

## Evaluasi Kinerja Algoritma Multi-Level Queue, Round Robin, dan Priority Scheduling dalam Simulasi Penjadwalan Proses Kinerja Sistem Operasi

Sugiarti<sup>1</sup>, Herman<sup>2</sup>, Fadri Suhendra<sup>3</sup>, Muh.Fardhan<sup>4</sup>

<sup>1,3,4</sup>Fakultas Ilmu Komputer, Program Studi Teknik Informatika

<sup>2</sup>Fakultas Ilmu Komputer, Program Studi Sistem Informasi

Universitas Muslim Indonesia

Sugiarti.sugiarti@umi.ac.id

### Abstrak

Penjadwalan proses bertujuan mengoptimalkan penggunaan CPU, menekan waktu tunggu, dan meningkatkan kinerja sistem, namun peningkatan jumlah serta kompleksitas proses menyulitkan pemeliharaan performa pada beban kerja dinamis, yang tidak selalu dapat diatasi oleh semua algoritma penjadwalan. Penelitian ini membandingkan tiga algoritma penjadwalan *Multi-Level Queue*, *Round Robin*, dan *Priority Scheduling* melalui simulasi dengan tiga skenario pengujian. Kebaruan penelitian terletak pada analisis komparatif kinerja ketiga algoritma berdasarkan karakteristik beban kerja berbeda, sehingga memberikan gambaran adaptivitas algoritma penjadwalan dalam kondisi sistem dinamis. Evaluasi dilakukan berdasarkan *turnaround time*, *waiting time*, *response time*, dan *throughput* pada tiga skenario. Pada beban kerja seimbang, *Priority Scheduling* unggul dalam TAT dan WT namun *Round Robin* terbaik untuk *Response Time*. Pada beban kerja berat, *Priority Scheduling* sangat efisien dengan TAT 17.6 ms. Pada beban kerja ringan dengan variasi prioritas, *Priority Scheduling* memberikan kinerja terbaik untuk TAT 6.6 ms dan WT 4.0 ms, namun *Round Robin* unggul dalam *Response Time* 2.0 ms. Hasil menunjukkan *Priority Scheduling* paling efisien dengan *turnaround time* 21% dan *waiting time* 29% lebih baik dari *Round Robin*, serta lebih unggul 13% dan 17% dibanding *Multi-Level Queue*. *Priority Scheduling* dapat mengeksekusi proses berprioritas tinggi terlebih dahulu tanpa *overhead context switching*. *Round Robin* unggul dalam *response time*, sedangkan *Multi-Level Queue* menunjukkan kinerja seimbang.

**Kata kunci:** Algoritma Penjadwalan CPU, Multi-Level Queue, Round Robin, Priority Scheduling, Simulasi

### Abstract

*Process scheduling aims to maximize CPU utilization, reduce waiting time, and improve system performance under dynamic workloads, where not all algorithms adapt effectively. This study compares Multi-Level Queue, Round Robin, and Priority Scheduling using simulation across three workload scenarios. The novelty lies in a comparative performance analysis based on different workload characteristics, highlighting algorithm adaptability under dynamic conditions. Performance is evaluated using turnaround time, waiting time, response time, and throughput. Under balanced workloads, Priority Scheduling achieves better turnaround time and waiting time, while Round Robin provides the best response time. In heavy workloads, Priority Scheduling shows high efficiency with a turnaround time of 17.6 ms. Under light workloads with priority variation, Priority Scheduling performs best with a turnaround time of 6.6 ms and a waiting time of 4.0 ms, whereas Round Robin excels in response time at 2.0 ms. Overall, Priority Scheduling is the most efficient algorithm, achieving 21% better turnaround time and 29% better waiting time than Round Robin, and outperforming Multi-Level Queue by 13% and 17%, respectively. Round Robin offers the best response time, while Multi-Level Queue demonstrates balanced performance.*

**Keywords:** *CPU Scheduling Algorithm, Multi-Level Queue, Round Robin, Priority Scheduling, Simulation.*

## PENDAHULUAN

Manajemen proses merupakan fungsi utama dalam sistem operasi yang bertanggung jawab mengatur pelaksanaan berbagai proses agar berjalan efisien dan adil (Lei Li, 2024). Dalam sistem operasi modern, pengelolaan proses mengoptimalkan sumber daya, mencegah CPU menganggur, dan menjaga stabilitas sistem (Afrianto et al., 2024).

Dalam lingkungan multitasking dan multiprocessing, penjadwalan CPU menjadi komponen penting karena menentukan bagaimana sistem membagi sumber daya di antara banyak proses yang berjalan secara bersamaan (Putra et al., 2021). Efektivitas sistem operasi dalam mengelola berbagai proses bergantung pada manajemen proses dan mekanisme penjadwalan yang diterapkan (Iqbal et al., 2023). Tujuan utama penjadwalan adalah memaksimalkan penggunaan CPU, mengurangi waktu tunggu, dan meningkatkan produktivitas sistem (Dalmia et al., 2022). Namun, peningkatan jumlah dan kompleksitas proses menimbulkan tantangan dalam menjaga performa di bawah beban kerja yang berubah-ubah. Tidak semua algoritma penjadwalan mampu beradaptasi dengan baik terhadap kondisi tersebut (Dalmia et al., 2022).

