

(Artikel Penelitian/Ulasan)

# Implementasi Program Penjadwalan CPU Menggunakan Algoritma Round Robin Untuk Penyelesaian Soal Penjadwalan Proses

Ridho Affandi<sup>1</sup>, Zulfahmi Indra<sup>2</sup>, Syti Salwaa Nafisah<sup>3\*</sup>, Shaqila Rahmayani<sup>4</sup>

<sup>12345</sup>Ilmu Komputer, FMIPA, Universitas Negeri Medan Jalan Willem Iskandar, Pasar V Medan Estate, Percut Sei Tuan, Deli Serdang

<sup>1</sup>[ridhoaffandi997@gmail.com](mailto:ridhoaffandi997@gmail.com), <sup>2</sup>[zulfahmi.indra@unimed.ac.id](mailto:zulfahmi.indra@unimed.ac.id), <sup>3</sup>[sytilsalwaanafisah@gmail.com](mailto:sytisalwaanafisah@gmail.com),

<sup>4</sup>[shaqilarahmayani269@gmail.com](mailto:shaqilarahmayani269@gmail.com)

\* Penulis : Syti Salwaa Nafisah

**Abstract:** In operating systems, CPU scheduling plays a crucial role in determining the execution order of processes to ensure efficient and fair processor utilization. One of the most commonly used scheduling algorithms in *time-sharing* systems is the *Round Robin* algorithm. This algorithm works by allocating an equal time slice (*time quantum*) to each process and then rotating unfinished processes back into the ready queue for subsequent execution. This mechanism makes *Round Robin* fair and responsive to all processes. This study discusses the implementation of a CPU scheduling program using the *Round Robin* algorithm in operating systems to solve process scheduling problems. The developed program simulates various process scenarios with different *arrival times*, *burst times*, and *time quanta* to analyze their effects on *waiting time* and *turnaround time*. The simulation results indicate that the selection of an appropriate *time quantum* significantly affects system performance: a too-small *time quantum* increases the frequency of *context switching*, while a too-large *time quantum* reduces system responsiveness. With an optimal *time quantum*, the *Round Robin* algorithm achieves a balance between efficiency and fairness in CPU process management, making it highly effective for modern *time-sharing* operating systems.

**Keywords:** Operating System, CPU Scheduling, Round Robin Algorithm, Time Quantum, Waiting Time, Turnaround Time

**Abstrak:** Dalam sistem operasi, penjadwalan CPU memiliki peranan penting untuk menentukan urutan eksekusi proses agar penggunaan prosesor menjadi efisien dan adil. Salah satu algoritma penjadwalan yang banyak digunakan pada sistem *time-sharing* adalah algoritma *Round Robin*. Algoritma ini bekerja dengan memberikan jatah waktu (*time quantum*) yang sama kepada setiap proses, kemudian menggilir proses yang belum selesai untuk dieksekusi kembali. Mekanisme ini membuat *Round Robin* dianggap adil dan responsif terhadap semua proses. Penelitian ini membahas implementasi program penjadwalan CPU menggunakan algoritma *Round Robin* dalam sistem operasi untuk penyelesaian soal penjadwalan proses. Program yang dikembangkan mensimulasikan berbagai skenario proses dengan variasi *arrival time*, *burst time*, dan *time quantum* untuk menganalisis pengaruhnya terhadap *waiting time* dan *turnaround time*. Hasil simulasi menunjukkan bahwa pemilihan *time quantum* yang tepat sangat berpengaruh terhadap performa sistem: *time quantum* yang terlalu kecil meningkatkan frekuensi *context switching*, sedangkan *time quantum* yang terlalu besar menurunkan responsivitas sistem. Dengan nilai *time quantum* yang optimal, algoritma *Round Robin* mampu memberikan keseimbangan antara efisiensi dan keadilan dalam pengelolaan proses CPU, sehingga sangat efektif diterapkan pada sistem operasi modern berbasis *time-sharing*.

**Kata kunci:** Sistem Operasi, Penjadwalan CPU, Algoritma Round Robin, Time Quantum, Waiting Time, Turnaround Time

Diterima: 12 Oktober 2025  
Direvisi: 3 November 2025  
Diterima: 4 November 2025  
Diterbitkan: 5 November 2025  
Versi sekarang: Januari 2026



Hak cipta: © 2025 oleh penulis.  
Diserahkan untuk kemungkinan  
publikasi akses terbuka  
berdasarkan syarat dan ketentuan  
lisensi Creative Commons  
Attribution (CC BY SA) (  
<https://creativecommons.org/licenses/by-sa/4.0/>)

