

BAB II

TINJAUAN PUSTAKA

2.1 Algoritma *Multilevel Feedback Queue* (MLFQ)

Multilevel Feedback Queue (MLFQ) merupakan algoritma penjadwalan yang kompleks yang menggunakan beberapa antrian dengan level prioritas dan *time slice* yang berbeda (Heryana et al., 2023). Kelebihan mendasar yang dimiliki oleh *Multilevel Feedback Queue* (MLFQ) adalah kemungkinan adanya suatu proses berpindah dari satu antrian ke antrian lainnya, misalnya dengan prioritas yang lebih rendah ataupun lebih tinggi. Algoritma ini didefinisikan melalui beberapa parameter, antara lain (Fitriana, 2020):

- a. Jumlah antrian.
- b. Algoritma penjadwalan tiap antrian.
- c. Kapan menaikkan proses ke antrian yang lebih tinggi.
- d. Kapan menurunkan proses ke antrian yang lebih rendah.
- e. Antrian mana yang akan dimasuki proses yang membutuhkan.

2.2 Aplikasi Android

2.2.1 Android

Android merupakan sebuah sistem operasi berbasis Linux yang ditujukan untuk perangkat mobile dengan layar sentuh seperti smartphone dan tablet. Awalnya dikembangkan oleh Android, Inc. dengan dukungan keuangan dari Google, yang kemudian diakuisisi oleh Google pada tahun 2005. Antarmuka Android biasanya memfasilitasi interaksi langsung melalui gerakan sentuhan yang meniru aktivitas di dunia nyata, seperti menggesek, mengetuk, dan mencubit untuk berinteraksi dengan objek pada layar, serta menggunakan keyboard virtual untuk menulis teks. Android merupakan sistem operasi yang bersifat sumber terbuka, dengan Google merilis kode-kodenya di bawah Lisensi Apache. Kode dan lisensi sumber terbuka Android memungkinkan produsen perangkat, operator, dan pengembang aplikasi untuk dengan bebas mengubah dan mendistribusikan perangkat lunak tersebut. Selain itu, Android memiliki komunitas pengembang

aplikasi yang besar yang memperluas fungsionalitas perangkat, biasanya menggunakan bahasa pemrograman Java yang telah disesuaikan (Suryadi Karim & Agarina, 2019).

Android merupakan suatu sistem operasi berbasis Linux yang dirancang khusus untuk perangkat seluler, yang meliputi sistem operasi, middleware, dan aplikasi. Dengan Android, para pengembang diberikan platform terbuka yang memungkinkan mereka untuk membuat aplikasi. Android dapat digunakan pada berbagai perangkat yang diproduksi oleh berbagai produsen. Platform ini juga menyediakan kit pengembangan perangkat lunak untuk menulis kode asli dan menyusun modul perangkat lunak guna menciptakan aplikasi bagi pengguna Android. Selain itu, Android menyediakan platform distribusi aplikasi. Secara keseluruhan, Android merepresentasikan ekosistem aplikasi seluler yang luas (Karman et al., 2019).

2.2.2 Bahasa Pemrograman Kotlin

Kotlin adalah sebuah bahasa pemrograman yang bersifat statis dan dijalankan di atas platform *Java Virtual Machine* (JVM). Kotlin menggunakan kompiler LLVM, yang memungkinkannya untuk dikompilasi menjadi kode JavaScript. Tim pengembang utamanya berasal dari JetBrains, sebuah perusahaan pengembangan perangkat lunak yang berbasis di Rusia. Bahasa pemrograman ini telah mendapatkan popularitas yang signifikan di kalangan para pengembang. Kotlin merupakan bahasa yang kuat dan sangat sesuai digunakan oleh pengembang dalam menulis aplikasi Android. Dikarenakan Kotlin dianggap sebagai bahasa terbaik untuk pengembangan aplikasi Android. Salah satu keunggulan utama dari Kotlin adalah (Aljundi dan Akbar, 2018):

- a. Kesederhanaan: Kotlin dapat mengurangi boilerplate dan kompleksitas kode yang biasa Anda tulis saat menggunakan bahasa Java.
- b. Aman: Kotlin dapat menjamin bahwa sintaks apa pun yang Anda buat selama proses kompilasi akan mencegah kemungkinan kesalahan. Misalnya, Anda dapat mencegah terjadinya *NullPointerException* saat pemrograman dalam bahasa Java.

