JURNAL MAHAJANA INFORMASI
e-ISSN 2527 – 8290
Vol. 8 No. 2, Desember 2023, Hal. 55 - 63

Implementasi Algoritma Penjadwalan Multilevel Feedback Queue pada Aplikasi Berbasis Flutter

Fransiska Artia[1*], I Gede Maha[2], M. Rakhmat Dramaga[3], Aqwam Rosadi Kardian[4]

[1,2,3]*Program Studi Rekayasa Kriptografi Politeknik Siber dan Sandi, Jl. Raya H. Usa, Putat Nutug, Kec. Ciseeng, Kabupaten Bogor, Jawa Barat*
[4]*Sistem Informasi, STMIK Jakarta STI&K, Jl. Bri Radio Dalam No.17, RT.14/RW.3, Gandaria Utara, Kec. Kby. Baru, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta*

*Penulis Korespondensi: E-mail: fransiska.artia@student.poltekssn.ac.id , Phone: +6281212743669

Article Info
Received : *03 Desember 2023*
Revised : *16 Desember 2023*
Accepted : *19 Desember 2023*

Abstract : *Application development has experienced a shift towards a cross-platform approach, where developers can write code once for multiple platforms. Flutter, as a fast-growing cross-platform framework with a wide community is one of the most popular cross-platform frameworks today. However, along with that, there is an increase in application complexity which makes the concept of multitasking very important. This article proposes the implementation of the Multilevel Feedback Queue (MLQ) scheduling algorithm in Flutter applications, which can help manage tasks and improve application efficiency. This research aims to examine the changes in application efficiency after the implementation of MLQ, as well as examine whether the changes remain relevant on different operating systems, namely Windows and Android. The implemented MLQ algorithm is an algorithm with adjustments to the calculation of time quantum and integration with the Shortest Job First (SJF) algorithm based on previous research. Tests were conducted using the Flutter benchmarking feature to measure the application frame rate before and after MLQ implementation on Windows and Android. The results of this study found that the implementation of MLQ increased application efficiency by 269% on the Windows operating system and 155% on the Android operating system.*

Abstrak : Pengembangan aplikasi telah mengalami pergeseran menuju pendekatan cross-platform, di mana pengembang dapat menulis kode satu kali untuk berbagai platform. Flutter, sebagai cross-platform framework yang berkembang pesat dengan komunitas yang luas menjadi salah satu cross-platform framework yang cukup populer pada masa ini. Namun, beriringan dengan itu terdapat peningkatan kompleksitas aplikasi yang menyebabkan konsep multitasking menjadi sangat penting. Artikel ini mengusulkan implementasi algoritma penjadwalan Multilevel Feedback Queue (MLQ) pada aplikasi Flutter, yang dapat membantu mengelola tugas dan meningkatkan efisiensi aplikasi. Penelitian ini bertujuan untuk menguji perubahan efisiensi aplikasi setelah penerapan MLQ, serta memeriksa apakah perubahan tersebut tetap relevan pada sistem operasi yang berbeda, yaitu Windows dan Android. Adapun algoritma MLQ yang diimplementasikan merupakan algoritma dengan penyesuaian pada kalkulasi time quantum dan integrasi dengan algoritma Shortest Job First (SJF) yang berdasar pada penelitian yang terdahulu. Pengujian dilakukan dengan menggunakan fitur benchmarking Flutter untuk mengukur kecepatan frame rate aplikasi sebelum dan setelah implementasi MLQ pada Windows dan Android. Hasil dari penelitian ini adalah ditemukan bahwa implementasi MLQ meningkatkan efisiensi aplikasi sebesar 269% pada sistem operasi Windows dan 155% pada sistem operasi Android.

Keyword : ***Scheduling Algorithm, Multilevel Feedback Queue, Dart, Flutter***

e-ISSN 2527 – 8290
Vol. 8 No. 2, Desember 2023, Hal. 55 - 63

PENDAHULUAN

Dewasa ini, tren dalam *framework* pengembangan aplikasi mengalami perubahan yang signifikan menuju pendekatan *cross-platform*. *Cross-platform framework* memungkinkan pengembang untuk menulis kode aplikasi satu kali dan menjalankannya pada berbagai platform, yang kemudian mengurangi kebutuhan akan kode untuk platform spesifik dan mengurangi biaya pengembangan (1). Salah satu *cross-platform framework* yang berkembang pesat saat ini adalah Flutter.