## 1. Pendahuluan

Sistem operasi adalah perangkat lunak sistem komputer yang dapat membantu perangkat keras dalam menjalankan fungsi-fungsinya termasuk manajemen proses atau kontrol proses. Proses adalah keadaan ketika suatu program dijalankan. Saat komputer sedang berjalan, ada banyak proses yang berjalan serentak. Ketika suatu proses dibuat maka proses tersebut dapat memperoleh sumber daya tersebut seperti waktu CPU, memori, file, atau perangkat I/O. (Putri 2021)

Penjadwalan CPU (*CPU Scheduling*) merupakan komponen utama dalam sistem operasi yang berfungsi mengatur urutan eksekusi proses agar penggunaan sumber daya prosesor menjadi efisien dan adil. Dalam sistem *multiprogramming*, banyak proses bersaing untuk mendapatkan waktu CPU, sehingga dibutuhkan algoritma penjadwalan yang dapat menyeimbangkan antara waktu tunggu, efisiensi, dan responsivitas sistem (Eka et al. 2022)

Dalam penerapannya, sistem operasi menggunakan beberapa algoritma penjadwalan CPU seperti First Come First Serve (FCFS), Shortest Job First (SJF), Priority Scheduling, dan Round Robin (RR). Masing-masing algoritma memiliki kelebihan dan kekurangan dalam hal efisiensi dan keadilan waktu proses. Algoritma Round Robin merupakan algoritma yang paling banyak digunakan dalam sistem operasi *time-sharing* karena memberikan keadilan terhadap semua proses dengan membagi waktu CPU secara merata. (Putra and Purnomo 2022)

Round Robin adalah algoritma penjadwalan yang preemptive dan berlangsung tanpa prioritas. Pada algoritma penjadwalan ini, secara bergiliran berdasarkan antrian, prosesor melayani sejenak setiap proses. Secara berturut-turut, proses yang telah dilayani prosesor dan belum rampung akan kembali ke akhir antrian yang ada pada saat itu, sehingga penggiliran ini berputar seperti gelang. Efisiensi algoritma Round Robin sangat bergantung pada penentuan nilai *time quantum* yang digunakan. Nilai *time quantum* yang terlalu kecil dapat menyebabkan frekuensi *context switching* meningkat, sehingga memperbesar beban kerja prosesor. Sebaliknya, jika *time quantum* terlalu besar, maka sistem menjadi kurang responsif dan proses dengan waktu eksekusi singkat harus menunggu lebih lama. Oleh karena itu, pemilihan nilai *time quantum* yang optimal menjadi faktor penting untuk menjaga keseimbangan antara efisiensi dan keadilan sistem. (Syauqi and Siregar 2025)

Dalam proses pembelajaran salah satu topik yang sering dianggap kompleks oleh mahasiswa adalah penjadwalan CPU (*CPU Scheduling*), terutama pada algoritma Round Robin. Kompleksitas ini muncul karena konsepnya melibatkan perhitungan matematis yang cukup rumit, seperti waktu tunggu (*waiting time*), waktu penyelesaian (*turnaround time*), serta pengaruh parameter *arrival time* dan *burst time* terhadap performa sistem. Pemahaman secara teoritis saja sering kali tidak cukup untuk menjelaskan bagaimana CPU melakukan pengalokasian waktu eksekusi secara bergilir terhadap setiap proses yang sedang berjalan dalam sistem. (Putra and Fadjriya 2021)

Oleh karena itu, diperlukan sebuah program simulasi interaktif yang dapat memvisualisasikan cara kerja algoritma Round Robin secara sistematis dan mudah dipahami. Melalui simulasi tersebut, pengguna dapat mengamati secara langsung bagaimana variasi nilai parameter *time quantum*, *arrival time*, dan *burst time* memengaruhi hasil penjadwalan CPU, baik dari sisi efisiensi maupun keadilan. Dengan adanya visualisasi tersebut, proses pembelajaran menjadi lebih efektif karena mahasiswa dapat melihat hubungan antara teori dan implementasi nyata secara langsung di layar komputer.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk mengimplementasikan program penjadwalan CPU menggunakan algoritma Round Robin dalam penyelesaian soal penjadwalan proses. Program ini dirancang agar mampu menerima input berupa *arrival time*, *burst time*, dan *time quantum*, kemudian menghitung secara otomatis hasil berupa *waiting time* dan *turnaround time* untuk setiap proses. Selain itu, penelitian ini juga menganalisis pengaruh variasi nilai *time quantum* terhadap performa sistem dalam konteks efisiensi dan keadilan alokasi CPU.

