

OPTIMASI TURN AROUND TIME PADA PENJADWALAN ROUND ROBIN DENGAN Mencari QUANTUM TIME OPTIMAL MENGGUNAKAN ALGORITMA SIMULATED ANNEALING

Asaziduhu Gea¹⁾

¹ *Fasilkom-TI Universitas Sumatera Utara, Medan*

Email : ¹asaziduhgea@gmail.com

ABSTRACT

A number of processes that are queuing need has managed by the operating system so that all processes can be served without any process that is in decline or too long. The duration of the response time results in a number of subsequent process failed because it ran Burst Time which has been allocated to each process. One of the responsibilities of the operating system is managing the process by making scheduling and expected around Turn Time is in getting smaller each time to execute a number of processes. Therefore, it is necessary to do the optimization of Turn around Time. Round Robin algorithm uses fair scheduling method by evenly dividing the execution time (quantum time) for each process but continue to serve all these processes sequentially. Allocation quantum appropriate time greatly affect the average waiting time for the entire process of the Round Robin. Because it is necessary to do an effort to get a proper quantum time can give a little time to execute a number of processes by implementing Simulated Annealing algorithms. Simulated Annealing is a heuristic algorithm that is oriented to find a settlement solution of a problem with the minimal use of energy (time), but the results are quite large (number of processes that a lot). Tests carried out on the process by varying amounts. Every process has a different burst time and tested with the alpha value varies. Tests in each group showed that there was a decrease in the average waiting time is quite significant in the early iteration until the final iteration thus obtained Turn around Time fewer, but the number of processes are executed more.

Keywords: Round Robin Scheduling, Quantum Time, Simulated annealing

1. Pendahuluan

Teknologi komputer pada saat ini, telah mendorong para *vendor* untuk membuat evolusi sistem operasi dengan menciptakan sistem operasi yang baru, ataupun menyempurnakan sistem operasi yang sudah ada sebelumnya. Beragam keunggulan dan fitur yang ditawarkan pada setiap sistem operasi, mulai dari tampilan grafis yang memikat sampai kemampuan menjalankan berbagai aplikasi dan kecepatan mengeksekusi sejumlah proses dalam waktu yang singkat, menjadi unggulan sekaligus menjadi fitur tambahan pada sistem operasi tersebut.

Menurut Arfiyanti (2012) banyaknya jumlah proses yang harus dieksekusi oleh CPU mengakibatkan kerumitan tersendiri dikarenakan alokasi waktu yang sangat terbatas. Kecepatan eksekusi sejumlah proses akan mempengaruhi efisiensi penggunaan waktu yang digunakan oleh CPU untuk mengeksekusi sejumlah proses yang sedang menunggu. Pada kasus penjadwalan, hasil yang optimum seringkali ditemui dengan jalan menganalisis dan mencoba aturan prioritas yang dinilai dapat memberikan solusi paling baik. Salah satunya menggunakan pendekatan komputasi dengan memanfaatkan modul yang tersedia pada perangkat lunak penjadwalan (Asmuliardi, 2010)

Algoritma Simulated Annealing adalah salah satu algoritma yang digunakan untuk teknik optimasi. Algoritma ini berorientasi pada penggunaan sumber daya minimal. Algoritma ini untuk bersifat generic, berdasarkan probabilitas dan mekanika statistik. Algoritma ini dapat digunakan untuk mencari pendekatan terhadap solusi optimum global dari suatu permasalahan seperti optimisasi kombinatorial, di mana ruang pencarian solusi yang ada terlalu besar, sehingga hampir tidak mungkin ditemukan solusi eksak terhadap permasalahan itu.

Kemampuan sistem operasi mengatasi kompleksitas waktu adalah hal utama yang perlu di perhatikan sehingga penggunaan waktu lebih efisien, sehingga CPU perlu memperbesar *throughput*. Oleh karena itu diperlukan sebuah teknik untuk mengalokasi *quantum time* yang tepat per masing masing proses. Alokasi *quantum time* yang tepat sangat mempengaruhi nilai *throughput* yang di dapatkan oleh CPU pada algoritma penjadwalan seperti pada penjadwalan *Round Robin* yang kinerjanya sangat tergantung pada *quantum time*. Menemukan nilai quantum yang bisa dialokasikan secara cepat dan dalam menyelesaikan eksekusi sejumlah proses diperlukan sebuah metode yang sifatnya heuristic search . Algoritma *simulated annealing* adalah metode yang tepat yang bisa menyelesaikan masalah ini, mengingat orientasi dari state yang dia

cari adalah state yang terkecil nilai bobotnya tetapi mengupayakan hasil yang maksimal.

2. Penjadwalan Proses

Menurut Tanenbaum (2001) Penjadwalan proses merupakan kumpulan kebijaksanaan dan mekanisme di sistem operasi yang berkaitan dengan urutan kerja yang dilakukan sistem komputer. Adapun penjadwalan bertugas memutuskan :

- a. Proses yang harus berjalan
- b. Kapan dan selama berapa lama proses itu berjalan

Menurut Tarore (2012) pengaturan waktu atau penjadwalan dari kegiatan-kegiatan yang terlibat didalamnya dimaksudkan agar suatu proyek dapat berjalan dengan lancar serta efektif. Oleh karena itu, pelaksana dari suatu kegiatan biasanya membuat suatu jadwal waktu kegiatan atau time schedule. Jadwal waktu kegiatan adalah urutan kerja yang berisi tentang :

- a. Jenis pekerjaan yang akan diselesaikan
- b. Waktu bilamana suatu pekerjaan dimulai dan diakhiri.

