

SUPER SMART OPTIMIZED ROUND ROBIN

Achmad Teguh Wibowo¹⁾

¹⁾Sistem Informasi Fakultas Sains dan Teknologi
 UIN Sunan Ampel
 Email : atw@uinsby.ac.id

Abstrak

Aspek penting dalam sistem operasi adalah multiprogramming. Multiprogramming adalah proses atau metode yang digunakan untuk mengeksekusi beberapa proses secara bersamaan dalam memori. Tujuan utamanya adalah untuk meminimalkan Average Waiting Time, Average Turnaround Time, dan memaksimalkan penggunaan CPU. Ada berbagai algoritma yang digunakan dalam multiprogramming seperti First Come First Serve (FCFS), Shortest Job First (SJF), Priority Scheduling (PS) dan Round Robin(RR). Diantara semua itu yang paling sering digunakan adalah Round Robin. Round Robin merupakan algoritma penjadwalan yang optimal dengan sistem timeshared. Dalam RR, waktu kuantum bersifat statis dan algoritma ini bergantung pada besarnya kuantum yang dipilih/digunakan. Kuantum inilah yang berpengaruh pada Average Waiting Time dan Average Turnaround Time nantinya. Tujuan dari makalah ini adalah mengusulkan algoritma yang lebih baik daripada Round Robin sederhana dan Smart Optimized Round Robin sebelumnya.

Kata Kunci: Context switching, CPU scheduling, Gantt chart, Response time, Round Robin CPU scheduling algorithm, Turnaround time, Waiting time.

1. PENDAHULUAN

Dalam sistem prosesor tunggal, prosesor hanya dapat menjalankan satu proses sementara proses yang lainnya harus menunggu hingga prosesor bebas lalu kemudian proses selanjutnya dapat dijadwalkan kembali. Tujuan dari multiprogramming adalah untuk menjalankan beberapa proses dalam satu waktu dan untuk memaksimalkan penggunaan CPU. Penjadwalan adalah fungsi fundamental dari sebuah sistem informasi, karena hampir semua sumber daya komputer dijadwalkan terlebih dahulu sebelum diproses. Dengan demikian penjadwalan merupakan desain inti dari sebuah sistem operasi.

2. ALGORITMA PENJADWALAN CPU

Berikut ini adalah algoritma penjadwalan CPU yang mendasar :

- First Come First Serve (FCFS)

Algoritma ini merupakan algoritma penjadwalan yang paling sederhana yang digunakan CPU. Dengan menggunakan algoritma ini setiap proses yang berada pada status ready dimasukkan ke dalam FIFO queue sesuai dengan waktu kedatangannya. Proses yang tiba terlebih dahulu yang akan dieksekusi.

- Shortest Job First (SJF)

Algoritma ini mempunyai cara penjadwalan yang

berbeda dengan FCFS. Dengan algoritma ini maka setiap proses yang ada di ready queue akan dieksekusi berdasarkan burst time terkecil. Hal ini mengakibatkan waiting time yang pendek untuk setiap proses dan karena hal tersebut maka waiting time rata-ratanya juga menjadi pendek, sehingga dapat dikatakan bahwa algoritma ini adalah algoritma yang optimal.

Algoritma ini dibagi menjadi dua, yakni :

- a. Preemptive

Jika ada proses yang sedang dieksekusi oleh CPU dan terdapat proses di ready queue dengan burst time yang lebih kecil daripada proses yang sedang dieksekusi tersebut, maka proses yang sedang dieksekusi oleh CPU akan digantikan oleh proses yang berada di ready queue tersebut. Preemptive SJF sering disebut juga Shortest-Remaining-Time-First scheduling.

- b. Non-Preemptive

CPU tidak memperbolehkan proses yang ada di ready queue untuk menggeser proses yang sedang dieksekusi oleh CPU meskipun proses yang baru tersebut mempunyai burst time yang lebih kecil.

- Priority Scheduling

Priority Scheduling merupakan algoritma penjadwalan yang mendahulukan proses yang memiliki prioritas tertinggi. Setiap proses memiliki prioritasnya masing-masing.

Prioritas suatu proses dapat ditentukan melalui beberapa karakteristik antara lain:

1. Time limit

2. Memory requirement
3. Akses file
4. Perbandingan antara I/O Burst dengan CPU Burst
5. Tingkat kepentingan proses

Priority Scheduling juga dapat dijalankan secara preemptive maupun non-preemptive.