Dengan adanya penelitian ini, diharapkan dapat memberikan kontribusi nyata dalam bidang pendidikan dan penelitian sistem operasi, khususnya dalam membantu mahasiswa maupun praktisi memahami konsep penjadwalan CPU secara lebih praktis. Program simulasi yang dikembangkan juga diharapkan dapat menjadi media pembelajaran interaktif yang mempermudah analisis dan penyelesaian soal-soal penjadwalan proses berbasis algoritma Round Robin, serta memperkuat pemahaman terhadap konsep dasar manajemen proses dalam sistem operasi modern.

## 2. Tinjauan Pustaka

### 2.1 Sistem Operasi

Sistem operasi merupakan perangkat lunak utama yang berfungsi mengelola seluruh sumber daya komputer dan memberikan layanan kepada pengguna maupun aplikasi. Salah satu fungsi penting sistem operasi adalah mengatur penggunaan prosesor melalui mekanisme penjadwalan CPU. Sistem operasi bertanggung jawab dalam melakukan manajemen proses, manajemen memori, serta manajemen perangkat keras agar penggunaan CPU dapat optimal. Penjadwalan yang baik membantu meningkatkan efisiensi sistem secara keseluruhan. Sistem operasi yang efisien harus mampu mengelola proses-proses secara adil dan responsif agar setiap proses memperoleh waktu eksekusi yang proporsional. (Insan et al. 2025)

### 2.2 Penjadwalan CPU

Penjadwalan CPU adalah proses menentukan urutan eksekusi dari sejumlah proses yang siap dijalankan agar penggunaan prosesor menjadi lebih efektif. Tujuan utama penjadwalan CPU adalah memaksimalkan throughput, meminimalkan waktu tunggu (*waiting time*), waktu penyelesaian (*turnaround time*), serta memastikan keadilan antar proses. Pemilihan algoritma penjadwalan CPU yang tepat dapat berpengaruh besar terhadap kinerja sistem operasi, khususnya dalam hal efisiensi waktu eksekusi dan pengelolaan sumber daya CPU. Penelitian mereka menunjukkan bahwa algoritma Round Robin memberikan hasil yang cukup baik pada sistem dengan jumlah proses yang banyak dan kebutuhan eksekusi yang hampir seimbang. (Al Ghifari and Siregar n.d.)

### 2.3 Algoritma Round Robin

Algoritma Round Robin merupakan algoritma penjadwalan CPU yang memberikan setiap proses waktu eksekusi yang sama secara bergiliran. Setiap proses akan dijalankan selama jangka waktu tertentu yang disebut *time quantum*. Jika dalam waktu tersebut proses belum selesai, maka proses akan dimasukkan kembali ke antrian untuk mendapatkan giliran berikutnya. Dengan demikian, tidak ada proses yang dibiarkan menunggu terlalu lama, dan sistem menjadi lebih responsif terhadap semua proses. Algoritma Round Robin menjadi pilihan ideal bagi sistem operasi yang membutuhkan interaktivitas tinggi seperti sistem *time-sharing*. Kelebihan algoritma ini adalah keadilannya, di mana setiap proses mendapatkan kesempatan yang sama untuk dijalankan. Namun, jika nilai *time quantum* terlalu kecil, maka akan terjadi *context switching* yang terlalu sering, yang justru menurunkan efisiensi CPU. Sebaliknya, jika terlalu besar, algoritma ini akan menyerupai FCFS (First Come First Served) dan menyebabkan peningkatan *waiting time* untuk proses yang lebih pendek. Untuk meningkatkan performa algoritma Round Robin, *time quantum* dapat diubah secara dinamis sesuai dengan panjang *burst time* dari proses. Dengan pendekatan ini, proses dengan durasi pendek dapat selesai lebih cepat tanpa harus menunggu giliran berikutnya, sehingga meningkatkan efisiensi waktu penyelesaian sistem secara keseluruhan. (Parinduri and Hutagalung 2019)

### 2.4 Time Quantum

Time Quantum adalah durasi waktu tetap yang diberikan CPU kepada setiap proses untuk dijalankan sebelum berpindah ke proses berikutnya. Pemilihan nilai *time quantum* yang tepat merupakan aspek krusial dalam algoritma Round Robin. Bila *quantum* terlalu pendek, sistem akan sering melakukan *context switching*, mengakibatkan pemborosan waktu CPU untuk

pergantian proses. Sebaliknya, jika terlalu panjang, waktu tanggap sistem akan menurun. Penggunaan dynamic time quantum yang menyesuaikan diri dengan rata-rata *burst time* dari proses terbukti dapat mengurangi waktu tunggu rata-rata (*waiting time*) serta meningkatkan efisiensi keseluruhan sistem. Penelitian ini menggunakan pendekatan *ascending burst time ordering* agar proses dengan waktu eksekusi lebih pendek diselesaikan terlebih dahulu, sehingga memperbaiki *turnaround time* total sistem.(Widiarto et al. n.d.)