Oleh karena itu penjadwalan yang baik harus memiliki ukuran agar proses-proses yang dijalankan lebih optimal. Untuk mengukur dan optimasi kinerja penjadwalan menurut Tarek (2006) bahwa ada beberapa hal yang perlu di perhatikan :

a. Adil (*fairness*)

Adalah proses-proses yang diperlakukan sama, yaitu mendapat jatah waktu pemroses yang sama dan tak ada proses yang tak kebagian layanan pemroses sehingga mengalami kekurangan waktu.

b. Efisiensi (*eficiency*)

Efisiensi atau utilisasi pemroses dihitung dengan perbandingan (rasio) waktu sibuk pemroses.

c. Waktu tanggap (*response time*)

Waktu tanggap adalah waktu yang dibutuhkan untuk merespon atau menanggapi permintaan layanan eksekusi dari sebuah proses. Waktu tanggap dibedakan atas dua hal yakni :

1). Sistem interaktif

Didefinisikan sebagai waktu yang dihabiskan dari saat karakter terakhir dari perintah dimasukkan atau transaksi sampai hasil pertama muncul di layar. Waktu tanggap ini disebut terminal response time.

2). Sistem waktu nyata didefinisikan sebagai waktu dari saat kejadian (internal atau eksternal) sampai instruksi pertama rutin layanan yang dimaksud dieksekusi, disebut *event response time*.

d. Turn around time

Adalah waktu yang dihabiskan dari saat program atau job mulai masuk ke sistem sampai proses diselesaikan sistem. Waktu yang dimaksud adalah waktu yang dihabiskan di dalam sistem, diekspresikan sebagai penjumlahan

waktu eksekusi (waktu pelayanan job) dan waktu menunggu, yaitu :

$$\text{Turn around time} = \text{waktu eksekusi} + \text{waktu menunggu.}$$

e. Throughput

Adalah jumlah kerja yang dapat diselesaikan dalam satu unit waktu. Cara untuk mengekspresikan throughput adalah dengan jumlah job pemakai yang dapat dieksekusi dalam satu unit/interval waktu.

Tipe penjadwalan

Menurut Hariyanto (2009), terdapat tiga tipe penjadwalan berada secara bersama-sama pada sistem operasi yang kompleks, yaitu:

1. Penjadwal jangka pendek (*short term scheduler*)
2. Penjadwal jangka menengah (*medium term scheduler*)
3. Penjadwal jangka panjang (*long term scheduler*)

Sasaran penjadwalan berdasarkan tipe-tipe penjadwalan :

- a. Memaksimumkan kinerja untuk memenuhi satu kumpulan kriteria yang diharapkan.
- b. Mengendalikan transisi dari *suspended to ready* (keadaan *suspend* ke *ready*)
- c. dari proses-proses *swapping*.
- d. Memberi keseimbangan *job-job* campuran.

Strategi Penjadwalan

Menurut Hariyanto (2009) Terdapat dua strategi penjadwalan, yaitu :

1. Penjadwalan non preemptive (*run to completion*)
Proses diberi jatah waktu oleh pemroses, maka pemroses tidak dapat diambil alih oleh proses lain sampai proses itu selesai.
2. Penjadwalan preemptive
Proses diberi jatah waktu oleh pemroses, maka pemroses dapat diambil alih proses lain, sehingga proses disela sebelum selesai dan harus dilanjutkan menunggu jatah waktu pemroses tiba kembali pada proses itu.

Algoritma-algoritma Penjadwalan

Ada beberapa jenis jenis algoritma penjadwalan. algoritma penjadwalan dibagi dalam dua konsep secara umum :

1. Nonpreemptive, menggunakan konsep :
 - a. FIFO (*First In First Out*) atau FCFS (*First Come First Serve*)
 - b. SJF (*Shortest Job First*)
 - c. HRN (*Highest Ratio Next*)
 - d. MFQ (*Multiple Feedback Queues*)
2. Preemptive, menggunakan konsep :
 - a. RR (*Round Robin*)
 - b. SRF (*Shortest Remaining First*)
 - c. PS (*Priority Scheduling*)
 - d. GS (*Guaranteed Scheduling*)

Klasifikasi lain selain berdasarkan dapat/tidaknya suatu proses diambil secara paksa adalah klasifikasi berdasarkan adanya prioritas di proses-proses, yaitu :

1. Algoritma penjadwalan tanpa berprioritas.
2. Algoritma penjadwalan berprioritas, terdiri dari :
 - a. Berprioritas statik
 - b. Berprioritas dinamis

3. Algoritma Round Robin

Konsep dasar dari algoritma Round Robin adalah dengan menggunakan *time-sharing*. Pada dasarnya algoritma ini sama dengan FCFS, hanya saja bersifat preemptive. Setiap proses mendapatkan waktu CPU yang disebut dengan waktu quantum (*quantum time*) untuk membatasi waktu proses, biasanya 1-100 milidetik.