Pada preemptive, jika ada suatu proses yang baru datang memiliki prioritas yang lebih tinggi daripada proses yang sedang dijalankan, maka proses yang sedang berjalan tersebut dihentikan, lalu CPU dialihkan untuk proses yang baru datang tersebut. Sementara itu, pada non-preemptive, proses yang baru datang tidak dapat mengganggu proses yang sedang berjalan, tetapi hanya diletakkan di depan queue. Kelemahan pada priority scheduling adalah dapat terjadinya indefinite blocking (starvation). Suatu proses dengan prioritas yang rendah memiliki kemungkinan untuk tidak dieksekusi jika terdapat proses lain yang memiliki prioritas lebih tinggi darinya. Solusi dari permasalahan ini adalah aging, yaitu meningkatkan prioritas dari setiap proses yang menunggu dalam queue secara bertahap.

- Round Robin

Algoritma penjadwalan ini mirip dengan algoritma First Come First Served, tetapi proses ini memberi suatu batasan waktu untuk setiap proses yang disebut dengan time quantum. Time quantum adalah suatu satuan waktu yang kecil. Jika proses yang sedang dieksekusi selesai dalam waktu kurang dari 1 time quantum, tidak ada masalah. Tetapi jika proses berjalan melebihi 1 time quantum, maka proses tersebut akan dihentikan, lalu digantikan oleh proses yang berikutnya. Proses yang dihentikan tersebut akan diletakkan di queue di urutan paling belakang.

3. KRITERIA PENJADWALAN

Suatu algoritma penjadwalan CPU yang berbeda dapat mempunyai nilai yang berbeda untuk sistem yang berbeda. Banyak kriteria yang bisa dipakai untuk menilai algoritma penjadwalan CPU.

Kriteria yang digunakan dalam menilai adalah:

- CPU Utilization. Kita ingin menjaga CPU sesibuk mungkin. CPU utilization akan mempunyai range dari 0 sampai 100 persen. Di sistem yang sebenarnya ia mempunyai range dari 40 sampai 100 persen.
- Throughput. Salah satu ukuran kerja adalah banyaknya proses yang diselesaikan per satuan waktu. Jika kita mempunyai beberapa proses yang sama dan memiliki beberapa algoritma penjadwalan yang berbeda, throughput bisa menjadi salah satu kriteria penilaian, dimana algoritma yang menyelesaikan proses terbanyak

mungkin yang terbaik.

- Turnaround Time. Dari sudut pandang proses tertentu, kriteria yang penting adalah berapa lama untuk mengeksekusi proses tersebut. Memang, lama pengeksekusian sebuah proses sangat tergantung dari hardware yang dipakai, namun kontribusi algoritma penjadwalan tetap ada dalam lama waktu yang dipakai untuk menyelesaikan sebuah proses. Misal kita memiliki sistem komputer yang identik dan proses-proses yang identik pula, namun kita memakai algoritma yang berbeda, algoritma yang mampu menyelesaikan proses yang sama dengan waktu yang lebih singkat mungkin lebih baik dari algoritma yang lain. Interval waktu yang diijinkan dengan waktu yang dibutuhkan untuk menyelesaikan sebuah proses disebut turnaround time. Turnaround time adalah jumlah periode untuk menunggu untuk dapat ke memori, menunggu di ready queue, eksekusi CPU, dan melakukan operasi M/K.

- Waiting Time. Algoritma penjadwalan CPU tidak mempengaruhi waktu untuk melaksanakan proses tersebut atau M/K, itu hanya mempengaruhi jumlah waktu yang dibutuhkan proses di antrian ready. Waiting time adalah jumlah waktu yang dibutuhkan proses di antrian ready.

- Response Time. Di sistem yang interaktif, turnaround time mungkin bukan waktu yang terbaik untuk kriteria. Sering sebuah proses dapat memproduksi output di awal, dan dapat meneruskan hasil yang baru sementara hasil yang sebelumnya telah diberikan ke pengguna. Ukuran lain adalah waktu dari pengiriman permintaan sampai respon yang pertama diberikan. Ini disebut response time, yaitu waktu untuk memulai memberikan respon, tetapi bukan waktu yang dipakai output untuk respon tersebut.

- Fairness. Suatu algoritma harus memperhatikan pengawasan nilai prioritas dari suatu proses (menghindari terjadinya starvation CPU time).

- Efisiensi. Rendahnya overhead dalam context switching, penghitungan prioritas dan sebagainya menentukan apakah suatu algoritma efisien atau tidak.

4. PENELITIAN TERKAIT

Penelitian terkait yang digunakan dalam penelitian ini tidak lain adalah Smart Optimized Round Robin itu sendiri, dimana dalam penelitian ini akan dijabarkan metode yang dapat menandingi optimasi dari SORR tersebut. Pada SORR, kuantum yang digunakan menggunakan formula jumlah mean dan burst time tertinggi lalu dibagi 2. Hal ini membuat dua kemungkinan. Kemungkinan pertama ialah mean akan sangat kecil apabila terdapat sebuah proses yang burst

SYSTEMIC

Vol. 02, No. 02, Desember 2016, 19-21

time nya sangat kecil pula. Kemungkinan kedua adalah meskipun mean sangat besar, proses akan memerlukan banyak waktu apabila proses yang dikerjakan banyak, dan masing masing proses memiliki burst time yang tinggi, sehingga kuantum tidak seluruhnya dapat meng-cover setiap proses dalam 1 waktu kuantum.

