BAB I

Pendahuluan

1.1 Latar belakang masalah

Sistem operasi adalah penghubung/jembatan antara manusia dan komputer. Beberapa fungsi yang disediakan oleh sistem operasi antara lain: manajemen memori, manajemen file, dll. Salah satu fungsi dari sistem operasi adalah mengatur penjadwalan pengeksekusian proses-proses yang berada di dalam komputer [4]. Proses-proses tersebut harus dijadwalkan agar dapat menghemat waktu untuk pengeksekusian proses dan agar semua proses dapat dieksekusi secara adil semua mendapat giliran. Agar semua proses dapat dijadwalkan dan dieksekusi oleh sistem operasi maka dibutuhkan sebuah algoritma yang sesuai untuk penjadwalan pada sistem operasi tersebut. Algoritma ini akan dipilih berdasarkan kecocokan sifat dari sistem operasi tersebut.

Terdapat banyak algoritma yang telah diciptakan dan digunakan untuk menangani penjadwalan proses pada sebuah sistem operasi seperti FCFS (*First Come First Served)*, RR (*Round Robin*), SJF (*Shortest Job First)* dan *priority scheduling [6]*. Biasanya algoritma baru akan diciptakan untuk mengatasi kekurangan dalam algoritma yang sebelumnya telah diciptakan atau bisa saja ingin mengoptimasi kinerja di dalam suatu algoritma penjadwalan [1]. Algoritma baru ini dibuat bisa dengan memodifikasi algoritma yang sudah ada, menggabungkan algoritma yang telah dibuat dengan referensi atau sumber ilmiah.

Penambahan aturan pada algoritma penjadwalan dibuat dengan memodifikasi aturan yang akan dipakai ke dalam algoritma yang akan ditetapkan. Modifikasi dilakukan secara parsial sehingga tidak mengubah struktur algoritma secara menyeluruh. Salah satu algoritma penjadwalan yang dikenal adalah RR. Algoritma RR dikenal algoritma yang adil karena ada pembatasan eksekusi oleh nilai quantum sehingga digunakan dalam tugas akhir ini. Di dalam algoritma RR tidak ada aturan yang mengatur urutan pengeksekusian di dalam antrian proses. Proses yang datang akan dieksekusi sesuai dengan waktu kedatangannya saja dan dieksekusi selama 1 quantum. Quantum adalah waktu yang diberikan oleh CPU untuk pengeksekusian proses. Akan terjadi masalah jika ada proses yang harus didahulukan pengeksekusian dan butuh prioritas. Maka dari itu penulis akan menambahkan aturan dimana proses dengan sisa waktu terbanyak akan mendapat giliran eksekusi lebih dulu, aturan seperti itu dikenal dengan *most time remaining* (MTR). Aturan tersebut bertujuan agar meminimasi waktu penyelesaian pada proses yang membutuhkan waktu layanan (*service time)* yang relatif besar. Diharapkan dengan meminimasi waktu penyelesaian pada proses dengan *service time* yang besar maka performansi penjadwalan akan semakin baik. Semakin kecil waktu penyelesaian pada proses maka proses itu semakin baik, sehingga penulis memberi prioritas tertinggi pada proses dengan sisa *service time* terbanyak. Maka dari itu penulis akan membandingkan parameter seperti *turn arround time*, average *waiting time, response time* dan CPU utilisation dengan algoritma seperti *time sharing* yang biasa digunakan pada sistem operasi Linux.

1.2 Perumusan masalah

Pada tugas akhir ini penulis melakukan penambahan aturan *most time remaining* pada algoritma penjadwalan RR. Perumusan masalah untuk tugas akhir ini dapat terklasifikasi dalam beberapa poin di bawah ini :

a) Kasus penjadwalan proses seperti apa yang cocok menggunakan algoritma round robin dengan aturan *most time remaining* ?
b) Bagaimana menerapkan aturan *most time remaining* pada algoritma RR ?

