



Evaluasi Algoritma Penjadwalan Proses: Perbandingan *Round Robin*, *Priority Scheduling*, dan *Multilevel Queue*

Fatur Rahman Triadi¹, Teti Desyani², Damar Bramantyo³, Dwi Ferdiansyah⁴, Hadayya Imtiyaz Suffit⁵, Riko Aria Putra⁶

^{1,2,3,4,5,6} Program Studi Teknik Informatika, Universitas Pamulang

faturtriadi01@gmail.com¹, dosen00839@unpam.ac.id², damararakaan5@gmail.com³, frdnsyh425@gmail.com⁴, hadavyaimtiyaz@gmail.com⁵, rikoarya287@gmail.com⁶

Abstrak

Penjadwalan proses merupakan salah satu aspek penting dalam sistem operasi karena berfungsi mengatur urutan eksekusi proses pada CPU agar sumber daya komputasi dapat dimanfaatkan secara optimal. Penelitian ini bertujuan untuk mengevaluasi dan membandingkan kinerja tiga algoritma penjadwalan proses, yaitu *Round Robin*, *Priority Scheduling*, dan *Multilevel Queue*, melalui pendekatan simulasi berbasis *Python*. Evaluasi dilakukan pada tiga skenario beban kerja, yaitu beban kerja seimbang, dominasi proses berprioritas tinggi, dan variasi *burst time* yang signifikan. Setiap skenario diuji menggunakan sejumlah proses dengan atribut *arrival time*, *burst time*, prioritas, dan level antrian. Parameter kinerja yang dianalisis meliputi rata-rata *turnaround time*, *waiting time*, *response time*, *throughput*, serta utilisasi CPU. Hasil penelitian menunjukkan bahwa *Priority Scheduling* memiliki performa paling efisien dalam menurunkan *turnaround time* dan *waiting time* karena proses dengan prioritas tinggi dieksekusi lebih awal. Sebaliknya, *Round Robin* unggul pada aspek *response time* karena mekanisme *time quantum* mampu memberikan kesempatan eksekusi awal secara lebih merata kepada setiap proses. Sementara itu, *Multilevel Queue* menunjukkan kinerja yang relatif seimbang karena membagi proses ke dalam beberapa antrian berdasarkan karakteristiknya, sehingga mampu mengakomodasi kebutuhan efisiensi dan keadilan. Berdasarkan hasil tersebut, dapat disimpulkan bahwa tidak terdapat satu algoritma yang paling optimal untuk seluruh kondisi. Pemilihan algoritma penjadwalan harus disesuaikan dengan karakteristik beban kerja, kebutuhan sistem, serta prioritas proses yang dijalankan secara keseluruhan.

Kata Kunci: Penjadwalan Proses, *Round Robin*, *Priority Scheduling*, *Multilevel Queue*, Simulasi Cpu.

1. Pendahuluan

Sistem operasi modern memiliki peran penting dalam mengelola berbagai proses yang berjalan secara bersamaan agar sumber daya komputer dapat dimanfaatkan secara optimal. Dalam lingkungan komputasi yang mendukung *multitasking* dan *multiprogramming*, CPU harus mampu menentukan proses mana yang memperoleh giliran eksekusi pada waktu tertentu. Mekanisme tersebut dikenal sebagai penjadwalan proses CPU, yaitu fungsi utama sistem operasi dalam mengatur alokasi sumber daya pemrosesan. Penjadwalan yang baik dapat meningkatkan efisiensi sistem, mempercepat penyelesaian proses, serta menjaga kestabilan kinerja komputer secara keseluruhan. Oleh karena itu, algoritma penjadwalan menjadi aspek penting dalam kajian sistem operasi modern (Carrasco-Ruiz et al., 2024).

Penjadwalan proses menjadi semakin penting karena setiap proses memiliki karakteristik yang berbeda-beda. Perbedaan tersebut dapat berupa waktu kedatangan atau *arrival time*, lama eksekusi atau *burst time*, tingkat prioritas, serta kebutuhan respons sistem. Kondisi ini menuntut sistem operasi untuk menggunakan algoritma penjadwalan yang sesuai dengan jenis beban kerja yang sedang diproses. Apabila algoritma yang digunakan kurang tepat, maka sistem dapat mengalami peningkatan waktu tunggu, rendahnya respons terhadap proses interaktif, atau ketidakseimbangan distribusi waktu CPU. Dengan demikian, pemilihan algoritma penjadwalan perlu dilakukan secara cermat berdasarkan karakteristik proses yang dijalankan (Al Ghifari & Siregar, 2025).

Secara umum, tujuan utama penjadwalan CPU adalah memaksimalkan utilisasi CPU, meningkatkan *throughput*, serta meminimalkan *turnaround time*, *waiting time*, dan *response time*. *Turnaround time* merupakan total waktu yang dibutuhkan suatu proses sejak masuk ke sistem hingga proses tersebut selesai dieksekusi. *Waiting time* menunjukkan lamanya proses berada dalam antrian sebelum atau selama menunggu giliran eksekusi CPU. Sementara itu, *response time* menggambarkan waktu yang dibutuhkan sistem untuk memberikan respons pertama terhadap suatu proses. Metrik-metrik tersebut menjadi indikator penting dalam menilai efektivitas algoritma penjadwalan proses (Omar et al., 2021).

