



Analisis Perbandingan Algoritma Penjadwalan Prioritas Preemptive dan Non-Preemptive Menggunakan Aplikasi Web Interaktif

Gilbert Aldrich Rumahorbo^{1*}, Zulfahmi Indra², Alfarizi Wijaya³, Melika Debiyana Putri⁴, Calvin Sahputra Buulolo⁵

¹⁻⁴Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Medan

Korespondensi penulis: aldrichgar@gmail.com

Abstract. CPU scheduling is a core function in modern operating systems that significantly impacts system performance and efficiency. Among various scheduling algorithms, Priority Scheduling is widely used and exists in two main variants: non-preemptive and preemptive. The non-preemptive mode allows a process to run to completion, while the preemptive mode can interrupt a running process for a higher-priority one. Understanding the behavioral differences between these modes is crucial but often challenging through manual calculations. To address this, an interactive web-based application was developed to simulate and visualize both preemptive and non-preemptive Priority Scheduling algorithms. The research method involved designing the system logic based on the core principles of each scheduling variant, followed by implementation using standard web technologies: HTML, Tailwind CSS, and JavaScript. The application allows users to input custom process data or load predefined case studies, select the scheduling mode, and instantly receive a comprehensive analysis. The results include a dynamically generated Gantt chart, a detailed performance metrics table (including turnaround time and waiting time), and a step-by-step execution log. Through a comparative analysis of a specific case study, the application is proven to be an effective educational tool. It accurately simulates both modes and visually demonstrates the impact of preemption on execution order, resource utilization, and key performance metrics, thereby simplifying the learning process for students and educators.

Keywords: CPU scheduling, non-preemptive, preemptive, priority scheduling, web application.

Abstrak. Penjadwalan CPU adalah fungsi inti dalam sistem operasi yang secara signifikan memengaruhi kinerja dan efisiensi sistem. Di antara berbagai algoritma penjadwalan, Penjadwalan Prioritas (Priority Scheduling) banyak digunakan dan hadir dalam dua varian utama: non-preemptive dan preemptive. Mode non-preemptive memungkinkan sebuah proses berjalan hingga selesai, sementara mode preemptive dapat menginterupsi proses yang sedang berjalan untuk proses lain dengan prioritas lebih tinggi. Memahami perbedaan perilaku antara kedua mode ini sangat penting, namun seringkali menjadi tantangan jika dilakukan melalui perhitungan manual. Untuk mengatasi masalah ini, sebuah aplikasi web interaktif dikembangkan untuk mensimulasikan dan memvisualisasikan algoritma Priority Scheduling, baik mode preemptive maupun non-preemptive. Metode penelitian yang digunakan meliputi perancangan logika sistem berdasarkan prinsip inti dari setiap varian penjadwalan, yang diikuti dengan implementasi menggunakan teknologi web standar: HTML, Tailwind CSS, dan JavaScript. Aplikasi ini memungkinkan pengguna memasukkan data proses kustom atau memuat studi kasus yang telah ditentukan, memilih mode penjadwalan, dan secara instan menerima analisis komprehensif. Hasil yang ditampilkan mencakup Gantt chart yang dibuat secara dinamis, tabel metrik kinerja terperinci (termasuk turnaround time dan waiting time), dan log eksekusi langkah demi langkah. Melalui analisis komparatif dari sebuah studi kasus spesifik, aplikasi ini terbukti menjadi alat bantu pembelajaran yang efektif. Aplikasi ini secara akurat mensimulasikan kedua mode dan menunjukkan secara visual dampak preemption pada urutan eksekusi, utilisasi sumber daya, dan metrik kinerja utama, sehingga menyederhanakan proses pembelajaran bagi mahasiswa dan pengajar.

Kata kunci: aplikasi web, non-preemptive, penjadwalan CPU, penjadwalan prioritas, preemptive.

1. LATAR BELAKANG

Dalam sistem komputasi modern, sistem operasi memegang peranan krusial sebagai manajer sumber daya yang mengatur seluruh komponen perangkat keras dan

perangkat lunak (Sari, 2018). Salah satu fungsi paling fundamental dari sebuah sistem operasi adalah penjadwalan Central Processing Unit (CPU), yang merupakan mekanisme untuk memutuskan proses mana yang akan dieksekusi oleh CPU dari sekian banyak proses yang berada dalam antrian siap (ready queue) (Mutasar, 2020). Penjadwalan yang efisien sangat penting untuk memaksimalkan utilitas CPU, meningkatkan throughput, dan memberikan waktu respons yang cepat kepada pengguna (Azzam & Siregar, 2025).

