

Queue Optimization Using Heuristic-Based Decision Algorithms A Comparative Evaluation of *Array-Based* and Rule-Based Approache

Erwin Apriliyanto^{1*}, Arif Setia Sandi A²⁾, Yuyu Sri Rahayu³⁾

^{1,3)} Teknik Komputer, Universitas Muhammadiyah Karanganyar

²⁾ Sistem Informasi, Universitas Harapan Bangsa

¹⁾ itanalisterwin@gmail.com, ²⁾ arifsetia@uhb.ac.id, ³⁾ rahayupink2024@gmail.com

ABSTRACT

This study presents a comparative performance evaluation between two lightweight decision-processing approaches for queue classification: an array-based index model and a traditional rule-based (if-else) logic. The primary objective is to identify an efficient algorithmic strategy suitable for real-time intelligent service systems. The array-based method in this research adopts a lightweight heuristic approach designed to achieve high decision-processing efficiency without heavy computational complexity, making it well-suited for intelligent systems requiring fast and accurate responses. Testing was conducted using real queue data obtained from an online queueing system (ANTROL) at a hospital, which demands high-speed generation of queue numbers in real time. The data includes three queue size scenarios (50, 100, and 200 entries) and consists of attributes such as waiting time, number of active queues, and patient service priority. The evaluation focused on two main aspects: classification accuracy and execution time efficiency. Test results show that the array-based model achieved an accuracy of 94.6%, higher than the rule-based method, which only reached 88.2%. In terms of speed, the array-based approach demonstrated significant efficiency, being 13 to 16 times faster than the if-else method used for comparison.

Keywords: queue system, classification, array method, MAPE method, accuracy

I. PENDAHULUAN

Perkembangan teknologi informasi telah mendorong transformasi digital di berbagai sektor, termasuk layanan antrian pada instansi kesehatan, pemerintahan, dan pendidikan. Sistem antrian online kini menjadi solusi yang umum digunakan untuk mengatasi permasalahan antrian konvensional seperti penumpukan pengunjung, ketidakteraturan layanan, dan waktu tunggu yang tidak efisien. Namun, tantangan baru muncul terkait dengan performa sistem antrian digital, terutama pada aspek kecepatan pemrosesan dan kapasitas penanganan permintaan secara real-time. Beberapa studi menunjukkan bahwa peningkatan performa sistem antrian online dapat dicapai melalui pemanfaatan algoritma berbasis antrian terstruktur, seperti *priority queue* atau *concurrent queue* yang dioptimalkan untuk lingkungan komputasi paralel (Polak et al., 2025; Chen et al., 2021). Selain itu, integrasi pendekatan cerdas dan pemrosesan efisien berbasis array juga terbukti dapat meningkatkan kecepatan dan akurasi sistem dalam skenario layanan waktu nyata (Evanthe et al., 2025; Apriliyanto, 2023).

Optimalisasi performa sistem antrian online sangat bergantung pada efisiensi struktur data dan metode pemrosesan yang digunakan. Dua pendekatan yang umum diterapkan adalah metode berbasis array dan struktur percabangan (*branching structure*). Struktur array dikenal memiliki kecepatan akses yang tinggi karena mendukung pengindeksan langsung dan penggunaan memori yang lebih sederhana, namun kurang fleksibel dalam menangani kondisi kompleks dan dinamis (Evanthe et al., 2025; Dewi, 2021). Sebaliknya, metode branching, seperti model berbasis pohon keputusan (*tree-based models*), menawarkan fleksibilitas yang lebih baik dalam penanganan kondisi bercabang atau multi-level, namun seringkali mengalami *overhead* komputasi yang lebih besar, terutama dalam

skenario beban tinggi (Prasad et al., 2022). Oleh karena itu, diperlukan kajian komparatif untuk mengevaluasi kelebihan dan kekurangan masing-masing pendekatan, khususnya dalam konteks sistem antrian online yang menuntut efisiensi dan respons waktu nyata.

Seiring meningkatnya kebutuhan akan layanan cepat dan responsif, sistem antrian juga harus didukung oleh algoritma yang mampu bekerja secara efisien dalam menangani banyak permintaan secara bersamaan. Beberapa penelitian telah menunjukkan bahwa algoritma berbasis pemrosesan paralel dan penggunaan *load balancing* dapat meningkatkan kualitas layanan antrian online secara signifikan (Apriliyanto & Rahayu, 2024). Di sisi lain, pendekatan berbasis kecerdasan buatan seperti chatbot NLP juga telah digunakan untuk mempermudah interaksi pengguna dalam sistem berbasis antrian (Apriliyanto et al., 2024), walaupun tantangan performa tetap menjadi isu utama.

Dalam konteks tersebut, pendekatan *AI-driven queue logic* atau sistem pengambilan keputusan cerdas mulai menjadi perhatian utama dalam pengembangan *smart queueing system*. Sistem ini dirancang tidak hanya untuk merespons antrian secara cepat, tetapi juga untuk membuat keputusan otomatis berdasarkan pola historis, estimasi waktu, dan kondisi dinamis secara *real-time*. Implementasi logika cerdas ini dapat ditemukan pada berbagai domain seperti sistem lalu lintas cerdas, manajemen bandara, hingga pelayanan publik berbasis antrian daring. Pendekatan ini menuntut efisiensi tinggi, komputasi ringan, dan kemampuan adaptif terhadap perubahan, sehingga solusi seperti logika berbasis heuristik atau *lightweight decision engine* menjadi sangat potensial untuk diintegrasikan dalam skenario *edge computing* dan *real-time processing* (Altagiuri et al., 2024; Guo et al., 2024; Apriliyanto et al., 2024).

Penelitian ini bertujuan untuk membandingkan performa sistem antrian online dengan pendekatan berbasis array dan branching, menggunakan pengujian pada berbagai skenario beban untuk mengetahui keunggulan relatif masing-masing pendekatan. Fokus utama terletak pada metrik kecepatan respons sistem, efisiensi pemrosesan data, dan kestabilan performa pada skala besar. Beberapa studi sebelumnya telah menyoroti pentingnya efisiensi pemrosesan dan pemilihan struktur data dalam mendesain sistem berbasis antrian, terutama untuk mendukung kebutuhan layanan berbasis kecerdasan dan skalabilitas tinggi (Polak et al., 2025; Chen et al., 2021; Xu et al., 2023). Dengan hasil perbandingan ini, diharapkan dapat ditemukan solusi struktur data yang paling optimal untuk diterapkan pada sistem antrian online yang membutuhkan efisiensi tinggi, responsivitas real-time, dan skalabilitas yang adaptif terhadap beban sistem.