Dalam kajian sistem operasi, terdapat berbagai algoritma penjadwalan yang dikembangkan untuk mengatur eksekusi proses secara efisien. Beberapa algoritma yang umum digunakan antara lain *First Come First Serve* atau *FCFS*, *Shortest Job First* atau *SJF*, *Round Robin*, *Priority Scheduling*, dan *Multilevel Queue*. Setiap algoritma memiliki pendekatan yang berbeda dalam menentukan urutan proses yang akan dieksekusi oleh CPU. Perbedaan

Evaluasi Algoritma Penjadwalan Proses: Perbandingan *Round Robin*, *Priority Scheduling*, dan *Multilevel Queue*

pendekatan tersebut menyebabkan setiap algoritma memiliki kelebihan dan kekurangan masing-masing. Oleh sebab itu, tidak semua algoritma dapat memberikan kinerja optimal pada seluruh kondisi beban kerja (Pemasinghe & Rajapaksha, 2022).

Salah satu algoritma penjadwalan yang banyak digunakan pada sistem *time-sharing* adalah *Round Robin*. Algoritma ini bekerja dengan memberikan jatah waktu eksekusi yang sama kepada setiap proses melalui interval tertentu yang disebut *time quantum*. Setelah jatah waktu tersebut habis, proses yang belum selesai akan dikembalikan ke antrian untuk menunggu giliran eksekusi berikutnya. Mekanisme ini membuat *Round Robin* dianggap adil karena setiap proses memperoleh kesempatan menggunakan CPU secara bergantian. Dengan demikian, algoritma ini banyak digunakan pada sistem yang membutuhkan pembagian waktu CPU secara merata (Putra & Purnomo, 2021).

Keunggulan utama *Round Robin* terletak pada kemampuannya memberikan respons awal yang relatif cepat kepada setiap proses. Hal ini terjadi karena setiap proses memiliki peluang untuk segera mendapatkan giliran eksekusi sesuai urutan antrian. Oleh karena itu, *Round Robin* cocok diterapkan pada sistem interaktif yang membutuhkan respons cepat dari CPU. Namun, kinerja algoritma ini sangat dipengaruhi oleh pemilihan nilai *time quantum*. Jika nilai *time quantum* terlalu kecil, maka sistem dapat mengalami terlalu banyak *context switch*, sedangkan jika terlalu besar, algoritma ini dapat berperilaku mendekati *FCFS* dan menurunkan efektivitas respons sistem (Omar et al., 2021).

Selain *Round Robin*, algoritma *Priority Scheduling* juga banyak digunakan karena mampu mengatur eksekusi proses berdasarkan tingkat kepentingannya. Pada algoritma ini, proses dengan prioritas lebih tinggi akan dieksekusi lebih dahulu dibandingkan proses dengan prioritas lebih rendah. Pendekatan tersebut sangat berguna pada sistem yang memiliki proses kritis atau proses yang membutuhkan penyelesaian lebih cepat. Dengan mekanisme ini, *Priority Scheduling* dapat membantu menurunkan *turnaround time* dan *waiting time* pada proses-proses yang memiliki prioritas tinggi. Oleh karena itu, algoritma ini sering digunakan pada sistem yang menekankan penyelesaian proses penting secara lebih cepat (Widiarto, 2024).

Meskipun memiliki keunggulan dalam aspek efisiensi, *Priority Scheduling* juga memiliki kelemahan yang perlu diperhatikan. Salah satu kelemahan utamanya adalah potensi terjadinya *starvation*, yaitu kondisi ketika proses berprioritas rendah menunggu terlalu lama karena terus didahului oleh proses berprioritas tinggi. Keadaan tersebut dapat menurunkan aspek keadilan dalam sistem penjadwalan. Untuk mengatasi permasalahan ini, dapat diterapkan mekanisme *aging*, yaitu peningkatan prioritas proses secara bertahap berdasarkan lamanya waktu tunggu. Dengan mekanisme tersebut, proses berprioritas rendah tetap memiliki peluang untuk dieksekusi setelah menunggu dalam waktu tertentu (Abu-Dalbouh, 2022).

Algoritma lain yang memiliki pendekatan lebih terstruktur adalah *Multilevel Queue*. Algoritma ini membagi proses ke dalam beberapa antrian berdasarkan karakteristik tertentu, seperti proses sistem, proses interaktif, dan proses *batch*. Setiap antrian dapat menggunakan algoritma penjadwalan yang berbeda sesuai kebutuhan proses di dalamnya. Pendekatan ini membuat *Multilevel Queue* lebih fleksibel dalam menghadapi variasi karakteristik proses. Dengan demikian, algoritma ini dapat digunakan untuk mengelola beban kerja campuran secara lebih adaptif dibandingkan algoritma tunggal (Harki, 2024).

Pada penerapannya, *Multilevel Queue* dapat menggabungkan keunggulan beberapa algoritma penjadwalan dalam satu mekanisme. Misalnya, antrian prioritas tinggi dapat menggunakan *Priority Scheduling*, antrian menengah dapat menggunakan *Round Robin*, sedangkan antrian rendah dapat menggunakan *FCFS*. Kombinasi tersebut memungkinkan sistem memberikan perlakuan berbeda sesuai karakteristik dan kebutuhan setiap proses. Dengan cara ini, sistem dapat mempertahankan efisiensi tanpa sepenuhnya mengabaikan aspek keadilan. Oleh karena itu, *Multilevel Queue* sering dipandang sebagai algoritma yang mampu menyeimbangkan efisiensi, respons, dan pembagian layanan CPU (Meing & Tandingan, 2024).