## 2.5 Waiting Time

Waiting Time adalah waktu yang dihabiskan oleh proses dalam antrian sebelum memperoleh giliran eksekusi CPU. Parameter ini sering digunakan sebagai ukuran efisiensi algoritma penjadwalan. Dalam konteks Round Robin, *waiting time* dipengaruhi oleh panjang *time quantum* dan jumlah proses yang menunggu.(Belferik and Banjarnahor 2025a)dalam penelitiannya pada *Jurnal JDMIS* meneliti perbandingan algoritma Round Robin dengan Shortest Remaining First (SRF). Hasil penelitian menunjukkan bahwa meskipun algoritma SRF memiliki *waiting time* lebih kecil, Round Robin tetap unggul dalam hal keadilan dan pemerataan alokasi waktu CPU. Penelitian ini juga menunjukkan bahwa modifikasi *time quantum* secara adaptif dapat membantu menurunkan *waiting time* rata-rata hingga 20% dibandingkan versi konvensional.

## 2.6 Turnaround Time

Turnaround time merupakan salah satu parameter utama dalam mengevaluasi performa algoritma penjadwalan CPU. Konsep ini menggambarkan lamanya waktu yang dibutuhkan oleh suatu proses sejak proses tersebut mulai masuk ke dalam sistem hingga proses tersebut selesai sepenuhnya. Dengan kata lain, turnaround time menunjukkan seberapa cepat sistem operasi mampu menyelesaikan seluruh tugas yang ada dalam antrian penjadwalan. Parameter ini penting karena dapat memberikan gambaran umum mengenai efisiensi waktu eksekusi dan tingkat respons sistem terhadap beban kerja yang diberikan. Menurut (Belferik and Banjarnahor 2025b) semakin kecil nilai turnaround time, maka semakin baik performa sistem karena setiap proses dapat diselesaikan dalam waktu yang lebih singkat dan efisien.

Dalam algoritma Round Robin, turnaround time sangat dipengaruhi oleh nilai *time quantum* yang digunakan. *Time quantum* merupakan batas waktu eksekusi sementara yang diberikan kepada setiap proses sebelum giliran proses tersebut dialihkan ke proses berikutnya. Jika nilai *time quantum* terlalu kecil, maka sistem akan melakukan *context switching* secara berulang-ulang, yang menyebabkan penambahan waktu total eksekusi dan memperbesar nilai turnaround time. Sebaliknya, jika *time quantum* terlalu besar, maka proses pertama akan mendominasi CPU untuk waktu yang lama dan proses lainnya harus menunggu, sehingga menyebabkan peningkatan waktu tunggu dan menurunkan efisiensi sistem secara keseluruhan. Hasil penelitian yang dilakukan oleh Purnomo dan Putra (2024) menunjukkan bahwa pemilihan *time quantum* yang ideal berpengaruh signifikan terhadap penurunan nilai turnaround time pada algoritma Round Robin.(Hermanto et al. 2025)

## 3. Metode Penelitian

Metode penelitian yang digunakan dalam studi ini adalah penelitian terapan dengan pendekatan pengembangan sistem berbasis web interaktif. Penelitian ini difokuskan pada implementasi algoritma *Round Robin* dalam proses penjadwalan CPU untuk menganalisis pengaruh parameter *arrival time*, *burst time*, dan *time quantum* terhadap performa sistem. Tujuan utama penelitian ini adalah untuk menghasilkan sebuah aplikasi simulasi yang mampu membantu pengguna memahami cara kerja algoritma *Round Robin* secara visual, akurat, dan mudah dipahami. Pendekatan penelitian ini dipilih karena selain menghasilkan pemahaman teoretis, juga menghasilkan produk nyata berupa aplikasi web yang dapat digunakan sebagai

media pembelajaran maupun alat bantu analisis sistem operasi. Aplikasi simulasi dikembangkan menggunakan teknologi HTML, CSS, dan JavaScript, serta diimplementasikan melalui platform Vercel sehingga dapat dijalankan langsung melalui browser tanpa instalasi tambahan. Sistem ini menampilkan hasil perhitungan secara otomatis, seperti *waiting time* dan *turnaround time*, serta divisualisasikan dalam bentuk diagram Gantt Chart real-time, yang menggambarkan urutan eksekusi proses secara bergilir sesuai prinsip algoritma *Round Robin*.