Penelitian ini mengevaluasi tiga algoritma penjadwalan dengan pendekatan berbeda, yaitu *Multi-Level Queue (MLQ)*, *Round Robin*, dan *Priority Scheduling*, untuk mengetahui efektivitas masing-masing terhadap variasi beban kerja melalui simulasi (Sharma et al., 2022). Algoritma MLQ membagi proses ke beberapa antrian berdasarkan jenis dan

prioritas, dengan Round Robin pada foreground, FCFS pada background, dan prioritas tetap antar-antrian. (Priambudi et al., 2024),(Huda, 2023).

Algoritma *Round Robin* cocok untuk proses interaktif karena memberikan waktu eksekusi merata melalui pembagian time quantum (Ghazy & Eldahshan, 2022),(Widiarto et al., 2024). Namun, quantum time yang terlalu besar memperlambat respon, sedangkan yang terlalu kecil meningkatkan *overhead* akibat seringnya *context switching* (Ardi Wijaya, 2021),(Goel & Garg, 2022). Priority Scheduling mengeksekusi proses berdasarkan prioritas secara preemptive atau non-preemptive, namun berisiko menunda proses berprioritas rendah. (Dalmia et al., 2022),(Simarmata & Harahap, 2021).

Beberapa penelitian menunjukkan peningkatan performa melalui modifikasi algoritma, seperti penggunaan *dynamic quantum* atau adaptasi prioritas (Koval et al., 2021). Namun, perbandingan komprehensif antara ketiga algoritma tersebut dalam konteks beban kerja variatif masih terbatas (Ghazy & Eldahshan, 2022).

Penelitian ini bertujuan merancang simulasi untuk mengukur kinerja ketiga algoritma berdasarkan waiting time, response time, turnaround time, dan throughput (Mutasar, 2020).

Meskipun berbagai penelitian sebelumnya telah mengkaji kinerja algoritma penjadwalan CPU, sebagian besar studi berfokus pada satu jenis beban kerja atau melakukan perbandingan terbatas tanpa mengaitkan karakteristik beban kerja secara eksplisit terhadap

kinerja algoritma. Dengan demikian, terdapat celah penelitian berupa kurangnya analisis komparatif yang sistematis untuk mengevaluasi adaptivitas Multi-Level Queue, Round Robin, dan Priority Scheduling pada kondisi beban kerja yang berbeda. Kebaruan penelitian ini terletak pada penyajian analisis komparatif kinerja ketiga algoritma berdasarkan variasi beban kerja seimbang, berat, dan ringan, sehingga memberikan gambaran yang lebih jelas mengenai kemampuan adaptasi masing-masing algoritma dalam lingkungan sistem yang dinamis.

### Rumusan Masalah

Bagaimana perbandingan kinerja antara algoritma *Multi-Level Queue*, *Round Robin*, dan *Priority Scheduling* dalam penjadwalan proses sistem operasi berdasarkan parameter evaluasi *response time*, *waiting time*, dan *turnaround time*?

### Tujuan Penelitian

Menganalisis kinerja algoritma *Multi-Level Queue*, *Round Robin*, dan *Priority Scheduling* dalam penjadwalan proses sistem operasi melalui simulasi, berdasarkan indikator waktu tunggu (*waiting time*) dan *turnaround time* untuk menentukan algoritma yang paling optimal dalam berbagai skenario beban proses.

### Manfaat Penelitian

#### 1. Manfaat Teoritis

Penelitian ini memberikan kontribusi ilmiah berupa pemahaman yang lebih mendalam mengenai kelebihan dan keterbatasan algoritma *Multi-Level Queue*, *Round Robin*, dan *Priority Scheduling* dalam penjadwalan proses sistem operasi, khususnya dalam menghadapi variasi beban kerja,

berdasarkan parameter *response time*, *waiting time*, dan *turnaround time*.

#### 2. Manfaat Praktis

Hasil penelitian ini dapat digunakan sebagai bahan pertimbangan dalam pemilihan algoritma penjadwalan proses yang sesuai pada sistem multitasking dan multiprocessing, sehingga dapat membantu meningkatkan efisiensi serta kestabilan kinerja CPU berdasarkan karakteristik beban proses yang dihadapi.

### Tinjauan Pustaka

Lei Li melakukan analisis dua jenis sistem operasi khusus, yaitu *Time-Sharing System* dan *Embedded System*, serta memperkenalkan algoritma penjadwalan yang relevan dengan kedua sistem tersebut beserta contoh penerapannya. Hasil tinjauan menunjukkan bahwa SJF dan SPTF memberikan kinerja lebih baik dibandingkan FCFS dan RR dalam hal waktu tunggu (*waiting time*), sementara HRRN juga secara signifikan mengungguli RR (Lei Li, 2024).

Hoger K. Omar et al. (Omar et al., 2021) Disimpulkan bahwa tidak ada satu algoritma penjadwalan CPU yang mampu memenuhi seluruh kebutuhan sistem. Setiap algoritma memiliki keterbatasan, seperti FCFS dan RR dengan waktu tunggu tinggi, SJF, LJFS, LRTE, dan SRTF yang berpotensi starvation, serta MFLQ yang memerlukan context switching tambahan, sehingga pemilihan algoritma harus disesuaikan dengan jenis pekerjaan dan karakteristik sistem.