Berbagai penelitian sebelumnya menunjukkan bahwa tidak terdapat satu algoritma penjadwalan yang selalu unggul pada seluruh kondisi. *Round Robin* umumnya memiliki keunggulan pada aspek *response time*, tetapi dapat menghasilkan *waiting time* dan *turnaround time* yang lebih tinggi ketika terdapat variasi *burst time* yang besar. Sebaliknya, *Priority Scheduling* dapat mempercepat penyelesaian proses yang memiliki prioritas tinggi, tetapi berpotensi menyebabkan proses prioritas rendah menunggu terlalu lama. Sementara itu, *Multilevel Queue* memberikan pendekatan kompromi dengan membagi proses ke dalam beberapa antrian berdasarkan karakteristiknya. Hal ini menunjukkan bahwa pemilihan algoritma penjadwalan harus disesuaikan dengan kebutuhan dan karakteristik sistem (IRJMETS, 2025).

Studi komparatif mengenai algoritma penjadwalan CPU juga menunjukkan bahwa setiap algoritma memiliki fokus kinerja yang berbeda. *Round Robin* lebih sesuai digunakan pada sistem yang membutuhkan keadilan distribusi waktu CPU dan respons awal yang cepat. *Priority Scheduling* lebih tepat digunakan pada sistem yang menekankan penyelesaian proses berdasarkan tingkat kepentingan. Sementara itu, *Multilevel Queue* dapat digunakan pada sistem yang memiliki variasi proses, baik proses interaktif, proses sistem, maupun proses *batch*. Perbedaan

karakteristik ini menunjukkan bahwa evaluasi algoritma perlu dilakukan dengan memperhatikan beberapa parameter kinerja secara menyeluruh (Zhang et al., 2024).

Perkembangan penelitian terkait penjadwalan proses juga menunjukkan adanya kecenderungan untuk mengembangkan algoritma hibrid. Salah satu bentuk pengembangan tersebut adalah *Priority Based Round Robin*, yaitu pendekatan yang menggabungkan mekanisme prioritas dengan pembagian waktu CPU secara bergilir. Model ini bertujuan untuk mempertahankan keunggulan *Round Robin* dalam memberikan keadilan, sekaligus meningkatkan efisiensi melalui pertimbangan prioritas proses. Pengembangan seperti ini menunjukkan bahwa algoritma penjadwalan masih menjadi topik penting dalam penelitian sistem operasi. Hal tersebut juga memperlihatkan bahwa algoritma tunggal sering kali perlu dimodifikasi agar lebih sesuai dengan kebutuhan sistem modern (Zouaoui et al., 2019).

Walaupun berbagai penelitian telah membahas algoritma penjadwalan CPU, kajian perbandingan secara langsung pada beberapa skenario beban kerja masih diperlukan. Perbandingan tersebut penting karena performa algoritma dapat berubah ketika karakteristik proses yang diuji juga berubah. Algoritma yang efisien pada beban kerja seimbang belum tentu memberikan hasil terbaik pada kondisi dominasi proses berprioritas tinggi. Demikian pula, algoritma yang baik pada proses dengan *burst time* seragam dapat mengalami penurunan kinerja ketika menghadapi variasi *burst time* yang ekstrem. Oleh karena itu, penelitian berbasis simulasi diperlukan untuk mengamati performa algoritma secara lebih terukur dan sistematis (Suhendra & Fardhan, 2026).

Berdasarkan uraian tersebut, penelitian ini berfokus pada evaluasi kinerja tiga algoritma penjadwalan proses, yaitu *Round Robin*, *Priority Scheduling*, dan *Multilevel Queue*. Evaluasi dilakukan melalui simulasi berbasis *Python* dengan menggunakan tiga skenario beban kerja, yaitu beban kerja seimbang, dominasi proses berprioritas tinggi, dan variasi *burst time* yang signifikan. Parameter yang digunakan dalam evaluasi meliputi rata-rata *turnaround time*, *waiting time*, *response time*, *throughput*, dan utilisasi CPU. Melalui penelitian ini, diharapkan dapat diperoleh gambaran yang lebih komprehensif mengenai keunggulan dan kelemahan setiap algoritma. Selain itu, hasil penelitian ini juga diharapkan dapat menjadi dasar rekomendasi dalam memilih algoritma penjadwalan yang sesuai dengan karakteristik beban kerja sistem.

2. Metode Penelitian

2.1 Desain Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan metode simulasi komputasional. Simulasi diimplementasikan menggunakan bahasa pemrograman *Python 3.x* tanpa bergantung pada *framework* simulasi eksternal, sehingga setiap detail implementasi algoritma dapat dikontrol secara penuh. Pendekatan ini sejalan dengan metodologi yang digunakan oleh Omar et al. dan penelitian komparatif di *Heliyon* yang menggunakan simulasi sebagai alat evaluasi utama (Omar et al., 2021; Carrasco-Ruiz et al., 2024).

2.2 Dataset dan Skenario Pengujian

Evaluasi dilakukan pada tiga skenario beban kerja yang masing-masing terdiri dari 10 proses dengan atribut: *Process ID* (PID), *arrival time*, *burst time*, *priority*, dan *queue level* untuk *Multilevel Queue*. Ketiga skenario dirancang untuk merepresentasikan kondisi nyata yang berbeda (Suhendra & Fardhan, 2026; Al Ghifari & Siregar, 2025).