Keunggulan Flutter sebagai *cross-platform framework* disebabkan oleh beberapa faktor. Pertama, Flutter memungkinkan pengembangan aplikasi mobile dengan kinerja tinggi yang dapat diandalkan untuk berbagai platform, termasuk iOS, Android, Web, dan Desktop (2). Dukungan multi-platform ini memungkinkan pengembang untuk menjangkau pengguna yang lebih luas dan memastikan pengalaman pengguna yang konsisten di berbagai perangkat yang berbeda . Selain itu, perkembangan yang cepat dan kemudahan pembelajaran Flutter menjadikannya pilihan yang menarik dibandingkan dengan *cross-platform framework* lainnya (3).

Namun, beriringan dengan berkembangnya beban kerja yang harus ditanggung oleh aplikasi yang semakin rumit, muncul suatu konsep yang disebut dengan multitasking. Multitasking adalah kemampuan suatu sistem untuk mengerjakan berbagai tugas dalam satu waktu atau kemampuan untuk berpindah dari satu tugas ke satu tugas yang lain dengan cepat yang memberikan kesan pengerjaan paralel (4). Oleh karena itu, diperlukan suatu solusi yang dapat membantu aplikasi untuk mengelola beban dan tugas tersebut agar aplikasi tetap dapat berjalan dengan baik. Salah satu solusinya yaitu algoritma penjadwalan atau *scheduling algorithm*.

Terdapat berbagai jenis scheduling algorithm, seperti First In First Out (FIFO), Shortest Job First (SJF), dan Round Robin. Dari berbagai scheduling algorithm yang ada, Multilevel Feedback Queue (MLQ) dinilai lebih baik daripada algoritma yang lain karena kemampuannya untuk meminimalkan rata-rata flow time tugas, sehingga meningkatkan daya tanggap sistem dan mengelola campuran proses dengan karakteristik eksekusi yang beragam secara efisien (5,6). Tidak hanya itu, karena terdapat berbagai campuran proses dan karakteristik eksekusi yang dapat dikembangkan dengan MLQ, hingga saat ini banyak penelitian yang dilakukan untuk mengembangkan algoritma MLQ yang lebih efisien. Adapun algoritma MLQ yang digunakan pada penelitian ini berdasar pada penelitian yang dilakukan oleh Thombare, dkk. yang melakukan pengembangan terhadap kalkulasi time quantum dan integrasi MLQ dengan SJF (7).

Penelitian ini hendak mengimplementasikan algoritma Multilevel Feedback Queue (MLQ) dalam suatu aplikasi yang dikembangkan menggunakan framework Flutter. Tujuan dari penelitian ini yaitu untuk menguji perubahan tingkat efisiensi aplikasi setelah penerapan algoritma MLQ. Selain itu, penelitian ini juga akan menguji apakah perubahan efisiensi tersebut tetap relevan ketika aplikasi dijalankan pada sistem operasi yang berbeda, yakni Windows dan Android.

METODE

Adapun metodologi penelitian yang digunakan pada penelitian ini yaitu metode kuantitatif yang menekankan pada pengumpulan dan analisis data numerik. Metode ini melibatkan penggunaan instrumen pengumpulan data terstruktur, seperti eksperimen, dan penerapan analisis statistik untuk menarik kesimpulan dari data tersebut (8).

Algoritma penjadwalan pada CPU adalah metode yang digunakan untuk menentukan urutan kerja yang akan dieksekusi oleh CPU. Algoritma penjadwalan bertujuan untuk menata sumber daya agar dapat beroperasi secara optimal meliputi beberapa proses dalam

e-ISSN 2527 – 8290
Vol. 8 No. 2, Desember 2023, Hal. 55 - 63

sistem operasi yang berhubungan dengan susunan operasi, umumnya terdapat pada sebuah sistem komputer. Menurut, Slamet Riadi & Faruq Ulum (2021), tujuan utama penjadwalan adalah menentukan urutan operasi yang perlu dijalankan lebih dahulu, kapan, dan rentang waktu yang digunakan untuk mengoperasikan prosedur tersebut. Tujuan lainnya yaitu untuk meningkatkan performa suatu sistem dengan mempertimbangkan beberapa kriteria tertentu. Algoritma penjadwalan mengevaluasi sejumlah parameter, seperti fairness, waktu tunggu rata-rata, waktu respons waktu putar rata-rata dan keluaran, agar mencapai hasil optimal (9).

