

Systematic Literature Review (SLR): Dampak Modifikasi Algoritma Round-Robin dalam Penjadwalan CPU pada Sistem Operasi

Hanif Ahmad Syauqi^{1*}, Herbert Siregar²

^{1,2}Universitas Pendidikan Indonesia, Indonesia

¹hanifsyauqi24@upi.edu, ²herbert@upi.edu



Histori Artikel:

Diajukan: 19 Juni 2025

Disetujui: 20 Juli 2025

Dipublikasi: 23 Juli 2025

Kata Kunci:

Penjadwalan CPU; Algoritma Round-Robin; Modifikasi; Kinerja; Sistem Operasi

Digital Transformation

Technology (Digitech) is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0).

Abstrak

Penjadwalan CPU adalah inti dari kinerja sistem operasi, dengan algoritma Round-Robin (RR) sebagai metode yang populer karena kesederhanaannya. Namun, RR klasik memiliki kelemahan: pemilihan *time quantum* yang statis dapat menyebabkan *overhead* tinggi atau waktu respons yang buruk. Tinjauan literatur sistematis ini mengidentifikasi, mengevaluasi, dan mensintesis berbagai modifikasi RR yang diusulkan untuk mengatasi keterbatasan ini, berdasarkan 15 publikasi terpilih. Hasilnya menunjukkan bahwa modifikasi dapat dikelompokkan ke dalam beberapa pendekatan, seperti berbasis pembelajaran mesin, dinamis (*Dynamic RR*), hibrida, dan statistik. Modifikasi ini secara signifikan meningkatkan kinerja dengan mengurangi metrik kunci seperti *average waiting time* (AWT), *average turnaround time* (ATT), dan jumlah peralihan konteks (*context switches*) dibandingkan RR biasa. Kesimpulannya, penerapan *time quantum* yang dinamis dan adaptif secara efektif mengatasi kelemahan RR standar, menghasilkan kinerja CPU yang lebih efisien dan responsif.

PENDAHULUAN

Sistem operasi (SO) modern sangat bergantung pada penjadwalan CPU untuk efisiensi dan keadilan dalam pengelolaan sumber daya (Heryana et al., 2023). Algoritma Round-Robin (RR) adalah metode penjadwalan yang populer karena kesederhanaan dan kemampuannya menyediakan keadilan waktu bagi setiap proses melalui alokasi *time quantum* yang bergiliran (Iqbal et al., 2023).

Namun, RR klasik memiliki keterbatasan. Pemilihan ukuran *time quantum* yang tidak tepat dapat menyebabkan *overhead* tinggi (jika terlalu kecil) atau waktu respons buruk (jika terlalu besar). (Lumbantoruan, 2016). Menyadari hal ini, banyak penelitian telah mengusulkan berbagai modifikasi dan peningkatan pada algoritma RR. Modifikasi ini bertujuan untuk mengatasi kelemahan tersebut dengan memperkenalkan mekanisme adaptif, prioritas, atau skema alokasi kuantum yang lebih cerdas, demi meningkatkan efisiensi dan responsivitas sistem.

Oleh karena itu, tinjauan literatur sistematis ini krusial untuk secara komprehensif mengidentifikasi, mengevaluasi, dan mensintesis modifikasi RR yang telah diusulkan. Penelitian ini akan mengkaji dampak modifikasi tersebut terhadap metrik kinerja penjadwalan CPU pada sistem operasi. Berdasarkan pada penjelasan mengenai latar belakang di atas, dibuat beberapa pertanyaan penelitian (*Research Question*) dengan sistem kerangka PICOS, yaitu: P (*Population/Problem*): Algoritma RR klasik dan modifikasinya yang digunakan. I (*Intervention*): Berbagai modifikasi algoritma RR. C (*Comparison*): Perbandingan antara algoritma RR biasa dengan RR modifikasi, atau perbandingan antara RR modifikasi itu sendiri. O (*Outcome*): Dampak kinerja penjadwalan CPU, yang dapat diukur seperti waktu tunggu (*waiting time*), waktu respons (*response time*), *throughput*, utilisasi CPU, jumlah peralihan konteks (*context switches*), atau *fairness*. S (Study Design): Studi yang membahas modifikasi RR. Berdasarkan kerangka PICOS yang telah dijabarkan, dirumuskan beberapa pertanyaan penelitian pada tabel 1.