Tabel 1. Konfigurasi Dataset Skenario Pengujian

Skenario	Nama Skenario	Karakteristik	Deskripsi
1	Beban Kerja Seimbang	Burst time: 2–10; Priority: 1–3	Distribusi proses merata di semua antrian
2	Dominasi Prioritas Tinggi	40% proses priority=1 (Q0)	Banyak proses interaktif berprioritas tinggi
3	Burst Time Variatif	Burst time: 1–20; Dispersi tinggi	Variasi burst time sangat besar (1–20 satuan)

2.3 Implementasi Algoritma

Ketiga algoritma diimplementasikan dalam Python dengan struktur modul yang terpisah. Setiap fungsi algoritma menerima daftar proses sebagai input dan mengembalikan hasil berupa daftar metrik per proses serta data Gantt chart untuk visualisasi.

Tabel 2. Spesifikasi Implementasi Algoritma

Algoritma	Tipe	Parameter Utama	Struktur Data
Round Robin	Preemptive	Time Quantum = 4	Queue (FIFO)
Priority Scheduling	Preemptive	Priority (1=Tertinggi)	Min-Heap berdasarkan prioritas

2.4 Pseudocode Algoritma

Pseudocode berikut mendeskripsikan logika inti dari masing-masing algoritma yang diimplementasikan:

Tabel 3. Pseudocode Algoritma Round Robin (Time Quantum = 4)

PROCEDURE RoundRobin(processes[], quantum)

```
Initialize: queue = [], time = 0
WHILE completed < n DO
  Tambahkan proses yang sudah tiba ke queue
  current_process = queue.dequeue()
  exec_time = MIN(quantum, remaining_burst)
  Eksekusi current_process selama exec_time
  time += exec_time
  IF remaining > 0 THEN queue.enqueue(current_process)
  ELSE catat finish_time, hitung TAT dan WT
END WHILE
END PROCEDURE
```

Tabel 4. Pseudocode Algoritma Priority Scheduling (Preemptive)

PROCEDURE PriorityScheduling(processes[])

```
Initialize: time = 0
WHILE completed < n DO
  available = [p | p.arrival <= time AND p.remaining > 0]
  current = MIN(available, key=priority)
  Eksekusi current selama 1 satuan waktu (preemptive)
  time += 1; current.remaining -= 1
  IF remaining == 0 THEN catat finish_time
END WHILE
END PROCEDURE
```

Tabel 5. Pseudocode Algoritma Multilevel Queue

PROCEDURE MultilevelQueue(processes[], quantum)

```
Bagi proses ke Q0 (Priority, prio=1), Q1 (RR, prio=2), Q2 (FCFS, prio=3)
WHILE completed < n DO
  IF Q0 tidak kosong THEN
    Jalankan Priority Scheduling pada Q0 (1 unit/time)
  ELSE IF Q1 tidak kosong THEN
    Jalankan Round Robin pada Q1 (quantum=4)
  ELSE IF Q2 tidak kosong THEN
    Jalankan FCFS pada Q2
  END IF
END WHILE
END PROCEDURE
```

2.5 Parameter Evaluasi

Parameter kinerja yang dievaluasi menggunakan rumus berikut (Carrasco-Ruiz et al., 2024; IRJMETS, 2025):

- (1) Turnaround Time (TAT) = Completion Time – Arrival Time
- (2) Waiting Time (WT) = Turnaround Time – Burst Time
- (3) Response Time (RT) = First Execution Time – Arrival Time
- (4) Throughput = Jumlah Proses Selesai / Total Waktu Eksekusi
- (5) CPU Utilization = (Total Burst Time / Total Time) × 100%

3. Hasil dan Pembahasan

3.1 Skenario 1: Beban Kerja Seimbang

Pada skenario pertama dengan distribusi proses yang merata (burst time 2–10 satuan, priority 1–3), hasil simulasi menunjukkan perbedaan yang signifikan antar algoritma. Tabel 6 menyajikan ringkasan metrik kinerja ketiga algoritma pada skenario ini.

Tabel 6. Hasil Metrik Kinerja – Skenario 1 (Beban Kerja Seimbang)

Metrik Kinerja	Round Robin	Priority Scheduling	Multilevel Queue
Rata-rata TAT	33,30	23,20	24,60
Rata-rata WT	27,50	17,40	18,80
Rata-rata RT	14,20	16,20	16,90
Throughput (proses/waktu)	0,1724	0,1724	0,1724
CPU Utilization (%)	100,00%	100,00%	100,00%

Hasil pada skenario 1 menunjukkan bahwa Priority Scheduling mencatat rata-rata TAT terendah sebesar 23,20 dan WT terendah sebesar 17,40. Hal ini konsisten dengan temuan yang menyatakan Priority Scheduling unggul 21% dalam TAT dibandingkan Round Robin (Suhendra & Fardhan, 2026). Round Robin mencatat TAT tertinggi (33,30) namun memiliki response time terbaik (14,20) karena setiap proses langsung mendapat jatah CPU sesuai posisi antrian. Multilevel Queue berada di antara keduanya dengan TAT 24,60 dan WT 18,80, mencerminkan keseimbangan antara keadilan dan efisiensi (Meing & Tandingan, 2024; Harki, 2024).