Berbagai algoritma penjadwalan telah dikembangkan, salah satunya adalah Priority Scheduling atau Penjadwalan Berbasis Prioritas. Algoritma ini bekerja dengan memberikan nilai prioritas pada setiap proses, di mana proses dengan prioritas tertinggi akan dieksekusi terlebih dahulu (Sari, 2018; Yanwari et al., 2023). Algoritma ini sangat relevan dalam berbagai implementasi, mulai dari sistem help desk hingga penjadwalan produksi, di mana tugas-tugas tertentu memerlukan penanganan yang lebih mendesak (Clarín, 2023; Akbar et al., 2015). Priority Scheduling sendiri terbagi menjadi dua varian utama: non-preemptive dan preemptive (Mutasar, 2020; Nasution et al., 2023). Dalam mode non-preemptive, sebuah proses yang telah mendapatkan alokasi CPU akan terus berjalan hingga selesai tanpa bisa diinterupsi (Suratun et al., 2025). Sebaliknya, dalam mode preemptive, proses yang sedang berjalan dapat dihentikan sementara (di-preempt) jika ada proses baru dengan prioritas lebih tinggi yang masuk ke dalam antrian (Putra & Purnomo, 2024).

Meskipun konsep Priority Scheduling tampak sederhana, memahami perbedaan perilaku dan implikasi kinerja antara kedua mode tersebut seringkali menjadi tantangan, terutama bagi mahasiswa yang baru mempelajari sistem operasi. Perhitungan manual untuk melacak alur eksekusi, waktu tunggu (waiting time), dan waktu penyelesaian (turnaround time) sangat rumit, memakan waktu, dan rentan terhadap kesalahan, khususnya pada skenario preemptive yang dinamis (Putra & Purnomo, 2022). Ketiadaan alat bantu interaktif yang dapat memvisualisasikan dan membandingkan kedua mode secara langsung menjadi sebuah celah (gap) dalam proses pembelajaran. Banyak penelitian telah membuktikan bahwa simulasi dan visualisasi merupakan metode yang efektif untuk memahami konsep-konsep kompleks dalam ilmu komputer (Hariyanto, 2005; Silberschatz et al., 2009).

Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun sebuah aplikasi web interaktif yang dapat menyimulasikan, menganalisis, dan memvisualisasikan algoritma Priority Scheduling dalam kedua modenya. Aplikasi ini dirancang sebagai sebuah alat bantu pembelajaran yang komprehensif, yang memungkinkan pengguna untuk secara dinamis membandingkan hasil dari kedua pendekatan dan memahami dampaknya melalui visualisasi Gantt chart dan rincian perhitungan yang dihasilkan secara otomatis.

2. KAJIAN TEORITIS

A. Sistem Operasi dan Penjadwalan CPU

Sistem operasi adalah sebuah program yang bertindak sebagai perantara antara pengguna komputer dan perangkat keras komputer (Sari, 2018). Salah satu tugas utamanya adalah manajemen proses, di mana penjadwalan CPU menjadi komponen sentral. Penjadwalan CPU adalah proses mengalokasikan sumber daya CPU untuk menyelesaikan tugas (Akbar et al., 2015). Tujuannya adalah untuk membuat sistem menjadi efisien, adil, dan cepat dalam merespons (Hariyanto, 2005). Ada banyak algoritma yang telah dikembangkan, seperti First-Come, First-Served (FCFS), Shortest

Job First (SJF), dan Round Robin (RR), yang masing-masing memiliki kelebihan dan kekurangan (Azzam & Siregar, 2025; Hariyanto, 2005).