Nurul Huda dan Novriyadi membangun aplikasi antrian online Metode *Multilevel Queue (MLQ)*. Hasil penelitian menunjukkan bahwa pendaftaran antrian dapat dilakukan melalui aplikasi

Android, sehingga administrasi manual tidak diperlukan. Algoritma Multi-Level Queue diterapkan dengan pengurutan berdasarkan waktu kedatangan dan penentuan quantum time untuk menghasilkan kinerja yang optimal (Huda, 2023).

Raghav Dalmia et al. (Dalmia et al., 2022) mengusulkan varian dari Dynamic Round Robin dengan memanfaatkan metode perhitungan *time quantum* dan penentuan prioritas untuk setiap proses. Pendekatan ini membantu mengurangi jumlah proses dalam antrian, sehingga meningkatkan efisiensi dan memengaruhi status penjadwalan secara keseluruhan.

Berdasarkan beberapa penelitian terdahulu, dapat disimpulkan bahwa kajian terkait algoritma penjadwalan CPU umumnya berfokus pada analisis algoritma secara individual, pengembangan varian algoritma tertentu, atau penerapan algoritma dalam studi kasus spesifik, dengan indikator kinerja seperti *waiting time*, *turnaround time*, *response time*, dan *throughput*. Namun demikian, masih terbatas penelitian yang melakukan analisis komparatif secara langsung terhadap beberapa algoritma penjadwalan populer, khususnya *Multi-Level Queue*, *Round Robin*, dan *Priority Scheduling*, dalam satu kerangka simulasi yang sama dan pada berbagai skenario beban proses. Oleh karena itu, penelitian ini fokus melakukan evaluasi dan perbandingan kinerja ketiga algoritma berdasarkan indikator waktu tunggu dan *turnaround time* untuk menentukan algoritma yang paling optimal sesuai karakteristik beban sistem.

## LANDASAN TEORI

### *Multi-Level Queue (MLQ)*

MLQ adalah algoritma penjadwalan CPU yang memisahkan proses ke dalam beberapa antrian berdasarkan kategori atau prioritas tertentu. Setiap antrian biasanya dikhususkan untuk jenis proses yang berbeda, misalnya proses interaktif, *batch*, atau system dan masing-masing antrian dapat menggunakan kebijakan penjadwalan yang berbeda. Kelebihan utama MLQ adalah kemampuannya mengelola proses secara terstruktur sesuai karakteristik beban kerja, sehingga pemanfaatan CPU dapat diatur lebih efisien. Hal ini menjadikan MLQ relevan untuk dianalisis berdasarkan *throughput*, karena pembagian antrian memungkinkan sistem menyelesaikan lebih banyak proses dalam satuan waktu tertentu. Namun, kelemahan MLQ terletak pada fleksibilitas proses tidak dapat berpindah antar antrian, sehingga proses dengan prioritas rendah bisa mengalami penundaan atau *starvation* apabila antrian dengan prioritas lebih tinggi terus aktif (Dalmia et al., 2022).

### *Round Robin (RR)*

*Round Robin* adalah algoritma penjadwalan *preemptive* yang mendistribusikan waktu CPU secara merata di antara semua proses menggunakan kuantum waktu (*time quantum*) tertentu. Setiap proses dieksekusi dalam interval waktu yang telah ditentukan, kemudian CPU berpindah ke proses berikutnya dalam antrian. Karakteristik ini menjadikan RR sangat sesuai dianalisis menggunakan indikator *response time*, karena algoritma ini dirancang untuk meminimalkan waktu respon awal proses, khususnya pada sistem time-sharing dan multi-pengguna. Proses

tidak perlu menunggu lama untuk pertama kali dieksekusi oleh CPU.

Namun, pemilihan nilai kuantum waktu yang tidak tepat dapat berdampak signifikan terhadap kinerja sistem, seperti meningkatnya *context switching* atau memburuknya *waiting time* dan *turnaround time* (Ardi Wijaya, 2021).

### **Priority Scheduling (PS)**

*Priority scheduling* adalah sebuah algoritma penjadwalan yang mengatur eksekusi proses berdasarkan tingkat prioritas yang telah ditentukan. CPU akan selalu mengeksekusi proses dengan prioritas tertinggi yang tersedia, sementara proses dengan prioritas lebih rendah harus menunggu hingga seluruh proses prioritas tinggi selesai dieksekusi.

Mekanisme ini menjadikan Priority Scheduling rawan mengalami starvation, terutama ketika sistem terus menerima proses baru dengan prioritas tinggi, sehingga proses berprioritas rendah dapat tertunda tanpa batas waktu. Prioritas dalam algoritma ini dapat ditentukan secara internal, seperti berdasarkan penggunaan CPU atau waktu tunggu, maupun secara eksternal, seperti kebijakan manajemen sistem atau kebutuhan pengguna. Apabila terdapat proses dengan prioritas yang sama, algoritma lain seperti First-Come, First-Served (FCFS) atau Round Robin dapat digunakan sebagai mekanisme penentu urutan eksekusi (Abas Ali Pangera, 2010).