Setelah waktu habis, proses ditunda dan ditambahkan pada *ready queue*. Jika suatu proses memiliki CPU burst lebih kecil dibandingkan dengan waktu quantum, maka proses tersebut akan melepaskan CPU jika telah selesai bekerja, sehingga CPU dapat segera digunakan oleh proses selanjutnya.

Sebaliknya, jika suatu proses memiliki CPU *burst* yang lebih besar dibandingkan dengan waktu quantum, maka proses tersebut akan dihentikan sementara jika sudah mencapai waktu quantum, dan selanjutnya mengantri kembali pada posisi ekor dari *ready queue*, CPU kemudian menjalankan proses berikutnya. Jika terdapat n proses pada *ready queue* dan waktu quantum q , maka setiap proses mendapatkan $1/n$ dari waktu CPU paling banyak q unit waktu pada sekali penjadwalan CPU. Tidak ada proses yang menunggu lebih dari $(n-1)q$ unit waktu (Tanenbaum, 2001)

Waktu *turnaround* juga tergantung ukuran waktu quantum. Rata-rata waktu *turnaround* tidak meningkat bila waktu quantum dinaikkan. Secara umum, rata-rata waktu *turnaround* dapat ditingkatkan jika banyak proses menyelesaikan CPU burst berikutnya sebagai satu waktu *quantum*. Sebagai contoh, terdapat tiga proses masing-masing 10 unit waktu dan waktu quantum 1 unit waktu, rata-rata waktu *turnaround* adalah 29. Jika waktu quantum 10, sebaliknya, rata-rata waktu *turnaround* turun menjadi 20.

4. Simulated Annealing

Menurut Mahlke (2006) Algoritma *Simulated Annealing* diperkenalkan oleh Metropolis *et al.* Pada tahun 1953, dan aplikasinya dalam masalah optimasi dilakukan pertama kali oleh Kirkpatrick *et al.* Tahun 1983. Algoritma ini beranalogi dengan proses *annealing* (pendinginan) yang diterapkan dalam pembuatan material *glassy* (terdiri dari butir kristal). Dari sisi ilmu fisika, tujuan sistem ini adalah untuk meminimasi energi potensial. Fluktuasi kinematika acak menghalangi sistem untuk mencapai energi potensial yang minimum

global, sehingga sistem dapat terperangkap dalam sebuah keadaan minimum lokal.

Dengan menurunkan temperatur sistem, menurut Henry (2012) diharapkan energi dapat dikurangi ke suatu level yang relatif rendah. Semakin lambat laju pendinginan ini, semakin rendah pula energi yang dapat dicapai oleh sistem pada akhirnya.

Algoritma Simulated Annealing berorientasi bagaimana menyelesaikan sebuah pekerjaan besar dengan pemakaian energi yang kecil. Berdasarkan teori tersebut, maka dianalogikan bahwa algoritma Round Robin membutuhkan sebuah perhitungan nilai *quantum time* yang tepat untuk bisa menyelesaikan sejumlah proses dengan waktu yang sangat sedikit.

Algoritma ini penulis analogikan sebagai sebuah algoritma yang mampu menyelesaikan eksekusi jumlah proses yang banyak tetapi membutuhkan waktu yang lebih sedikit atau sedikit. Jika temperatur dalam algoritma *simulated annealing* sebagai factor penentu keberhasilan pendinginan maka dalam kasus yang penulis teliti ini, penentuan quantum time yang menjadi penentu keberhasilan pencapaian waktu optimal.

Panggabean (2002) mengatakan bahwa algoritma Simulated Annealing secara umum adalah sebagai berikut:

- A. Pilih sebuah solusi awal x_0 secara acak dan tetapkan nilai temperatur awal. Pada langkah ke - i , solusi yang current disebut x_i . Parameter kontrol adalah c_i dan $f_i = f(x_i)$.
- B. Ulangi langkah-langkah berikut :
 1. Buat sebuah neighbour x_p dari solusi current x_i dan hitung nilai fungsi objektifnya. State x adalah sebuah kandidat potensial untuk state $x_{(i+1)}$
 2. Set $x_{(i+1)} = x_p$ dengan probabilitas $\min\{1, \exp((f_i - f_p)/c_i)\}$. Jika tidak, set $x_{i+1} = x_i$. Turunkan nilai temperatur berdasarkan faktor d tertentu :
 $c_i = c_i + dc_i$. Tambahkan 1 pada jumlah iterasi : $i = i + 1$.

Parameter Simulated Annealing

Menurut Pahwa (2004) parameter yang di butuhkan dalam Simulated Annealing adalah sebagai Berikut :

1. Starting Temperature - t
2. Cooling Schedule - α
3. Final temperature / Stopping Rule - $S_n()$
4. Iterations at given Temperature - N
5. Initial (starting) configuration - $S[a,b,c]$
6. Transition rule - $Tr()$
7. New configuration derivation rule - $Ex(S[a,b,c])$

Optimasi nilai quantum time pada penjadwalan round robin dapat di lakukan dengan cara memilih secara acak angka yang tepat yang digunakan untuk

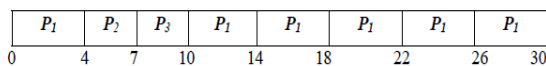
menyelesaikan sejumlah proses tersebut. Hasil dari penyelesaian proses itu nantinya akan di jadikan sebagai acuan untuk melaksanakan eksekusi terhadap proses berikutnya, sehingga penggunaan *quantum time* yang sesuai dapat tepat dapat di temukan. (Tanenbaum, 2001)

Performans algoritma round robin dapat dijelaskan sebagai berikut, jika q besar, maka yang digunakan adalah algoritma FIFO, tetapi jika q kecil maka sering terjadi context switch. Misalkan ada 3 proses: *P1*, *P2*, dan *P3* yang meminta pelayanan CPU dengan *quantum-time* sebesar 4 milidetik.