3.2 Skenario 2: Dominasi Proses Berprioritas Tinggi

Skenario 2 memiliki 40% proses di antrian berprioritas tinggi (priority=1). Pada kondisi ini, Priority Scheduling dan MLQ menunjukkan keunggulan yang lebih signifikan karena dapat melayani proses-proses kritis terlebih dahulu (Widiarto, 2024; Zouaoui et al., 2019).

Tabel 7. Hasil Metrik Kinerja – Skenario 2 (Dominasi Prioritas Tinggi)

Metrik Kinerja	Round Robin	Priority Scheduling	Multilevel Queue
Rata-rata TAT	37,70	28,70	28,80
Rata-rata WT	31,70	22,70	22,80
Rata-rata RT	14,60	22,70	21,40
Throughput (proses/waktu)	0,1667	0,1667	0,1667
CPU Utilization (%)	100,00%	100,00%	100,00%

Pada skenario 2, terdapat temuan menarik: meskipun Priority Scheduling unggul dalam TAT (28,70) dan WT (22,70), algoritma ini justru menghasilkan response time tertinggi (22,70) karena proses berprioritas rendah harus menunggu sangat lama untuk mendapatkan eksekusi pertama. Round Robin kembali mencatat response time terbaik (14,60) karena mekanisme time quantum memastikan setiap proses mendapat respons awal yang cepat. MLQ hampir setara dengan PS (TAT 28,80; WT 22,80) namun dengan response time yang lebih baik (21,40), menunjukkan keseimbangan yang lebih baik antara efisiensi dan keadilan (Meing & Tandingan, 2024; Pemasingshe & Rajapaksha, 2022).

3.3 Skenario 3: Burst Time Sangat Bervariasi

Skenario 3 menguji ketahanan algoritma dengan variasi burst time ekstrem (1–20 satuan waktu). Kondisi ini merupakan tantangan terbesar bagi Round Robin karena time quantum tetap kurang optimal untuk proses dengan burst time yang sangat berbeda (Ejurnal BITS, 2021; Omar et al., 2021).

Tabel 8. Hasil Metrik Kinerja – Skenario 3 (Burst Time Sangat Bervariasi)

Metrik Kinerja	Round Robin	Priority Scheduling	Multilevel Queue
Rata-rata TAT	40,80	27,70	28,40
Rata-rata WT	32,70	19,60	20,30
Rata-rata RT	11,70	18,80	17,30
Throughput (proses/waktu)	0,1235	0,1235	0,1235
CPU Utilization (%)	100,00%	100,00%	100,00%

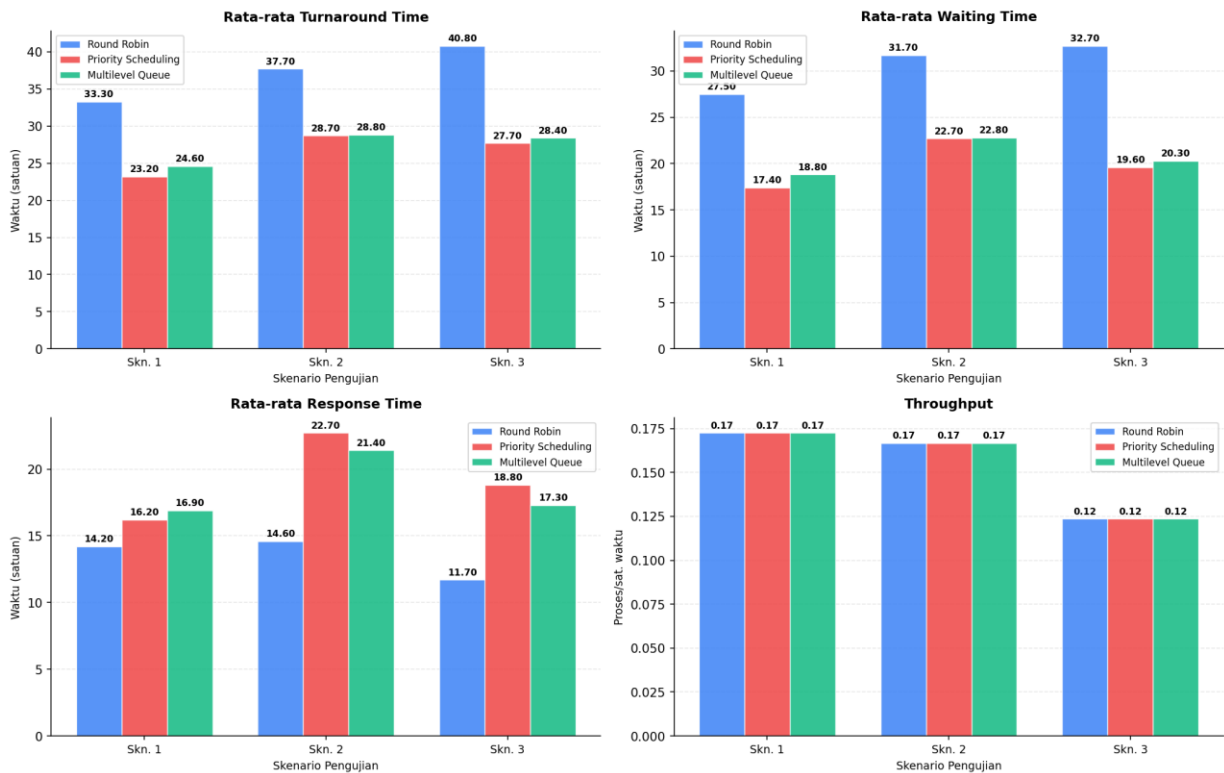
Skenario 3 mengungkapkan kelemahan paling nyata dari Round Robin: dengan variasi burst time yang ekstrem, TAT melonjak hingga 40,80 dan WT mencapai 32,70 — nilai tertinggi di antara semua skenario. Hal ini disebabkan oleh banyaknya context switch yang terjadi akibat time quantum tetap tidak mampu mengakomodasi heterogenitas proses (Ejurnal UMRI, 2024; Omar et al., 2021). Priority Scheduling kembali unggul dengan TAT 27,70 dan WT 19,60. Sementara MLQ menunjukkan performa yang mendekati PS (TAT 28,40; WT 20,30) karena strategi pembagian antrian berhasil mengelola proses dengan burst time berbeda secara lebih adaptif (Meing & Tandingan, 2024; Zhang et al., 2024).