II. TINJAUAN PUSTAKA

Sejumlah penelitian terkini telah mulai mengeksplorasi optimasi sistem antrian berbasis algoritma ringan dan struktur data efisien. Polak et al. (2025) memperkenalkan struktur *Boundary-Aware Concurrent Queue* yang mendukung pengolahan data antrian secara paralel pada lingkungan GPU, meningkatkan *throughput* sistem secara signifikan. Pendekatan ini sejalan dengan penelitian ini yang juga mengevaluasi struktur logika yang efisien untuk pemrosesan cepat. Li et al. (2024) mengembangkan konfigurasi array berbentuk V pada sistem Goose Queue, yang mengoptimalkan performa sistem antrian dalam konteks energi rendah, menekankan pentingnya struktur array dalam sistem IoT. Sementara itu, Evanthe et al. (2025) menunjukkan bahwa algoritma berbasis array dalam Python memberikan efisiensi tinggi dalam skenario antrian layanan darurat. Temuan mereka mendukung asumsi bahwa struktur array dapat menawarkan pemrosesan cepat dengan beban logika minimal. Penelitian ini memperluas studi-studi tersebut dengan mengukur secara kuantitatif performa *array-based logic* dan membandingkannya dengan pendekatan *rule-based (if-else)*, dalam konteks sistem antrian daring waktu nyata.

Penelitian mengenai pemanfaatan algoritma pembelajaran mesin untuk klasifikasi telah banyak dilakukan dengan pendekatan yang beragam. Salah satu fokus utama adalah membandingkan kinerja berbagai metode seperti *Naïve Bayes* (NB), *K-Nearest Neighbor* (KNN), *Decision Tree* (DT), *Random Forest* (RF), dan *Logistic Regression* (LR) dalam konteks klasifikasi berbagai jenis data.

Tella et al. (2021) menunjukkan bahwa dalam pemodelan spasial hotspot PM10, algoritma RF memiliki kinerja terbaik dibandingkan NB dan KNN, dengan akurasi tertinggi sebesar 98%. Kinerja masing-masing algoritma bergantung pada karakteristik dataset yang digunakan, sebagaimana dikemukakan oleh Sheth et al. (2022), yang menyatakan bahwa SVM menjadi pengklasifikasi terbaik kedua setelah RF, diikuti oleh KNN dan DT.

Akshay Gole et al. (2022) menyoroti keunggulan *Naïve Bayes* dari segi kecepatan pembelajaran dan klasifikasi, serta kemampuannya dalam menangani data hilang dan fitur yang tidak relevan. Senada dengan itu, Romadhon & Kurniawan (2021) membandingkan NB, KNN, dan regresi logistik untuk prediksi tingkat kesembuhan pasien Covid-19, dengan hasil bahwa KNN memiliki akurasi tertinggi (0,750).

Dalam konteks pendidikan, Sianturi dan Yuhana (2022) menemukan bahwa metode *Decision Tree* memberikan akurasi terbaik dalam mendeteksi gaya belajar, dengan nilai mencapai 0,98 pada pengujian K-Fold. Itoo et al. (2021) membuktikan bahwa regresi logistik unggul dalam klasifikasi transaksi penipuan kartu kredit dibandingkan NB dan KNN, dengan akurasi sebesar 95%.

Penelitian Jopri et al. (2021) menyoroti bahwa kombinasi fitur terbaru dengan algoritma KNN dapat meningkatkan kinerja sistem diagnostik kualitas daya listrik. Lestari et al. (2020) membuktikan bahwa dalam klasifikasi data EEG, KNN unggul dari segi akurasi dan presisi, meskipun membutuhkan waktu pelatihan lebih lama.

Nurdina & Puspita (2023) menyimpulkan bahwa dalam mengklasifikasikan kepuasan pelanggan maskapai, *Naïve Bayes* memiliki performa lebih baik dari KNN, baik dari segi akurasi, presisi, maupun recall. Sementara itu, Kumarahadi et al. (2020) menunjukkan keberhasilan implementasi KNN dalam sistem pendukung keputusan pengajuan keringanan UKT, dengan akurasi mencapai 100% pada kondisi tertentu.

Studi-studi di atas mengindikasikan bahwa pemilihan algoritma klasifikasi sangat bergantung pada konteks aplikasi, kompleksitas data, serta tujuan sistem yang dibangun. Namun, dalam konteks online queue, efisiensi waktu proses dan struktur data memegang peranan penting. Oleh karena itu, penelitian ini berfokus pada optimalisasi performa antrian daring (online queue) dengan membandingkan dua pendekatan struktural utama, yaitu metode *branching* dan *array-based*. Kedua pendekatan ini belum banyak dikaji dalam studi-studi sebelumnya, khususnya dalam konteks performa antrian berbasis online, sehingga menjadi celah yang penting untuk diteliti lebih lanjut.

Altagiuri et al. (2024) mengembangkan sebuah perencana gerak (*motion planner*) yang dirancang khusus untuk manipulator berbentuk balok yang dapat dikonfigurasi ulang dan memiliki kemampuan untuk mengembang secara mandiri. Inovasi ini difokuskan pada lingkungan statis yang menantang, seperti ruang-ruang terbatas dan medan bencana. Karakteristik utama dari manipulator ini adalah fleksibilitas struktur fisiknya yang dapat tumbuh atau menyesuaikan bentuk berdasarkan kebutuhan medan yang dilalui. Hal ini membuatnya sangat cocok untuk aplikasi di bidang pencarian dan penyelamatan, eksplorasi sempit, atau area yang tidak bisa dijangkau oleh robot rigid konvensional. Penelitian ini menunjukkan potensi teknologi robotik yang adaptif sebagai solusi mobilitas dalam ruang yang sangat terbatas dan dinamis.