Tabel 1 Antrian Proses

Process	Burst Time
<i>P1</i>	24
<i>P2</i>	3
<i>P3</i>	3

Penjadwalan proses dengan algoritma round robin dapat dilihat pada gant chart berikut :



Gbr. 1 *Ghant chart* algoritma Round Robin

Waktu tunggu untuk *P1* adalah 6, *P2* adalah 4, dan *P3* adalah 7 sehingga rata-rata waktu tunggu adalah $(6 + 4 + 7)/3 = 5.66$ milidetik. Algoritma Round-Robin ini di satu sisi memiliki keuntungan, yaitu adanya keseragaman waktu. Namun di sisi lain, algoritma ini akan terlalu sering melakukan *switching* seperti yang terlihat pada Gambar 3.1.

Semakin besar *quantum-time* maka *switching* yang terjadi akan semakin sedikit. Waktu *turnaround* juga tergantung ukuran waktu *quantum*. Rata-rata waktu *turnaround* tidak meningkat bila waktu quantum dinaikkan. Secara umum, rata-rata waktu *turnaround* dapat ditingkatkan jika banyak proses menyelesaikan CPU burst berikutnya sebagai satu waktu quantum. Sebagai contoh, terdapat tiga proses masing-masing 10 unit waktu dan waktu quantum 1 unit waktu, rata-rata waktu *turnaround* adalah 29. Jika waktu quantum 10, sebaliknya, rata-rata waktu *turnaround* turun menjadi 20.

Oleh karena di perlukan suatu metode atau cara baru untuk bisa memilih atau menentukan nilai quantum time yang tepat untuk menyelesaikan eksekusi sejumlah proses. Dengan asumsi jumlah proses yang berhasil di eksekusi banyak tetapi waktu yang digunakan sedikit.

Menurut Suwarjono (2013) penggunaan Algoritma simulated annealing dapat memberikan nilai yang optimal untuk menyelesaikan kasus yang memiliki urutan jadwal order seperti yang terdapat di dalam penjadwalan round robin. Algoritma Simulated Annealing adalah sebuah algoritma yang berorientasi pada pencapaian hasil yang maksimal tetapi menggunakan waktu yang sedikit. Metode penyelesaian masalah yang terdapat dalam

algoritma simulated annealing memiliki korelasi yang baik jika di diterapkan untuk menyelesaikan masalah waktu yang cukup kompleks pada penjadwalan round- robin yakni mencari nilai *Turnaround Time minimal*. Peranan algoritma Simulated Annealing dalam hal ini adalah mencari nilai quantum time yang mampu meminimalkan average waiting time untuk keseluruhan proses yang sedang mengantri sehingga di dapatkan Turnaround Time yang minimal.

Pada Penelitian ini, pengaruh nilai quantum time optimal akan terlihat pada opitmasi minimum variable yang mempengaruhi waktu pada penjadwalan, yaitu :

- a. Average Waiting Time
- b. Turnaround Time
- c. Context Switching

Dengan menggunakan parameter nilai alpha yang digunakan pada simulated annealing dilakukan penelusuran pada penelitian ini untuk membuktikan bahwa average waiting time, Turn around time, dan context switching bisa menurun atau waktu yang digunakan lebih sedikit. Namun nilai alpha tersebut akan dipengaruhi oleh nilai quantum time yang dipilih secara acak untuk menentukan best solution.

5. Data Yang Digunakan

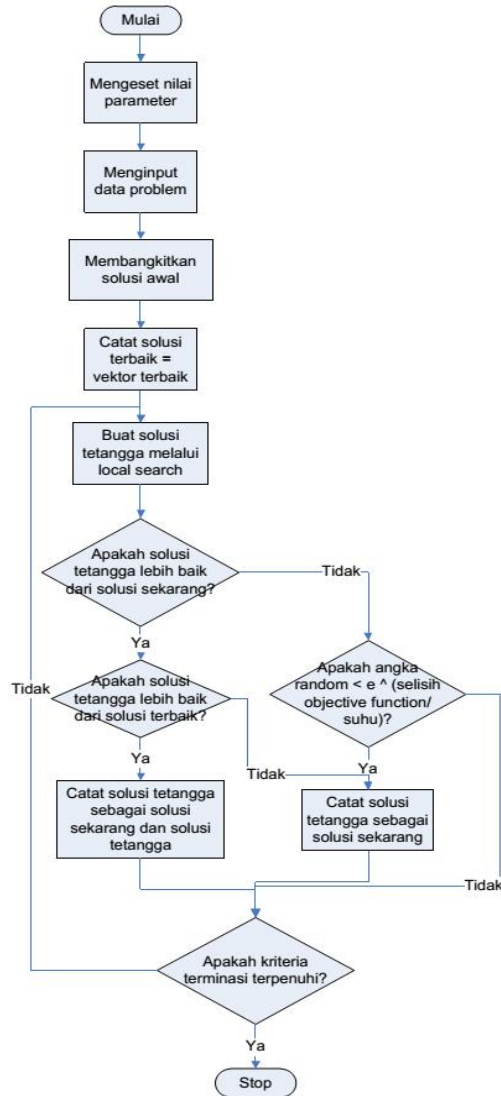
Pada penelitian ini, penulis membutuhkan beberapa data input yang terdiri dari:

1. Jumlah proses
Jumlah proses dalam hal ini adalah banyaknya jumlah proses yang sedang mengantri dalam satu tumpukan.
2. Arrival Time
Adalah urutan kedatangan sebuah proses yang akan menunggu untuk di eksekusi
3. Quantum Time
Adalah nilai yang di berikan atau lamanya waktu eksekusi yang di alokasikan oleh sistem operasi pada masing masing proses secara keseluruhan. dimana setiap proses akan berganti dengan proses berikutnya apabila quantum time-nya habis.
4. Burst Time
Adalah alokasi lamanya waktu eksekusi yang telah di alokasikan kepada masing masing proses sejak proses itu di buat.
5. Time Slice
Adalah jumlah iterasi waktu yang di gunakan untuk mencari nilai quantum yang mampu memberikan nilai *turnaround time* minimal. Jumlah time slice yang optimal adalah maksimal 100.
6. Context Switching
Adalah nilai peralihan waktu antar proses ketika di eksekusi lalu di tunda dengan proses berikutnya. Semakin sedikit nilai context Switching semakin banyak jumlah

proses yang bisa disedikit rata rata waktu tunggu untuk masing masing proses

6. Optimasi Quantum Time Dengan Simulated Annealing

Prosedur penyelesaian masalah optimasi quantum time dengan simulated annealing dapat dijelaskan dengan menggunakan diagram alir 2 dibawah ini.



Gambar 2. Optimasi Quantum Time Dengan Simulated Annealing

Mengeset Nilai Parameter

Langkah awal dalam penyelesaian masalah untuk mencari nilai quantum time dengan algoritma Simulated annealing adalah dengan menentukan parameter yang meliputi :

1. Penentuan Jumlah Proses (t1)
 Dalam penelitian ini, jumlah proses yang akan di dieksekusi terlebih dahulu diset sebagai standar ukuran proses yang mempengaruhi waktu eksekusi. Semakin

2. Burst Time (α)
 Burst Time digunakan sebagai jumlah waktu yang di alokasikan pada masing masing proses sebelum di eksekusi.
3. Time Slice (S_n)
 Adalah batasan atau waktu toleransi yang digunakan untuk menyelesaikan sebuah proses.
4. Quantum Time (N)
 Adalah nilai quantum time awal yang diberikan dan nilai ini akan di ganti dengan nilai yang lain jika hasil yang di dapatkan belum maksimal.
5. Initial (starting) configuration (S[a,b,c])
 adalah nilai array yang digunakan untuk dipilih secara acak oleh sistem untuk menentukan nilai quantum time yang akan di pilih berikutnya.
6. Transition rule (Tr)
 Adalah aturan transisi nilai dari quantum time yang akan digunakan.
7. Current Solution (Ex (S[a,b,c]))
 Adalah konfigurasi alternatif yang lain yang digunakan jika nilai turnaround time yang di dapatkan tidak minimal.
8. Nilai Alpha
 Adalah nilai yang mempengaruhi ruang pencarian solusi yang terbaik
9. Context Switching
 Adalah rata rata peralihan waktu antar proses ketika quantum time telah di alokasikan pada masing masing proses. Nilai Context Switching semakin kecil semakin bagus.

Menginput Data Problem.

Input data Problem adalah penentuan nilai dari setiap variable yang akan mempengaruhi eksekusi proses seperti jumlah proses, *burst time*, *arrival time* dan *nilai quantum time*.

Membangkitkan Solusi Awal.

Solusi awal dalam penyelesaian kasus dalam algoritma round robin adalah penentuan nilai *quantum time* yang akan di gunakan untuk mengesekusi sejumlah proses yang sedang mengantri serta menghitung berapa nilai *average waiting time* awal. Kemudian nilai *average waiting time* yang di didapatkan di awal iterasi (*time slice*) di uji pada iterasi yang kedua dengan membangkitkan nilai quantum time secara acak.

Menampilkan Solusi Yang Terbaik.

Setelah nilai *quantum time* di tentukan dan eksekusi di jalankan, akan ditampilkan berapa lama waktu yang digunakan untuk menyelesaikan sejumlah

proses tadi dengan harapan nilai yang didapatkan adalah nilai yang terbaik saat itu (*current solution*).

Kriteria Terpenuhi

Jika hasil yang di dapatkan adalah nilai yang optimal maka tujuan tercapai dan pencarian solusi terbaik di hentikan.