- c. *Versatile*: Kotlin sendiri berasal dari bahasa aslinya yaitu Java, jadi Kotlin sebenarnya sama dengan Java. Artinya Kotlin juga bisa digunakan untuk mengembangkan aplikasi di web dan perangkat seluler.
- d. *Interoperabilitas*: Kotlin tidak sama dengan bahasa turunan Java lainnya (seperti Scala atau Clojure) dan tidak dapat berjalan bersamaan dengan kode yang ditulis di Java. Kotlin dapat membaca kode dan pustaka lama yang digunakan atau ditulis di Java, dan sebaliknya.

2.2.3 MySQL

MySQL adalah pengembangan lanjutan dari proyek UNIREG yang dikerjakan oleh Michael Monty Widenius dan TcX (perusahaan perangkat lunak asal Swedia). *MySQL* adalah DBMS yang *open source* dengan dua bentuk lisensi, yaitu *free software* (perangkat lunak bebas) dan *shareware* (perangkat lunak berpemilik yang penggunaannya terbatas). Jadi, *MySQL* adalah *database server* yang gratis dengan lisensi GNU *General Public License* (GPL) sehingga dapat dipakai untuk keperluan pribadi atau komersial tanpa harus membayar lisensi yang ada (Fitri, 2020).

Seperti yang sudah disebutkan sebelumnya, *MySQL* masuk ke dalam jenis RDBMS (*Relational Database Management Sistem*). Maka dari itu, istilah semacam baris, kolom, tabel dipakai pada *MySQL*. *MySQL* merupakan *server* yang melayani *database*, kita dapat mempelajari pemrograman khusus yang disebut *query* (perintah) *SQL*. *MySQL* merupakan *database engine* atau *server database* yang mendukung bahasa *SQL* sebagai bahasa interaktif dalam mengelola data. *MySQL* adalah sebuah perangkat lunak sistem manajemen basis data *SQL* atau DBMS yang *multithread*, *multi-user* (Enterprise & Agung, 2018).

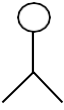



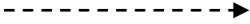
2.3 UML (*Unified Modeling Language*)

UML adalah teknik pemodelan visual yang berfungsi sebagai sarana untuk merancang sistem berorientasi objek. UML dapat digunakan untuk desain, dokumentasi, atau perencanaan perangkat lunak (Rosa & Shalahuddin, 2018). Jenis pemodelan UML yang digunakan dalam pengembangan sistem pada penelitian ini adalah *use case diagram*, *activity diagram*, dan *class diagram*.

2.3.1 Use Case Diagram

Sebuah *use case* menggambarkan interaksi antara satu atau lebih aktor dan sistem informasi yang dibuat. Secara garis besar *use case* digunakan untuk mengetahui fungsi apa saja yang ada pada suatu sistem informasi dan siapa saja yang berhak menggunakan fungsi tersebut. Prasyarat untuk memberi nama pada *use case* adalah bahwa nama tersebut harus didefinisikan secara sederhana dan mudah dipahami. Sebuah *use case* memiliki dua elemen utama: definisi perilaku aktor dan *use case*.

Tabel 2. 1 Simbol *Use Case Diagram*

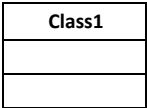
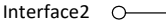




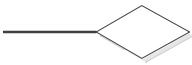
No	Simbol	Keterangan Fungsi
1.	<i>Actor</i> 	Aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.
2.	<i>Use Case</i> 	<i>Use Case</i> adalah deskripsi dari urutan aksi- aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
3.	Asosiasi 	Asosiasi adalah apa yang menghubungkan antara objek satu dengan objek yang lainnya.
4.	Generalisasi 	Generalisasi adalah hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya atau sebaliknya dari bawah ke atas.
5.	<i>Defendency</i> 	<i>Defendency</i> (ketergantungan) adalah hubungan dimana perubahan yang terjadi pada suatu elemen defenden (mandiri) akan mempengaruhi elemen yang bergantung padanya (<i>independen</i>).

