

## ANALISIS PERFORMANSI WEB SERVER DENGAN ALGORITMA FASTEST CONNECTION FIRST

Niko Wilfritz Sianipar<sup>1</sup>, Tri Brotoharsono<sup>2</sup>, Fazmah Arif Yulianto<sup>3</sup>

<sup>1</sup>Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

---

### Abstrak

Web server merupakan sebuah program komputer yang berada di server yang bertugas melayani request dari banyak user. Bagaimana jika request-request tersebut dilayani berdasarkan kecepatan koneksinya masing-masing? Apakah performansi web server akan meningkat? Pertanyaan-pertanyaan inilah yang melatarbelakangi Tugas Akhir ini. Tugas Akhir ini mengemukakan sebuah algoritma penjadwalan untuk pemrosesan request HTTP statik pada web server. Algoritma ini, yang disebut dengan Fastest Connection First (FCF), memberikan prioritas kepada request HTTP berdasarkan pada throughput koneksi user. Request yang melalui koneksi lebih cepat akan menerima prioritas lebih tinggi. Tugas Akhir ini membandingkan FCF dengan FIFO. Implementasi dilakukan pada web server Apache dan sistem operasi Linux. Simulasi dilakukan pada lingkungan LAN. Hasil simulasi mengindikasikan bahwa FCF menghasilkan pengurangan pada delay server yang berimbas pada response time rata-rata. Request yang melalui koneksi lambat mengalami sedikit perlambatan atau starvation.

Kata Kunci : web server, throughput, fastest connection first, performansi.

---

### Abstract

Web server is a server side computer program that services requests from many users. How if the requests scheduled according to their connection's throughput? Is the web server's performance improved? These are the questions which be the background of this final project. This final project proposes a scheduling policy for processing of the static HTTP requests in web server. This policy, called Fastest Connection First (FCF), gives priority to HTTP requests based on the throughput of the user's connection. The requests issued through faster connections receive the higher priorities. This final project compares FCF with FIFO policy. The implementation is on Apache web server and Linux operating system. Experiments are executed in LAN environment. Results indicate that FCF yields some reductions in delay at web server. These results are in mean response time. The request from low connection experiments a small penalization.

Keywords : web server, throughput, fastest connection first, performance.

Telkom  
University

# 1. PENDAHULUAN

## 1.1 Latar Belakang Masalah

Banyak sekali algoritma penjadwalan yang telah didesain untuk memperbaiki *delay* pada *web server*, yang mana (*delay*) mungkin saja merupakan bagian penting dari *response time*. Apakah cukup untuk mengurangi *response time* dari setiap *request* hanya dengan mengubah urutan penjadwalan request-request tersebut atau dengan hanya memberikan prioritas yang tinggi terhadap *request* yang mempunyai ukuran *file* yang lebih kecil? Bagaimanapun penjadwalan-penjadwalan yang berhubungan dengan hal-hal yang ditanyakan di atas tidak mempertimbangkan interaksi antara sistem *server* dan implementasi TCP pada tiap *host* yang terhubung ke *server*. *Web server* digunakan terutama dalam *wide-area* Internet dimana terdapat variasi yang cukup banyak dalam hal *bandwidth*, *RTT* dan *packet lost characteristics*.

Tugas Akhir ini mencoba untuk mengimplementasikan sebuah *scheduling policy* untuk pemrosesan *request* HTTP statik pada *web server*. Penjadwalan ini, yang disebut dengan *Fastest Connection First* (FCF), memberikan prioritas bagi koneksi *user*. *Request* untuk file-file yang berukuran lebih kecil serta melewati koneksi yang lebih cepat diberikan prioritas yang lebih tinggi. Tujuan Tugas Akhir ini adalah untuk memperbaiki *response time* rata-rata. Yang mana jika koneksi yang paling cepat menerima prioritas paling tinggi pada *server*, maka mereka akan dapat mengakhiri proses transaksi dengan cepat, hal inilah yang pada akhirnya membuat *response time* menjadi semakin kecil. Penjadwalan ini mempertimbangkan interaksi antara sistem *server* dan implementasi TCP pada *server*. Tugas Akhir ini membandingkan FCF dengan algoritma penjadwalan *First In First Out* (FIFO).

Disamping kelebihan yang diberikan, FCF juga memiliki kelemahan. Satu alasannya adalah bahwa algoritma ini memberikan prioritas kepada koneksi cepat, lalu bagaimana dengan paket yang melalui *slow connection*. Mungkin saja *request* yang melalui *slow connection* harus menunggu sangat lama (*starvation*) karena mendapatkan prioritas yang rendah dari *server*. Salah satu tujuan dari Tugas Akhir ini adalah untuk menyelidiki apakah *request* yang melalui *slow connection* mengalami *starvation* yang terlalu berlebihan.

