

BAB I

PENDAHULUAN

1.1. Latar Belakang

Sebuah *client* yang mengakses web server yang sedang sibuk mungkin dapat sangat lama menunggu respon dari server. Dalam Tugas Akhir ini penulis mengajukan cara bagaimana agar dapat mengurangi waktu tunggu tersebut. Tipe *request* yang dibahas dalam Tugas Akhir dibatasi hanya pada *request* yang bersifat statis (*GET request*).

Dalam Tugas Akhir ini penulis menggunakan *client response time* sebagai parameter utama. *Client response time* didefinisikan sebagai rentang waktu yang berawal sejak *client* mengirimkan paket SYN untuk membuka koneksi dengan server sampai ketika *client* menerima *byte* data terakhir dari file yang diminta.

Idenya adalah sebagai berikut. Biasanya, setiap *request* yang ada di web server akan menerima bagian yang sama dari *resource* yang dimiliki oleh web server. Metode penjadwalan seperti ini sering disebut dengan *FAIR scheduling*. Penulis mencoba untuk menerapkan pola penjadwalan yang lain yaitu *unfair scheduling*, dimana prioritas akan diberikan kepada *request* yang berukuran kecil. Metode ini disebut juga *size based scheduling*. Secara intuitif, *size based scheduling* dapat meminimalkan waktu tunggu *request*. Untuk itu *request* yang berukuran kecil tentunya lebih diutamakan daripada *request* yang berukuran besar. Membuat *request* berukuran besar menunggu di belakang *request* berukuran kecil dapat mengurangi *response time* rata-rata dibandingkan jika *request* berukuran kecil yang menunggu di belakang *request* berukuran besar. *Size-based scheduling* diharapkan dapat mengurangi total *response time* dari sistem.

Disamping kelebihan yang ditawarkan, *size-based scheduling* juga mempunyai kekurangan. Banyak aplikasi yang menghindari penggunaan metode antrian ini dengan satu alasan. Mungkin saja *request* yang berukuran besar harus menunggu sangat lama sebelum akhirnya memperoleh kesempatan untuk dilayani oleh server (*starvation*). Salah satu tujuan dari Tugas Akhir ini adalah untuk menyelidiki apakah kekhawatiran ini masuk akal.

Dalam konteks web server, *scheduling* dapat mempunyai arti yang luas. Web server bukanlah sistem dengan *resource* tunggal. Ada banyak kemungkinan dimana *scheduling policy* tadi dapat diterapkan. Yang menjadi pertimbangan utama adalah *resource* yang mengalami *bottleneck*-lah yang harus menerapkan *scheduling policy*. Dalam Tugas Akhir ini, *resource* dimana *scheduling policy* diterapkan adalah *bandwidth* server (bukan prosesor atau memori). Pada situs yang melayani *request* statis, umumnya *bottleneck* terjadi pada bagian *bandwidth* yang terbatas. Membeli lebih banyak *bandwidth* memerlukan biaya yang lebih besar daripada membeli lebih banyak prosesor atau memori.

Oleh karena itu, fokus dari Tugas Akhir ini adalah bagaimana menerapkan *size based scheduling* pada *bandwidth* server dan efeknya terhadap performa web server. Untuk itu, *size based scheduling* harus diterapkan pada level *network*.

1.2. Perumusan Masalah

Dari latar belakang di atas maka dapat dirumuskan masalah sebagai berikut:

1. Bagaimana merancang dan menerapkan sebuah metode antrian *size based scheduling* pada *bandwidth*/jaringan web server.
2. Bagaimana merancang sebuah uji coba dan simulasi untuk mendapatkan data yang diperlukan untuk proses analisa terhadap performa *size based scheduling*.
3. Bagaimana menganalisa dan menginterpretasikan data hasil simulasi sehingga menjadi sebuah kesimpulan sesuai dengan tujuan masing-masing simulasi.

1.3. Tujuan Pembahasan

Maksud dan tujuan dari Tugas Akhir ini adalah untuk:

1. Membandingkan *FAIR scheduling* dengan *size based scheduling*.
2. Mengetahui pengaruh *size based scheduling* terhadap performansi web server yaitu pada parameter berikut:
 - *Mean Response Time*
 - *Mean Slowdown*

3. Mengetahui pengaruh jumlah antrian pada *size based scheduling* terhadap performansi web server.

1.4. Batasan Masalah

Agar dalam pengerjaan tugas akhir ini didapatkan hasil yang optimal, maka masalah akan dibatasi sebagai berikut:

1. *Request* bersifat statis dengan distribusi *request* yang telah ditentukan
2. *Scheduling policy* diterapkan pada *bandwidth* server (di level *network*) bukan prosesor atau memori.
3. Implementasi di Linux.

1.5. Metodologi Penyelesaian Masalah

Metode penelitian yang dilakukan dalam tugas akhir ini adalah sebagai berikut:

1. Studi Literatur
Studi Literatur dengan mempelajari berbagai literatur yang relevan dengan permasalahan yang meliputi: buku referensi, bahan dari internet, paper, perangkat lunak dan lain-lain yang dapat digunakan sebagai bahan pendukung dari materi tugas akhir ini.
2. Perancangan Ujicoba dan Simulasi
Membuat pemodelan sistem, merancang skenario simulasi dan membangun lingkungan simulasi (*server, network, client*) sesuai dengan tujuan yang ingin dicapai.
3. Implementasi dan Pengumpulan Data
Melakukan proses simulasi yang telah dibuat sebelumnya dan melakukan pengumpulan data.
4. Analisa dan Evaluasi
Mengevaluasi dan menganalisa data hasil ujicoba.
5. Pengambilan kesimpulan dan penyusunan laporan.

1.6. Sistematika Penulisan

Adapun sistematika dari penulisan tugas akhir ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang, perumusan masalah, batasan masalah, tujuan, metodologi dan sistematika penulisan.

BAB II DASAR TEORI

Bab ini membahas tentang teori pendukung penelitian.

BAB III PERANCANGAN DAN IMPLEMENTASI SISTEM

Bab ini berisi tentang mekanisme perancangan lingkungan ujicoba dan implementasi simulasi.

BAB IV ANALISIS

Bab ini berisi analisis hasil simulasi.

BAB V KESIMPULAN DAN SARAN

Bab terakhir ini berisi kesimpulan dan saran yang diperoleh dari keseluruhan tugas akhir ini.