Tabel 1
Rumusan Pertanyaan

No	Research Question (RQ)
1	Apa saja jenis modifikasi algoritma RR yang telah dibahas?
2	Bagaimana dampak modifikasi algoritma RR terhadap kinerja penjadwalan CPU dibandingkan dengan algoritma RR biasa?
3	Modifikasi algoritma RR mana yang menunjukkan peningkatan signifikan dalam skenario tertentu?

STUDI LITERATUR

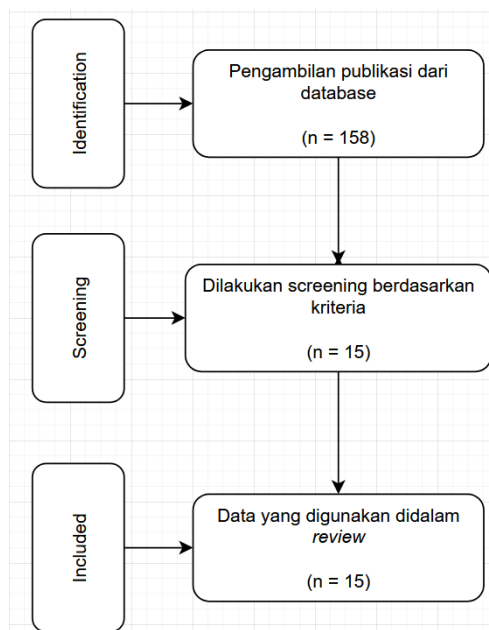
Tinjauan terhadap penelitian-penelitian yang ada menunjukkan bahwa kelemahan utama dari algoritma RR klasik terletak pada *time quantum* (TQ) yang bersifat statis. TQ yang terlalu kecil dapat menyebabkan *overhead* sistem yang tinggi akibat seringnya terjadi *context switch*, sementara TQ yang terlalu besar membuat RR cenderung berperilaku seperti algoritma *First Come First Serve* (FCFS), yang pada akhirnya memperpanjang waktu tunggu (Putra & Purnomo, 2021). Akibatnya, sebagian besar penelitian yang ada berupaya memodifikasi RR dengan mengembangkan mekanisme TQ yang dinamis. Tujuan utamanya adalah untuk mengoptimalkan metrik-metrik kinerja penting seperti *Average Waiting Time* (AWT), *Average Turnaround Time* (ATT), dan jumlah *context switch* (Biswas, Samsuddoha, Al Asif, & Ahmed, 2023).

Salah satu pendekatan yang paling banyak diteliti adalah penentuan TQ dinamis menggunakan metode statistik berdasarkan atribut proses yang ada di dalam *ready queue*. Beberapa algoritma seperti *Median-Average Round Robin* (MARR) dan *Modified Median Mean Round Robin* (MMRR) berhasil menggunakan kombinasi nilai rata-rata (*mean*) dan median dari *burst time* untuk menghitung TQ yang lebih adaptif (Abdelkader et al., 2022; Purnomo et al., 2025). Pendekatan lain yang juga populer adalah dengan menciptakan algoritma hibrida yang menggabungkan kekuatan RR dengan algoritma penjadwalan fundamental lainnya. Penelitian telah menunjukkan keberhasilan dalam mengintegrasikan RR dengan *Shortest Job First* (SJF) untuk memprioritaskan pekerjaan yang lebih pendek (AL-SAFAR, 2021) dan dengan penjadwalan berbasis prioritas untuk menangani proses-proses krusial serta mengatasi masalah *starvation* (Abu-Dalbouh, 2022).

Dalam beberapa tahun terakhir, teknik kecerdasan buatan telah diterapkan untuk menghasilkan optimasi TQ yang lebih canggih. Model pembelajaran mesin seperti *K-Nearest Neighbors* (KNN) telah digunakan untuk memprediksi TQ yang optimal secara *real-time* berdasarkan karakteristik proses (Abdelhameed, Alhode, Alkikly, & Alkharbash, 2024). Selain itu, teknik klusterisasi seperti *K-Means* juga terbukti efektif untuk mengelompokkan proses-proses dengan atribut serupa, di mana setiap kluster kemudian dapat diberi TQ yang telah disesuaikan (Mostafa & Amano, 2020b). Pendekatan berbasis grup ini, seperti yang ditunjukkan oleh model AE-IGBQT, juga berhasil mengurangi AWT dan ATT dengan membagi proses ke dalam grup-grup yang lebih mudah dikelola (Al-Khatib et al., 2023). Secara keseluruhan, beragam pendekatan modifikasi ini secara konsisten menunjukkan peningkatan kinerja yang signifikan dibandingkan RR klasik, dengan beberapa metode melaporkan peningkatan efisiensi antara 12% hingga 62% (Hussein & Kareem, 2024).