B. Algoritma Priority Scheduling

Priority Scheduling adalah algoritma penjadwalan di mana setiap proses diberi nomor prioritas (integer). CPU dialokasikan untuk proses dengan prioritas tertinggi (biasanya nilai integer terkecil) (Sari, 2018). Jika beberapa proses memiliki prioritas yang sama, maka akan digunakan aturan FCFS (Suratun et al., 2025; Fitri et al., 2016). Algoritma ini sangat efektif diimplementasikan dalam berbagai sistem, seperti penjadwalan produksi pakaian, sistem pelaporan kerusakan, hingga penjadwalan tenaga keperawatan, karena kemampuannya mengakomodasi tugas-tugas dengan tingkat urgensi berbeda (Budiyantoro & Kerlooza, 2018; Kustanto et al., 2024; Harihayati et al., 2025).

a. Priority Scheduling Non-Preemptive

Pada varian ini, sekali CPU dialokasikan ke suatu proses, maka proses tersebut akan tetap memakai CPU sampai selesai (Hariyanto, 2005). Proses baru yang datang, meskipun memiliki prioritas lebih tinggi, tidak dapat mengganggu proses yang sedang berjalan, melainkan hanya diletakkan di bagian depan antrian (head of queue) (Suratun et al., 2025; Mutasar & Niesa, 2021).

b. Priority Scheduling Preemptive

Pada varian ini, jika ada proses baru yang datang dengan prioritas lebih tinggi daripada proses yang sedang dijalankan, maka proses yang sedang berjalan tersebut dihentikan, dan CPU dialihkan untuk proses yang baru datang tersebut (Sari, 2018; Mutasar, 2020). Proses yang diinterupsi akan kembali ke ready queue untuk menunggu gilirannya lagi (Putra & Purnomo, 2024).

C. Dynamic Priority Scheduling

Beberapa penelitian mengembangkan konsep Priority Scheduling menjadi dinamis (Dynamic Priority Scheduling), di mana prioritas sebuah proses dapat berubah seiring waktu berdasarkan aturan-aturan tertentu (Nasution et al., 2023; Masyfa et al., 2023). Salah satu teknik yang umum adalah aging, yaitu meningkatkan prioritas dari setiap proses yang menunggu dalam antrian secara bertahap untuk mencegah terjadinya indefinite blocking atau starvation (Mutasar, 2020).

D. Metrik Evaluasi Kinerja

Untuk mengevaluasi performa algoritma penjadwalan, beberapa kriteria digunakan (Hariyanto, 2005; Sari, 2018):

- 1) CPU Utilization: Menjaga agar CPU selalu dalam keadaan sibuk.
- 2) Throughput: Banyaknya proses yang selesai dikerjakan dalam satu satuan waktu.
- 3) Turnaround Time (TAT): Waktu yang dibutuhkan untuk mengeksekusi proses secara keseluruhan, dari awal hingga selesai ($TAT = Completion\ Time - Arrival\ Time$).
- 4) Waiting Time (WT): Waktu yang dihabiskan proses untuk menunggu di ready queue ($WT = TAT - Burst\ Time$).
- 5) Response Time: Waktu dari saat sebuah permintaan dikirimkan hingga respons pertama diterima.

3. METODE PENELITIAN

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah System Development Life Cycle (SDLC) dengan model waterfall. Pendekatan ini dipilih karena memiliki tahapan yang jelas dan terstruktur, mulai dari analisis kebutuhan hingga pengujian sistem (Muhammadiyah & Komul, 2025; Kustanto et al., 2024).

Analisis Kebutuhan

Tahap ini berfokus pada identifikasi kebutuhan fungsional dan non-fungsional dari aplikasi yang akan dibangun. Kebutuhan utamanya adalah sebuah sistem yang mampu:

- a. Menerima input data proses (Arrival Time, Burst Time, Priority).
- b. Menyediakan opsi untuk memilih mode penjadwalan (Non-Preemptive atau Preemptive).
- c. Menampilkan hasil simulasi dalam bentuk Gantt Chart yang visual dan mudah dibaca.
- d. Menampilkan tabel hasil perhitungan yang memuat metrik kinerja (CT, TAT, WT) untuk setiap proses.
- e. Menampilkan jejak atau langkah-langkah perhitungan secara detail untuk tujuan edukasi.

Perancangan Sistem

Aplikasi ini dirancang sebagai sebuah single-page application (SPA) untuk memberikan pengalaman pengguna yang mulus dan responsif. Desain antarmuka pengguna dibagi menjadi dua area utama: panel kontrol untuk input data dan pengaturan di sebelah kiri, dan panel hasil untuk menampilkan visualisasi dan data di sebelah kanan. Logika algoritma untuk kedua mode dijabarkan dalam bentuk flowchart untuk memastikan implementasi yang akurat.

Implementasi

Aplikasi ini diimplementasikan menggunakan teknologi web standar yang bersifat client-side untuk memastikan portabilitas dan kemudahan akses tanpa memerlukan instalasi khusus.