7. Pengujian

Penerapan Algoritma Simulated Annealing pada penjadwalan round robin terdiri dari beberapa tahapan. Salah satu tahapan yang pada penelitian ini adalah penentuan jumlah proses dan alokasi burst time pada masing masing proses tersebut. Pada penelitian ini optimasi nilai *turnaround time* yang dipengaruhi secara langsung oleh penggunaan nilai quantum time yang tepat menggunakan beberapa variable yaitu

1. Nilai alpha (0,90 dan 0,99)
2. Inisial Context Switching (nilai awal 110 dan nilai akhir 0,5)
3. Jumlah Proses
4. Burst Time
5. Iterasi

8. Hasil Uji Coba

Pada penelitian ini, penentuan *Quantum time* yang optimal ditentukan dengan menggunakan algoritma Simulated annealing. Pengujian dilakukan dengan menerapkan dua jenis parameter yang berbeda dan masing-masing parameter diuji pada tiga jenis proses yang berbeda yaitu 100, 150 dan 200 proses.

9. Uji Coba Pada 100 Proses

Parameter 1

Pada pengujian ini, parameter yang diterapkan adalah koefisien *alpha* atau factor pendinginan sebesar 0.90, jumlah proses sebesar 100, initial Context Switching atau nilai Context Switching awal sebesar 110, *final Context Switching* atau Context Switching akhir sebesar 0,5 dan iterasi bernilai satu. Parameter-parameter tersebut ditampilkan pada table 1 berikut ini :

Tabel 2. Parameter 1 100 proses

Alpha	Jumlah Proses	Context Switching max	Context Switching Min	Iterasi
0,90	100	110	0,5	1

Hasil Pengujian

Tabel 3. Hasil Pengujian Parameter 1 dengan 100 proses

Atribut	Hasil
Quantum time	314
AWT awal	3663.88
Context Switching	110

Quantum time	100
AWT	3441.09
Current Solution	3371.00
<i>Turnarround Time</i>	3371.00
Context Switching	8.77
Quantum time	56
AWT	3512.73
Current Solution	3371.00
<i>Turnarround Time</i>	3371.00
Context Switching	7.90
Quantum time	14
AWT	3329.47
Current Solution	3329.47
<i>Turnarround Time</i>	3329.47
Context Switching	7.11
Quantum time	171
AWT	3717.14
Current Solution	3329.47
<i>Turnarround Time</i>	3329.47
Context Switching	6.40
Quantum time	99
AWT	3496.07
Current Solution	3329.47
<i>Turnarround Time</i>	3329.47
Context Switching	0.57
Quantum time	64
AWT	3571.84
Current Solution	3329.47
<i>Turnarround Time</i>	3329.47
Context Switching	0.51
Quantum time	40
AWT	3415.50
Current Solution	3329.47
<i>Turnarround Time</i>	3329.47
Context Switching	0.46

Best seeking

Best seeking adalah proses penelusuran solusi terbaik mulai dari iterasi / Context Switching awal sampai iterasi/Context Switching (CS) akhir.

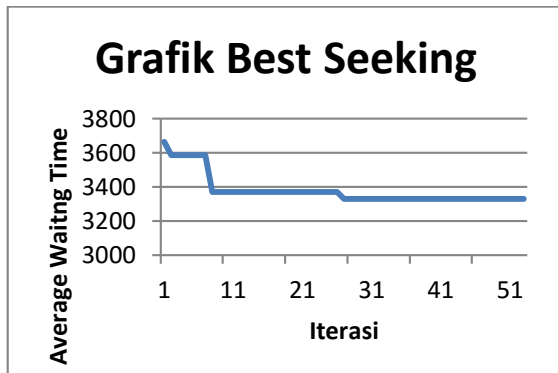
Tabel 4. *Turnarround Time* Parameter 1 dengan 100 proses

Ite rasi	CS	<i>TT</i>	Ite rasi	CS	<i>TT</i>
1	110	3663.88	29	5.76	3329.47
2	99	3586.58	30	5.18	3329.47
3	89.1	3586.58	31	4.66	3329.47
4	80.2	3586.58	32	4.2	3329.47
5	72.2	3586.58	33	3.78	3329.47
6	65	3586.58	34	3.4	3329.47
7	58.5	3586.58	35	3.06	3329.47

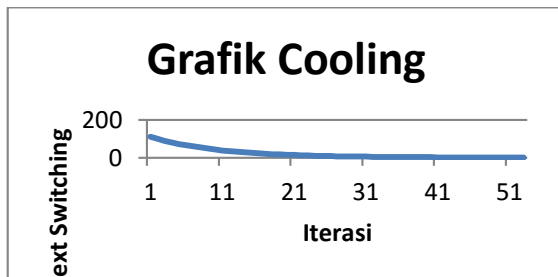
8	52.6	3371	36	2.75	3329.47
9	47.4	3371	37	2.48	3329.47
10	42.6	3371	38	2.23	3329.47
20	14.9	3371	45	1.07	3329.47
28	6.4	3329.47	53	0.46	3329.47

Pada tabel 4.4 di atas ditampilkan bahwa pada iterasi awal diperoleh *Turnaround Time* sebesar **3663.88**. Namun pada iterasi ke dua didapatkan *Turnaround Time* lebih kecil dari hasil iterasi pertama yakni sebesar **3586.58**. Hasil ini juga di uji pada iterasi ketiga, dan di dapatkan nilai *Turnaround Time* yang sama pada iterasi yang ke tujuh. Pada iterasi delapan diperoleh nilai average waiting time yang lebih kecil yaitu sebesar **3371** nilai tersebut dipertahankan sampai pada iterasi ke 27. Pada iterasi yang ke 28 di dapatkan hasil pengujian sebesar 3329.47. nilai ini terus di uji coba pada sampai iterasi ke 53 di dapatkan nilai 3329.47. Pengujian terhenti sampai pada iterasi yang ke 53 karena nilai Context Switching akhir sudah terpenuhi yakni 0,46

Proses penelusuran atau best seeking pada tabel 4.4 di atas dapat ditampilkan pada gambar 3 berikut ini.