METODE

Penulis menggunakan metode pencarian sistematis dengan pendekatan *Preferred Reporting Items for Systematic Literature Reviews* (PRISMA). PRISMA adalah panduan atau checklist yang digunakan untuk membantu peneliti dalam menyusun dan melaporkan *systematic review* (termasuk SLR) dan meta-analisis secara transparan dan lengkap. Ada beberapa tahapan dalam melakukan analisis dengan pendekatan ini.



Gambar 1. Diagram Alur Tahapan

Dari gambar 1, tahapan pencarian dimulai dengan melakukan pengambilan publikasi dari database. Setelah itu, publikasi dilakukan *screening* sesuai dengan kriteria yang ditentukan. Jika publikasi sesuai dengan kriteria, maka publikasi tersebut akan dijadikan sebagai referensi. Ada beberapa kriteria inklusi yang dicantumkan oleh penulis agar pencarian sesuai dengan apa yang ingin dicapai pada *paper* ini. Kriterianya berada pada tabel 2.

Tabel 2
Kriteria Inklusi

No	Kriteria
1	Publikasi berada pada jangka tahun 2020-2025
2	Meliputi penjelasan modifikasi algoritma RR
3	Meliputi dampak penggunaan pada modifikasi algoritma <i>Round-Robin</i>
4	Publikasi berbahasa Inggris atau Indonesia
5	Publikasi <i>open access</i> dan bisa di- <i>download</i>

HASIL

Untuk pencarian publikasi, penulis menggunakan database dari Semantic Scholar dengan keyword "CPU Scheduling" AND "round robin" AND "modified". Dari pencarian tersebut terdapat sebanyak 158 publikasi yang relevan dengan keyword tersebut. Setelah mendapatkan pencarian, penulis melakukan *screening* dengan kriteria yang ada pada tabel 2 diatas. Setelah dilakukan *screening*, terdapat 15 publikasi yang sesuai dengan kriteria. Data tersebut berada pada tabel 3.

Tabel 3
Pilihan Publikasi

No	Penulis	Judul	Tujuan Utama
1	(Abdelhameed et al., 2024)	Optimizing Round Robin Using Machine Learning Models (KNN) in CPU scheduling	Mengusulkan peningkatan algoritma RR dengan menggabungkan <i>machine learning</i> KNN untuk menentukan TQ secara dinamis dan optimal.
2	(Abdelkader et al., 2022)	MMRR: a Modified Median Mean Round Robin Algorithm for Task Scheduling	Mengusulkan algoritma MMRR untuk meningkatkan kinerja RR dengan menemukan TQ dinamis yang ideal berdasarkan sisa burst time proses.
3	(Abu-Dalbouh, 2022)	A New Combination Approach to CPU Scheduling based on Priority and Round-Robin Algorithms for Assigning a Priority to a Process and Eliminating Starvation	Mengembangkan algoritma baru (mix PI-RR) yang menggabungkan RR dan penjadwalan berbasis prioritas untuk mengatasi kelemahan kedua algoritma serta menghilangkan masalah <i>starvation</i> .
4	(Cpu et al., 2025)	An Enhanced and Improved Half-Life Variable Quantum Time Round Robin (EImHLVQTRR) CPU Scheduling Algorithm	Menyempurnakan algoritma ImHLVQTRR dengan memodifikasi TQ untuk meningkatkan kinerja sistem, termasuk AWT, ATT, dan ART (Average Response Time).
5	(Ahmad, 2022)	Improved Round Robin CPU Scheduling Algorithm with Different Arrival Times Based on Dynamic Quantum	Mengusulkan algoritma Different Arrival-Dynamic Quantum Round Robin (DADQRR) untuk mencapai waktu tunggu yang singkat sambil mempertahankan waktu respons yang wajar.
6	(AL-SAFAR, 2021)	Hybrid CPU Scheduling algorithm SJF-RR In Static Set of Processes	Mengusulkan algoritma hibrida (Hybrid SJF-RR) yang menggabungkan fitur terbaik dari SJF dan RR untuk memaksimalkan utilisasi CPU serta meminimalkan AWT dan ATT.
7	(Al-Khatib et al.,	A new enhanced IGBTQ-based	Mengembangkan algoritma baru (AE-