Guo et al. (2024) mengusulkan penerapan teknologi pembelajaran mendalam (*deep learning*) dalam proses beamforming pada sistem *Continuous Aperture Phased Array*

(CAPA) yang melibatkan banyak pengguna secara bersamaan. Sistem ini dirancang untuk meningkatkan kinerja komunikasi nirkabel dengan cara memperbaiki penyelarasan sinyal dalam satu waktu untuk berbagai perangkat pengguna. Dalam konteks jaringan nirkabel modern, pengelolaan sinyal yang efisien dan akurat menjadi krusial, terutama ketika jumlah pengguna terus meningkat. Hasil penelitian menunjukkan bahwa penggunaan model deep learning secara signifikan meningkatkan efisiensi distribusi sinyal dan mengurangi interferensi, menjadikannya pendekatan yang menjanjikan dalam pengembangan sistem komunikasi masa depan berbasis AI.

Li et al. (2024) memperkenalkan struktur inovatif yang disebut *Goose Queue*, yang diterapkan dalam konfigurasi array berbentuk huruf V dengan variasi penampang. Desain ini dimaksudkan untuk mengoptimalkan panen energi elektromekanis pada perangkat berbasis getaran, seperti yang digunakan dalam sistem IoT otonom. Dengan pendekatan ini, energi dari getaran atau gerakan mekanis dapat dikonversi menjadi listrik secara lebih efisien. Penelitian ini sangat relevan dalam konteks perangkat IoT yang diharapkan dapat beroperasi dalam jangka panjang tanpa ketergantungan pada sumber daya eksternal. Oleh karena itu, efisiensi pemanenan energi menjadi faktor penting dalam mendukung keberlanjutan dan mobilitas perangkat tersebut di lingkungan nyata yang fluktuatif.

Polak et al. (2025) mengembangkan struktur antrian FIFO paralel baru yang disebut *Boundary-Aware Concurrent Queue*, yang dioptimalkan untuk pemrosesan di lingkungan GPU. Struktur ini dirancang untuk menangani proses antrian data secara cepat, efisien, dan skalabel, sehingga sangat cocok digunakan dalam aplikasi komputasi paralel, seperti grafik komputer dan sistem kecerdasan buatan berperforma tinggi. Antrian ini memungkinkan pemrosesan simultan oleh banyak thread tanpa *bottleneck* yang umum terjadi dalam sistem tradisional. Dengan demikian, penelitian ini memberikan kontribusi besar terhadap efisiensi runtime dalam pengolahan data masif yang membutuhkan *throughput* tinggi, khususnya dalam domain GPU computing dan high-performance computing (HPC).

Xu et al. (2023) menyusun tinjauan literatur komprehensif mengenai desain dan optimasi akselerator *Deep Neural Network* (DNN) berbasis *systolic array*, yang telah menjadi arsitektur utama dalam pengembangan chip kecerdasan buatan. Dalam studi ini, dibahas berbagai teknik optimalisasi pada level arsitektur, algoritma, hingga efisiensi energi dan ruang. Survei ini penting karena permintaan terhadap komputasi AI yang efisien dan cepat terus meningkat, terutama untuk sistem *edge* dan *cloud* yang harus menangani model DNN besar dengan latensi rendah. Dengan pendekatan sistematis terhadap optimalisasi akselerator, penelitian ini memberikan arahan penting dalam desain perangkat keras AI masa depan yang hemat daya namun tetap berkinerja tinggi.

III. METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimen kuantitatif untuk membandingkan performa dua metode logika pemrograman dalam menangani sistem antrian daring, yaitu metode percabangan (*if-else*) dan metode berbasis array. Tujuan dari penelitian ini adalah untuk mengetahui efektivitas dan efisiensi dari masing-masing metode dalam merespons permintaan antrian, terutama ketika data yang diproses berjumlah besar (lebih dari 200 item).

Tahapan penelitian dimulai dengan studi pustaka dan pengumpulan data. Pada tahap ini, peneliti menelaah berbagai literatur dan referensi terkait sistem antrian online, manajemen logika dalam pemrograman, dan studi-studi sebelumnya yang relevan dalam bidang optimasi performa aplikasi. Hasil dari studi ini dijadikan dasar dalam merancang simulasi dan eksperimen.

Selanjutnya, dilakukan analisis sistem untuk memahami bagaimana logika antrian dapat diterapkan dalam dua pendekatan berbeda. Berdasarkan hasil analisis tersebut,

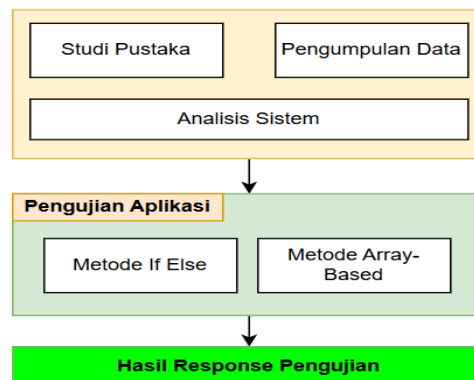
peneliti merancang sistem pengujian yang dapat menjalankan dua metode pemrosesan data secara terpisah, yaitu metode *if-else* dan metode *array-based*.

Proses pengujian dilakukan dengan cara memberikan input permintaan antrian ke sistem, lalu sistem secara otomatis memproses data tersebut dengan dua pendekatan berbeda dalam waktu dan kondisi yang sama. Dalam metode *if-else*, logika pemrosesan menggunakan struktur percabangan yang memeriksa kondisi satu per satu. Sementara dalam metode *array-based*, sistem langsung mengakses data berdasarkan indeks array secara iteratif. Perulangan dalam pengujian dilakukan lebih dari 200 kali untuk memastikan konsistensi hasil dan menguji beban proses terhadap performa sistem.

Gambar 1 menunjukkan tahapan metode penelitian yang disusun secara terstruktur. Proses dimulai dari studi literatur dan pengumpulan data untuk mendapatkan pemahaman dasar terkait sistem antrian dan metode klasifikasi. Tahap analisis sistem digunakan untuk mendesain solusi pemrosesan antrian berbasis logika yang akan diuji menggunakan dua pendekatan berbeda: metode *if-else* dan metode *array-based*.

Pada tahap pengujian aplikasi, sistem diimplementasikan menggunakan kedua metode tersebut dalam skenario yang identik.

