma *Multilevel Feedback Queue* Pada Layanan Pemesanan Pada Restoran Ayam Penyet Pemuda Semarang terlihat pada gambar 3.11.



Gambar 3. 10 Perancangan Antarmuka Menu Daftar Pesanan Akses Kepala Koki

3.5.3 Pengkodean

Setelah dilakukan pembuatan desain sistem, tahap selanjutnya adalah pembuatan program menggunakan aplikasi Android Studio dengan bahasa pemrograman Kotlin. Untuk penyimpanan data dilakukan menggunakan MySQL. Hasil implementasi akan dibahas pada Bab IV.

3.5.4 Pengujian

Setelah dilakukan pembuatan program, tahap implementasi dibarengi dengan pengujian agar supaya sistem atau aplikasi yang dibuat berjalan sebagaimana semestinya dan sesuai dengan fungsinya. Pengujian dilakukan menggunakan *black box testing*. Hasil pengujian akan dibahas pada Bab IV.