

Analisis Penjadwalan Proses Pada Sistem Operasi Untuk Efisiensi Prosesor

Juniar Cicilia Sitohang^{3*}, Endang Pramita Nainggolan², Ekha Gabena Siahaan³, Indra Gunawan⁴

¹ Teknik Informatika, STIKOM Tunas Bangsa

^{1*}ekhasiahaan@gmail.com, ²mitanainggolan00@gmail.com, ³juniarsitohang339@gmail.com, ⁴indra@amiktunasbangsa.ac.id

Abstrak

Penjadwalan proses merupakan salah satu fungsi penting dalam sistem operasi karena menentukan urutan proses dalam memperoleh akses terhadap prosesor. Pemilihan algoritma penjadwalan yang kurang tepat dapat meningkatkan waktu tunggu dan waktu penyelesaian sehingga menurunkan efisiensi pemrosesan. Penelitian ini bertujuan menganalisis dan membandingkan kinerja algoritma First Come First Served (FCFS), Shortest Job First (SJF), Priority Scheduling, dan Round Robin dalam penjadwalan proses. Metode penelitian menggunakan pendekatan eksperimen berbasis simulasi dengan enam proses yang memiliki variasi waktu kedatangan, waktu eksekusi, dan prioritas. Evaluasi dilakukan berdasarkan Average Waiting Time, Average Turnaround Time, dan Response Time. Hasil simulasi menunjukkan bahwa SJF menghasilkan Average Waiting Time terendah sebesar 9,17 satuan waktu dan Average Turnaround Time sebesar 13,83 satuan waktu. FCFS menghasilkan Average Waiting Time 10,00 dan Average Turnaround Time 14,67, sedangkan Priority Scheduling menghasilkan masing-masing 11,67 dan 16,33. Round Robin dengan *time quantum* 3 menghasilkan Average Waiting Time 13,17 dan Average Turnaround Time 17,83. Berdasarkan skenario penelitian, SJF memberikan efisiensi waktu terbaik, sedangkan Round Robin lebih menekankan pemerataan akses prosesor.

Kata Kunci: CPU Scheduling, FCFS, SJF, Round Robin, Priority Scheduling

PENDAHULUAN

Sistem operasi merupakan perangkat lunak yang berfungsi sebagai pengelola sumber daya komputer sekaligus penghubung antara perangkat keras dan pengguna. Salah satu sumber daya utama yang dikelola sistem operasi adalah Central Processing Unit (CPU). Dalam lingkungan *multitasking*, terdapat banyak proses yang membutuhkan akses terhadap CPU pada waktu yang bersamaan. Oleh karena itu, sistem operasi memerlukan mekanisme penjadwalan untuk menentukan proses yang memperoleh kesempatan dieksekusi terlebih dahulu. Penjadwalan CPU menjadi bagian penting karena keputusan tersebut berpengaruh terhadap waktu tunggu, waktu penyelesaian, respons sistem, dan pemanfaatan prosesor.

Penjadwalan CPU pada dasarnya bertugas memilih proses dari *ready queue* untuk diberikan kepada CPU. Pemilihan tersebut dapat dilakukan menggunakan berbagai algoritma dengan karakteristik berbeda. FCFS memilih proses berdasarkan urutan kedatangan, SJF memilih proses dengan waktu eksekusi paling pendek, Priority Scheduling mempertimbangkan tingkat prioritas proses, sedangkan Round Robin memberikan jatah waktu tertentu kepada setiap proses secara bergantian. Pemilihan algoritma tidak dapat dilakukan hanya berdasarkan satu ukuran karena setiap algoritma mempunyai kelebihan dan kekurangan. Pemilihan yang tidak sesuai karakteristik beban kerja dapat menyebabkan peningkatan waktu tunggu atau jumlah pergantian konteks.