- a. HTML: Digunakan untuk membangun struktur kerangka dan elemen-elemen halaman web.
- b. Tailwind CSS: Sebuah framework CSS modern yang digunakan untuk merancang antarmuka pengguna yang estetis dan responsif.
- c. JavaScript: Menjadi inti dari aplikasi, digunakan untuk mengimplementasikan seluruh logika fungsional. Dua fungsi utama dikembangkan:

Calculatenonpreemptive(): Fungsi ini mengimplementasikan algoritma non-preemptive dengan logika memilih proses prioritas tertinggi dari ready queue dan menjalankannya hingga selesai.

Calculatepreemptive(): Fungsi ini mengimplementasikan algoritma preemptive. Simulasi berjalan per satuan waktu, di mana pada setiap satuan waktu, fungsi akan mengevaluasi kembali ready queue. Jika ada proses baru dengan prioritas lebih tinggi, proses yang sedang berjalan akan diinterupsi.

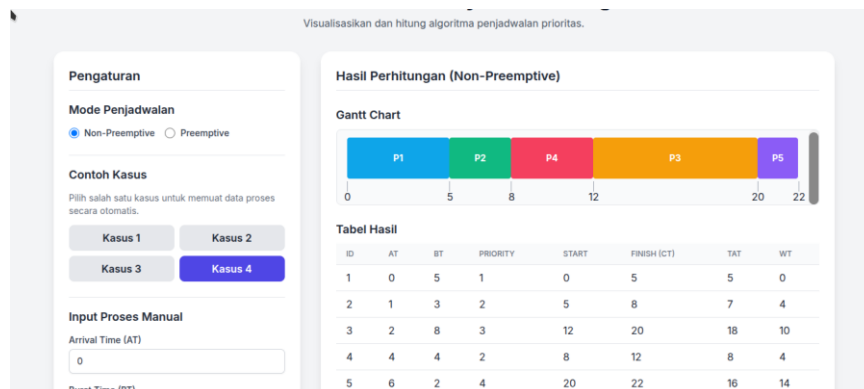
Pengujian

Pengujian fungsionalitas sistem dilakukan menggunakan metode black-box testing (Muhammadiyah & Komul, 2025; Rumampuk et al., 2025). Pengujian ini berfokus pada verifikasi output berdasarkan input yang diberikan. Empat contoh kasus yang sudah ada di dalam aplikasi digunakan sebagai skenario pengujian untuk memastikan akurasi perhitungan pada kedua mode. Hasil yang dikeluarkan oleh aplikasi kemudian divalidasi dengan membandingkannya terhadap perhitungan manual yang telah dilakukan sebelumnya.

4. HASIL DAN PEMBAHASAN

Untuk mendemonstrasikan kapabilitas aplikasi dan melakukan analisis perbandingan antara mode non-preemptive dan preemptive, digunakan data dari Kasus

4 yang telah disematkan di dalam aplikasi. Kasus ini ideal untuk analisis karena mencakup proses-proses dengan waktu kedatangan dan tingkat prioritas yang bervariasi.



Data Input (Kasus 4):

| ID Proses | Arrival Time (AT) | Burst Time (BT) | Priority |
|-----------|-------------------|-----------------|----------|
| P1 | 0 | 5 | 1 |
| P2 | 1 | 3 | 2 |
| P3 | 2 | 8 | 3 |
| P4 | 4 | 4 | 2 |
| P5 | 6 | 2 | 4 |

(Catatan: Sesuai dengan implementasi pada aplikasi dan konvensi umum, nilai prioritas yang lebih kecil menunjukkan prioritas yang lebih tinggi)

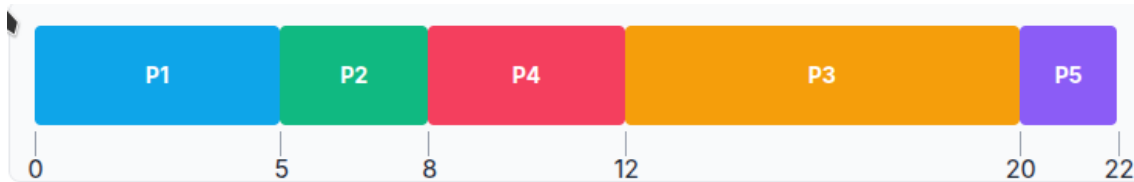
Analisis Mode Non-Preemptive

Dalam mode non-preemptive, proses yang sedang dieksekusi akan berjalan hingga selesai tanpa interupsi. Berdasarkan data Kasus 4, aplikasi menghasilkan urutan eksekusi dan perhitungan sebagai berikut:

- a. $t=0$: Hanya P1 (Prio: 1) yang berada di ready queue. P1 dieksekusi hingga selesai pada $t=5$.
- b. $t=5$: P1 selesai. Proses yang sudah tiba adalah P2 (AT=1, Prio: 2), P3 (AT=2, Prio: 3), dan P4 (AT=4, Prio: 2). Di antara P2 dan P4 yang memiliki prioritas sama (2), P2 dieksekusi terlebih dahulu karena memiliki arrival time yang lebih awal (aturan FCFS). P2 berjalan hingga selesai pada $t=8$.
- c. $t=8$: P2 selesai. Proses yang tersedia di ready queue adalah P3, P4, dan P5 (yang tiba pada $t=6$). P4 (Prio: 2) dipilih karena memiliki prioritas tertinggi. P4 berjalan hingga selesai pada $t=12$.
- d. $t=12$: P3 (Prio: 3) dipilih untuk dieksekusi. P3 berjalan hingga selesai pada $t=20$.
- e. $t=20$: P5 (Prio: 4) sebagai proses terakhir dieksekusi dan selesai pada $t=22$.

B. Gantt Chart (Non-Preemptive):

Tabel Hasil (Non-Preemptive):



Tabel Hasil (Non-Preemptive):

| ID | AT | BT | Priority | Start | Finish (CT) | TAT | WT |
|----|----|----|----------|-------|-------------|-----|----|
| P1 | 0 | 5 | 1 | 0 | 5 | 5 | 0 |
| P2 | 1 | 3 | 2 | 5 | 8 | 7 | 4 |
| P3 | 2 | 8 | 3 | 12 | 20 | 18 | 10 |
| P4 | 4 | 4 | 2 | 8 | 12 | 8 | 4 |
| P5 | 6 | 2 | 4 | 20 | 22 | 16 | 14 |

Average Turnaround Time (ATAT): $(5+7+18+8+16) / 5 = 10.80$

Average Waiting Time (AWT): $(0+4+10+4+14) / 5 = 6.40$

Analisis Mode Preemptive

Dalam mode preemptive, proses yang sedang berjalan dapat diinterupsi oleh proses baru dengan prioritas yang lebih tinggi.

- a. $t=0$: P1 (Prio: 1) tiba dan mulai dieksekusi.
- b. $t=1$: P2 (Prio: 2) tiba. Prioritas P1 (1) lebih tinggi dari P2, sehingga P1 terus berjalan.
- c. $t=2$: P3 (Prio: 3) tiba. Prioritas P1 (1) masih yang tertinggi, P1 terus berjalan.
- d. $t=4$: P4 (Prio: 2) tiba. Prioritas P1 (1) masih yang tertinggi, P1 terus berjalan.
- e. $t=5$: P1 selesai. Proses di ready queue: P2 (Prio: 2), P3 (Prio: 3), P4 (Prio: 2). Antara P2 dan P4, P2 dipilih karena tiba lebih dulu. P2 mulai berjalan.
- f. $t=6$: P5 (Prio: 4) tiba. Prioritas P2 (2) lebih tinggi dari P5, sehingga P2 terus berjalan.
- g. $t=8$: P2 selesai. Proses yang tersisa: P3 (Prio: 3), P4 (Prio: 2), P5 (Prio: 4). P4 dipilih karena prioritasnya tertinggi (2).
- h. $t=12$: P4 selesai. P3 (Prio: 3) dieksekusi selanjutnya.
- i. $t=20$: P3 selesai. P5 (Prio: 4) dieksekusi terakhir.
- j. $t=22$: Semua proses selesai.

Pada studi kasus spesifik ini, tidak ada momen di mana proses yang baru datang memiliki prioritas lebih tinggi daripada proses yang sedang berjalan. Akibatnya, tidak terjadi preemption, dan urutan eksekusi serta hasilnya identik dengan mode non-preemptive.

Pembahasan Komparatif dan Nilai Edukatif Aplikasi

Aplikasi web yang dikembangkan berhasil menyimulasikan kedua mode algoritma dengan akurat, yang divalidasi melalui fitur "Langkah-langkah Perhitungan" dan perbandingan dengan perhitungan manual. Kemampuan untuk beralih antara mode

non-preemptive dan preemptive dengan satu klik adalah fitur utama yang memberikan nilai edukatif.