No	Penulis	Judul	Tujuan Utama
	2023)	model for CPU scheduling	IGBQT) sebagai penyempurnaan dari model <i>Improved Group Based Time Quantum</i> (IGBTQ) untuk meningkatkan AWT dan ATT dalam penjadwalan CPU.
8	(Faizan, Marikal, & Anil, 2020)	A Hybrid Round Robin Scheduling Mechanism for Process Management	Mengusulkan mekanisme penjadwalan hibrida (HYRR) yang mewarisi sifat-sifat RR, SJF, dan FCFS dengan TQ dinamis untuk meminimalkan AWT, ATT, ART, dan NCS (Number of Context Switches).
9	(Hussein & Kareem, 2024)	A Proposed CPU Job-Scheduling Technique Based on Round Robin Method Using Dual Synchronized Time-Slices	Mengusulkan metode penjadwalan CPU yang dimodifikasi menggunakan teknik RR dan dual synchronized time-slices untuk mengurangi waktu tunggu pekerjaan.
10	(Jamil, Yar, & Ijaz, 2024)	Dynamic Time Quantum Computation for Improved Round Robin Scheduling Algorithm Using Quartiles and Randomization (IRRQR)	Mengusulkan algoritma IRRQR yang menggunakan kuartil dan randomisasi untuk menghitung TQ dinamis dengan tujuan mengoptimalkan alokasi waktu CPU.
11	(Mostafa & Amano, 2020a)	An adjustable variant of round robin algorithm based on clustering technique	Mengusulkan versi modifikasi RR yang menggabungkan keunggulan overhead rendah dari RR dan mendukung proses singkat untuk meminimalkan biaya waktu, dengan memanfaatkan teknik klusterisasi.
12	(Mostafa & Amano, 2020b)	Dynamic Round Robin CPU Scheduling Algorithm Based on K-Means Clustering Technique	Mengusulkan versi modifikasi dari algoritma RR yang menggunakan teknik <i>clustering K-Means</i> untuk mengelompokkan proses serupa dan meminimalkan AWT, ATT, dan NCS.
13	(Purnomo et al., 2025)	Median-Average Round Robin (MARR) Algorithm for Optimal CPU Task Scheduling	Menganalisis dan mengusulkan algoritma MARR di mana TQ dihitung secara dinamis dari <i>median</i> dan rata-rata <i>burst time</i> untuk mendapatkan ATT dan AWT yang lebih rendah.
14	(Putra & Fadriya, 2021)	Comparison Between Simple Round Robin and Intelligent Round Robin Algorithms in CPU Scheduling	Membandingkan efisiensi antara algoritma simple round robin dan intelligent round robin untuk membuktikan bahwa versi intelligent lebih efisien.
15	(Qazi, Agha, Naseem, Badar, & Khan, 2023)	Improving Round Robin Scheduling with Dynamic Time Quantum (IRRDRQ)	Memperkenalkan algoritma IRRDRQ untuk mengurangi ATT, AWT, dan <i>context switching</i> dengan mengatur proses berdasarkan <i>burst time</i> minimum dan menetapkan TQ secara dinamis.

PEMBAHASAN

RQ1. Apa saja jenis modifikasi algoritma Round-Robin yang telah dibahas?

Berbagai jenis modifikasi algoritma RR telah dibahas untuk meningkatkan kinerja penjadwalan CPU: Algoritma Round Robin Berbasis Pembelajaran Mesin

Pendekatan modern dalam optimasi algoritma RR telah mengintegrasikan teknik pembelajaran mesin untuk meningkatkan efisiensi dan adaptivitas penjadwalan. Salah satu implementasinya adalah penggunaan KNN untuk memprediksi waktu kuantum optimal secara dinamis berdasarkan karakteristik proses, yang didukung oleh manipulasi data menggunakan *NumPy*. Selain KNN, model regresi dan klasifikasi lain seperti regresi linier,

polinomial, dan *Random Forest* juga dimanfaatkan untuk tujuan serupa (Abdelhameed et al., 2024). Di sisi lain, algoritma *Dynamic Round Robin* (DRR) yang berbasis *K-Means Clustering* menerapkan pendekatan berbeda dengan mengelompokkan proses-proses yang memiliki atribut serupa, seperti periode layanan CPU dan bobot. Berdasarkan pengelompokan ini, algoritma kemudian menetapkan irisan waktu yang seragam untuk setiap kluster (Mostafa & Amano, 2020b).