Pemasinghe dan Rajapaksha (2022) membahas perbandingan FCFS, SJF, SRTF, Round Robin, Priority Based, dan beberapa algoritma *multilevel queuing*. Penelitian tersebut menunjukkan bahwa setiap algoritma mempunyai karakteristik dan kriteria kinerja yang berbeda sehingga penggunaannya perlu disesuaikan dengan kebutuhan sistem. Tufegdžić, Jevremović, dan Petrović (2022) juga membahas estimasi efisiensi algoritma penjadwalan menggunakan pemrograman berorientasi objek dengan mempertimbangkan beberapa algoritma klasik, seperti FCFS, SJF, SRTF, dan Round Robin.

Sonata, Hutagalung, dan Rachmad (2023) melakukan analisis waktu tunggu pada algoritma Round Robin, SJF, dan FIFO. Penelitian tersebut menunjukkan pentingnya Average Waiting Time sebagai salah satu parameter dalam mengevaluasi kinerja penjadwalan CPU. Purnomo dan Putra (2023) melakukan studi komparatif terhadap Preemptive SJF dan Round Robin dengan mempertimbangkan Average Waiting Time, Turnaround Time, dan *time slice*. Penelitian tersebut menunjukkan bahwa perubahan karakteristik algoritma dapat memberikan hasil berbeda terhadap kinerja proses.

González-Rodríguez et al. (2024) mengembangkan simulator untuk mempelajari dan membandingkan algoritma penjadwalan CPU dalam lingkungan yang terkontrol. Penelitian tersebut menekankan pentingnya simulasi untuk memahami perbedaan kinerja berbagai algoritma secara objektif. Putra dan Purnomo (2024) juga menggunakan OS-SIM untuk melakukan simulasi algoritma Priority Scheduling secara *preemptive*, menunjukkan bahwa simulasi dapat digunakan untuk mempelajari perilaku algoritma penjadwalan secara lebih sistematis.

Hajjar et al. (2024) melakukan penilaian kinerja FCFS, Round Robin, dan SJF menggunakan pendekatan berbasis skenario. Penelitian tersebut memperlihatkan bahwa karakteristik skenario proses dapat memengaruhi hasil evaluasi algoritma. Hussein

dan Abdul Kareem (2024) mengembangkan teknik penjadwalan berbasis Round Robin dengan dua *time slice* tersinkronisasi untuk meningkatkan kinerja penjadwalan. Hal tersebut menunjukkan bahwa pemilihan *time quantum* merupakan aspek penting pada algoritma Round Robin.

Penelitian Al Ghifari dan Siregar (2025) menggunakan pendekatan *Systematic Literature Review* untuk membandingkan FCFS, SJF, Round Robin, dan Priority Scheduling. Hasil kajian menunjukkan bahwa SJF cenderung memberikan waktu tunggu dan *turnaround time* rendah pada lingkungan *batch*, sementara varian Round Robin dapat memberikan *throughput* dan respons yang baik dengan konsekuensi tambahan *context-switching overhead*. Syauqi dan Siregar (2025) secara khusus mengkaji berbagai modifikasi Round Robin dan menunjukkan bahwa *time quantum* statis dapat menjadi salah satu keterbatasan algoritma tersebut.

Berdasarkan penelitian terdahulu tersebut, terdapat *gap* berupa kebutuhan evaluasi yang sederhana, terukur, dan dapat direplikasi oleh mahasiswa dengan menggunakan dataset proses yang sama pada beberapa algoritma klasik. Sebagian penelitian berfokus pada pengembangan algoritma baru, simulasi tertentu, atau kajian literatur, sedangkan penelitian ini membandingkan FCFS, SJF, Priority Scheduling, dan Round Robin menggunakan satu skenario proses yang sama. Dengan demikian, perbedaan hasil dapat diamati secara langsung berdasarkan Average Waiting Time, Average Turnaround Time, dan Response Time.