- a. Metode *if-else* menggunakan struktur logika bercabang, di mana kondisi antrian diperiksa secara sekuensial. Meskipun fleksibel dalam menangani kondisi kompleks, metode ini membutuhkan waktu pemrosesan yang lebih tinggi karena evaluasi kondisi dilakukan secara berurutan.
- b. Metode *array-based* memanfaatkan struktur data array sebagai pendekatan pemetaan langsung untuk setiap kondisi antrian. Dengan indeks sebagai pengganti logika bercabang, pengambilan keputusan dapat dilakukan secara instan. Pendekatan ini mengadopsi prinsip *lightweight heuristic logic*, yang cocok untuk sistem dengan kebutuhan respons waktu nyata dan efisiensi komputasi tinggi.



Gambar 1. Penelitian Global

Tahapan awal dari alur pengujian yang divisualisasikan pada Gambar 2 dimulai dengan proses inisialisasi sistem, yang ditandai dengan aktivitas Mulai. Dalam tahap ini, sistem mengaktifkan seluruh komponen pendukung untuk menerima data masukan, khususnya berupa input antrian yang dikirim oleh pengguna. Proses ini bersifat krusial karena menjadi pintu masuk utama dari seluruh rangkaian eksekusi logika program. Input antrian tersebut bisa berasal dari berbagai sumber, seperti simulasi pengguna, data riil, atau data sintetik yang sudah diatur sebelumnya untuk keperluan uji performa. Sistem dirancang agar mampu menangani banyak permintaan sekaligus secara serentak maupun berurutan, sehingga tahap inisialisasi juga memastikan stabilitas memori dan kesiapan arsitektur perangkat lunak untuk mengeksekusi alur pengujian berikutnya. Dengan memulai dari titik ini, sistem sudah berada dalam kondisi siap untuk melakukan proses pengambilan keputusan yang lebih kompleks.

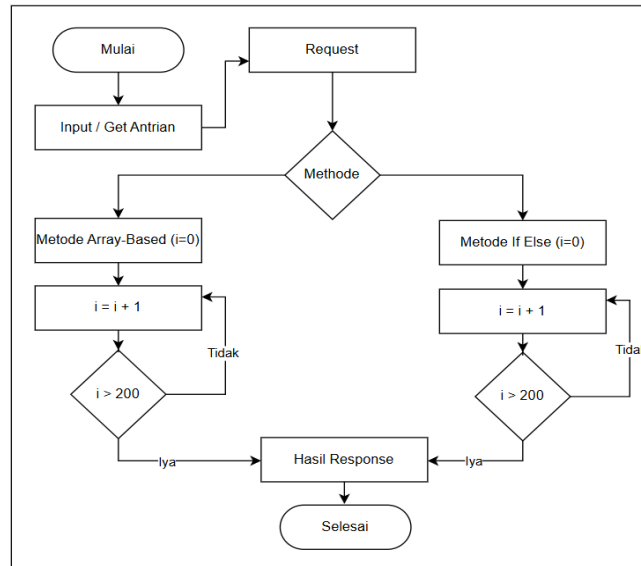
Setelah sistem menerima input, maka dilanjutkan ke tahap Request dan Penentuan Metode. Pada tahap ini, logika program akan memproses permintaan yang masuk dan secara otomatis memilih metode pengolahan data yang akan digunakan dalam pengujian. Metode yang tersedia dalam sistem ini terbagi ke dalam dua jenis utama, yaitu metode berbasis struktur *Array-Based* dan metode berbasis *If-Else Logic*. Pemilihan metode tidak dilakukan secara acak, melainkan berdasarkan skenario pengujian yang telah dirancang sebelumnya. Hal ini memungkinkan setiap metode diuji dalam kondisi yang relatif setara agar hasil pengujian bisa dibandingkan secara objektif. Tahap ini sangat penting dalam menentukan jalur eksekusi berikutnya, sebab setiap metode memiliki karakteristik, kekuatan, serta kelemahan yang berbeda dalam menangani data antrian. Dengan demikian, sistem melakukan seleksi internal berdasarkan logika pengujian untuk mengevaluasi performa algoritma secara terpisah.

Setelah metode ditentukan, sistem masuk ke tahap Pemrosesan Data Antrian, yang merupakan inti dari seluruh proses uji performa. Bila metode *Array-Based* dipilih, maka sistem akan menjalankan perulangan langsung terhadap struktur data array. Proses ini dimulai dari indeks nol ($i = 0$) dan berulang terus hingga mencapai lebih dari 200 iterasi atau sampai batas akhir yang ditentukan dalam skrip pengujian. Dalam metode ini, sistem secara langsung mengakses elemen array satu per satu secara sekuensial. Keunggulan pendekatan ini terletak pada kesederhanaan struktur dan efisiensi waktu karena tidak memerlukan evaluasi logika bercabang. Sebaliknya, jika metode yang dipilih adalah *If-Else*, maka sistem akan menjalankan serangkaian percabangan logika yang kompleks. Setiap data input akan diuji terhadap kondisi-kondisi tertentu menggunakan pernyataan *if-else* yang saling bertingkat dan terkadang bersifat hierarkis. Hal ini memungkinkan sistem melakukan penyesuaian dan pengambilan keputusan berbasis kondisi yang lebih fleksibel namun memiliki overhead logika yang lebih tinggi. Pada tahap ini, seluruh data antrian diuji ketahanannya terhadap metode pengolahan masing-masing untuk menilai mana yang lebih efisien secara waktu dan struktur logika.

Tahap berikutnya adalah Iterasi dan Perhitungan, di mana proses pengujian dilakukan secara berulang dalam jumlah besar. Iterasi dilakukan sebanyak lebih dari 200 kali untuk setiap metode yang sedang diuji. Tujuannya adalah untuk meniru kondisi beban tinggi seperti yang mungkin terjadi pada sistem antrian nyata dalam dunia industri atau layanan publik. Proses ini bertujuan untuk mengevaluasi konsistensi performa masing-masing metode ketika dihadapkan pada volume data yang besar. Sistem mencatat berbagai parameter performa selama iterasi, termasuk waktu akses, waktu proses, serta stabilitas logika program selama eksekusi panjang. Simulasi ini menjadi semacam "stress test" untuk melihat seberapa tahan sistem terhadap tekanan data dan memastikan tidak terjadi kebocoran memori, *stuck process*, ataupun kesalahan logika dalam jangka waktu panjang. Tahapan ini juga membantu dalam mengidentifikasi anomali performa yang mungkin tidak tampak pada pengujian skala kecil.