3.4 Analisis Visual dan Perbandingan Komprehensif

Gambar 1 menyajikan perbandingan visual keempat metrik kinerja utama pada ketiga skenario pengujian. Dari grafik batang dapat diamati pola yang konsisten: Priority Scheduling dan Multilevel Queue secara konsisten

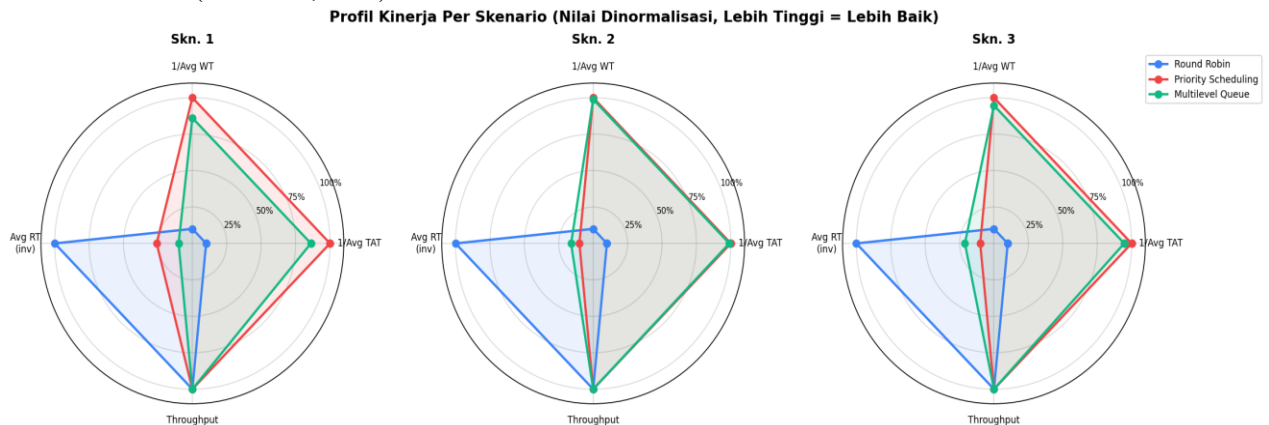
menghasilkan TAT dan WT yang lebih rendah dibanding Round Robin, sementara Round Robin unggul kedua dalam response time (Omar et al., 2021; Pemasinghe & Rajapaksha, 2022).

Perbandingan Kinerja Algoritma Penjadwalan Proses CPU (Round Robin vs Priority Scheduling vs Multilevel Queue)



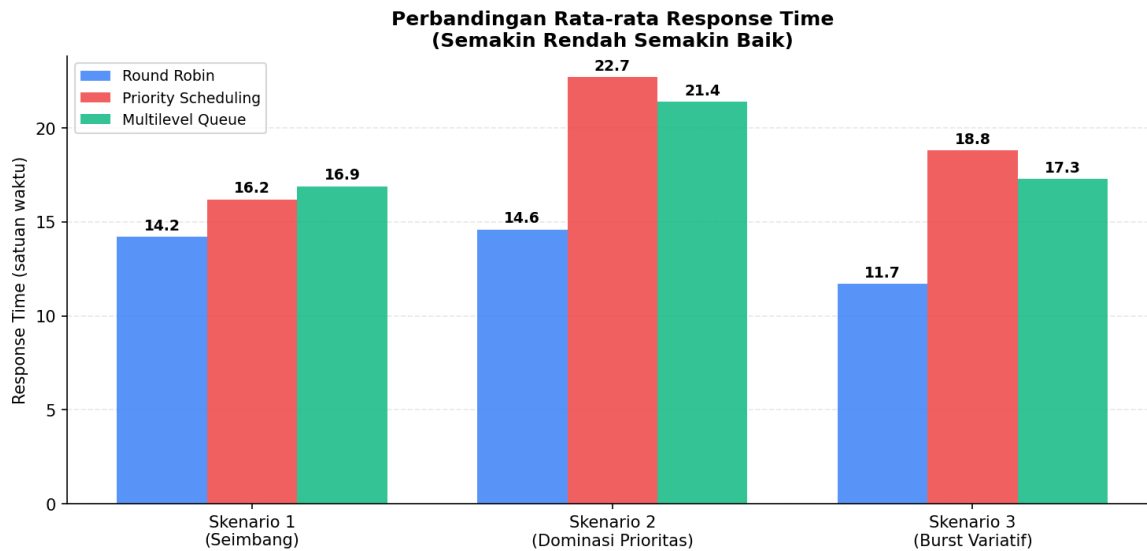
Gambar 1. Perbandingan Metrik Kinerja Tiga Algoritma pada Tiga Skenario