Algoritma Dinamis

Berbagai metode telah dikembangkan untuk mengimplementasikan TQ yang dinamis pada algoritma RR guna meningkatkan kinerja sistem. Sebagai contoh, algoritma ImHLVQTRR terbukti secara signifikan memperbaiki AWT dan waktu penyelesaian sekaligus mengurangi *context switching* (Cpu et al., 2025). Serupa dengan itu, *DRR with Controlled Preemption* (DRRCP) juga fokus pada pengurangan *context switching* yang tidak perlu dengan menerapkan TQ dinamis (Abdelhameed et al., 2024). Terdapat pula beragam variasi lain yang diusulkan oleh Mostafa & Amano (2020a), seperti *Variable Time Round-Robin* (VTRR) yang mengalokasikan irisan waktu berdasarkan total beban kerja, *burst time* proses, dan jumlah proses dalam antrian. Ada pula *Changeable Time Quantum* (CTQ) yang memilih kuantum dengan AWT terkecil di setiap putaran, serta *Adaptive80* RR yang menetapkan irisan waktu berdasarkan *burst time* proses pada persentil ke-80. Pendekatan lainnya mencakup *Proportional Weighted Round Robin* (PWRR) yang memberikan irisan waktu proporsional terhadap *burst time*, *Adjustable Round Robin* (ARR) yang menggabungkan *overhead* rendah dengan dukungan untuk proses singkat, dan *Amended DDR* (ADRR) yang menyesuaikan irisan waktu secara siklis.

Algoritma Hibrida

Konsep hibrida menggabungkan kekuatan algoritma RR dengan algoritma penjadwalan lainnya untuk menghasilkan kinerja yang lebih unggul. Sebagai contoh, *Variant On Round Robin* (VORR-KNN) memodifikasi RR dengan menentukan TQ berdasarkan median dari *burst time* (Abdelhameed et al., 2024). Algoritma *SJF and RR with dynamic quantum* (SRDQ) membagi antrian menjadi dua—untuk tugas singkat dan panjang—dan menetapkan irisan waktu berdasarkan median dan *burst time* (Mostafa & Amano, 2020a). Demikian pula, HYRR mewarisi sifat dari RR, SJF, dan FCFS, di mana TQ nya dihitung secara dinamis menggunakan rata-rata dan nilai minimum dari *burst time* (Faizan et al., 2020). Algoritma lain seperti *mix* PI-RR mengombinasikan RR dengan penjadwalan berbasis prioritas (Abu-Dalbouh, 2022), sedangkan SJF-RR mengintegrasikan properti kernel SJF dengan fitur terbaik RR, menggunakan persamaan optimal untuk mengadaptasi faktor TQ di setiap putaran (AL-SAFAR, 2021).

Algoritma Berbasis Statistik/Matematika:

Pendekatan berbasis statistik dan matematika menawarkan metode kuantitatif untuk menentukan TQ yang optimal. Algoritma IRRQR memanfaatkan kuartil median untuk mengelompokkan proses dan kuartil atas sebagai ambang batas untuk proses yang berjalan lama, dengan elemen pengacakan untuk pemilihan nilai (Jamil et al., 2024). Sementara itu, MARR menghitung TQ dengan mengambil nilai tengah dari penjumlahan median dan rata-rata *burst time* (Purnomo et al., 2025). Algoritma DADQRR mengadaptasi TQ berdasarkan *burst time* kumulatif dari proses yang ada di antrian siap (Ahmad, 2022). Selain itu, IRR menggunakan formula yang lebih kompleks, mencakup komponen seperti *Original Time Slice* (OTS), *Priority Component* (PC), *Shortness Component* (SC), dan *Context Switch Component* (CSC) (Putra & Fadriya, 2021). Terakhir, *Improving Round Robin Scheduling with Dynamic Time Quantum* IRRDQ mengatur proses berdasarkan *burst time* minimum dan menetapkan TQ optimal menggunakan prinsip SJF (Qazi et al., 2023).