Meskipun pada Kasus 4 hasilnya sama, nilai pembelajaran terletak pada kemampuan pengguna untuk bereksperimen. Pengguna (mahasiswa) dapat dengan mudah mengubah parameter, misalnya dengan memberikan prioritas sangat tinggi (contoh: 0) pada P5 yang datang belakangan. Dalam skenario tersebut, aplikasi akan secara visual menunjukkan pada Gantt chart bagaimana P4 (yang berjalan dari $t=8$) akan di-preempt oleh P5. Hal ini secara langsung mengilustrasikan konsep preemption dan dampaknya pada metrik kinerja seperti waiting time P4 yang menjadi lebih panjang. Dengan demikian, aplikasi ini berfungsi sebagai "laboratorium virtual" yang memfasilitasi pembelajaran aktif dan pemahaman konseptual yang lebih mendalam, yang sulit dicapai hanya dengan teori dan perhitungan di atas kertas (Pradana et al., 2023; Kusuma, 2020; Yanwari et al., 2023).

5. KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan hasil penelitian, pengembangan, dan analisis, dapat ditarik beberapa kesimpulan:

- a. Aplikasi web interaktif untuk simulasi algoritma Priority Scheduling telah berhasil dikembangkan dengan fungsionalitas penuh untuk mode preemptive dan non-preemptive, sesuai dengan tujuan penelitian.
- b. Aplikasi ini terbukti akurat dalam menghitung metrik kinerja (TAT dan WT) dan menyajikan visualisasi Gantt Chart yang dinamis, yang telah divalidasi menggunakan data studi kasus dan perhitungan manual.
- c. Sebagai alat bantu pembelajaran, aplikasi ini sangat efektif. Fitur perbandingan mode, dikombinasikan dengan log perhitungan langkah demi langkah, memberikan platform yang kuat bagi mahasiswa untuk bereksperimen dan memahami perbedaan konseptual serta dampak praktis antara kedua varian algoritma.

Saran

Untuk pengembangan di masa depan, beberapa peningkatan dapat dipertimbangkan untuk memperkaya fungsionalitas dan nilai edukatif aplikasi:

- a. Mengintegrasikan algoritma penjadwalan CPU lainnya (misalnya, FCFS, SJF, Round Robin) untuk menciptakan sebuah platform toolkit penjadwalan yang komprehensif (Azzam & Siregar, 2025).
- b. Menambahkan fitur visualisasi context switch pada Gantt Chart untuk secara eksplisit menandai kapan preemption terjadi, sehingga memberikan pemahaman yang lebih jelas.
- c. Mengembangkan "Mode Perbandingan" di mana hasil (Gantt Chart dan tabel) dari kedua mode ditampilkan secara berdampingan dalam satu layar untuk analisis komparatif yang lebih mudah dan langsung.

DAFTAR REFERENSI

- Afrianto, M. I., Fauziah, F., & Wijaya, Y. F. (2024). Kombinasi Algoritma Priority Scheduling dan Earliest Due Date untuk Sistem Penjadwalan Slitting Produk Berbasis Web. *TEKNOKOM: Jurnal Teknologi dan Rekayasa Sistem Komputer*, 7(1), 180–186