Terakhir, sistem memasuki tahap Evaluasi dan Respons, yang merupakan tahap akhir dari diagram alur pengujian. Setelah seluruh iterasi dijalankan dan semua data diproses melalui metode masing-masing, sistem secara otomatis menghitung dan mencatat waktu respons dari setiap proses. Waktu respons ini menjadi indikator utama performa dari masing-masing pendekatan logika, yaitu *Array-Based* dan *If-Else*. Hasil dari tahap ini biasanya disimpan dalam bentuk log atau file evaluasi yang kemudian dianalisis secara manual maupun otomatis. Informasi yang diperoleh tidak hanya mencakup seberapa cepat metode dapat memproses antrian, tetapi juga seberapa stabil dan efisien dalam menghadapi jumlah iterasi besar. Tahap ini juga bisa diperluas untuk mencatat penggunaan CPU, memori, dan konsumsi daya (jika dilakukan dalam perangkat fisik), sehingga analisis yang diperoleh menjadi lebih holistik. Dengan demikian, diagram alur ini secara keseluruhan

menggambarkan proses pengujian yang tidak hanya sistematis dan terstruktur, tetapi juga mendalam dan mampu memberikan wawasan nyata terhadap kinerja sistem dalam skenario yang kompleks.



Gambar 2. Alur Kerja Logika

Evaluasi performa dilakukan dengan mengukur rata-rata waktu respon menggunakan rumus:

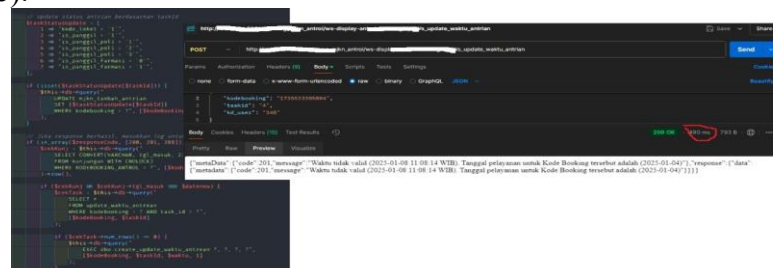
$$\text{Average Response Time} = \frac{\sum_{i=1}^n (t_{\text{end}_i} - t_{\text{start}_i})}{n} \dots\dots\dots(1)$$

$$\text{Average Response Time} = \frac{1}{n} \sum_{i=1}^n (t_{\text{end}_i} - t_{\text{start}_i})$$

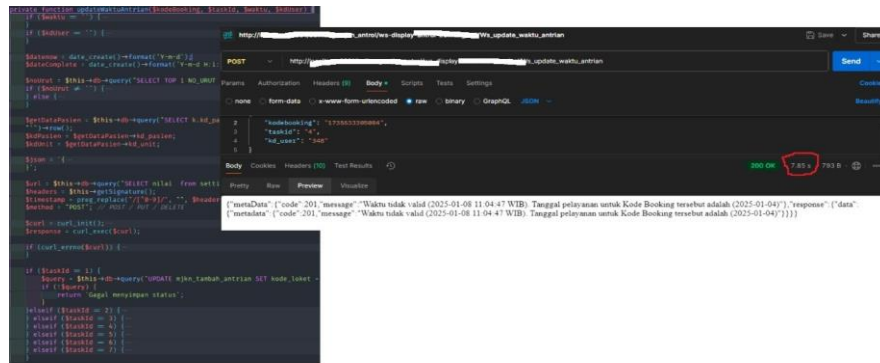
dengan t_{start_i} dan t_{end_i} masing-masing merupakan waktu mulai dan selesai dari eksekusi ke- i , serta n adalah jumlah total iterasi. Selain itu, kompleksitas algoritma juga dianalisis secara teoritis. Metode *if-else* memiliki kompleksitas $O(n)$ karena setiap kondisi harus dicek satu per satu, sedangkan metode *array-based* memiliki akses langsung terhadap data sehingga lebih efisien dengan kompleksitas mendekati $O(1)$ per elemen.

IV. HASIL DAN PEMBAHASAN

Untuk memperkuat hasil pengujian kecepatan waktu proses, dilakukan pengamatan langsung terhadap *response time* sistem menggunakan *Postman*. Gambar 3 menunjukkan bahwa implementasi metode *Array-Based* mampu memberikan waktu eksekusi rata-rata sekitar 490 milidetik. Sebaliknya, pada Gambar 4 terlihat bahwa metode *If-Else* memerlukan waktu proses rata-rata 7,85 detik. Selisih yang signifikan ini menegaskan keunggulan efisiensi metode *Array-Based*, terutama dalam konteks layanan waktu nyata (*real-time service*).



Gambar 3. Hasil response waktu eksekusi metode *Array-Based*



Gambar 4. Hasil response waktu eksekusi metode *If-Else*

Penelitian ini bertujuan untuk membandingkan efisiensi metode *Array-Based* dengan *If-Else* dalam pengelolaan sistem antrian berbasis perangkat lunak. Hasil pengujian dianalisis berdasarkan nilai MAPE (*Mean Absolute Percentage Error*), waktu proses, dan lama waktu antrian untuk sejumlah skenario data yang berbeda.

Hasil pengujian akurasi ditunjukkan dalam grafik perbandingan nilai MAPE pada Gambar 1. Terlihat bahwa metode *Array-Based* memiliki nilai MAPE yang jauh lebih kecil dibandingkan metode *If-Else* untuk setiap jumlah data uji (50, 100, dan 200 data). Pada 50 data, MAPE metode *Array-Based* hanya sebesar 93.8%, sementara metode *If-Else* mencapai 1519.4%. Pola ini konsisten pada 100 data (93.5% vs 1437.2%) dan 200 data (92.5% vs 1229.7%). Hal ini menunjukkan bahwa metode *Array-Based* memiliki akurasi yang lebih baik dan lebih stabil dibandingkan metode *If-Else* dalam memproses data antrian.