Algoritma Round Robin Berbasis Kelompok:

Salah satu pendekatan unik dalam pengembangan algoritma RR adalah dengan berbasis kelompok. Algoritma IGBTQ merupakan penyempurnaan dari metode sebelumnya, yaitu *Group Based Time Quantum* (GBTQ). Dalam versi yang diusulkan, yaitu AE-IGBQT, jumlah kelompok dibatasi secara dinamis. Mekanisme ini bekerja dengan mengadaptasi 80% dari total proses untuk menentukan jumlah akhir kelompok yang akan digunakan dalam penjadwalan (Al-Khatib et al., 2023).

RQ2. Bagaimana dampak modifikasi algoritma Round-Robin terhadap kinerja penjadwalan CPU dibandingkan dengan algoritma Round-Robin biasa?

Modifikasi algoritma RR secara konsisten menunjukkan dampak signifikan terhadap peningkatan kinerja penjadwalan CPU dibandingkan dengan algoritma RR klasik, terutama dalam meminimalkan metrik kunci seperti AWT, ATT, dan NCS. Sebagai contoh, algoritma IRR terbukti lebih efisien dengan jumlah *context switching* yang lebih rendah (11 berbanding 16 pada kasus uji), AWT yang lebih singkat (25,4ms berbanding 32,6ms), dan ATT yang lebih cepat (33,4ms berbanding 44,2ms) dibandingkan RR standar (Putra & Fadriya, 2021). Peningkatan performa yang lebih besar ditunjukkan oleh algoritma DADQRR, yang mengungguli RR dan varian lainnya

dengan peningkatan AWT antara 6,155% hingga 31,409% (Ahmad, 2022). Demikian pula, implementasi IRRQR berhasil menekan NCS hingga 25%, AWT hingga 13,7%, dan ATT sebesar 8,4% (Jamil et al., 2024). Selain itu, sebuah teknik yang berbasis *Dual Synchronized Time-Slices* juga memperlihatkan efisiensi optimasi yang tinggi, dengan persentase rata-rata mencapai 12,4% hingga 62,07% di atas RR asli, yang berdampak langsung pada penurunan AWT secara drastis (Hussein & Kareem, 2024). Secara keseluruhan, berbagai modifikasi ini membuktikan bahwa penyesuaian TQ secara dinamis dan cerdas mampu mengatasi kelemahan fundamental dari algoritma RR klasik.

RQ3. Modifikasi algoritma Round-Robin mana yang menunjukkan peningkatan signifikan dalam skenario tertentu?

Beberapa modifikasi algoritma RR menunjukkan peningkatan kinerja yang signifikan dengan memanfaatkan teknik pembelajaran mesin dan pengelompokan (*clustering*). Sebagai contoh, sebuah pendekatan yang mengoptimalkan RR menggunakan model KNN berhasil menunjukkan peningkatan substansial pada parameter AWT, ATT, dan NCS dibandingkan dengan RR tradisional (Abdelhameed et al., 2024). Serupa dengan itu, algoritma yang berbasis *K-Means Clustering* juga memberikan hasil yang lebih baik secara keseluruhan, di mana dalam sebuah contoh ilustratif, algoritma ini berhasil mengurangi AWT dari 31,75 menjadi 26,5 satuan waktu dan ATT dari 50 menjadi 44,75 satuan waktu (Mostafa & Amano, 2020b). Pendekatan lain yang berbasis *clustering*, yaitu *Dynamic Time Slice* (DTS), juga secara konsisten terbukti mengungguli algoritma lain yang dijadikan pembandingan dalam penelitiannya (Mostafa & Amano, 2020a).

Kekokohan beberapa algoritma modifikasi teruji dalam berbagai skenario dinamis yang berbeda. IRRQR menunjukkan keunggulan yang konsisten di berbagai kondisi: pada kasus dengan *burst time* menaik, ia mengurangi *context switch*, AWT, dan ATT masing-masing sebesar 25%, 8,4%, dan 5,6%; pada *burst time* menurun, pengurangannya adalah 11,1%, 7,7%, dan 5,6%; dan pada *burst time* acak, pengurangannya mencapai 12,5%, 8,7%, dan 6,3%. Kinerja superior ini juga teramati dalam skenario dengan waktu kedatangan proses yang tidak serentak (Jamil et al., 2024). Demikian pula, ImHLVQTRR secara signifikan mengungguli RR, baik pada skenario waktu kedatangan nol (AWT 13,4ms vs 20,4ms) maupun pada waktu kedatangan non-nol (AWT 18,6ms vs 28,6ms) (Cpu et al., 2025). DADQRR juga menunjukkan keunggulannya pada skenario dengan rentang *burst time* spesifik (2-10), di mana ia mengungguli algoritma lain dalam hal AWT dengan selisih 15,862% hingga 31,409% (Ahmad, 2022).