Algoritma Multilevel Feedback Queue adalah salah satu metode yang menggunakan format antrean dengan model Multi Channel Single Server. Keunggulan utama yang dimiliki oleh algoritma Multilevel Feedback Queue antara lain adanya potensi suatu proses beralih dari suatu jenjang antrean menuju jenjang antrean lain, seperti perpindahan dengan preferensi yang lebih rendah atau lebih tinggi (10).

Seperti yang disiratkan oleh namanya, algoritma Multilevel Feedback Queue terdiri dari seperangkat dua atau lebih antrian proses First in First Out (FIFO), dan suatu aturan penjadwalan. Terkait dengan setiap tingkat dalam Multilevel Feedback Queue adalah sebuah quantum waktu maksimum yang dibatasi setiap proses untuk berjalan sebelum harus melepaskan CPU. Antrean menyimpan Process Control Block, masing-masing merepresentasikan suatu proses kepada sistem operasi, menyimpan data status tentang proses terkait dan mempertahankan referensi ke lokasi memori. Mulai dari titik ini, deskripsi proses dalam konteks memasuki dan meninggalkan antrean berkaitan dengan Process Control Block terkait (11).

Flutter adalah salah satu Software Development Kit yang dirancang agar dapat mengembangkan program mobile dengan performa tinggi. Program yang dikembangkan menggunakan Flutter dapat dijalankan pada platform Android maupun iOS walaupun hanya menggunakan basis kode tunggal yang dikelola oleh Google dan bersifat open source. Tujuan utama dari Flutter adalah memberikan kemampuan kepada pengembang untuk menciptakan aplikasi yang memiliki kinerja tinggi dan memberikan pengalaman pengguna yang alami, meskipun dijalankan pada platform yang berbeda (12). Menurut Austin D. Latture & Grand Valley (2020), Keunggulan Flutter jelas terlihat karena memungkinkan tim pengembangan menggunakan satu kode sumber untuk aplikasi seluler sambil tetap melayani pengguna pada kedua sistem operasi iOS dan Android (13).

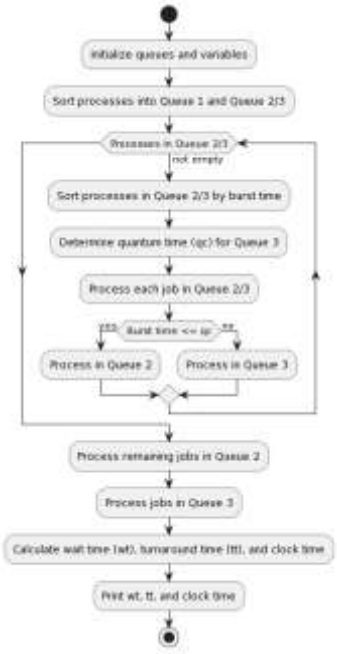
Menurut Daniele Palumbo (2021), Flutter ditulis dalam Dart dan merupakan sebuah cross-platform framework yang didasarkan pada pendekatan terkompilasi. Seluruh jalur penggambaran Flutter diimplementasikan langsung oleh framework dan bersifat independen dari komponen asli sistem operasi yang mendasarinya. Oleh karena itu, Flutter berbeda dari cross-platform framework lainnya seperti React Native atau Ionic. Dart adalah bahasa prosedural dengan aspek fungsional yang dirilis pada tahun 2011. Selain itu, Dart dapat dieksekusi baik dalam bentuk terkompilasi maupun diinterpretasi. Hal ini memungkinkan Flutter memiliki lingkungan pengembangan yang cepat dan dinamis, menawarkan fitur seperti Hot Reload. Selanjutnya, kode dapat dikompilasi menjadi kode mesin dan dieksekusi tanpa beban interpretasi (14).