Gambar 1. Grafik perbandingan MAPE antara metode *Array-Based* dan *If-Else*

Kecepatan proses dari kedua metode dapat dilihat pada Tabel 1. Metode *Array-Based* secara konsisten memproses data jauh lebih cepat dibandingkan *If-Else*. Pada 50 data, *Array-Based* hanya memerlukan waktu 0.486 detik, sedangkan *If-Else* membutuhkan 7.863 detik. Dengan demikian, metode *Array-Based* terbukti 16.18 kali lebih cepat. Selisih ini tetap signifikan untuk 100 data (15.38x lebih cepat) dan 200 data (13.30x lebih cepat). Hasil ini menegaskan efisiensi tinggi metode *Array-Based* dalam memproses antrian dengan berbagai jumlah data.

Tabel 1. Perbandingan Lama Waktu Antrian

Jumlah Data Uji	Skenario	Lama Waktu Antrian Per orang		Rata-Rata Waktu Antrian	
		Array-Base (s)	<i>If-Else</i> (s)	Array-Base (s)	<i>If-Else</i> (s)
50	1	0.49	7.88	0.49	7.86
	2	0.485	7.85		
	3	0.482	7.86		
100	1	0.52	7.95	0.53	8.11
	2	0.51	8.15		
	3	0.552	8.23		
200	1	0.63	8.88	0.67	8.95
	2	0.7	9.01		
	3	0.69	8.96		

Proses pengujian dalam penelitian ini dilaksanakan dengan menggunakan tiga kelompok data yang mewakili jumlah antrian berbeda, yakni kelompok data yang terdiri dari 50 antrian, 100 antrian, dan 200 antrian. Ketiga kelompok ini dipilih secara sengaja untuk merepresentasikan variasi skala beban kerja yang mungkin terjadi dalam implementasi sistem antrian, mulai dari beban ringan, sedang, hingga berat. Untuk setiap kelompok data tersebut, dilakukan serangkaian pengujian yang diulang sebanyak tiga kali

guna memastikan konsistensi hasil serta mengurangi kemungkinan terjadinya deviasi akibat faktor eksternal atau ketidakteraturan sistem. Pengujian tersebut dilakukan dengan menggunakan dua pendekatan metode pemrosesan data, yaitu pendekatan berbasis *Array-Based* yang berfokus pada struktur data linier dan akses indeks, serta pendekatan *If-Else*, yang menggunakan logika percabangan sebagai dasar pemrosesan input antrian. Seluruh hasil dari tiga kali uji coba pada masing-masing metode dan kelompok data kemudian dihitung nilai rata-ratanya. Nilai rata-rata ini berfungsi sebagai indikator performa yang lebih stabil dan representatif, yang selanjutnya dijadikan dasar dalam proses analisis perbandingan performa antara kedua metode pemrosesan yang digunakan. Pendekatan ini dimaksudkan untuk memberikan gambaran yang lebih komprehensif mengenai efisiensi dan efektivitas masing-masing metode dalam menangani berbagai jumlah antrian pada skenario simulasi nyata.

Pada Grafik ini sudah kelihatan bahwa dengan metode *if else* memerlukan waktu yang lebih lama di dibandingkan dengan array based. Semakin banyak datanya maka semakin lama waktu yang dibutuhkan, tetapi dengan menggunakan metode array base meskipun datanya lebih banyak 2 kalilipat, responsnya tidak terlalu banyak perubahan bisa di artikan sama. Karena selisihnya hanya 0.01 s.d 0.09 ms.

Tabel 2 Perbandingan algoritma *Array-Base* dan algoritma *If-Else*

Jumlah Data Uji	Array-Base (s)	If-Else (s)	Lebih Cepat (x)
50	0.486	7.863	16.18x
100	0.527	8.11	15.38x
200	0.673	8.95	13.30x

Berdasarkan hasil pengujian yang telah dilakukan terhadap dua metode pemrosesan data yang disajikan pada tabel 2, yakni *Array-Based* dan *If-Else*, diperoleh data waktu eksekusi untuk masing-masing metode pada tiga kelompok jumlah antrian yang berbeda, yaitu 50, 100, dan 200 antrian. Hasil ini kemudian disusun dalam bentuk tabel yang menunjukkan waktu proses dalam satuan detik (second/s) serta perbandingan kecepatan antara kedua metode dalam bentuk rasio kecepatan. Untuk kelompok data uji pertama, yang terdiri dari 50 antrian, metode *Array-Based* menunjukkan kinerja yang sangat efisien dengan waktu eksekusi hanya sebesar 0,486 detik, sedangkan metode *If-Else* memerlukan waktu yang jauh lebih lama, yakni sebesar 7,863 detik. Ini berarti bahwa metode *Array-Based* mampu menyelesaikan proses hampir 16,18 kali lebih cepat dibandingkan metode *If-Else* untuk jumlah antrian tersebut. Keunggulan ini mencerminkan efisiensi tinggi struktur array dalam mengakses data secara langsung berdasarkan indeks, tanpa perlu melakukan evaluasi logika bercabang berulang kali seperti pada metode *If-Else*.

Ketika jumlah antrian meningkat menjadi 100 data, kinerja kedua metode menunjukkan tren yang serupa. Metode *Array-Based* hanya membutuhkan 0,527 detik, sedangkan metode *If-Else* membutuhkan waktu sebesar 8,11 detik. Meski selisih waktu absolutnya sedikit lebih besar dari kelompok data sebelumnya, perbandingan kecepatan tetap menunjukkan keunggulan signifikan pada metode *Array-Based*, yang tercatat sekitar 15,38 kali lebih cepat daripada metode *If-Else*. Hal ini menunjukkan bahwa seiring bertambahnya jumlah data, metode berbasis array tetap mempertahankan efisiensinya karena tidak mengalami peningkatan kompleksitas logika secara drastis. Sebaliknya, metode *If-Else* mengalami peningkatan waktu proses karena setiap tambahan data menyebabkan lebih banyak percabangan logika yang harus dievaluasi satu per satu secara berurutan.