Pendekatan hibrida dan berbasis kelompok juga terbukti sangat efektif. Model AE-IGBQT, yang merupakan penyempurnaan dari penjadwalan berbasis kelompok, berhasil mencapai AWT dan ATT terendah dibandingkan RR, DRRC, GBTQ, dan IGBTQ dalam dua kasus uji yang berbeda (Al-Khatib et al., 2023). Algoritma hibrida SJF-RR menunjukkan superioritasnya dalam set proses statis dengan mencatatkan waktu tunggu 19,918 dan waktu penyelesaian 29,205, jauh lebih rendah dari RR tradisional (AL-SAFAR, 2021). Begitu pula dengan algoritma *mix* PI-RR yang mengombinasikan RR dengan penjadwalan berbasis prioritas, yang secara konsisten menghasilkan AWT dan ATT yang lebih rendah di seluruh sampel yang diuji (Abu-Dalbouh, 2022).

Beberapa modifikasi lain menunjukkan keunggulan optimal pada kondisi analisis tertentu atau untuk jenis pekerjaan spesifik. Algoritma MARR berhasil mencapai AWT dan ATT terendah dalam salah satu analisis komparatifnya (Purnomo et al., 2025). Sementara itu, sebuah teknik yang menggunakan *Dual Synchronized Time-Slices* menunjukkan efisiensi optimasi yang sangat tinggi, dengan rata-rata 12,4% hingga 62,07% lebih baik dari RR asli, dan terbukti sangat menguntungkan untuk menangani pekerjaan (proses) yang lebih panjang (Hussein & Kareem, 2024).

KESIMPULAN

Penjadwalan CPU adalah inti dari kinerja sistem operasi modern, namun efisiensi algoritma RR standar terhambat oleh ketergantungannya pada TQ yang statis, dimana nilai yang tidak optimal dapat meningkatkan *overhead* atau memperpanjang waktu tunggu. Untuk mengatasi keterbatasan ini, berbagai modifikasi RR telah diusulkan dengan fokus utama pada penentuan TQ yang dinamis dan adaptif. Pendekatan-pendekatan ini sangat beragam, mulai dari algoritma berbasis pembelajaran mesin yang menggunakan model seperti KNN dan *K-Means Clustering* untuk memprediksi kuantum optimal, hingga algoritma dinamis seperti ImHLVQTRR dan VTRR yang mengubah TQ berdasarkan karakteristik proses. Selain itu, dikembangkan pula algoritma hibrida yang menggabungkan fitur terbaik RR dengan algoritma lain seperti SJF atau penjadwalan berbasis prioritas dalam metode SJF-RR dan *mix* PI-RR, serta algoritma berbasis statistik/matematis seperti IRRQR dan MARR yang menggunakan perhitungan formula untuk optimasi. Secara keseluruhan, semua modifikasi ini bertujuan sama yaitu meningkatkan efisiensi waktu dan sumber daya secara signifikan dengan mengurangi AWT, ATT, dan jumlah *context switches*, sehingga menghasilkan kinerja CPU yang lebih unggul dalam lingkungan komputasi yang dinamis.