## 1.2 Perumusan Masalah

Pada Tugas Akhir ini, yang pertama kali ingin dibuktikan apakah performansi *web server* akan meningkat bila diterapkan algoritma penjadwalan FCF dibandingkan algoritma penjadwalan yang lain (dalam hal ini algoritma FIFO). Parameter peningkatan performansi tersebut dalam hal ini antara lain : *delay* pada *server*, *throughput* koneksi, *response time* rata-rata, serta *starvation* pada *request* melalui *slow connection*.

Algoritma FCF memberikan *service* terutama sekali kepada *user* yang memiliki *throughput* koneksi yang paling cepat. Permasalahan yang kemudian muncul adalah bagaimana menentukan *throughput* koneksi tiap *user* untuk kemudian ditentukan *user* mana yang memiliki *throughput* koneksi yang paling cepat. Dalam jaringan komunikasi, *throughput* merupakan jumlah data digital per satuan waktu yang dikirimkan melalui baik *link* fisik maupun logic, atau yang mengalir dari suatu node ke node lain. Sebagai contoh, *throughput* bisa merupakan jumlah data yang disampaikan ke sebuah terminal jaringan tertentu atau sebuah *host*, atau antara dua komputer yang saling terhubung. *Throughput* biasanya diukur dalam bit per detik (bps), kadang-kadang dalam banyak paket per detik. Sehingga *throughput* dianalogikan juga sebagai besar pemakaian *bandwidth*. Penjelasan lebih lanjut mengenai *throughput* dapat dilihat pada Bab Dasar Teori.

Permasalahan yang muncul berikutnya adalah bagaimana merancang sebuah ujicoba dan simulasi untuk mendapatkan data yang diperlukan untuk proses analisa terhadap performansi FCF serta bagaimana menganalisis dan menginterpretasikan data hasil simulasi sehingga menjadi sebuah kesimpulan sesuai dengan tujuan masing-masing.

Asumsi sekaligus batasan masalah yang dipakai dalam Tugas Akhir ini sebagai berikut:

- *Throughput client* mulai dari saat *request* diterima *server* sampai *request* terpenuhi oleh *server* selalu sama.
- *Request* bersifat statis.

### 1.3 Tujuan Pembahasan

Tujuan dari Tugas Akhir ini adalah:

- Menerapkan penjadwalan FCF pada *web server* sehingga diharapkan terjadi perbaikan dari segi *delay*, yang merupakan bagian penting dari *response time*.
- Menyimpulkan hasil perbandingan algoritma penjadwalan FCF dengan algoritma penjadwalan FIFO dari segi mean *response time*, *delay* pada *server*, *throughput* koneksi, analisis *starvation* pada *request* yang berada pada *slow connection* yang dihasilkan oleh masing-masing algoritma penjadwalan tersebut.

Dugaan awal (hipotesis awal) yang ingin dibuktikan dalam Tugas Akhir ini sebagai berikut:

- Memberikan prioritas kepada *request* yang melalui *fast connection* maka akan dapat memperbaiki mean *response time*. *Response time* di sini adalah waktu mulai dari *client* mengirimkan *request* sampai *client* mendapatkan *byte* terakhir dari *file* yang di-*request*.

## 1.4 Metodologi Penyelesaian Masalah

Metodologi yang digunakan untuk menyelesaikan masalah dalam Tugas Akhir ini adalah:

1. Studi Literatur  
Studi literatur dari beberapa buku, jurnal, artikel yang membahas algoritma penjadwalan terutama FCF dan FIFO serta penerapannya pada *platform* Linux dan *Apache web server*.
2. Perancangan Model Simulasi  
Model simulasi yang dirancang diantaranya model *server*, workload generator, dan model LAN.
3. Implementasi  
Penerapan algoritma penjadwalan FCF dan FIFO untuk kemudian dibandingkan dari segi *mean response time*. Implementasi dilakukan terutama pada konfigurasi Apache pada *platform* Linux.
4. Pengujian  
Pengujian dilakukan dengan mengirimkan *request* dari beberapa *user* dengan menggunakan workload generator kepada *web server* yang sudah dimodifikasi dengan algoritma penjadwalan FCF dan FIFO. Kemudian dicatat waktu *delay* pada *server* serta *response time* rata-rata.
5. Analisis  
Analisis terhadap hasil pengujian yang telah didapatkan untuk kemudian dicapai simpulan perbandingan algoritma penjadwalan FCF dan FIFO dari segi *mean response time*, *delay* pada *server*, *throughput* koneksi, analisis *starvation* pada *request* yang melalui *slow connection* yang dihasilkan oleh masing-masing algoritma penjadwalan tersebut.
6. Penyusunan Laporan  
Hasil penelitian akan disusun menjadi suatu laporan yang meliputi aspek-aspek dalam penelitian yaitu teori dan implementasinya.