Penelitian ini bertujuan menganalisis kinerja empat algoritma penjadwalan CPU, yaitu FCFS, SJF, Priority Scheduling, dan Round Robin. Analisis dilakukan melalui simulasi menggunakan dataset proses yang sama dan mengukur waktu tunggu, waktu penyelesaian, serta respons proses. Hasil penelitian diharapkan dapat memberikan gambaran mengenai karakteristik setiap algoritma serta membantu menentukan algoritma yang paling sesuai untuk kondisi proses tertentu.

METODE

Tahapan Penelitian

Penelitian ini menggunakan metode eksperimen berbasis simulasi. Tahapan penelitian terdiri atas identifikasi masalah, studi literatur, penentuan algoritma, penyusunan dataset, simulasi, perhitungan parameter, perbandingan hasil, analisis, dan penarikan kesimpulan.



Gambar 1. Tahapan Penelitian

Tahap identifikasi masalah dilakukan untuk menentukan permasalahan utama dalam penjadwalan CPU, yaitu bagaimana menentukan algoritma yang dapat memberikan waktu tunggu dan waktu penyelesaian yang rendah. Selanjutnya dilakukan studi literatur terhadap penelitian mengenai algoritma penjadwalan CPU. Tahap berikutnya adalah menentukan empat algoritma yang akan dibandingkan. Dataset kemudian disusun dengan memberikan nilai *arrival time*, *burst time*, dan prioritas yang berbeda pada setiap proses.

Setiap algoritma disimulasikan menggunakan dataset yang sama. Hasil simulasi digunakan untuk menghitung *completion time*, *turnaround time*, *waiting time*, dan *response time*. Hasil tersebut kemudian dibandingkan untuk menentukan karakteristik dan performa masing-masing algoritma.

Metode Penyelesaian Masalah

FCFS

FCFS (*First Come First Served*) merupakan algoritma *non-preemptive* yang menjalankan proses berdasarkan urutan kedatangan. Proses yang datang lebih dahulu akan memperoleh CPU lebih dahulu. Kelebihan FCFS adalah mekanismenya sederhana dan mudah diterapkan. Kekurangannya adalah munculnya *convoy effect*, yaitu proses pendek harus menunggu ketika proses panjang berada di depan antrian.

SJF

SJF (*Shortest Job First*) memilih proses dengan *burst time* paling pendek dari proses yang tersedia. Pada penelitian ini digunakan SJF *non-preemptive*. Algoritma ini secara teori dapat memberikan rata-rata waktu tunggu yang rendah. Namun, proses dengan *burst time* panjang dapat mengalami penundaan apabila terus terdapat proses pendek yang datang.

Priority Scheduling

Priority Scheduling menentukan proses berdasarkan nilai prioritas. Pada penelitian ini digunakan Priority Scheduling *non-preemptive* dengan nilai prioritas yang lebih kecil menunjukkan prioritas yang lebih tinggi. Kelebihan metode ini adalah proses penting dapat diproses lebih dahulu. Kekurangannya adalah proses dengan prioritas rendah dapat mengalami *starvation*.

Round Robin

Round Robin merupakan algoritma *preemptive* yang memberikan setiap proses jatah waktu yang disebut *time quantum*. Apabila proses belum selesai ketika jatah waktunya habis, proses dikembalikan ke bagian belakang *ready queue*. Algoritma ini memberikan kesempatan yang relatif merata kepada proses, tetapi *time quantum* yang terlalu kecil dapat meningkatkan jumlah *context switch*. Kajian terbaru juga menunjukkan bahwa pemilihan *time quantum* merupakan faktor penting dalam kinerja Round Robin.

Data Penelitian

Dataset penelitian terdiri atas enam proses. Dataset yang sama digunakan pada seluruh algoritma untuk menjaga keadilan perbandingan.