Gambar 2 menampilkan profil kinerja dalam bentuk radar chart yang menggambarkan trade-off antara kelima dimensi kinerja secara lebih komprehensif. Nilai yang lebih tinggi pada setiap sumbu mengindikasikan performa yang lebih baik (nilai sudah dinormalisasi). Terlihat bahwa tidak ada satu algoritma pun yang dominan di semua dimensi secara bersamaan, mengonfirmasi pernyataan bahwa pemilihan algoritma harus disesuaikan dengan kebutuhan sistem (IRJMETS, 2025).



Gambar 2. Profil Kinerja Ternormalisasi (Radar Chart) Per Skenario

Gambar 3 secara khusus menyoroti perbandingan response time di antara ketiga algoritma. Terlihat jelas bahwa Round Robin secara konsisten menghasilkan response time terendah pada semua skenario, dengan keunggulan paling menonjol pada Skenario 3 (11,70 vs 18,80 untuk PS dan 17,30 untuk MLQ). Keunggulan ini terjadi karena mekanisme time quantum memastikan setiap proses mendapat slot CPU pertama dalam waktu yang relatif singkat (Zouaoui et al., 2019; Abu-Dalbouh, 2022).



Gambar 3. Perbandingan Rata-rata Response Time Ketiga Algoritma

3.5 Rekomendasi Pemilihan Algoritma

Berdasarkan hasil simulasi komprehensif, dapat dirumuskan rekomendasi pemilihan algoritma sebagaimana disajikan pada Tabel 9 berikut.

Tabel 9. Rekomendasi Pemilihan Algoritma Berdasarkan Karakteristik Sistem

Kondisi Sistem	Algoritma Terbaik	Alasan Utama	Referensi
Butuh TAT & WT minimal (batch processing)	Priority Scheduling	TAT & WT terendah di semua skenario	Suhendra & Fardhan (2026); Widiarto (2024); Zouaoui et al. (2019)
Butuh response time cepat (sistem interaktif)	Round Robin	RT terbaik di seluruh skenario	Ejurnal UMRI (2024); Omar et al. (2021); Pemasinghe & Rajapaksha (2022)
Campuran proses (interaktif + batch + real-time)	Multilevel Queue	Mendekati PS dalam TAT/WT, RT lebih baik dari PS	Meing & Tandingan (2024); Harki (2024); Zhang et al. (2024)
Sistem dengan heterogenitas burst time tinggi	Priority Scheduling / MLQ	RR tidak efisien pada variasi besar	Ejurnal BITS (2021); Al Ghifari & Siregar (2025); IRJMETS (2025)

4. Kesimpulan

Penelitian ini telah berhasil mengimplementasikan dan membandingkan tiga algoritma penjadwalan CPU (*Round Robin*, *Priority Scheduling*, dan *Multilevel Queue*) melalui simulasi *Python* pada tiga skenario beban kerja yang berbeda. Berdasarkan hasil simulasi dan analisis yang telah dilakukan, dapat ditarik beberapa kesimpulan berikut:

- Priority Scheduling* secara konsisten mengungguli algoritma lain dalam metrik *turnaround time* dan *waiting time* pada seluruh skenario pengujian, dengan rata-rata TAT 23,20 (Skenario 1), 28,70 (Skenario 2), dan 27,70 (Skenario 3). Hal ini mengkonfirmasi temuan penelitian sebelumnya (Suhendra & Fardhan, 2026; Omar et al., 2021).
- Round Robin* unggul dalam *response time* di semua skenario (14,20; 14,60; dan 11,70) karena mekanisme *time quantum* yang menjamin setiap proses mendapat eksekusi perdana secara cepat dan adil, menjadikannya pilihan ideal untuk sistem *time-sharing* dan aplikasi interaktif (Zouaoui et al., 2019; Pemasinghe & Rajapaksha, 2022).
- Multilevel Queue* menawarkan keseimbangan terbaik antara efisiensi (TAT dan WT mendekati PS) dan keadilan (RT lebih baik dari PS), menjadikannya pilihan optimal untuk sistem dengan beban kerja campuran yang terdiri dari proses interaktif, sistem, dan *batch* (Meing & Tandingan, 2024; Harki, 2024).
- Tidak ada satu algoritma yang *universally optimal* untuk semua skenario, mengonfirmasi pernyataan bahwa pemilihan algoritma harus mempertimbangkan karakteristik beban kerja dan prioritas sistem secara menyeluruh (IRJMETS, 2025).