REFERENSI

- Abdelhameed, S. T., Alhode, N., Alkikly, M., & Alkharbash, F. O. (2024). *Optimizing round robin using machine learning models KNN in CPU scheduling*. (Ii), 1–10.
- Abdelkader, A., Ghazy, N., Zaki, M. S., & ElDahshan, K. A. (2022). MMMRR: a modified median mean round robin algorithm for task scheduling. *International Journal of Intelligent Engineering and Systems*, 15(6), 599–608. <https://doi.org/10.22266/ijies2022.1231.53>
- Abu-Dalbouh, H. M. (2022). A new combination approach to CPU scheduling based on priority and round-robin algorithms for assigning a priority to a process and eliminating starvation. *International Journal of Advanced Computer Science and Applications*, 13(4), 541–546. <https://doi.org/10.14569/IJACSA.2022.0130463>
- Ahmad, ̣a. (2022). Improved round robin CPU scheduling algorithm with different arrival times based on dynamic quantum. *Journal of Education and Science*, 31(4), 105–115. <https://doi.org/10.33899/edusj.2022.135082.1273>
- Al-Khatib, R. M., Al-Khateeb, A., Al-Daom, E., Al-Dagamseh, I. T., Tawalbeh, A., & Abualigah, L. (2023). A new enhanced IGBTQ-based model for CPU scheduling. *Applied and Computational Engineering*, 8(1), 411–417. <https://doi.org/10.54254/2755-2721/8/20230207>
- AL-SAFAR, A. (2021). Hybrid CPU scheduling algorithm SJF-RR in static set of processes. *Journal of Al-Rafidain University College For Sciences (Print ISSN: 1681-6870 ,Online ISSN: 2790-2293)*, (1), 36–60. <https://doi.org/10.55562/jruc.s.v29i1.377>
- Biswas, D., Samsuddoha, M., Al Asif, M. R., & Ahmed, M. M. (2023). Optimized round robin scheduling algorithm using dynamic time quantum approach in cloud computing environment. *International Journal of Intelligent Systems and Applications*, 15(1), 22–34. <https://doi.org/10.5815/ijisa.2023.01.03>
- Cpu, E., Algorithm, S., & Isa, I. (2025). *A periodical of the Faculty of Natural and Applied Sciences , UMYU , Katsina an enhanced and improved half-life variable quantum time round robin*. 4(1), 305–316.
- Faizan, K., Marikal, A., & Anil, K. (2020). A hybrid round robin scheduling mechanism for process management. *International Journal of Computer Applications*, 177(36), 14–19. <https://doi.org/10.5120/ijca2020919851>
- Heryana, N., Kom, M., Mabruri, A., Kom, S., Kesuma, L. I., Si, S., ... others. (2023). *Sistem operasi*. CV Rey Media Grafika.
- Hussein, S. A., & Kareem, E. I. A. (2024). A proposed CPU job-scheduling technique based on round robin method using dual synchronized time-slices. *Ingenierie Des Systemes d'Information*, 29(5), 1847–1858. <https://doi.org/10.18280/isi.290517>
- Iqbal, M., Ullah, Z., Khan, I. A., Aslam, S., Shaheer, H., Humayon, M., ... Mehmood, A. (2023). Optimizing task execution: the impact of dynamic time quantum and priorities on round robin scheduling. *Future Internet*, 15(3), 104.
- Jamil, B., Yar, A., & Ijaz, H. (2024). Dynamic time quantum computation for improved round robin scheduling algorithm using quartiles and randomization (IRRQR). *Sukkur IBA Journal of Computing and Mathematical Sciences*, 7(2), 25–37. <https://doi.org/10.30537/sjcms.v7i2.1340>
- Lumbantoruan, G. (2016). Modifikasi algoritma round robin dengan dynamic quantum time dan pengurutan proses secara ascending. *Jurnal ISD*, 2(2), 2528–5114.
- Mostafa, S. M., & Amano, H. (2020a). An adjustable variant of round robin algorithm based on clustering technique. *Computers, Materials and Continua*, 66(3), 3253–3270. <https://doi.org/10.32604/cmc.2021.014675>
- Mostafa, S. M., & Amano, H. (2020b). Dynamic round robin CPU scheduling algorithm based on K-means clustering technique. *Applied Sciences (Switzerland)*, 10(15). <https://doi.org/10.3390/app10155134>
- Purnomo, R., Putra, T. D., Bhayangkara, U., & Raya, J. (2025). *Median-average round robin (MARR) algorithm for optimal CPU task scheduling*. 9(1), 90–95.
- Putra, T. D., & Fadriya, A. (2021). Comparison between simple round robin and intelligent round robin algorithms in CPU scheduling. *Ijarccce*, 10(4), 86–90. <https://doi.org/10.17148/ijarccce.2021.10413>
- Putra, T. D., & Purnomo, R. (2021). Analisis algoritma round robin pada penjadwalan CPU. *Jurnal Ilmiah Teknologi Informasi Asia*, 15(2), 85–90. <https://doi.org/10.32815/jitika.v15i2.481>
- Qazi, F., Agha, D.-S., Naseem, M., Badar, S., & Khan, F. H. (2023). Improving round robin scheduling with dynamic time quantum (IRRDQ). *Journal of Applied Engineering & Technology (JAET)*, 7(2), 70–82. <https://doi.org/10.55447/jaet.07.02.115>