Tabel 1. Data Proses

Process	Arrival Time	Burst Time	Priority
P1	0	8	3
P2	1	4	1
P3	2	2	4
P4	3	5	2
P5	4	3	5
P6	5	6	2

Dataset tersebut dipilih untuk menghasilkan kondisi proses yang bervariasi. P1 datang paling awal dengan *burst time* relatif panjang, sedangkan P3 mempunyai *burst time* paling pendek. Perbedaan prioritas juga memungkinkan Priority Scheduling menghasilkan urutan berbeda dengan FCFS dan SJF.

Parameter Pengukuran

- **Turnaround Time (TAT):** Total waktu yang dibutuhkan suatu proses dari awal datang hingga selesai dieksekusi

$$TAT = CT - AT$$

- **Waiting Time (WT):** Total waktu yang dihabiskan proses untuk menunggu di *ready queue* sebelum atau selama dieksekusi

$$WT = TAT - BT$$

- **Average Waiting Time (AWT):** Rata-rata waktu tunggu seluruh proses

$$AWT = \frac{\sum WT}{n}$$

- **Average Turnaround Time (ATAT):** Rata-rata waktu penyelesaian seluruh proses

$$ATAT = \frac{\sum TAT}{n}$$

- **Response Time (RT):** Waktu yang dibutuhkan sejak proses datang sampai pertama kali mendapatkan respons/akses dari CPU

$$RT = FST - AT$$

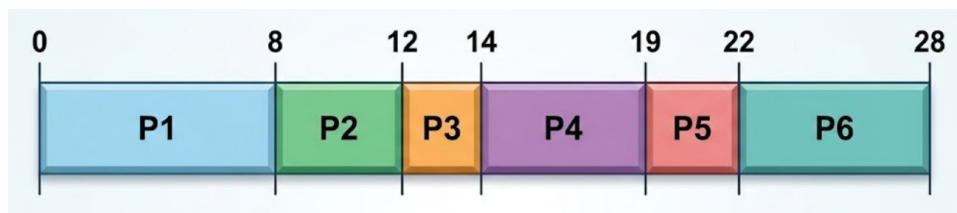
Keterangan:

- **CT** = *Completion Time* (waktu selesai proses)
- **AT** = *Arrival Time* (waktu kedatangan proses)
- **BT** = *Burst Time* (lama waktu eksekusi proses)
- **FST** = *First Start Time* (waktu pertama kali proses mulai dieksekusi CPU)
- **n** = Jumlah total proses

HASIL DAN PEMBAHASAN

Simulasi FCFS

Pada FCFS, proses dijalankan berdasarkan urutan kedatangan sehingga urutannya adalah P1, P2, P3, P4, P5, dan P6.



Gambar 1. Gantt Chart FCFS

Tabel 2. Hasil Simulasi FCFS

Process	CT	TAT	WT	RT
P1	8	8	0	0
P2	12	11	7	7
P3	14	12	10	10
P4	19	16	11	11
P5	22	18	15	15
P6	28	23	17	17

- Jumlah Waiting Time adalah 60 sehingga:

$$AWT = \frac{60}{6} = 10$$

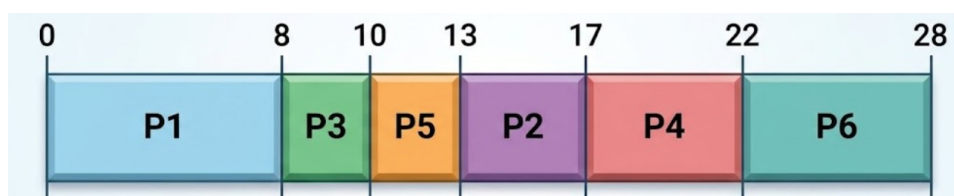
- Jumlah Turnaround Time adalah 88 sehingga:

$$ATAT = \frac{88}{6} = 14,67$$

Hasil tersebut menunjukkan bahwa proses-proses yang datang setelah P1 mengalami waktu tunggu cukup tinggi karena P1 memiliki *burst time* 8 satuan waktu.