Gambar 3. Grafik Penelusuran Nilai quantum time pada parameter 1 dengan 100 proses



Gambar 4. Grafik Penurunan Context Switching dengan alpha 0.90 pada 100 proses

9.3 Parameter 2

Pada pengujian ini, parameter yang diterapkan adalah koefisien *alpha* atau factor pendinginan sebesar 0.99, jumlah proses sebesar 100, initial Context Switching atau nilai Context Switching awal sebesar 110, *final Context Switching* atau Context Switching akhir sebesar 0,5 dan iterasi

bernilai satu. Parameter-parameter tersebut ditampilkan pada table 4 berikut ini :

Tabel 5 Parameter dua 100 proses dengan alpha 0,99

Alpha	Jumlah proses	Context Switching max	Context Switching Min	Iterasi
0,99	100	110	0,5	1

Hasil Pengujian Parameter 2

Iterasi	CS	TT	Iterasi	CS	TT
1	110.00	3674.29	51	66.55	3365.72
2	108.90	3658.69	52	65.89	3365.72
3	107.81	3606.38	53	65.23	3365.72
4	106.73	3606.38	54	64.57	3365.72
5	105.67	3606.38	55	63.93	3365.72
10	100.49	3606.38	60	60.80	3292.57
20	90.88	3579.19	70	54.98	3292.57
30	82.19	3371	80	49.72	3292.57
40	74.33	3365.72	90	44.97	3292.57
50	67.22	3365.72	100	40.67	3292.57
230	11.01	3292.57	280	6.66	3292.57
240	9.96	3292.57	290	6.03	3292.57
250	9.01	3292.57	300	5.45	3292.57
320	4.46	3292.57	370	2.70	3292.57
330	4.03	3292.57	380	2.44	3292.57
340	3.65	3292.57	390	2.21	3292.57
350	3.30	3292.57	400	1.99	3292.57
510	0.66	3292.57	529	0.55	3292.57
516	0.62	3292.57	535	0.51	3292.57
517	0.62	3292.57	536	0.51	3292.57
518	0.61	3292.57	537	0.50	3292.57

Tabel 6. Hasil Pengujian Parameter 2 dengan 100 proses

Quantum time	330
AWT awal	3674.29
Quantum time	306
AWT	3658.69
Current Solution	3658.69
<i>Turnaround Time</i>	3658.69
Context Switching	108.90
Quantum time	11
AWT	3292.57
Current Solution	3292.57
<i>Turnaround Time</i>	3292.57
Context Switching	62.66
Quantum time	62
AWT	3591.78
Current Solution	3292.57
<i>Turnaround Time</i>	3292.57
Context Switching	62.03

Quantum time	152
AWT	3760.13
Current Solution	3293.66
<i>Turnarround Time</i>	3292.57
Context Switching	0.51
Quantum time	195
AWT	3649.52
Current Solution	3293.66
<i>Turnarround Time</i>	3292.57
Context Switching	0.50

Pada tabel 5 di atas ditampilkan sebagian data bahwa pada awal iterasi, nilai average waiting time awal adalah sebesar 3674.29 ms, nilai ini diperoleh dari Quantum time sebesar 330 pada Context Switching 110

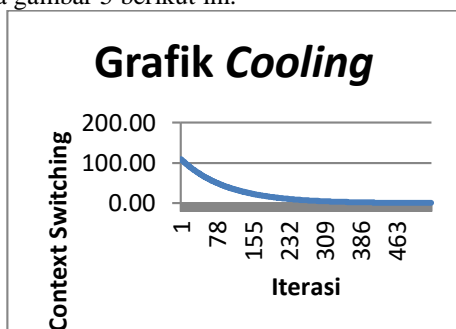
Pada tabel 6 berikut ini ditampilkan hasil penelusuran *Turnarround Time* untuk 100 proses menggunakan parameter 2 dan burst time yang berbeda-beda.

Best Seeking

Pada tabel 6 di atas ditampilkan bahwa pada iterasi awal diperoleh *Turnarround Time* sebesar 3674.29. Namun pada iterasi ke dua didapatkan *Turnarround Time* lebih kecil dari hasil iterasi pertama yakni sebesar 3658.69. Hasil ini juga di uji pada iterasi ketiga, dan di dapatkan nilai *Turnarround Time* yakni 3606.38. Nilai ini terus bertahan sampai pada iterasi yang ke 11.

Proses penelusuran terhadap *best seeking* dapat dilihat pada gambar 4.3 di bawah ini :

Pada pengujian ini, nilai Context Switching awal adalah 110 dan nilai Context Switching ini akan diturunkan dengan factor 0.99 sampai diperoleh nilai Context Switching akhir. Grafik penurunan Context Switching dapat ditampilkan pada gambar 5 berikut ini.