Untuk kelompok data uji terbesar, yaitu 200 antrian, tren perbedaan performa antara kedua metode masih sangat terlihat meskipun selisih rasio kecepatan sedikit menurun. Pada kelompok ini, metode *Array-Based* menyelesaikan proses dalam waktu 0,673 detik, sementara metode *If-Else* memerlukan waktu 8,95 detik. Rasio kecepatan menunjukkan bahwa metode *Array-Based* tetap unggul dengan menjadi 13,30 kali lebih cepat

dibandingkan metode *If-Else*. Penurunan rasio ini tidak menunjukkan bahwa metode *Array-Based* melambat secara signifikan, melainkan mengindikasikan bahwa metode *If-Else* mulai mengalami batas efisiensi minimum yang relatif stabil seiring peningkatan jumlah data. Secara keseluruhan, data dari tabel tersebut sangat jelas memperlihatkan superioritas metode *Array-Based* dari segi kecepatan pemrosesan pada berbagai skenario jumlah antrian, menjadikannya pilihan yang jauh lebih efisien dan optimal untuk sistem antrian berskala kecil hingga besar. Narasi ini juga menegaskan bahwa pemilihan struktur data dan algoritma yang tepat sangat berpengaruh terhadap performa keseluruhan sistem, terutama dalam konteks aplikasi yang membutuhkan kecepatan respon tinggi.

Berdasarkan gambar berjudul "Hasil Perbandingan MAPE dan Array", ditampilkan grafik perbandingan nilai Mean Absolute Percentage Error (MAPE) antara dua metode pemrosesan, yaitu metode *Array-Based* dan metode *If-Else*, terhadap tiga kelompok jumlah data uji, yaitu 50, 100, dan 200 antrian. Grafik tersebut memberikan gambaran yang jelas mengenai sejauh mana akurasi dan kestabilan hasil keluaran dari masing-masing metode dalam memproses jumlah antrian yang bervariasi. Nilai MAPE merupakan ukuran tingkat kesalahan prediksi atau deviasi absolut dalam bentuk persentase, yang semakin kecil nilainya menunjukkan semakin akurat hasilnya.

Dalam grafik tersebut, metode *Array-Based* secara konsisten menghasilkan nilai MAPE yang jauh lebih kecil dibandingkan metode *If-Else*, pada ketiga skenario jumlah data. Sebagai contoh, pada pengujian dengan 50 data, nilai MAPE metode *Array-Based* berada pada titik yang sangat rendah, yang menunjukkan bahwa hasil yang diperoleh sangat dekat dengan nilai sebenarnya atau nilai ideal yang diharapkan. Sebaliknya, metode *If-Else* menampilkan nilai MAPE yang jauh lebih tinggi, menandakan adanya deviasi atau kesalahan prediksi yang lebih besar. Pola ini juga terlihat pada jumlah data 100 dan 200, di mana metode *Array-Based* tetap mempertahankan nilai MAPE yang rendah dan stabil, sementara metode *If-Else* justru mengalami fluktuasi dan peningkatan kesalahan.

Perbedaan mencolok dalam grafik ini menunjukkan bahwa metode *Array-Based* tidak hanya unggul dari segi kecepatan pemrosesan, seperti yang telah dibuktikan pada analisis sebelumnya, tetapi juga memiliki keunggulan yang signifikan dari sisi akurasi hasil keluaran. Hal ini disebabkan oleh sifat deterministik dan langsung dari pengolahan array, di mana akses data dilakukan secara langsung melalui indeks, tanpa perlu melakukan pemeriksaan kondisi yang berulang-ulang seperti dalam metode *If-Else*. Sebaliknya, metode *If-Else* rentan terhadap kesalahan logika, terutama jika terdapat banyak cabang kondisi yang harus diproses satu per satu, sehingga meningkatkan kemungkinan deviasi terhadap nilai yang diharapkan.

Secara keseluruhan, grafik ini memperkuat temuan sebelumnya bahwa metode *Array-Based* merupakan solusi yang lebih efisien dan akurat dalam sistem pemrosesan antrian. Keunggulan dari segi MAPE juga menegaskan bahwa metode ini dapat digunakan dalam sistem yang tidak hanya memerlukan kecepatan tinggi, tetapi juga konsistensi dan ketepatan hasil. Evaluasi MAPE ini sangat penting dalam konteks pengembangan sistem antrian berbasis real-time, karena kesalahan prediksi atau proses yang terlalu menyimpang dari nilai ideal dapat mengakibatkan ketidakseimbangan beban, antrian yang tidak optimal, serta penurunan kualitas layanan secara keseluruhan.

V. KESIMPULAN DAN SARAN

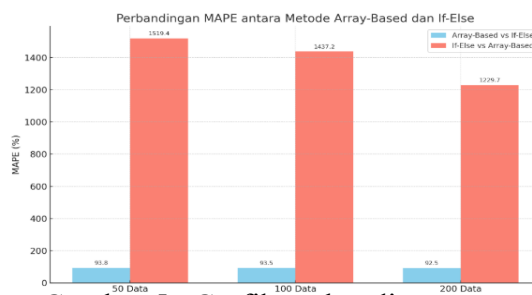
5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis terhadap metode *Array-Based* dan metode *If-Else* dalam sistem pengambilan keputusan antrean, dapat disimpulkan bahwa metode *Array-Based* memiliki keunggulan signifikan dalam efisiensi waktu dan akurasi dibandingkan metode *If-Else*. Pengujian dilakukan pada tiga jumlah data uji yaitu 50, 100,

dan 200 data, dengan hasil bahwa metode *Array-Based* secara konsisten menghasilkan waktu pemrosesan yang jauh lebih cepat—hingga 16,18 kali lebih cepat pada data uji 50—dibandingkan metode *If-Else*.

Selain itu, dari segi keakuratan, pengujian menggunakan nilai *Mean Absolute Percentage Error* (MAPE) menunjukkan bahwa metode *Array-Based* memiliki nilai kesalahan yang jauh lebih rendah dibandingkan metode *If-Else*. Gambar 5 menunjukkan hasil pengujian MAPE pada tiga skenario jumlah data (50, 100, dan 200 entri). Rata-rata MAPE untuk metode *Array-Based* berkisar antara 92% hingga 94%, menandakan kestabilan dan efektivitas dalam klasifikasi. Sebaliknya, metode *If-Else* menunjukkan nilai MAPE yang sangat tinggi, mencapai lebih dari 1200%, bahkan hingga 1500%, yang menandakan performa prediksi yang buruk serta inkonsistensi terhadap variasi beban data.