Saran

Berdasarkan hasil penelitian dan analisis yang telah dilakukan, beberapa saran untuk pengembangan penelitian

selanjutnya adalah sebagai berikut:

1. Penelitian selanjutnya dapat mengembangkan simulasi dengan menambahkan algoritma lain seperti *Shortest Job First* (SJF), *Highest Response Ratio Next* (HRRN), dan *Multilevel Feedback Queue* (MLFQ) untuk perbandingan yang lebih komprehensif (Al Ghifari & Siregar, 2025; Omar et al., 2021).
2. Implementasi mekanisme *aging* pada *Priority Scheduling* perlu dikaji lebih lanjut untuk mengatasi masalah *starvation* yang dapat terjadi pada proses berprioritas rendah dalam skenario sistem *real-time* (Abu-Dalbouh, 2022; Zhang et al., 2024).
3. Penelitian lanjutan dapat mengeksplorasi penggunaan *dynamic time quantum* pada *Round Robin* dan *Multilevel Queue* untuk meningkatkan adaptivitas terhadap variasi beban kerja, sebagaimana disarankan oleh Meing dan Tandingan serta penelitian terkait (Meing & Tandingan, 2024; Ejournal UMRI, 2024; Ejournal BITS, 2021).
4. Evaluasi pada beban kerja nyata (*real workload*) menggunakan data *trace* dari sistem operasi Linux atau Windows dapat memberikan hasil yang lebih representatif dibandingkan dataset simulatif (Carrasco-Ruiz et al., 2024; IRJMETS, 2025).
5. Pengembangan algoritma hibrid yang menggabungkan keunggulan *Priority Scheduling* dalam efisiensi TAT/WT dengan keunggulan *Round Robin* dalam *response time* merupakan arah penelitian yang menjanjikan (Zouaoui et al., 2019; Abu-Dalbouh, 2022; Meing & Tandingan, 2024).

Reference

- 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). <https://doi.org/10.14569/IJACSA.2022.0130463>
- 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>
- Carrasco-Ruiz, D., Herrera, O., & López, E. (2024). Study and evaluation of CPU scheduling algorithms. *Heliyon*. <https://doi.org/10.1016/j.heliyon.2024.e31905>
- Ejournal BITS. (2021). Analisis perbandingan algoritma penjadwalan CPU First Come First Serve (FCFS) dan Round Robin. *Building of Informatics, Technology and Science (BITS)*. <https://ejournal.seminar-id.com/index.php/bits/article/view/1047>
- Ejournal UMRI. (2024). Implementasi algoritma Round Robin dan Priority pada penjadwalan proses. *Jurnal Ilmu Komputer (JIK)*, 14(2), 507–513. <https://ejournal.umri.ac.id/index.php/JIK/article/download/7334/3079/>
- Harki, N. A. (2024). Multi-level feedback queue scheduling technique: Model proposal to reduce risk and enhance performance of healthcare systems. *Cihan University-Erbil Scientific Journal*, 8(2). <https://journals.cihanuniversity.edu.iq/index.php/cuesj/article/view/1140>
- IRJMETS. (2025). CPU scheduling algorithms. *International Research Journal of Modernization in Engineering, Technology and Science*. https://www.irjmets.com/upload_newfiles/irjmets71000010264/paper_file/irjmets71000010264.pdf
- Meing, L. W. C., & Tandingan, D. R. (2024). Exploring multilevel feedback queue combinations and regression-based time quanta in scheduling algorithms. *Journal of Information Systems Engineering and Management*, 9. <https://www.jisem-journal.com/index.php/journal/article/view/699>
- Omar, H. K., Jihad, K. H., & Hussein, S. F. (2021). Comparative analysis of the essential CPU scheduling algorithms. *Bulletin of Electrical Engineering and Informatics*, 10(5), 2742–2750. <https://pdfs.semanticscholar.org/0a24/8215a55473199cc1254c3f9c32ae2510e541.pdf>
- Pemasinghe, S., & Rajapaksha, S. (2022). Comparison of CPU scheduling algorithms: FCFS, SJF, SRTF, Round Robin, Priority Based, and Multilevel Queuing. *2022 IEEE 10th Region 10 Humanitarian Technology Conference (R10-HTC)*, 4, 318–323. <https://doi.org/10.1109/R10-HTC54060.2022.9929533>
- Putra, T. D., & Purnomo, R. (2021). Analisis algoritma Round Robin pada penjadwalan CPU. *Jurnal Ilmiah Teknologi Informasi Asia*, 15(2), 85–90. <https://jurnal.asia.ac.id/index.php/jitika/article/download/481/291>
- Suhendra, F., & Fardhan, M. (2026). Evaluasi kinerja algoritma Multi-Level Queue, Round Robin, dan Priority Scheduling dalam simulasi penjadwalan proses kinerja sistem operasi. *Jurnal Ilmiah ILKOMINFO: Ilmu Komputer & Informatika*, 9(1), 159–170.
- Widiarto. (2024). Analisis perbandingan penjadwalan proses menggunakan algoritma Round Robin dan Priority Preemptive. *Progresif: Jurnal Ilmiah Komputer*. <https://ojs.stmik-banjarbaru.ac.id/index.php/progresif/article/view/2169>
- Zhang, J., et al. (2024). Enhancing task execution: A dual-layer approach with multi-queue adaptive priority scheduling. *PeerJ Computer Science*. <https://pmc.ncbi.nlm.nih.gov/articles/PMC11623127/>
- Zouaoui, S., Boussaid, L., & Mtibaa, A. (2019). Priority based Round Robin (PBRR) CPU scheduling algorithm. *International Journal of Electrical and Computer Engineering*, 9(1), 190–202. <https://doi.org/10.11591/ijece.v9i1.pp190-202>