### **Sistem Operasi**

Sistem operasi merupakan perangkat lunak sistem yang berfungsi mengelola sumber daya komputer, seperti prosesor, memori, dan perangkat input/output, serta mengontrol proses yang berjalan sebagai penghubung antara perangkat keras dan

pengguna. Seiring perkembangannya, sistem operasi yang awalnya menjalankan proses secara tunggal dan sekuensial telah berkembang mendukung multitasking, sehingga pemanfaatan sumber daya menjadi lebih efisien (Putra et al., 2021).

### **METODE PENELITIAN**

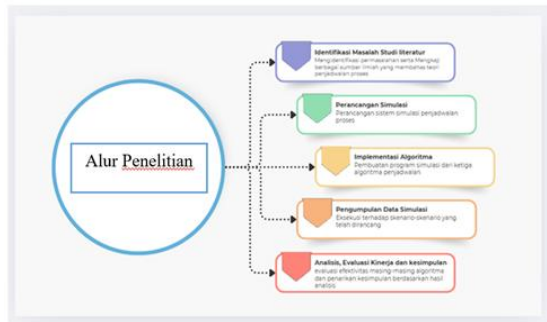
Pendekatan yang digunakan dalam penelitian ini adalah pendekatan eksperimental dengan melakukan simulasi kinerja algoritma penjadwalan CPU, yaitu *Multi-Level Queue (MLQ)*, *Round Robin (RR)*, dan *Priority Scheduling*. Tujuan utama pendekatan ini adalah untuk memperoleh data empiris mengenai tingkat efektivitas masing-masing algoritma dalam mengelola proses di sistem operasi.

Penelitian ini menggunakan bahasa pemrograman C++ untuk membangun tiga modul simulasi yang masing-masing merepresentasikan algoritma *Multi-Level Queue*, *Round Robin*, dan *Priority Scheduling*. Simulasi dijalankan pada sistem operasi Windows, dengan setiap modul diuji menggunakan data proses yang identik agar hasil perbandingan bersifat adil dan objektif.

Hasil simulasi dianalisis dengan membandingkan nilai rata-rata parameter kinerja, yaitu *waiting time*, *turnaround time*, dan *response time*, dari masing-masing algoritma pada setiap skenario beban kerja guna menentukan perbedaan kinerja dan tingkat optimalitas algoritma yang diuji. Variabel yang diukur meliputi *turnaround time*, *waiting time*, *response time*, dan *throughput*, sebagai parameter utama untuk menilai performa sistem.

### **Alur Penelitian**

Pada penelitian ini alur penelitian dapat dilihat pada gambar 1.



Gambar 1. Alur penelitian

Diagram alir di atas bertujuan menggambarkan alur penelitian yang dilakukan secara bertahap yang terdiri dari:

### 1. Input

Pada tahap awal, input yang digunakan dalam penelitian ini meliputi:

- 1) Studi literatur terkait algoritma penjadwalan proses dalam sistem operasi.
- 2) Penelusuran referensi mengenai algoritma *Multi-Level Queue*, *Round Robin*, dan *Priority Scheduling*.
- 3) Penentuan dataset simulasi proses yang mencakup parameter seperti *burst time*, *arrival time*, dan prioritas proses.
- 4) Pemilihan *tools* atau *software* simulasi yang mendukung implementasi algoritma menggunakan Bahasa Pemrograman C++.

### 2. Proses

Tahapan proses merupakan inti dari penelitian, yang terdiri atas:

- 1) Perancangan model simulasi penjadwalan proses.
- 2) Implementasi masing-masing algoritma (*Multi-Level Queue*, *Round Robin*, dan *Priority Scheduling*) dalam simulasi.
- 3) Pengujian simulasi untuk menjalankan proses penjadwalan menggunakan data input yang telah disiapkan.
- 4) Evaluasi dan perbandingan hasil dari masing-masing algoritma berdasarkan

metrik kinerja seperti *turnaround time*, *waiting time*, dan *CPU utilization*.

### 3. Output

Hasil dari proses penelitian ini menghasilkan beberapa keluaran penting, di antaranya:

- 1) Data hasil simulasi dari ketiga algoritma yang telah diterapkan.
- 2) Analisis komparatif efektivitas dan efisiensi tiap algoritma dalam konteks penjadwalan proses.
- 3) Kesimpulan mengenai algoritma yang paling optimal untuk kondisi skenario tertentu.
- 4) Rekomendasi pemilihan algoritma penjadwalan yang dapat diterapkan pada sistem operasi dengan beban kerja tertentu.

## ANALISIS DAN PERANCANGAN

### Analisis Karakteristik Algoritma

#### a. *Multi-Level Queue (MLQ)*

Proses dibagi ke dalam beberapa antrian berdasarkan prioritas atau tipe tugas. Setiap antrian memiliki algoritma penjadwalan tersendiri, misalnya antrian *foreground* menggunakan *Round Robin*, sedangkan *background* menggunakan FCFS. Kelebihannya adalah efisiensi tinggi pada sistem dengan proses yang beragam, namun kekurangannya adalah proses tidak dapat berpindah antar antrian.

#### b. *Round Robin (RR)*

Setiap proses mendapatkan jatah waktu (*time quantum*) yang sama secara bergiliran. Cocok untuk sistem *time-sharing* karena memberikan kesempatan eksekusi yang adil. Namun jika *quantum time* terlalu kecil, CPU akan mengalami *context switching* berlebih.