Pengujian juga menunjukkan bahwa metode *Array-Based* menghasilkan rata-rata waktu tunggu per orang yang lebih efisien di seluruh skenario pengujian, dengan rata-rata waktu tunggu berkisar antara 0,49 detik hingga 0,67 detik, jauh lebih rendah dibandingkan metode *If-Else* yang berkisar antara 7,86 detik hingga 8,95 detik.



Gambar 5. Grafik perbandingan e_{rms}

Dengan demikian, dapat disimpulkan bahwa metode *Array-Based* lebih efektif dan efisien untuk digunakan dalam sistem pengambilan keputusan antrian, baik dari sisi waktu pemrosesan maupun tingkat keakuratannya.

5.2 SARAN

Berdasarkan hasil penelitian yang telah dilakukan, beberapa saran yang dapat disampaikan antara lain:

Metode *Array-Based* yang efisien dan akurat cocok diterapkan pada sistem antrian nyata yang memerlukan respons cepat dan keputusan real-time.

Penelitian selanjutnya dapat menggabungkan metode *Array-Based* dengan kecerdasan buatan, seperti pembelajaran mesin, untuk meningkatkan akurasi serta kemampuan adaptasi terhadap pola antrian yang dinamis.

Pengujian dapat diperluas dengan jumlah data yang lebih besar dan kondisi yang lebih kompleks untuk menguji kestabilan dan skalabilitas metode *Array-Based* dalam berbagai skenario penggunaan yang lebih nyata.

Untuk meningkatkan fleksibilitas penggunaan, pengembangan selanjutnya dapat mengadaptasi metode ini dalam berbagai platform perangkat seperti sistem berbasis web, mobile, maupun embedded system untuk kebutuhan perangkat IoT.

Perlu juga dipertimbangkan faktor-faktor eksternal seperti perilaku pengguna, jenis layanan, serta waktu puncak antrian

DAFTAR PUSTAKA

Altagiuri, R. E., Zaghloul, O. H., Do, B. H., & Stroppa, F. (2024). A Motion Planner for Growing Reconfigurable Inflated Beam Manipulators in Static Environments. *IEEE Robotics and Automation Letters*.

- Apriliyanto, E., & Rahayu, Y. S. (2024). Comparison of Sentiment Analysis from Twitter Data Collection with Naïve Bayes, Decision Tree, and k-Nearest Neighbor Methods. *Jurnal Ilmiah SINUS*, 22(2), 1-12.
- Apriliyanto, E. (2023). Comparison of Response Time Database RDBMS with NoSQL on Electronic Medical Records (EMR). *Jurnal Ilmiah SINUS*, 21(2), 65-78.
- Apriliyanto, E., Putra, R. I., & Rahayu, Y. S. (2024). Peran AI Chatbots dalam Layanan Mahasiswa Ilmu Komunikasi Menggunakan Metode Natural Language Processing (NLP)
- Chen, Y., Hua, F., Jin, Y., & Zhang, E. Z. (2021, August). BGPQ: A Heap-Based Priority Queue Design for GPUs. In *Proceedings of the 50th International Conference on Parallel Processing* (pp. 1-10).
- Dewi, D. S. K. (2021). Pelaksanaan E-Government Di Pemerintah Daerah Kabupaten Ponorogo: Array. *MODERAT: Jurnal Ilmiah Ilmu Pemerintahan*, 7(2), 357-369.
- Evanthe, H., Sinaga, R. M., Halawa, S. P. P., Priscillia, S. A., & Ramadhani, F. (2025). Implementasi Algoritma Antrian Prioritas Menggunakan Array di Python Untuk Sistem Antrian Layanan Darurat. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 9(1), 90-95.
- Guo, J., Liu, Y., Shin, H., & Nallanathan, A. (2024). Deep learning for beamforming in multi-user continuous aperture array (CAPA) systems. *arXiv preprint arXiv:2411.09104*.
- Han, S., Oh, J., Oh, K., & Ha, J. (2022). Deep-learning for breaking the trapping sets in low-density parity-check codes. *IEEE Transactions on Communications*, 70(5), 2909-2923.
- Igarashi, Y., Dozono, M., Honda, R., Kobayashi, N., Lin, C. S., Ota, S., ... & Takahashi, T. N. (2024). Implementations of streaming DAQ on actual detector systems. *IEEE Transactions on Nuclear Science*.
- Li, Z., Wang, C., Shen, F., Peng, Y., & Wang, M. (2024). Goose Queue Inspired V-Array With Cross-Section Variations for Enhanced Electromechanical Energy Harvesting Behaviors. *IEEE/ASME Transactions on Mechatronics*.
- Polak, M. S. H., Troendle, D. A., & Jang, B. (2025). Boundary-Aware Concurrent Queue: A Fast and Scalable Concurrent FIFO Queue on GPU Environments. *Applied Sciences*, 15(4), 1834.
- Prasad, A., Rajendra, S., Rajan, K., Govindarajan, R., & Bondhugula, U. (2022, October). Treebeard: An optimizing compiler for decision tree based ML inference. In *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)* (pp. 494-511). IEEE.
- Tan, B. L., Mok, K. M., Chang, J. J., Lee, W. K., & Hwang, S. O. (2021). RISC32-LP: Low-power FPGA-based IoT sensor nodes with energy reduction program analyzer. *IEEE Internet of Things Journal*, 9(6), 4214-4228.
- Tang, Y., Zhou, H., Ji, Z., & Wang, C. L. (2025). Cube-fx: Mapping Taylor Expansion Onto Matrix Multiplier-Accumulators of Huawei Ascend AI Processors. *IEEE Transactions on Parallel and Distributed Systems*.
- Thievenaz, H., Kimura, K., & Alias, C. (2021). Towards Trace-Based Array Contraction (Doctoral dissertation, Inria; Waseda University).
- Sanjel, A., Khanal, B., Rivas, P., & Speegle, G. (2024, December). Automated Synthesis of Distributed Code from Sequential Snippets Using Deep Learning. In *2024 IEEE International Conference on Big Data (BigData)* (pp. 4119-4126). IEEE.
- Xu, R., Ma, S., Guo, Y., & Li, D. (2023). A survey of design and optimization for systolic *array-based* dnn accelerators. *ACM Computing Surveys*, 56(1), 1-37.