Implementasi algoritma penjadwalan Multilevel Feedback Queue pada aplikasi berbasis Flutter dilakukan dengan urutan proses dan tingkat antrian dapat bertransisi dari tingkat antrean awal ke tingkat antrean yang lebih tinggi. Adapun algoritma MLQ yang digunakan pada penelitian ini adalah algoritma dari Thombare, dkk. yaitu prosedur dieksekusi dalam tiga antrean, yaitu q1, q2, dan q3. Time quantum dari antrean pertama, q1, dijaga agar tetap pendek sehingga proses yang pendek dieksekusi dalam antrean ini sendiri. Time

JURNAL MAHAJANA INFORMASI
e-ISSN 2527 – 8290
Vol. 8 No. 2, Desember 2023, Hal. 55 - 63

quantum dari q1 diatur ke 10, sehingga proses dengan burst time 10 atau kurang dapat dieksekusi, dan burst time ditambahkan ke variabel qc. Variabel qc dibuat untuk menghitung time quantum antrean kedua dan ketiga. Jika burst time melebihi 10, proses-proses ini akan berpindah ke antrean kedua. Di dalam antrean kedua, proses-proses diatur menurut algoritma penjadwalan Shortest Job First (SJF) dan dieksekusi dengan tepat. Jika burst time melebihi time quantum yang dihitung, proses kemudian memasuki antrean ketiga (8).

Pseudocode dari algoritma penjadwalan Multilevel Feedback Queue yang digunakan dalam bentuk flowchart ditunjukkan oleh Gambar 1.



Gambar 1. *Flowchart* Algoritma Penjadwalan Multilevel Feedback Queue

Berikut adalah spesifikasi perangkat yang digunakan untuk proses implementasi dan pengujian pada Tabel 1 dan Tabel 2.

a. Windows

Tabel 1. Spesifikasi Perangkat Windows

Kategori	Keterangan
Prosesor	Intel (R) Core i5-12450H ~2.0GHz
RAM	16GB DDR4
Penyimpanan	500GB SSD SATA
Sistem Operasi	Windows 11 Home 64-bit

b. Android

Tabel 2. Spesifikasi Perangkat Android

Kategori	Keterangan
Prosesor	Mediatek MT6785V/CC Helio G90T (12nm)
RAM	8GB RAM
Penyimpanan	64GB
Sistem Operasi	Android 9.0 (Pie)

Pengujian *benchmark* adalah proses membandingkan dan mengevaluasi kinerja produk, layanan, atau proses terhadap standar industri atau praktik terbaik untuk mengidentifikasi area yang perlu ditingkatkan dan mencapai kinerja yang unggul (15,16). Hal ini melibatkan identifikasi, penyebaran, dan implementasi pengetahuan dan praktik terbaik untuk mencapai peningkatan dan daya saing yang berkelanjutan (17,18). *Benchmarking* dapat dilakukan dalam berbagai bentuk, seperti *benchmarking* kinerja, yang melibatkan perbandingan data kinerja dari proses atau aktivitas serupa, dan *benchmarking* praktik terbaik, yang berfokus pada identifikasi, adaptasi, dan implementasi praktik yang menghasilkan hasil kinerja terbaik (16).

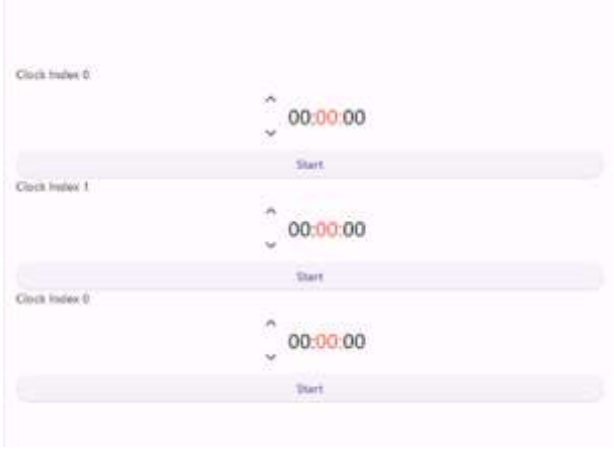
Pada penelitian ini, uji *benchmark* yang akan dilakukan berupa *benchmark* kinerja dengan menggunakan fitur benchmarking yang ada pada *framework* Flutter yaitu fitur *Developer Tools* untuk mencari kecepatan *frame rate* dair aplikasi sebelum dan sesudah menggunakan algoritma MLQ saat menggunakan Windows maupun Android.