2.3.2 Class Diagram

Diagram kelas menggambarkan struktur sistem dalam kaitannya dengan definisi kelas-kelas yang dibuat untuk membangun sistem. Kelas memiliki apa

yang disebut atribut dan metode atau operasi. Atribut adalah variabel yang dimiliki suatu kelas, dan operasi atau metode adalah fungsi yang dimiliki suatu kelas.

Tabel 2. 2 Simbol *Class Diagram*

No	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem.
2.	Natarmuka/ <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	Asosiasi 	Relasi antar kelas dalam makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	Asosiasi berarah 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
6.	Kebergantungan 	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.	Agregasi 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

2.4 Black Box Testing

Black box testing merupakan pengujian untuk mengetahui fungsi perangkat lunak yang telah berjalan sesuai dengan kebutuhannya. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan pengeluaran perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian kotak hitam (*black box*) juga disebut pengujian perilaku yang berfokus pada persyaratan fungsional perangkat

lunak, artinya teknik pengujian *black box* memungkinkan untuk membuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan semua kebutuhan fungsional untuk program (Rosa & Shalahuddin, 2018). *Black box testing* berupaya untuk menemukan kesalahan dalam kategori berikut:

- a. Fungsi yang salah atau hilang
- b. Kesalahan antarmuka
- c. Kesalahan dalam struktur data atau akses basis data eksternal
- d. Kesalahan perilaku atau kinerja
- e. Kesalahan instalasi dan penghentian

2.5 Penelitian terkait

Jurnal atau naskah publikasi yang terkait dengan penelitian *Multilevel Feedback Queue* (MLFQ) terlihat pada tabel 2.3.

Tabel 2. 3 Penelitian Terkait

No	Judul	Penulis	Metode/Aplikasi	Kesimpulan
1.	Push-Notification Order Menu Menggunakan Metode Multilevel Feedback Queue Pada Pondok Lesehan dan Resto Lembah Dempo Kota Metro (Fitriana, 2020)	Amelia Fitriana	Prototype Web mobile	Aplikasi notifikasi order makanan berbasis web mobile dengan Metode Multilevel Feedback Queue membantu mengatur antrian pesanan di restoran.
2.	Sistem E-Booking Pelayanan Jasa Salon Menggunakan Metode Multilevel Feedback Queue	Mahdalena Serumpun, Imam Ahmad, Maulana	Prototype Web	Aplikasi ini memfasilitasi proses booking studio foto dan meningkatkan hubungan dengan

No	Judul	Penulis	Metode/Aplikasi	Kesimpulan
	(Study Kasus : Salon Griya) (Serumpun et al., 2022)	Aziz Assuja, Renaldy Hermanto, Farid Jeck Hendri		konsumen melalui sistem e-booking. Pelanggan diwajibkan mengisi feedback atau memberikan rating untuk meningkatkan kualitas layanan.
3.	Implementasi Algoritma Multilevel Feedback Queue Scheduling Untuk Sistem Antrian Booking Menggunakan Framework Flutter (Studi Kasus : Bengkel Putra Jaya Motor) (Mujab & Zakaria, 2023)	Saiful Mujab, Hadi Zakaria	Rapid Application Development (RAD) Android	Aplikasi antrian booking ini meningkatkan efisiensi waktu tunggu dan kualitas layanan, memberikan pengalaman pengguna yang lebih baik.
4.	Penerapan Metode Multilevel Queue Antrian Pendaftaran Pasien Pada Klinik Talang Kelapa Berbasis Android (Huda & Novriyadi, 2023)	Nurul Huda, Novriyadi	Prototype Android	Aplikasi Android untuk antrian pendaftaran pasien di Klinik Talang Kelapa memudahkan administrasi dan pencatatan data.