5. ALGORITMA PENGEMBANGAN

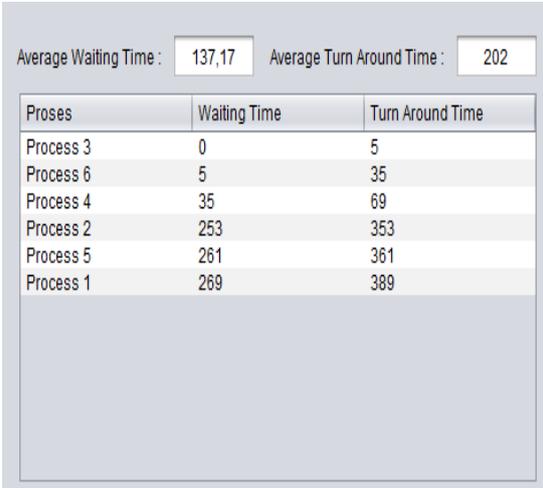
Jika sebelumnya pada SORR yang sudah ada, quantum ditentukan dengan cara menjumlahkan antara rata-rata burst time dengan burst time tertinggi lalu dibagi dengan 2. Hal ini tidak begitu efektif saat dalam deretan proses terdapat selisih burst time antar proses yang tinggi, sehingga membuat rata-rata dari burst time total menjadi kecil. Dengan demikian, quantum yang disediakan bisa saja kecil dan berimbas pada Average Waiting Time dan Average Turn Around Time yang dihasilkan.

Pada SORR-Re-Designed ini, penulis ingin menawarkan rumus sederhana untuk menentukan kuantum yang dihasilkan, yakni dengan merata-rata dua proses dengan burst time tertinggi. Ini akan menghasilkan kuantum yang mampu menyelesaikan 80% proses dalam satu kali waktu kuantum.

$$\text{Quantum} = \frac{(\text{Highest Burst Time [1]} + \text{Highest Burst Time [2]})}{2}$$

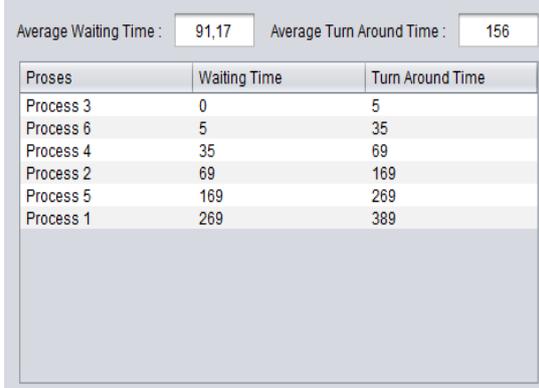
6. ANALISIS DAN PERCOBAAN

Dalam percobaan ini akan dilakukan dengan menginputkan 6 proses yakni P1, P2, P3, P4, P5, P6 dengan masing-masing burst time 120, 100, 5, 34, 100, 30. Dan berikut adalah hasil dari SORR dan SORR Re-designed.



Proses	Waiting Time	Turn Around Time
Process 3	0	5
Process 6	5	35
Process 4	35	69
Process 2	253	353
Process 5	261	361
Process 1	269	389

Gambar 1 SORR



Proses	Waiting Time	Turn Around Time
Process 3	0	5
Process 6	5	35
Process 4	35	69
Process 2	69	169
Process 5	169	269
Process 1	269	389

Gambar 2 SORR Re-designed

7. KESIMPULAN

Pada bagian ini penulis ingin menyimpulkan bahwa dengan semakin besar kuantum yang dapat disediakan oleh CPU dalam proses Round-Robin, akan berdampak pada minimumnya Average Waiting Time dan Average Turn Around Time yang dihasilkan. Dengan demikian SORR Re-designed ini menghasilkan AWT dan ATAT yang lebih minimal dibandingkan dengan SORR sebelumnya.

DAFTAR PUSTAKA

- Frada Burstein, dan Clyda W. Holsapple (Editors). 2008. *Handbook Of Decision Support System 1 Basic Theme*. Springer: Verlag Berlin Heidelberg.
- Kusumadewi, Sri. 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.
- Kusumadewi, Sri dan Hari Purnomo. 2005. *Penyelesaian Masalah Optimasi dengan Teknik-teknik Heuristik*. Yogyakarta: Graha Ilmu.