Tahapan penelitian yang dilakukan dalam pengembangan aplikasi simulasi ini terdiri dari beberapa langkah sistematis, yaitu:

#### 1. Studi Literatur

Pada tahap awal dilakukan pengumpulan referensi dan teori yang relevan terkait sistem operasi, penjadwalan CPU, dan algoritma *Round Robin*. Literatur diperoleh dari buku, jurnal nasional, dan sumber daring yang mendukung pemahaman tentang konsep *time quantum*, *waiting time*, serta *turnaround time*. Tahap ini menjadi dasar dalam merancang logika dan struktur perhitungan algoritma yang diterapkan dalam aplikasi.

#### 2. Analisis Kebutuhan Sistem

Analisis dilakukan untuk mengidentifikasi kebutuhan fungsional dan nonfungsional dari sistem yang akan dikembangkan. Kebutuhan utama meliputi kemampuan sistem untuk menerima input data berupa *arrival time*, *burst time*, dan *time quantum*, serta menghitung hasil keluaran berupa *waiting time* dan *turnaround time*. Selain itu, sistem juga harus mampu menampilkan hasil simulasi secara visual melalui Gantt Chart agar pengguna dapat dengan mudah memahami urutan eksekusi proses.

#### 3. Perancangan Sistem

Tahap ini meliputi pembuatan desain alur kerja program dan struktur antarmuka pengguna. Desain dibuat sederhana dan interaktif agar mudah digunakan oleh mahasiswa atau peneliti yang ingin mempelajari algoritma *Round Robin*. Pada tahap ini dirancang pula logika dasar algoritma, termasuk bagaimana sistem akan menangani proses yang belum selesai dan dikembalikan ke antrian (*ready queue*).

#### 4. Implementasi Sistem

Implementasi dilakukan dengan membangun aplikasi berbasis web menggunakan JavaScript sebagai bahasa pemrograman utama. Algoritma *Round Robin* diimplementasikan dalam kode untuk mengatur eksekusi proses berdasarkan jatah waktu (*time quantum*) yang diberikan secara bergiliran. Program ini kemudian diintegrasikan dengan antarmuka berbasis HTML dan CSS untuk menghasilkan tampilan yang responsif. Aplikasi dikembangkan dan diunggah ke platform Vercel dengan alamat:

<https://simulator-penjadwalan-round-robin.vercel.app/>

#### 5. Pengujian Sistem

Setelah implementasi selesai, dilakukan pengujian dengan beberapa skenario simulasi untuk memastikan program berjalan sesuai dengan teori algoritma *Round Robin*. Dua skenario utama diuji, yaitu:

➤ Skenario 1: Semua proses tiba pada waktu yang sama (*arrival time* serentak).

➤ Skenario 2: Proses tiba secara bertahap dengan *arrival time* berbeda.

Hasil dari setiap simulasi dibandingkan dengan perhitungan manual untuk memverifikasi keakuratan sistem.

#### 6. Analisis Data dan Evaluasi

Data hasil pengujian dianalisis dengan menghitung nilai rata-rata *waiting time* dan *turnaround time* dari setiap skenario. Analisis ini digunakan untuk mengetahui bagaimana variasi nilai *time quantum* memengaruhi efisiensi dan responsivitas sistem. Evaluasi dilakukan dengan menilai kecepatan perhitungan, keakuratan hasil, dan k\*3emudahan penggunaan aplikasi sebagai media pembelajaran interaktif.

Dengan mengikuti tahapan tersebut, penelitian ini menghasilkan sebuah sistem simulasi berbasis web yang mampu menampilkan proses penjadwalan CPU menggunakan algoritma *Round Robin* secara akurat dan interaktif. Pendekatan ini diharapkan dapat membantu pengguna memahami hubungan antara teori dan praktik penjadwalan CPU, sekaligus memberikan kontribusi dalam pengembangan media pembelajaran berbasis teknologi di bidang sistem operasi.

#### 4. Hasil dan Pembahasan

Program simulasi penjadwalan CPU menggunakan algoritma Round Robin ini memiliki keunggulan utama pada kemudahan akses dan interaktivitasnya karena berbasis web. Pengguna dapat langsung melakukan simulasi tanpa instalasi tambahan melalui tautan <https://simulator-penjadwalan-round-robin.vercel.app/>. Program mampu menghitung otomatis waktu tunggu (*waiting time*), waktu penyelesaian (*turnaround time*), serta menampilkan hasilnya dalam bentuk diagram Gantt Chart secara real-time. Tampilan yang sederhana namun informatif memudahkan pengguna memahami konsep *time sharing* dalam algoritma Round Robin. Selain cepat dan akurat, aplikasi ini juga ringan karena seluruh proses dijalankan di sisi klien, sehingga sangat efektif digunakan sebagai alat bantu pembelajaran interaktif untuk memahami mekanisme penjadwalan CPU.