No	Judul	Penulis	Metode/Aplikasi	Kesimpulan
				<p>Dengan algoritma multilevel queue, aplikasi ini mengurutkan pasien berdasarkan waktu kedatangan dan menghitung quantum time, meningkatkan efisiensi sistem.</p>
5.	<p>Implementasi Algoritma Penjadwalan Multilevel Feedback Queue pada Aplikasi Berbasis Flutter (Artia et al., 2023)</p>	<p>Fransiska Artia, I Gede Maha, M. Rakhmat Dramaga, Aqwam Rosadi Kardian</p>	<p>Kuatitatif Desktop dan Android</p>	<p>Algoritma MFQ mengurangi masalah seperti process starvation dan mengoptimalkan sumber daya, menghasilkan kinerja berbeda dalam hal frame rate pada kedua platform.</p>
6.	<p>Penerapan Metode Multilevel Feedback Queue Pada Sistem Informasi Pemesanan Paket Haji dan Umrah Di Pt.Aubaine Kabuhayan</p>	<p>Muhammad Arif Suhada, Ilka Zufria, Ali Ikhwan</p>	<p>Waterfall Website</p>	<p>Metode Multilevel Feedback Queue cocok untuk sistem pemesanan paket haji dan umrah di PT. Aubaine Kabuhayan, memberikan antrian prioritas</p>

No	Judul	Penulis	Metode/Aplikasi	Kesimpulan
	(Suhada et al., 2020)			berdasarkan keberangkatan jamaah.
7.	Perancangan Sistem Informasi Penjadwalan Program Kerja Penjaminan Mutu Universitas AMIKOM Dengan Metode Multilevel Feedback Queue (Verawati & Sulistiyono, 2017)	Ike Verawati, Mulia Sulistiyo no	Website	Sistem penjadwalan ini menggunakan algoritma Multilevel Feedback Queue, mengelompokkan kegiatan berdasarkan prioritas dan memungkinkan perpindahan proses penjadwalan.
8.	Perancangan Sistem Informasi Pemesanan Makanan Pada Coffee Shop Dengan Menerapkan Algoritma Multilevel Feedback Queue (Zen et al., 2024)	Irwan, Muhammad Zen, Muhammad Davy Anggara Saragih	Website	Sistem informasi pemesanan makanan berbasis website ini mempermudah pelanggan mendapatkan informasi menu dan melakukan pemesanan tanpa menunggu pelayan.
9.	Algoritma Content-Based Filtering dan Algoritma Multilevel Feedback Queue	Aei Dhelwis Okdhianty Putri, Fauziah,	Waterfall Website	Kedua metode tersebut cocok untuk sistem pemesanan makanan dan

No	Judul	Penulis	Metode/Aplikasi	Kesimpulan
	Pada Proses Pemesanan Makanan Berbasis Web (Putri et al., 2023)	Agus Iskandar		minuman karena memberikan hasil dan akurasi rekomendasi yang tinggi.
10.	Aplikasi Mobile Pendaftaran Pasien Klinik Berbasis Algoritma Multilevel Queue Dan FIFO Untuk Meningkatkan Layanan Antrian (Priambudi et al., 2024)	Abiyoga Priambudi, Yunan Fauzi Wijaya, Ucuk Darusalam3	Android	Algoritma Multilevel Queue dapat digunakan untuk aplikasi daftar pasien di klinik dengan mengurutkan pasien berdasarkan waktu dan jenis pembayaran.
11.	Penjadwalan Perbaikan Kerusakan Perangkat Lab Komputer IIB Darmajaya Menggunakan Metode Multilevel Feedback Queue Berbasis Mobile (Sefriadi, 2020)	Rian Sefriadi	Prototype Web Mobile	Aplikasi ini memudahkan asisten laboratorium dalam melaporkan kerusakan perangkat dan admin laboran dalam memantau laporan.
12.	E-Booking Personal Fotografer dan Videografer Menggunakan	Wayan Rida Arista	Prototype Android	Aplikasi ini memudahkan pencarian fotografer dan

No	Judul	Penulis	Metode/Aplikasi	Kesimpulan
	Metode Multilevel Feedback Queue Berbasis Android (Rida, 2020)			videografer dengan portofolio yang tersedia, mempercepat pemesanan dengan metode terintegrasi, dan memungkinkan pembayaran langsung dengan bukti transfer.