#### c. *Priority Scheduling*

Setiap proses dijalankan berdasarkan tingkat prioritas. Algoritma ini dapat

bersifat *preemptive* maupun *non-preemptive*. Kekurangannya adalah potensi starvation pada proses prioritas rendah.

d. Simulasi dilakukan dengan membuat tiga modul program menggunakan bahasa pemrograman C++. Setiap modul merepresentasikan satu algoritma penjadwalan dan diuji menggunakan data proses identik untuk menjaga validitas perbandingan.

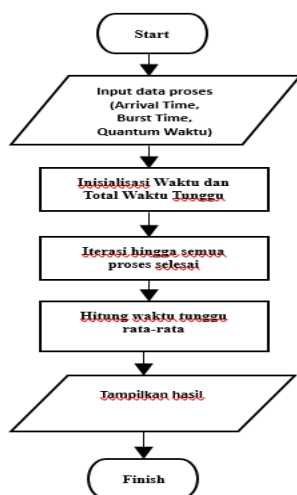
Proses diuji berdasarkan empat parameter performa utama:

- Turnaround Time (TAT)* – total waktu penyelesaian proses.
- Waiting Time (WT)* – total waktu proses menunggu giliran CPU.
- Response Time (RT)* – waktu dari kedatangan hingga pertama kali mendapat CPU.
- Throughput (TP)* – jumlah proses yang diselesaikan dalam satuan waktu.

### Perancangan Flowchart

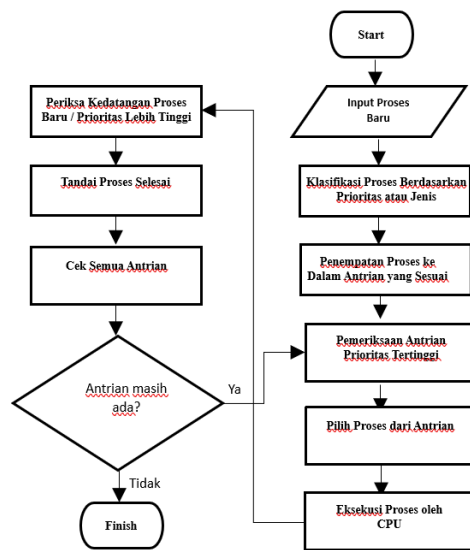
Berdasarkan analisis masalah, maka diterapkan 3 (tiga) Algoritma beserta rancangannya masing-masing:

#### a. Round Robin (RR)



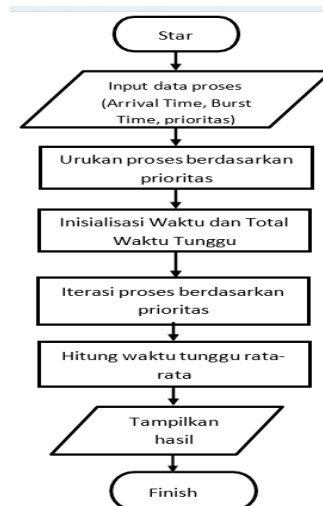
Gambar 2. Flowchart Round Robin

#### b. Algoritma Multi-Level Queue (MLQ)



Gambar 3. Flowchart MLQ

#### c. Priority Scheduling



Gambar 4. Flowchart Scheduling

## IMPLEMENTASI DAN PEMBAHASAN Implementasi Sistem

Simulasi dilakukan dengan 3 (tiga) skenario pengujian berbeda untuk mengevaluasi kinerja sistem operasi dengan menerapkan algoritma *Multi-Level Queue (MLQ)*, *Round Robin*, dan *Priority Scheduling* berdasarkan *waiting time*, *turnaround time*, *response time*, dan *throughput*.

### Skenario 1:

Tabel 1. Beban Kerja Seimbang

Proses	AT	BT	Priority
P1	0	4	2
P2	1	3	1
P3	2	5	3
P4	3	2	4
P5	4	4	2

#### a. Round Robin (TQ=2)

Proses eksekusi dengan metode RR diperoleh hasil seperti tabel di bawah ini.

Tabel 2. Proses eksekusi RR

Proses	CT	TAT	WT	RT
P1	10	10	6	0
P2	11	10	7	1
P3	16	14	9	2
P4	8	5	3	3
P5	18	14	10	9
<b>Rata-rata</b>	<b>12.6</b>	<b>10.6</b>	<b>7.0</b>	<b>3.0</b>

#### b. Priority Scheduling

Proses eksekusi dengan metode *Priority Scheduling* diperoleh hasil seperti tabel di bawah ini.