##### 1. Tampilan Awal Program

Tampilan awal Simulator Penjadwalan Round Robin menampilkan halaman utama tempat pengguna memasukkan data proses yang akan disimulasikan. Pada bagian atas terdapat kolom Time Quantum untuk menentukan lama waktu eksekusi tiap proses sebelum bergantian. Di bawahnya terdapat tabel berisi kolom Proses, Waktu Tiba (Arrival Time), dan Waktu Eksekusi (Burst Time) sebagai input utama. Contoh pada tampilan menunjukkan empat proses (P1–P4) dengan waktu tiba 0 dan waktu eksekusi 1, menandakan semua proses datang bersamaan. Tiga tombol di bagian bawah Tambah Proses, Hapus Proses Terakhir, dan Reset Tabel digunakan untuk mengatur data input. Tampilan ini dirancang sederhana, interaktif, dan mudah dipahami agar pengguna dapat langsung memulai simulasi penjadwalan CPU dengan cepat.

**Simulator Penjadwalan Round Robin** ⚙️

Masukkan data proses pada tabel di bawah, tentukan Time Quantum, lalu klik "Hitung" untuk melihat simulasi.

Time Quantum:

Proses	Waktu Tiba (Arrival Time)	Waktu Eksekusi (Burst Time)
P1	<input type="text" value="0"/>	<input type="text" value="1"/>
P2	<input type="text" value="0"/>	<input type="text" value="1"/>
P3	<input type="text" value="0"/>	<input type="text" value="1"/>
P4	<input type="text" value="0"/>	<input type="text" value="1"/>

Gambar 1

##### 2. Studi Kasus

###### 2.1. Skenario 1 Semua Proses Tiba Bersamaan

###### 2.1.1 Tampilan Input Data Proses

Gambar ini menampilkan halaman awal simulator, tempat pengguna memasukkan data proses yang akan disimulasikan. Nilai *Time Quantum* diatur sebesar 3, dan keempat proses (P1–P4) memiliki waktu tiba 0 dengan waktu eksekusi berbeda (10, 5, 8, dan 6). Tampilan ini menunjukkan tahap awal sebelum simulasi dijalankan.

**Simulator Penjadwalan Round Robin**

Masukkan data proses pada tabel di bawah, tentukan Time Quantum, lalu klik "hitung" untuk melihat simulasi.

Time Quantum: 3

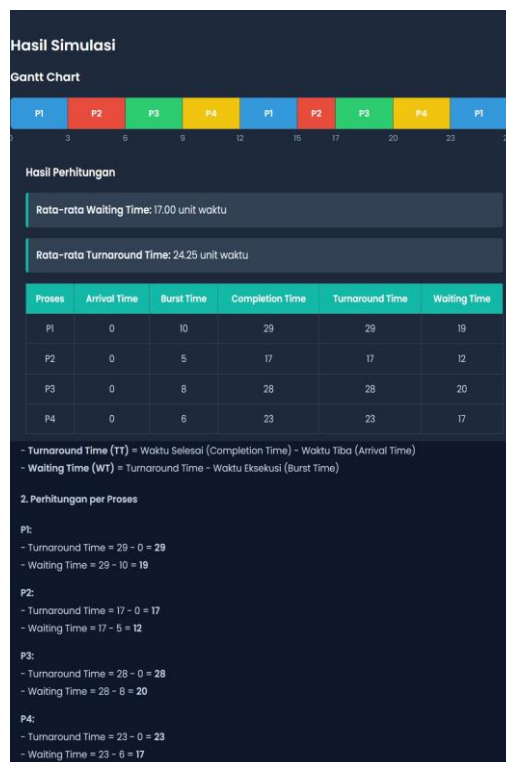
Proses	Waktu Tiba (Arrival Time)	Waktu Eksekusi (Burst Time)
P1	0	10
P2	0	5
P3	0	8
P4	0	6