HASIL DAN PEMBAHASAN

Aplikasi yang menjadi target uji implementasi algoritma adalah aplikasi timer yang menampilkan tiga timer yang bekerja

JURNAL MAHAJANA INFORMASI
e-ISSN 2527 – 8290
Vol. 8 No. 2, Desember 2023, Hal. 55 - 63

secara paralel. Tampilan aplikasi dapat dilihat pada Gambar 2.



Gambar 2. Tampilan awal aplikasi

Perubahan pada kode program untuk melakukan implementasi algoritma terlihat pada Gambar 3.

Gambar 3. Implementasi Algoritma pada Program

Pengujian yang dilakukan terhadap aplikasi dilakukan menggunakan Flutter Developer Tools pada Flutter. Dart DevTools menyediakan alat untuk memahami kinerja aplikasi dan memecahkan masalah, termasuk pemantauan kinerja, analisis memori, dan alat debugging. Hot Reload adalah fitur yang sangat berguna, memungkinkan pengembang mengganti kode sumber tanpa memulai ulang aplikasi, mempercepat siklus pengembangan. Command Line Tools Flutter menyederhanakan tugas-tugas pengembangan melalui perintah baris perintah untuk instalasi paket, menjalankan tes, dan membuat proyek baru. Pengintegrasian Flutter DevTools di Visual Studio Code dan Android Studio memberikan antarmuka pengguna terpadu dalam lingkungan pengembangan terintegrasi, memungkinkan akses mudah dan penggunaan alat pengembangan tanpa harus meninggalkan IDE.

JURNAL MAHAJANA INFORMASI
e-ISSN 2527 – 8290
Vol. 8 No. 2, Desember 2023, Hal. 55 - 63

Tabel 3. Hasil Uji Benchmark Aplikasi Desktop

Tanpa Algoritma MLFQ

Frame	Frame Rate (ms)
1	20.6
2	24.8
3	22.2
4	24.4
5	17.1
6	22.8
7	19.6
8	19.3
9	23.0
10	24.0

Berikut ditampilkan hasil pengujian *benchmark* pada aplikasi Desktop dengan sistem operasi Windows sebanyak 15 frame dengan algoritma Multilevel Feedback Queue pada Tabel 4.

Tabel 4. Hasil Uji Benchmark Aplikasi Desktop dengan Algoritma MLFQ

Frame	Frame Rate (ms)
1	2.2
2	8.0
3	5.1
4	6.4
5	8.8
6	8.2
7	8.8
8	8.9
9	7.5
10	11.0

Berikut ditampilkan hasil pengujian *benchmark* pada aplikasi yang dijalankan pada sistem operasi berbasis Android sebanyak 15 frame tanpa algoritma Multilevel Feedback Queue pada Tabel 5.

Tabel 5. Hasil Uji Benchmark Aplikasi Androin Tanpa Algoritma MLFQ

Frame	Frame Rate (ms)
1	14.0
2	12.7
3	10.4
4	10.2
5	15.6
6	9.9
7	26.0
8	10.2
9	10.2
10	20.5

Berikut ditampilkan hasil pengujian *benchmark* pada aplikasi yang dijalankan pada sistem operasi berbasis Android sebanyak 15 frame dengan algoritma Multilevel Feedback Queue pada Tabel 6.

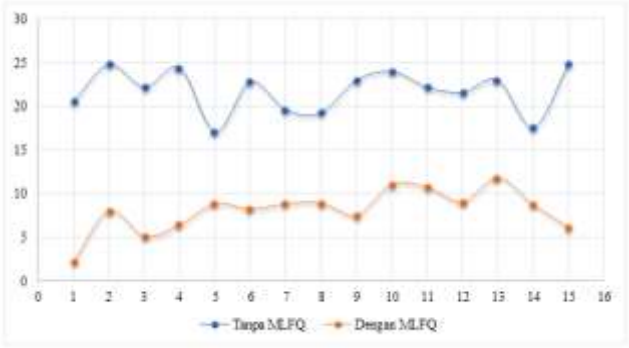
Tabel 6. Hasil Uji Benchmark Aplikasi Androin dengan Algoritma MLFQ

Frame	Frame Rate (ms)
1	6.2
2	8.9
3	7.6
4	21.1
5	6.9
6	7.8
7	13.9
8	8.8
9	6.4
10	8.5

PEMBAHASAN

Berikut adalah grafik hasil uji aplikasi Desktop tanpa dan dengan algoritma MLQ pada Gambar 4.