Simulasi SJF

SJF memilih proses dengan *burst time* terpendek dari proses yang sudah tersedia. Setelah P1 selesai pada waktu 8, proses yang tersedia adalah P2, P3, P4, P5, dan P6. P3 memiliki *burst time* terpendek sehingga diproses terlebih dahulu, kemudian P5, P2, P4, dan P6.



Gambar 2. Gantt Chart SJF

Tabel 3. Hasil Simulasi SJF

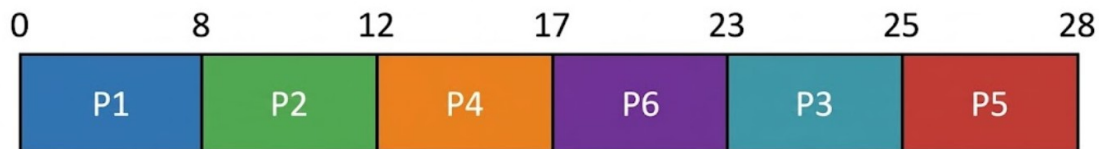
Process	CT	TAT	WT	RT
P1	8	8	0	0
P3	10	8	6	6
P5	13	9	6	6
P2	17	16	12	12
P4	22	19	14	14
P6	28	23	17	17

- Jumlah Waiting Time adalah 55 sehingga:
 $AWT = \frac{55}{6} = 9,17$
- Jumlah Turnaround Time adalah 83 sehingga:
 $ATAT = \frac{83}{6} = 13,83$

SJF menghasilkan Average Waiting Time dan Average Turnaround Time paling rendah dibandingkan tiga algoritma lain pada dataset penelitian. Hasil tersebut sesuai dengan karakteristik SJF yang mengutamakan proses dengan *burst time* lebih pendek.

Simulasi Priority Scheduling

Pada Priority Scheduling, nilai prioritas terkecil dianggap sebagai prioritas tertinggi. Setelah P1 selesai, P2 dengan prioritas 1 dijalankan, kemudian P4 dan P6 yang memiliki prioritas 2. Selanjutnya P3 dan P5 dijalankan berdasarkan prioritas yang lebih rendah.



Gambar 3. Gantt Chart Priority Scheduling

Tabel 4. Hasil Simulasi Priority Scheduling

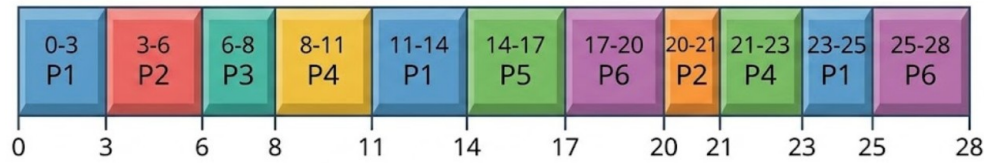
Process	CT	TAT	WT	RT
P1	8	8	0	0
P2	12	11	7	7
P4	17	14	9	9
P6	23	18	12	12
P3	25	23	21	21
P5	28	24	21	21

- Jumlah Waiting Time adalah 70 sehingga:
 $AWT = \frac{70}{6} = 11,67$
- Jumlah Turnaround Time adalah 98 sehingga:
 $ATAT = \frac{98}{6} = 16,33$

Hasil menunjukkan bahwa Priority Scheduling pada dataset ini tidak memberikan Average Waiting Time serendah SJF. Hal tersebut terjadi karena P3 yang memiliki *burst time* paling pendek mempunyai prioritas rendah sehingga harus menunggu sampai proses dengan prioritas lebih tinggi selesai.

Simulasi Round Robin

Round Robin menggunakan *time quantum* sebesar 3 satuan waktu. Proses yang belum selesai setelah memperoleh tiga satuan waktu dikembalikan ke bagian belakang *ready queue*.