Tambah Proses Hapus Proses Terakhir Reset Tabel

Gambar 2.1.1

### 2.1.2 Hasil Simulasi

Hasil simulasi algoritma Round Robin dengan *Time Quantum* sebesar 3. Pada bagian atas terlihat diagram Gantt Chart yang menunjukkan urutan eksekusi proses P1, P2, P3, dan P4 secara bergantian sesuai jatah waktu CPU. Warna berbeda menggambarkan proses yang dijalankan dalam tiap interval waktu, mencerminkan prinsip *time sharing* pada penjadwalan CPU. Bagian bawah gambar memperlihatkan tabel hasil perhitungan yang mencakup Completion Time, Turnaround Time, dan Waiting Time untuk setiap proses, dengan rata-rata waktu tunggu sebesar 17,00 unit waktu dan rata-rata waktu penyelesaian sebesar 24,25 unit waktu. Perhitungan manual yang ditampilkan di sisi akhir menunjukkan kesesuaian hasil simulasi dengan teori, di mana proses P1 memiliki waktu tunggu 19 unit, P2 sebesar 12 unit, P3 sebesar 20 unit, dan P4 sebesar 17 unit. Hasil ini membuktikan bahwa simulator mampu menampilkan perhitungan secara akurat dan interaktif, serta menggambarkan kinerja algoritma Round Robin dalam membagi waktu CPU secara adil di antara proses yang memiliki waktu eksekusi berbeda.



Gambar 2.1.2

## 2.2. Skenario 2 Arrival time bertahap

### 2.2.1. Tampilan Input Data Proses dengan Arrival Time Berbeda

Gambar ini menampilkan tahap input data proses pada simulator Round Robin dengan *Time Quantum* sebesar 3. Pada skenario ini, setiap proses memiliki

waktu kedatangan berbeda—P1 pada waktu 0, P2 pada waktu 2, P3 pada waktu 4, dan P4 pada waktu 6—serta waktu eksekusi bervariasi yaitu 6, 4, 5, dan 7 satuan waktu. Variasi ini digunakan untuk menggambarkan kondisi penjadwalan yang lebih realistis, di mana proses tidak selalu tiba bersamaan. Tampilan ini memperlihatkan bagaimana pengguna dapat menyesuaikan data untuk melihat pengaruh perbedaan waktu tiba dan durasi eksekusi terhadap hasil simulasi penjadwalan CPU.

Simulator Penjadwalan Round Robin

Masukkan data proses pada tabel di bawah, tentukan Time Quantum, lalu klik "Hitung" untuk melihat simulasi.

Time Quantum: 3

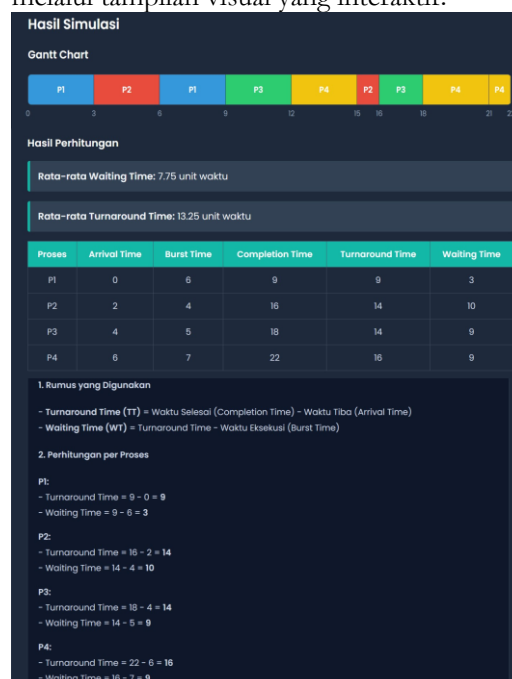
Proses	Waktu Tiba (Arrival Time)	Waktu Eksekusi (Burst Time)
P1	0	6
P2	2	4
P3	4	5
P4	6	7

Tambah Proses Hapus Proses Terakhir Reset Tabel

Gambar 2.2.1

### 2.2.2. Hasil Simulasi

Hasil simulasi algoritma Round Robin dengan *Time Quantum* sebesar 3, di mana diagram Gantt Chart memperlihatkan urutan eksekusi proses P1, P2, P3, dan P4 yang dijalankan secara bergantian sesuai waktu kedatangannya. Setiap warna pada diagram menunjukkan giliran eksekusi masing-masing proses dalam pembagian waktu CPU yang adil. Berdasarkan tabel hasil perhitungan, diperoleh rata-rata waktu tunggu sebesar 7,75 unit waktu dan rata-rata waktu penyelesaian sebesar 13,25 unit waktu, yang menunjukkan efisiensi sistem dalam menangani proses dengan waktu tiba dan durasi eksekusi berbeda. Perhitungan manual memperkuat hasil simulasi dengan nilai Turnaround Time dan Waiting Time masing-masing, yaitu P1 (9, 3), P2 (14, 10), P3 (14, 9), dan P4 (16, 9). Nilai-nilai ini membuktikan bahwa simulator bekerja akurat dan sesuai teori, mampu menampilkan proses penjadwalan CPU secara adil, efisien, serta mudah dipahami melalui tampilan visual yang interaktif.