Tabel 3. Proses eksekusi PS

Proses	CT	TAT	WT	RT
P1	4	4	0	0
P2	7	6	3	3
P3	16	14	9	9
P4	18	15	13	13
P5	11	7	3	3
<b>Rata-rata</b>	<b>11.2</b>	<b>9.2</b>	<b>5.6</b>	<b>5.6</b>

#### c. Multi-Level Queue

Proses eksekusi dengan metode *Multi-Level Queue* dengan Config: Q1 (Pri 1-2): RR TQ=2 | Q2 (Pri 3-4): FCFS

Tabel 4. Proses eksekusi MLQ

Proses	CT	TAT	WT	RT
P1	6	6	2	0
P2	9	8	5	1
P3	16	14	9	9
P4	18	15	13	13
P5	11	7	3	2
<b>Rata-rata</b>	<b>12.0</b>	<b>10.0</b>	<b>6.4</b>	<b>5.0</b>

Analisis skenario 1 dapat dilihat pada tabel berikut:

Tabel 5. Proses Analisis skenario 1

Algoritma	Avg TAT	Avg WT	Avg RT	Throughput
Round Robin	10.6	7.0	3.0	0.278
Priority Scheduling	9.2	5.6	5.6	0.278
Multi-Level Queue	10.0	6.4	5.0	0.278

### Skenario 2: Beban Kerja Berat

Skenario 2 dengan karakteristik beban kerja berat dengan *burst time* panjang (4-8 ms), mensimulasikan kondisi sistem dengan beban tinggi dan proses CPU-intensive, total *burst time*: 30 ms (tinggi dibanding skenario lain), serta *time Quantum* untuk *Round Robin* = 3 ms.

Tabel 6. Proses beban kerja berat

Proses	AT	BT	Priority
P1	0	8	3
P2	1	6	1
P3	2	7	2
P4	3	5	4
P5	4	4	2

#### a. Round Robin (TQ=3)

Proses eksekusi dengan metode RR (putaran-per putaran) diperoleh hasil seperti tabel di bawah ini.

Tabel 7. Proses RR

Proses	CT	TAT	WT	RT
P1	28	28	20	0
P2	21	20	14	2
P3	29	27	20	4
P4	26	23	18	6
P5	30	26	22	8
<b>Rata-rata</b>	<b>26.8</b>	<b>24.8</b>	<b>18.8</b>	<b>4.0</b>

#### b. Priority Scheduling

Proses eksekusi dengan metode *Priority Scheduling* diperoleh hasil seperti tabel di bawah ini.

Tabel 8. Proses PS

Proses	CT	TAT	WT	RT
P1	8	8	0	0
P2	14	13	7	7
P3	21	19	12	12
P4	30	27	22	22

P5	25	21	17	17
<b>Rata-rata</b>	<b>19.6</b>	<b>17.6</b>	<b>11.6</b>	<b>11.6</b>

c. *Multi-Level Queue*

Proses eksekusi dengan metode *Multi-Level Queue* dengan Config: Q1 (Pri 1-2): RR TQ=3 | Q2 (Pri 3-4): FCFS

Tabel 9. Proses *MLQ*

Proses	AT	BT	Priority	Queue	CT	TAT	WT	RT
P1	0	8	3	Q2	26	18	18	18
P2	1	6	1	Q1	13	6	0	0
P3	2	7	2	Q1	18	9	3	3
P4	3	5	4	Q2	30	22	2	6
P5	4	4	2	Q1	17	9	9	9
<b>Rata-rata</b>					<b>18.8</b>	<b>12.8</b>	<b>11.2</b>	

Analisis skenario 2 dapat dilihat pada tabel berikut:

Tabel 10. Proses Analisis skenario 2

Algoritma	Avg TAT	Avg WT	Avg RT	Throughput
Round Robin	24.8	18.8	4.0	0.167
Priority Scheduling	17.6	11.6	11.6	0.167
Multi-Level Queue	18.8	12.8	9.4	0.167

**Skenario 3:** Beban Kerja Ringan Dengan Variasi Prioritas.

Skenario 3 dengan karakteristik yaitu beban kerja ringan dengan burst time pendek (2-4 ms), variasi prioritas yang beragam untuk menguji efektivitas algoritma berbasis prioritas, *arrival time* berurutan untuk mensimulasikan kedatangan proses secara teratur serta *time quantum* untuk RR = 2 ms.

Tabel 11. Beban Kerja Ringan

Proses	AT	BT	Priority
P1	0	3	1

P2	1	2	4
P3	2	4	2
P4	3	3	3
P5	4	2	1

a. *Round Robin* (TQ=2)

Proses eksekusi dengan metode RR diperoleh hasil seperti tabel di bawah ini.

Tabel 12. Proses *RR*

Proses	AT	BT	CT	TAT	WT	RT
P1	0	3	11	11	8	0
P2	1	2	4	3	1	1
P3	2	4	13	11	7	2
P4	3	3	14	11	8	3
P5	4	2	10	6	4	4
<b>Rata-rata</b>				<b>8.4</b>	<b>5.6</b>	<b>2.0</b>

b. *Priority Scheduling*

Proses eksekusi dengan metode *Priority Scheduling (Non-Preemptive)* diperoleh hasil seperti tabel di bawah ini.

Tabel 13. Proses *PS*

Proses	AT	BT	Priority	CT	TAT	WT	RT
P1	0	3	1	3	3	0	0
P2	1	2	4	14	13	11	11
P3	2	4	2	9	7	3	3
P4	3	2	3	12	9	6	6
P5	4	2	1	5	1	0	0
<b>Rata-rata</b>					<b>6.6</b>	<b>4.0</b>	<b>4.0</b>

c. *Multi-Level Queue*

Proses eksekusi dengan metode *Multi-Level Queue* dengan Config: Q1 (Pri 1-2): RR TQ=3 | Q2 (Pri 3-4): FCFS