Gambar 4. Gantt Chart Round Robin

Tabel 5. Hasil Simulasi Round Robin

Process	CT	TAT	WT	RT
P1	25	25	17	0
P2	21	20	16	2
P3	8	6	4	4
P4	23	20	15	5
P5	17	13	10	10
P6	28	23	17	12

- jumlah Waiting Time adalah 79 sehingga:

$$AWT = \frac{79}{6} = 13,17$$
- Jumlah Turnaround Time adalah 107 sehingga:

$$ATAT = \frac{107}{6} = 17,83$$

Round Robin memberikan kesempatan kepada setiap proses untuk memperoleh CPU secara bergantian. Pada dataset ini, nilai Average Waiting Time lebih tinggi dibandingkan FCFS dan SJF. Namun, karakteristik Round Robin tidak hanya dinilai dari waktu tunggu. Algoritma ini memberikan pemerataan akses CPU sehingga sesuai untuk lingkungan interaktif dan sistem *time-sharing*.

Perbandingan Hasil

Tabel 6. Perbandingan Kinerja Algoritma

Algoritma	Average Waiting Time	Average Turnaround Time	Average Response Time	CPU Utilization
FCFS	10	14,67	10	100%
SJF	9,17	13,83	8,17	100%
Priority	11,67	16,33	11,67	100%
Round Robin	13,17	17,83	5,5	100%

KESIMPULAN

Penelitian ini menganalisis empat algoritma penjadwalan CPU, yaitu FCFS, SJF, Priority Scheduling, dan Round Robin menggunakan satu dataset proses yang sama. Berdasarkan hasil simulasi, SJF memberikan kinerja terbaik berdasarkan Average Waiting Time dan Average Turnaround Time dengan nilai masing-masing 9,17 dan 13,83 satuan waktu. FCFS menghasilkan Average Waiting Time 10,00 dan Average Turnaround Time 14,67, Priority Scheduling menghasilkan 11,67 dan 16,33, sedangkan Round Robin dengan *time quantum* 3 menghasilkan 13,17 dan 17,83. Meskipun SJF memberikan efisiensi waktu paling baik pada skenario penelitian, Round Robin mempunyai keunggulan dalam pemerataan akses CPU. Dengan demikian, algoritma yang paling sesuai tidak dapat ditentukan secara mutlak karena bergantung pada karakteristik proses, tujuan sistem, dan parameter kinerja yang diprioritaskan. Penelitian ini dapat menjadi dasar pembelajaran bagi mahasiswa untuk memahami pengaruh algoritma penjadwalan terhadap kinerja prosesor dan dapat dikembangkan menggunakan dataset yang lebih besar serta skenario yang lebih kompleks.

UCAPAN TERIMA KASIH

Terima kasih disampaikan kepada STIKOM Tunas Bangsa, dosen pengampu mata kuliah Sistem Operasi, serta seluruh pihak yang telah memberikan dukungan dan masukan selama proses penyusunan penelitian ini.