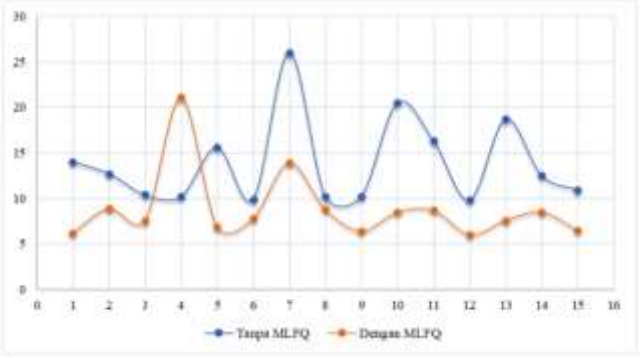
JURNAL MAHAJANA INFORMASI
e-ISSN 2527 – 8290
Vol. 8 No. 2, Desember 2023, Hal. 55 - 63



Gambar 4. Grafik Hasil Uji Aplikasi Desktop

Berdasarkan hasil uji yang sudah dilakukan, ditemukan bahwa rata-rata kecepatan *frame rate* dari aplikasi Desktop tanpa menggunakan algoritma MLQ adalah **21,8ms** dan **8,08ms** dengan menggunakan algoritma MLQ. Terdapat peningkatan kecepatan *frame rate* aplikasi sebesar **269%** dengan menggunakan algoritma MLQ.

Berikut adalah grafik hasil uji aplikasi Android tanpa dan dengan algoritma MLQ pada Gambar 5.



Gambar 5. Grafik Hasil Uji Aplikasi Android

Berdasarkan hasil uji yang sudah dilakukan, ditemukan bahwa rata-rata kecepatan *frame rate* dari aplikasi Android tanpa menggunakan algoritma MLQ adalah **13,86ms** dan **8,89ms** dengan menggunakan algoritma MLQ. Terdapat peningkatan kecepatan *frame rate* aplikasi sebesar **155%** dengan menggunakan algoritma MLQ.

Terdapat lonjakan *frame rate* pada *frame* ke-4 di aplikasi yang menggunakan algoritma MLQ. Hal ini disebabkan oleh adanya *frame drop* pada saat aplikasi dijalankan. *Frame drop* sendiri dapat disebabkan oleh banyak faktor, seperti kesalahan bit pada *frame* yang diterima, keterbatasan perangkat keras selama kondisi beban tinggi, dan masalah terminal tersembunyi (18).

KESIMPULAN DAN SARAN

Berdasarkan penelitian yang sudah dilakukan, dengan hasil peningkatan sebesar 269% dan 155% untuk aplikasi Windows dan Android, terlihat jelas bahwa penggunaan algoritma penjadwalan Multilevel Feedback Queue yang dalam hal ini berupa algoritma yang diajukan oleh Thombare, dkk. dapat meningkatkan efisiensi aplikasi yang dibangun menggunakan framework Flutter. Perbedaan persentase peningkatan frame rate antara aplikasi Windows dan Android disebabkan oleh kemampuan algoritma untuk beradaptasi dengan karakteristik spesifik dan alokasi sumber daya masing-masing platform. Kemampuan algoritma MFQ untuk mengurangi masalah seperti process starvation dan mengoptimalkan pemanfaatan sumber daya dapat menghasilkan kinerja yang berbeda dalam hal frame rate pada kedua platform (19).

Penelitian berikutnya dapat mencoba menerapkan modifikasi lain dari algoritma Multilevel Feedback Queue yang lebih baik serta menerapkan algoritma pada aplikasi yang lebih kompleks. Selain itu, aplikasi juga dapat diuji pada lingkungan platform sistem operasi lain seperti Linux dan iOS.

DAFTAR PUSTAKA

1. Biørn-Hansen A, Rieger C, Grønli T-M, Majchrzak TA, Ghinea G. An empirical investigation of performance overhead in cross-platform mobile development frameworks. Empir Softw Eng. 2020 Jul 9;25(4):2997–3040.