Tabel 14. Proses *MLQ*

Proses	AT	BT	Priority	Queue	CT	TAT	WT	RT
P1	0	3	1	Q1	7	7	4	0
P2	1	2	4	Q2	14	11	11	11
P3	2	4	2	Q1	9	7	3	0
P4	3	3	3	Q2	12	9	6	6
P5	4	2	1	Q1	6	2	0	0
<b>Rata-rata</b>					<b>7.6</b>	<b>4.8</b>	<b>3.4</b>	

Analisis komparatif skenario 3 dapat dilihat pada tabel berikut:

Tabel 15. Analisis komparatif

Algoritma	Avg TAT	Avg WT	Avg RT	Through-put
Round Robin	8.4	5.6	2.0	0.357
Priority Scheduling	6.6	4.0	4.0	0.357
Multi-Level Queue	7.6	4.8	3.4	0.357

## Pembahasan

### Skenario 1

Berdasarkan hasil analisis skenario 1 diperoleh kesimpulan bahwa *Priority Scheduling* unggul dalam TAT dan WT namun *Round Robin* terbaik untuk untuk *Response Time*.

### Skenario 2

Berdasarkan hasil analisis skenario 2 diperoleh kesimpulan pada beban berat, *Priority Scheduling* sangat efisien (TAT 17.6 ms). *Round Robin* mengalami overhead tinggi akibat *context switching* berulang.

### Skenario 3

Berdasarkan hasil analisis skenario 3 diperoleh hasil bahwa (1) *Priority Scheduling* memberikan kinerja terbaik untuk TAT (6.6 ms) dan WT (4.0 ms). Pada beban ringan dengan variasi prioritas yang jelas, algoritma ini sangat efisien karena dapat mengeksekusi proses berprioritas tinggi terlebih dahulu tanpa *overhead context switching*. (2) *Round Robin* unggul dalam Response Time (2.0 ms), membuktikan keunggulannya untuk sistem interaktif. Setiap proses mendapat giliran eksekusi cepat, memberikan pengalaman responsif bagi pengguna. (3) *Multi-Level Queue* menunjukkan keseimbangan yang baik dengan TAT 7.6 ms dan RT 3.4 ms, memposisikan diri di antara kedua algoritma lainnya. Pemisahan queue berdasarkan prioritas

memungkinkan proses penting (Q1) mendapat layanan lebih cepat. (4) Throughput ketiga algoritma identik (0.357 proses/ms) karena total waktu eksekusi yang sama dan beban kerja yang ringan. (5) Proses dengan prioritas rendah (P2) mengalami *starvation* pada *Priority Scheduling* dan MLQ dengan WT 11 ms, menunjukkan *trade-off* antara efisiensi dan *fairness*.

Berdasarkan hasil simulasi, terlihat bahwa setiap algoritma penjadwalan menunjukkan karakteristik kinerja yang berbeda. Algoritma Round Robin memberikan nilai response time yang relatif lebih baik, sehingga lebih sesuai digunakan pada sistem time-sharing yang menuntut respons cepat terhadap proses interaktif. Sementara itu, *Priority Scheduling* menunjukkan kinerja yang efektif dalam menangani proses dengan tingkat kepentingan tinggi, namun berpotensi menimbulkan *starvation* pada proses berprioritas rendah apabila tidak disertai mekanisme penyeimbang. Di sisi lain, *Multi-Level Queue* mampu mengelola beban kerja secara lebih terstruktur dan menunjukkan performa yang baik dalam hal throughput, sehingga cocok diterapkan pada sistem dengan klasifikasi proses yang jelas, seperti pemisahan antara proses sistem dan proses pengguna.

Hasil ini secara langsung menjawab tujuan penelitian, yaitu menganalisis kinerja algoritma penjadwalan CPU, dengan menunjukkan adanya trade-off kinerja pada masing-masing algoritma terhadap indikator performa yang digunakan dan pada kondisi sistem yang berbeda.

Implikasi praktis dari hasil pengujian ini menunjukkan bahwa tidak terdapat satu algoritma penjadwalan yang paling unggul untuk seluruh kondisi, melainkan

pemilihan algoritma perlu disesuaikan dengan karakteristik beban kerja dan kebutuhan sistem, seperti kebutuhan respons cepat, prioritas proses, atau efisiensi penyelesaian proses secara keseluruhan.

## KESIMPULAN

Penelitian ini membandingkan kinerja tiga algoritma penjadwalan CPU *Multi-Level Queue (MLQ)*, *Round Robin (RR)*, dan *Priority Scheduling* berdasarkan *turnaround time*, *waiting time*, *response time*, dan *throughput* melalui tiga skenario beban kerja (seimbang, berat, dan ringan).

Hasil simulasi menunjukkan bahwa *Priority Scheduling* memberikan kinerja paling efisien, dengan *turnaround time* 21% lebih baik dari *Round Robin* dan 13% lebih baik dari *Multi-Level Queue*, serta *waiting time* 29% lebih baik dari *Round Robin* dan 17% lebih baik dari *Multi-Level Queue*. *Round Robin* unggul pada *response time*, menjadikannya ideal untuk sistem interaktif yang menuntut respons cepat. Sementara itu, *Multi-Level Queue* menampilkan performa seimbang di seluruh parameter dengan kombinasi efisiensi dan pemerataan sumber daya CPU.