Gambar 2.2.2



## 5. Kesimpulan

Penelitian ini berhasil mengimplementasikan algoritma Round Robin dalam bentuk simulasi penjadwalan CPU berbasis web yang interaktif dan mudah digunakan. Aplikasi mampu menghitung *waiting time* dan *turnaround time* secara otomatis serta menampilkan hasilnya melalui *Gantt Chart* real-time, sehingga membantu pengguna memahami prinsip *time sharing* pada sistem operasi. Hasil simulasi menunjukkan bahwa nilai *time quantum* sangat memengaruhi performa sistem. *Time quantum* yang terlalu kecil meningkatkan frekuensi *context switching* dan menurunkan efisiensi, sedangkan nilai yang terlalu besar membuat sistem kurang responsif. Dengan pemilihan *time quantum* yang tepat, algoritma Round Robin dapat mencapai keseimbangan antara efisiensi dan keadilan. Secara keseluruhan, aplikasi yang dikembangkan bersifat ringan, akurat, dan efektif sebagai media pembelajaran interaktif. Implementasi ini memberikan kontribusi nyata dalam meningkatkan pemahaman konsep penjadwalan CPU melalui pendekatan praktis berbasis web.

## References

- 1 Belferik, Ronald, and Evander Banjarnahor. 2025a. "Analisis Average Waiting Time Penjadwalan CPU Menggunakan Algoritma Shortest Remaining First Dan Algoritma Round Robin." *JDMIS: Journal of Data Mining and Information Systems* 3(1):43–53. doi:10.54259/jdmis.v3i1.4076.
- 2 Belferik, Ronald, and Evander Banjarnahor. 2025b. "Analisis Average Waiting Time Penjadwalan CPU Menggunakan Algoritma Shortest Remaining First Dan Algoritma Round Robin." *JDMIS: Journal of Data Mining and Information Systems* 3(1):43–53. doi:10.54259/jdmis.v3i1.4076.
- 3 Eka, Meyti, Apriyani Elok Nur, Hamdana Rinanza, and Zulmy Alhamri. 2022. *SISTEM OPERASI*.
- 4 Al Ghifari, Abdurrahman, and Herbert Siregar. n.d. "Systematic Literature Review: Perbandingan Kinerja Algoritma Penjadwalan CPU FCFS, SJF, Round Robin, Dan Priority." *Jurnal Ilmiah Sistem Informasi Dan Ilmu Komputer* 5(2):530–48. doi:10.55606/juisik.v5i2.1404.
- 5 Hermanto, Nandang, Irma Darmayanti, Dimas Saputra, and Aden Hidayatuloh. 2025. "Development of Mobile Application by Applying Content-Based Filtering." *Sinkron* 9(1):232–38. doi:10.33395/sinkron.v9i1.14320.
- 6 Insan, Raihan, Pratama Siagian, Azima Lubis, Muhammad Abid Syuja, and Dedy Kiswanto. 2025. *ANALISIS PERFORMA SISTEM OPERASI MANJARO LINUX DALAM LINGKUNGAN KOMPUTASI DESKTOP VIRTUAL*. Vol. 9.
- 7 Parinduri, Ikhsan, and Siti Nurhabibah Hutagalung. 2019. "Prosiding Seminar Nasional Riset Information Science (SENARIS) Teknik Penjadwalan Prosesor FIFO, SJF Non Preemptive, Round Robin."
- 8 Putra, Tri Dharma, and Andry Fadriya. 2021. "Comparison Between Simple Round Robin and Intelligent Round Robin Algorithms in CPU Scheduling." *IJARCCCE* 10(4). doi:10.17148/ijarccce.2021.10413.
- 9 Putra, Tri Dharma, and Rakhmat Purnomo. 2022. "Simulation of Priority Round Robin Scheduling Algorithm." *Sinkron* 7(4):2170–81. doi:10.33395/sinkron.v7i4.11665.
- 10 Putri, Raissa Amanda. 2021. "Aplikasi Simulasi Algoritma Penjadwalan Sistem Operasi." *Jurnal Teknologi Informasi* 5(1):98–102. doi:10.36294/jurti.v5i1.2215.

- 
- Syauqi, Hanif Ahmad, and Herbert Siregar. 2025. "Systematic Literature Review (SLR): Dampak Modifikasi Algoritma Round-Robin Dalam Penjadwalan CPU Pada Sistem Operasi." *Digital Transformation Technology* 5(1):368–74. doi:10.47709/digitech.v5i1.6171.
- Widiarto, Wisnu, Desinta Maheswari, Dewi Puspita Sari, Kezia Jazzlyn Arianto, and Program Studi Sains Data. n.d. "Implementasi Algoritma Round Robin Dan Priority Pada Sistem Antrian Rumah Sakit."