- Akbar, R. J., Setyawan, H. B., & Kusumawati, W. I. (2015). Rancang Bangun Sistem Penjadwalan Produksi Menggunakan Aturan Prioritas Pada PT. Unichem Candi Indonesia. *JSIKA*, 4(2), 1–8.
- Azzam, A. I., & Siregar, H. (2025). Analisis Perbandingan Algoritma Penjadwalan CPU pada Sistem Operasi Linux. *Remik: Riset dan E-Jurnal Manajemen Informatika Komputer*, 9(3), 818–825.
- Budiyantoro, & Kerlooza, Y. Y. (2018). Strategi Penjadwalan Produksi Pakaian Metode Quantum-Based Dan Preemptive Priority Scheduling. *Jurnal Tata Kelola dan Kerangka Kerja Teknologi Informasi*, 4(2), 77–83.
- Clarín, J. A. (2023). Priority-Based Scheduling Algorithm for Help Desk Support System. *International Journal of Intelligent Systems and Applications in Engineering*, 11(4), 299–307.
- Fitri, A., Permana, I., & Marsal, A. (2016). Penerapan Constraint Satisfaction Problem pada metode Priority Scheduling untuk Penjadwalan Khutbah Jum'at para Mubaligh di IKMI Pekanbaru. *Jurnal Sains. Teknologi dan Industri*, 13(2), 190–194.
- Harihayati, T., Lubis, R., & Abdulloh, M. (2025). Desain Sistem Penjadwalan Tenaga Keperawatan Rumah Sakit Menggunakan Algoritma Priority Scheduling. *KOMPUTA: Jurnal Ilmiah Komputer dan Informatika*, 14(1), 108–119.
- Hariyanto, B. (2005). *Sistem Operasi*. Informatika.
- Kusuma, I. G. N. A. (2020). Priority Based Scheduling untuk Peningkatan Efisiensi Konsumsi Energi pada Jaringan Sensor Nirkabel. *Jurnal Teknologi Informasi dan Komputer*, 6(3), 288–294.
- Kustanto, P., Kurniawan, W. A., Noe'man, A., Nurfiyah, & Ridwan. (2024). Implementation of Priority Scheduling Algorithm in Designing Infrastructure Damage Reporting System. *Greenation International Journal of Engineering Science*, 2(3), 98–112.
- Masyfa, F. H., Kartikasari, D. P., & Tibyani. (2023). Penjadwalan dan Pelaporan Menggunakan Dynamic Priority Scheduling dan Geolocation untuk Keamanan Lingkungan. *Techno.COM*, 22(1), 195–206.
- Muhammadiyah, N., & Komul, T. M. P. (2025). Peningkatan Layanan IT Melalui Optimasi Aplikasi Android Pelaporan Keluhan dengan Konsep Algoritma Priority Scheduling: Studi Kasus PT Infomedia Nusantara. *TEKINFO*, 26(1), 58–65.
- Mutasar. (2020). Optimasi Alokasi Waktu Proses CPU pada Linux dengan Algoritma Priority Scheduling. *Jurnal Elektronika dan Teknologi Informasi*, 1(2), 14–16.
- Mutasar, & Niesa, C. (2021). Optimasi Basis Data Terdistribusi dengan Algoritma Priority Scheduling. *Jurnal TIKFA Fakultas Ilmu Komputer Universitas Almuslim*, 06(02), 141–151.
- Nasution, M. F. A., Suendri, & Harahap, A. M. (2023). Customer Service Information System Using Dynamic Priority Scheduling Algorithm At PT Sumatra Sistem Integrasi. *AIRA Journal of Information System and Technology Research*, 2(1), 25–37.
- Pradana, M. I., Trisnawan, P. H., & Siregar, R. A. (2023). Implementasi Priority Packet Scheduling di Sisi Node Pengirim pada Mobile Ad-hoc Network berdasarkan Ukuran Paket Data. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 7(3), 1156–1162.

- Putra, T. D., & Purnomo, R. (2022). Simulation of Priority Round-Robin Scheduling Algorithm. *Sinkron: Jurnal dan Penelitian Teknik Informatika*, 6(4), 2170–2181.
- Putra, T. D., & Purnomo, R. (2023). Average Max Round Robin Algorithm: A Case Study. *Sinkron: Jurnal dan Penelitian Teknik Informatika*, 7(4), 2662–2669.
- 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.
- Rumampuk, V. G., Ardiansyah, R., Joeffie, Y. Y., Laila, R., & Pusadan, M. Y. (2025). Design and Build an Online Reservation System for Health Services at Pet Clinics Using the Priority Scheduling Algorithm. *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 10(2), 1482–1490.
- Sari, M. W. (2018). *Sistem Operasi*. UPY Press.
- Silberschatz, A., Galvin, P. B., & Gagne, G. (2009). *Operating System Concepts* (8th ed.). John Wiley & Sons, Inc.
- Suratun, Ginting, N. B., Syarif, Z. N., & Gofur, R. A. (2025). Application of Non-Preemptive Priority Scheduling Method for Work Order Scheduling System. *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, 10(3), 698–709.
- Yanwari, M., Prabuwo, A. S., Yudiantoro, T. R., Aji, N. B., Wiktasari, & Handoko, S. (2023). Priority Scheduling Implementation for Exam Schedule. *Indonesian Journal of Information Systems (IJIS)*, 5(2), 80–89.