DAFTAR PUSTAKA

- Al Ghifari, A., & Siregar, H. (2025). Systematic literature review: Perbandingan kinerja algoritma penjadwalan CPU FCFS, SJF, Round Robin, dan Priority. *Jurnal Ilmiah Sistem Informasi dan Ilmu Komputer*, 5(2), 530–548. <https://doi.org/10.55606/juisik.v5i2.1404>
- Barman, D., Paul, B., Bhattacharya, S., De, S., & Arya, G. P. (2024). Optimized Round Robin CPU scheduling for critical processes. *Global Journal of Computer Science and Technology*, 23(H3), 13–18.
- González-Rodríguez, M., Otero-Cerdeira, L., González-Rufino, E., & Rodríguez-Martínez, F. J. (2024). Study and evaluation of CPU scheduling algorithms. *Heliyon*, 10(9), e29959. <https://doi.org/10.1016/j.heliyon.2024.e29959>
- Hajjar, O., Mekhallalati, E., Annwty, N., Alghayadh, F., et al. (2024). Performance assessment of CPU scheduling algorithms: A scenario-based approach with FCFS, RR, and SJF. *Journal of Computer Science*, 20(9), 972–985. <https://doi.org/10.3844/jcssp.2024.972.985>
- Hussein, S. A., & Abdul Kareem, E. I. (2024). A proposed CPU job-scheduling technique based on Round Robin method using dual synchronized time-slices. *Ingénierie des Systèmes d'Information*, 29(5), 1847–1858. <https://doi.org/10.18280/isi.290517>
- Indra, Z., Zidan, A. A., Silaen, A. A., Habibi, A. N., & Sitepu, K. P. (2025). Analisis komparatif dan evaluatif terhadap algoritma First-Come First-Served (FCFS) dalam penjadwalan CPU di era komputasi modern. *Jurnal Penelitian Inovatif*, 5(4), 3197–3206. <https://doi.org/10.54082/jupin.1957>
- Pan, W. (2024). Comparison and analysis of scheduling algorithms: Exploring performance, time, fairness, and applicable scenarios. *Highlights in Science, Engineering and Technology*, 120, 1–7. <https://doi.org/10.54097/4asdt07>
- Pemasinghe, S., & Rajapaksha, S. (2022). Comparison of CPU scheduling algorithms: FCFS, SJF, SRTF, Round Robin, priority based, and multilevel queuing. In *2022 IEEE 10th Region 10 Humanitarian Technology Conference (R10-HTC)* (pp. 318–323). IEEE. <https://doi.org/10.1109/R10-HTC54060.2022.9929533>
- Purnomo, R., & Putra, T. D. (2023). Comparative study: Preemptive Shortest Job First and Round Robin algorithms. *Sinkron: Jurnal dan Penelitian Teknik Informatika*, 8(2). <https://doi.org/10.33395/sinkron.v8i2.12525>
- Putra, T. D., & Purnomo, R. (2024). Simulation and modelling of pre-emptive Priority CPU Scheduling Algorithm. *Sinkron: Jurnal dan Penelitian Teknik Informatika*, 8(3), 1631–1640. <https://doi.org/10.33395/sinkron.v9i3.13352>
- Rachmad, A., Sonata, F., & Hutagalung, J. (2023). Analysis average waiting time search performance in the queue process on CPU scheduling using the Round Robin, shortest job first and first in first out algorithm. *AIP Conference Proceedings*, 2679, 020010. <https://doi.org/10.1063/5.0111352>
- Rizal, A., & Hasibuan, N. (2022). Implementasi penjadwalan CPU menggunakan algoritma First Come First Served (FCFS). *InfoTekJar: Jurnal Nasional Informatika dan Teknologi Jaringan*, 7(1). <https://doi.org/10.30743/infotekjar.v7i1.9638>
- Sidiq, W. A. B., Gani, E. U., Djuniadi, & Ardhiansyah, A. (2026). Analisis komparatif efisiensi algoritma CPU scheduling (FCFS, Round Robin, dan Priority) berbasis simulasi pada lingkungan OS Simulator. *Elkom: Jurnal Elektronika dan Komputer*, 19(1). <https://doi.org/10.51903/elkom.v19i1.3403>
- Syauqi, H. A., & Siregar, H. (2025). Systematic literature review (SLR): Dampak modifikasi algoritma Round-Robin dalam penjadwalan CPU pada sistem operasi. *Digital Transformation Technology*, 5(1), 368–374. <https://doi.org/10.47709/digitech.v5i1.6171>
- Tufegdžić, M., Jevremović, V., & Petrović, Z. (2022). Estimation of CPU scheduling algorithms efficiency using object oriented programming. In *9th International Scientific Conference Technics and Informatics in Education*. <https://doi.org/10.46793/TIE22.182T>