Gambar 5 Grafik Penurunan nilai Context Switching dengan nilai alpha 0,99 pada 100 proses

10. Kesimpulan Dan Saran

10.1 Kesimpulan

Berdasarkan percobaan yang telah dilakukan oleh untuk mengoptimalkan penjadwalan *round robin* dengan mencari nilai *quantum time* yang optimal dengan menggunakan algoritma *simulated annealing* di dapat kesimpulan sebagai berikut :

1. Untuk mencari nilai quantum time yang optimal pada algoritma *simulated annealing* membutuhkan nilai alpha yang lebih besar.
2. Penggunaan Algoritma *simulated annealing* untuk membangkitkan nilai quantum time secara acak mampu mempersingkat waktu pencarian quantum time optimum
3. Penggunaan penjadwalan *round robin* untuk mengeksekusi sejumlah proses yang mengantri akan lebih optimal jika nilai *quantum time* yang digunakan lebih tepat karena dapat memperkecil *Average Waiting Time*.
4. *Turnarround Time* yang dibutuhkan untuk mengeksekusi sejumlah proses yang mengantri di dalam CPU akan lebih sedikit jika nilai quantum time yang digunakan lebih dioptimalkan.
5. Terdapat perbedaan yang signifikan antara *Average Waiting Time* Sebelum di terapkan *Simulated Annealing* dengan *Average Waiting Time* sesudah digunakan Algoritma *Simulated Annealing*.
6. Penerapan Algoritma *Simulated Annealing* untuk mencari nilai quantum time pada penjadwalan *round robin* sangat mempengaruhi kinerja CPU dan Sistem Operasi, sehingga proses yang banyak dapat di eksekusi dengan menggunakan waktu yang sangat sedikit.
7. Context Switching akan semakin sedikit jika nilai quantum time yang digunakan lebih optimal

10.2 Saran

1. Pada penelitian lain yang berkaitan dengan penjadwalan CPU yang berkaitan dengan penjadwalan *round robin* perlu di optimasi lagi dengan menggunakan berbagai algoritma lain yang bersifat heuristic seperti algoritma *simulated annealing*, tetapi dengan jumlah proses yang lebih banyak sehingga waktu yang dibutuhkan untuk menjalankan proses lebih sedikit sementara jumlah proses yang dijalankan lebih banyak.
2. Diharapkan algoritma *simulated annealing* bisa diterapkan pada teknik penjadwalan sehingga ada efisiensi penggunaan waktu yang lebih optimal.

11. DAFTAR PUSTAKA

- Benny, R.Joshi. Simulated Annealing untuk mencari rute jalan terpendek dengan biaya minimal. Jurnal vol 1 No 2.2010
- Budi, S. Gregorius. Activity Based Costing dan Simulated Annealing Untuk Pencarian Rute Pada Flexible Manufacturing System. Jurnal vol 4 No 1.2003. Teknik Informatika. Universitas Kristen Petra.
- Dunham, M.2003. Operating System : Introductory And Advanced Topics. Prentice Hall International: New Jersey.
- Danish, M. 2011. Virtual CPU Scheduling In The Quest Operating System. Computer Science. Jurnal. Boston University.
- Hariyanto, B. 2009. Sistem Operasi. Edisi I. Informatika: Bandung
- Jogiyanto.2010. Desain dan Praktek Sistem Operasi. Gramedia. Jakarta
- Kanagaraj, G. Adaptive Load Balancing Algorithm Using Service Queue. Jurnal. ICCSIT 2012. Singapore.
- Kusumadewi, S. 2004. Aplikasi Simulated Annealing Untuk Penentuan Tata Letak Mesin. Teknik Industri. Jurnal. Universitas Islam Indonesia
- Muluk, Asmuliardi. Penjadwalan Job Shop Dengan Kriteria Minimasi Maskepan. Jurnal. 2010. Teknik Industri. Universitas Andalas.
- Panggabean, H.P. 2002. Penjadwalan job shop statik dengan algoritma simulated annealing. Master Tesis.Universitas Parhyangan.
- Pahwa , M. 2004. A simulated annealing model of optimal hub-and-spoke airline networks. Ph. D Dissertation . University of Delaware
- Santoso, B.H. 2005. Perancangan Sistem Operasi. Edisi I. Andi : Yogyakarta
- Suwarjono, W. 2013. Penjadwalan Produksi Pada Unit Produksi PT X Dengan Menggunakan Algoritma Simulated Annealing Untuk Meminimasi Waktu Makespan. Teknik Industri. Jurnal. Universitas Sumatera Utara.
- Tanenbaum, A. 2001. *Operating System and Design*. Prentice Hall : New York
- Tarek, M.Sobh. 2006. Parametric Optimization Of Some Critical Operating System Functions – An Alternative Approach To The Study Of Operating Systems Design. Jurnal. University of Bridgeport
- Tarore, H.2012. Analisis Optimalisasi Waktu Dan Biaya Dengan Program Primavera 6.0. Teknik Sipil. Jurnal. Universitas Sam Ratulangi.
- Yasan, H. 2012. Aplikasi Pembelajaran Dan Sistem Operasi Dalam Materi Proses Penjadwalan FCFS, SJF dan Round Robin. Teknik Informatika. Jurnal Vol 1. STMIK Budi Dharma. Medan.