1.3 Batasan masalah

Adapun batasan masalah dalam tugas akhir ini diantaranya sebagai berikut:

a) Algoritma yang akan dioptimasi adalah algoritma penjadwalan round robin.
b) Aturan yang akan ditambahkan adalah aturan *most time remaining*.
c) Implementasi simulasi algoritma akan menggunakan simulator dengan tool CPUSS dengan menggunakan bahasa C#.
d) Simulasi dilakukan terhadap data uji CPU bound dan I/O bound.

1.4 Tujuan

Tujuan yang diharapkan penulis dapat tercapai pada pembuatan tugas akhir ini diantaranya :

a) Memodifikasi algoritma *Round Robin* dengan prioritas berdasarkan proses dengan sisa waktu terbanyak.
b) Menganalisis performansi average *waiting time, turn around time, response time*, CPU utilization dan NTAT (*Normalized Turn Arround Time*) pada algoritma *Round Robin* dengan aturan *most time remaining* pada penjadwalan proses dan membandingkannya dengan algoritma LTS (*Linux Time Sharing*)

1.5 Hipotesa

Setelah ditambahkannya aturan *most time remaining* (MTR) pada algoritma *Round Robin* (RR), penulis berhipotesa:

a) Algoritma RR yang ditambahkan aturan MTR akan memiliki nilai NTAT lebih kecil dari LTS sehingga performansi lebih baik daripada LTS.
b) Waktu total yang dibutuhkan untuk mengeksekusi proses-proses menjadi lebih singkat dibandingkan LTS.

1.6 Metodologi penyelesaian masalah

Metode yang digunakan untuk menyelesaikan permasalahan pada Tugas Akhir ini adalah sebagai berikut:

a) Studi Literatur

Pada tahap ini dilakukan pencarian referensi serta berbagai sumber lain yang digunakan sebagai penunjang dan acuan dalam proses pengerjaan tugas akhir ini.Studi ini dilakukan terhadap sumber-sumber yang mempunyai informasi tentang algoritma *Round Robin*,

b) IdentifikasiPermasalahan

Pada tahap ini dilakukan identifikasi terhadap masalah yang dialami oleh algoritma penjadwalan R*ound Robin.* Identifikasi tersebut meliputi parameter yang terdapat pada algoritma RR seperti penetapan *arrival time, time quantum, service time, dll* .

c) Perancangan Simulasi Algoritma

Pada tahap ini akan dibuat simulasi dari algoritma tersebut. Pembuatan simulasi akan menggunakan *software* yaitu CPUSS dan akan dibuat menggunakan bahasa pemograman C# serta menggunakan visual studio dalam implementasi algoritma di dalam CPUSS.

d) Pengujian Algoritma

Setelah tahap dibuat simulasinya kemudian dilakukan pengujian untuk mendapatkan data valid dari hasil kinerja algoritma tersebut. Pengujian akan dilakukan dengan 2 kasus skenario uji. Pada skenario uji yang pertama akan dilakukan pengujian pengaruh jumlah I/O bound dan pengaruh besarnya nilai quantum pada performansi simulasi penjadwalan. Pada skenario uji yang kedua akan diuji pengaruh jumlah proses terhadap performansi algoritrma *Round Robin Most Time Remaining*. Pada skenario uji yang kedua akan digunakan jumlah proses I/O bound dan nilai quantum dari hasil pengujian skenario uji pertama yang menghasilkan NTAT terbaik sebagai standar penilaian performansi penjadwalan yang baik.

e) Analisis Hasil

Pada tahap ini data valid dari hasil uji akan dianalisa, di bandingkan dengan Algoritma Time sharing pada linux. Parameter yang akan dianalisa adalah *average waiting time, turn around time, response time, CPU Utilisation dan NTAT.*

f) Penarikan kesimpulan dan pembuatan laporan

Setelah dianalisa lalu akan disimpulkan secara jelas kemudian di buat laporannya ke dalam sebuah dokumentasi buku laporan tugas akhir.