Ketiga algoritma memiliki *throughput* yang relatif sama, namun *Priority Scheduling* dan *MLQ* berpotensi menyebabkan *starvation* pada proses berprioritas rendah. Secara keseluruhan, *Priority Scheduling* paling efisien, *Round Robin* paling responsif, dan *MLQ* paling stabil untuk sistem operasi dengan beban kerja campuran.

## Saran

Berdasarkan hasil analisis dan temuan penelitian ini, beberapa rekomendasi penelitian selanjutnya:

- Adanya kombinasi mekanisme *Priority Scheduling* dan *Round Robin* dalam bentuk algoritma hibrid dinamis dapat menjadi solusi untuk mengurangi *starvation* sekaligus mempertahankan efisiensi waktu dan responsivitas sistem.
- Menambahkan parameter CPU *utilization* dan *context switching overhead* untuk memberikan gambaran yang lebih komprehensif terhadap efisiensi sistem.

## DAFTAR PUSTAKA

- Abas Ali Pangera, D. A. (2010). *Sistem Operasi* (O. HS (ed.)). ANDI Yogyakarta.
- 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*, 7(1), 180–186. <https://doi.org/10.31943/teknokom.v7i1.176>
- Ardi Wijaya, G. (2021). *Implementasi Algoritma Round Robin Pada Sistem Penjadwalan Mata Kuliah ( Studi Kasus : Universitas Muhammadiyah Bengkulu)*. *Jurnal Informatika UPGRIS (JIU)*, 4(1), 64–71. <http://journal.gris.ac.id/index.php/JIU/article/view/2336/1885up>
- Dalmia, R., Sinha, A., Verma, R., & Gupta, P. K. (2022). *Dynamic Ready Queue Based Process Priority Scheduling Algorithm*. *ArXiv*.
- Ghazy, N., & Eldahshan, K. (2022). *A New Round Robin Algorithm for Task Scheduling in Real-time System A New Round Robin Algorithm for Task Scheduling in Real-time System*. *International Journal of Intelligent Engineering & Systems*, October.

- <https://doi.org/10.22266/ijies2022.1031.59>
- Goel, N., & Garg, R. B. (2022). *A Comparative Study of CPU Scheduling Algorithms*. International Journal of Graphics & Image Processing |Vol 2|issue, 4(4), 245. [www.ifrsa.org](http://www.ifrsa.org)
- Huda, N. (2023). *Penerapan Metode Multilevel Queue Antrian Pendaftaran Pasien pada Klinik Talang Kelapa Berbasis Android Method Application Multilevel Queue Patient Registration Queue at The Talang Kelapa Clinic Based Android*. SMATIKA : STIKI Informatika Jurnal, 13, 62–72.
- Iqbal, M., Ullah, Z., Khan, I. A., Aslam, S., Shaheer, H., Humayon, M., Salahuddin, M. A., & Mehmood, A. (2023). *Optimizing Task Execution: The Impact of Dynamic Time Quantum and Priorities on Round Robin Scheduling*. Future Internet, 15(3). <https://doi.org/10.3390/fi15030104>
- Koval, N., Alistarh, D., & Nadiradze, G. (2021). *Multi-Queues Can Be State-Of-The-Art Priority Schedulers*. *ArXiv:2109.00657v1*.
- Lei Li. (2024). *CPU Scheduling Algorithms in Operating Systems*. Highlights in Science, Engineering and Technology, 120, 8–13.
- Mutasar. (2020). *Optimasi Alokasi Waktu Proses CPU Pada Linux*. Jurnal Ilmu Komputer Dan Multimedia, UNIKI, 1(2), 14–16.
- 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://doi.org/10.11591/eei.v10i5.2812>
- Priambudi, A., Wijaya, Y. F., & Darusalam, U. (2024). *Aplikasi Mobile Pendaftaran Pasien Klinik Berbasis Algoritma Multilevel Queue Dan FIFO Untuk Meningkatkan Layanan Antrian*. Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK), 9, 472–480.
- Putra, M. T. D., Hidayat, H., Septian, N., & Afriani, T. (2021). *Analisis Perbandingan Algoritma Penjadwalan CPU First Come First Serve (FCFS) Dan Round Robin*. Building of Informatics, Technology and Science (BITS), 3(3), 207–212. <https://doi.org/10.47065/bits.v3i3.1047>
- Sharma, C., Sharma, S., Kautish, S., Alsallami, S. A. M., Khalil, E. M., & Mohamed, A. W. (2022). *A new median-average round Robin scheduling algorithm: An optimal approach for reducing turnaround and waiting time*. Alexandria Engineering Journal, 61(12), 10527–10538. <https://doi.org/10.1016/j.aej.2022.04.006>
- Simarmata, A. M., & Harahap, M. (2021). *Sistem Penjadwalan Iklan Menggunakan Metode Priority Scheduling pada PT. Kidung Indah Selaras Suara (Radio Kiss FM) untuk Efektivitas dan Efisiensi Produksi Siaran*. Jurnal Teknologi Dan Ilmu Komputer Prima (JUTIKOMP), 2(1), 66–75. <https://doi.org/10.34012/jutikomp.v2i1.564>
- Widiarto, W., Maheswari, D., Sari, D. P., & Arianto, K. J. (2024). *Implementasi Algoritma Round Robin dan Priority Pada Sistem Antrian Rumah Sakit*. 14(2), 507–513.