2. AL-atraqchi OMA. A Proposed Model for Build a Secure Restful API to Connect between Server Side and Mobile Application Using Laravel Framework with Flutter Toolkits. Cihan Univ Sci J. 2022 Aug 20;6(2):28–35.
3. mohammed ahmed O. Backend as a Service Cloud Computing Integrated with Cross-platform Mobile Development Framework to Create an E-learning application that works in Mobile and Web with a single codebase. In: 4th International Conference on Communication Engineering and Computer Science (CIC-COCOS'2022). Cihan University; 2022. p. 50–7.
4. Setiawan A, Sebastian D, Adi Nugraha K. Implementasi Fitur Sinkronisasi Basis Data Pada Aplikasi Monitoring Keuangan Pada Wilayah Dengan Keterbatasan Jaringan Internet. J Terap Teknol Inf. 2022 Apr 30;6(1):63–74.
5. Adler RF, Benbunan-Fich R. The Effects of Task Difficulty and Multitasking on Performance. Interact Comput. 2015 Jul;27(4):430–9.
6. Becchetti L, Leonardi S. Nonclairvoyant scheduling to minimize the total flow time on single and parallel machines. J ACM. 2004 Jul;51(4):517–39.
7. Van Houdt B, Van Velthoven J, Blondia C. QBD Markov chains on binomial-like trees and its application to multilevel feedback queues. Ann Oper Res. 2008 Apr 29;160(1):3–18.
8. Thombare M, Sukhwani R, Shah P, Chaudhari S, Raundale P. Efficient implementation of Multilevel Feedback Queue Scheduling. In: 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). IEEE; 2016. p. 1950–4.
9. Creswell JW, Creswell JD. Research Design: Qualitative, Quantitative, and Mixed Methods Approaches. 5th ed. SAGE Publications, Inc; 2018.
10. Riadi S, Ulum F. ANALISIS PENERAPAN ALGORITMA FIRST COME FIRST SERVED (FCFS) DALAM PROSES PESANAN PADA APLIKASI GOJEK. J Inform dan Rekayasa Perangkat Lunak. 2021;2(2):268–75.
11. Verawati I, Sulistiyono M. PERANCANGAN SISTEM INFORMASI PENJADWALAN PROGRAM KERJA PENJAMINAN MUTU UNIVERSITAS AMIKOM DENGAN METODE MULTILEVEL FEEDBACK QUEUE. In 2018. Available from: https://api.semanticscholar.org/CorpusID:196157330
12. Brown J. On Intelligent Mitigation of Process Starvation In Multilevel Feedback Queue Scheduling. 2017 Dec;
13. Tjandra S, Chandra GS. Pemanfaatan Flutter dan Electron Framework pada Aplikasi Inventori dan Pengaturan Pengiriman Barang. J Inf Syst Hosp Technol. 2020 Dec 10;2(02):76–81.
14. Scholarworks@gvsu S, Latture AD. Backdrop: An Exploration of Flutter [Internet]. Available from: https://scholarworks.gvsu.edu/cistechlib
15. Malik R, Mann R, Knapman R. Rapid benchmarking: the case of a multinational dairy company. Benchmarking An Int J. 2021 Mar 29;28(3):1031–58.
16. Wahyudi H. Values Adoption of Benchmarking to Best Practice. In 2021.
17. Beretta S, Dossi A, Grove H. Methodological strategies for benchmarking accounting processes. Benchmarking Qual Manag Technol. 1998 Sep 1;5(3):165–83.
18. Jardosh AP, Ramachandran KN, Almeroth KC, Belding-Royer EM. Understanding Congestion in IEEE 802.11b Wireless Networks. In: Proceedings of the 5th ACM SIGCOMM Conference on Internet

e-ISSN 2527 – 8290
Vol. 8 No. 2, Desember 2023, Hal. 55 - 63

Measurement - IMC '05. 2005.
19. Andrian Ginting Student D, Dwi Endah O. Implementation of Multilevel Feedback Queue Algorithm in Restaurant Order Food Application Development for Android and iOS Platforms. Vol. 80, International Journal of Computer Applications. 2013.