Telkom  
University

## 1.5 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.

Telkom  
University

## 5. KESIMPULAN DAN SARAN

Pada bab ini, penulis mencoba menyimpulkan hasil dari pengujian yang dilakukan dan telah dijelaskan serta dianalisis hasilnya pada bab sebelumnya. Pada bab ini juga tertulis saran untuk pengembangan dan perbaikan lebih lanjut.

### 5.1 Kesimpulan

Berdasarkan pengujian dan analisis yang telah dilaksanakan dan diuraikan pada bab IV, maka dapat disimpulkan hal-hal sebagai berikut:

1. Penjadwalan *request* pada *web server* dengan mendahulukan *request* yang berasal dari koneksi dengan *throughput* terbesar dapat menurunkan *response time* rata-rata terhadap *request user*. Penurunan *response time* rata-rata (dalam simulasi rata-rata penurunan sebesar 0.4 detik) tersebut didapat dari penurunan *delay server*.
2. Dengan penjadwalan *request* berdasarkan kecepatan koneksi maka *throughput* sistem juga meningkat (dalam simulasi rata-rata kenaikan sebesar 0.73 Mbyte/sec).
3. Penurunan *response time* rata-rata dan peningkatan *throughput* tersebut didapat dengan mendahulukan koneksi cepat dan mengorbankan koneksi lambat. *Request* dengan koneksi lambat tersebut mengalami penurunan performansi tapi dalam batas yang masih bisa ditoleransi. Masih dapat ditoleransi karena dibandingkan dengan FIFO, perlambatan yang dialami oleh koneksi lambat (dalam simulasi untuk *concurrent user* 140 mengalami perlambatan sebesar 4.38 kali dan untuk *concurrent user* 190 mengalami perlambatan sebesar 3.44 kali pada *throughput* koneksi 1.8 KBps) lebih kecil dari percepatan yang dialami oleh koneksi cepat (dalam simulasi untuk *concurrent user* 140 mengalami percepatan sebesar 10.06 kali dan untuk *concurrent user* 190 mengalami percepatan sebesar 7.29 kali pada *throughput* koneksi 1310.7 KBps).

## 5.2 Saran

1. Pada Tugas Akhir ini, jenis *request* dibatasi pada jenis *file* statis. Untuk pengembangan lebih lanjut, dapat diadakan pengujian pada *file* non statis.
2. Tugas Akhir ini hanya menggunakan *file* uji berukuran 7 KB, ke depannya dapat diteliti dengan ukuran yang lain.
3. Perhitungan *throughput* koneksi seperti yang dilakukan pada Tugas Akhir ini dirasakan cukup memakan waktu untuk trafik yang tidak terlalu sibuk dan keakuratannya juga tidak mendekati 100%. Perhitungan *throughput* koneksi juga mengasumsikan *throughput* sebuah koneksi selalu bernilai konstan. Lebih lanjut dapat dilakukan penelitian cara menghitung *throughput* koneksi yang lebih cepat dan efektif serta nilainya lebih bersifat *real-time*.
4. Pada penelitian lebih lanjut dapat dicoba pada *platform* sistem operasi lainnya, *web server* open source lainnya, dan area jaringan yang lebih luas.





## Daftar Pustaka

- [1] Apache Group, Apache Web server, <http://www.apache.org>, didownload pada tanggal 10 Oktober 2007.
- [2] Crovella, Mark E.. *Heavy-Tailed Probability Distributions in the World Wide Web*, Boston University.
- [3] Differentiated Services on Linux, <http://diffserv.sourceforge.net/>, didownload tanggal 20 September 2007.
- [4] First-come, first served, <http://en.wikipedia.org/wiki/FIFO>, didownload tanggal 25 Maret 2008.
- [5] Jogiyanto. Prof. Dr, 1992, “*Konsep Dasar Pemrograman Bahasa C*”, Penerbit Andi, Yogyakarta.
- [6] Linux Reviews, <http://www.linuxreviews.org>.
- [7] Murta. C, Corlassoli. T, “*Fastest Connection First: A New Scheduling Policy for Web servers*”, Federal University of Parana, Brazil.
- [8] setsockopt(2) – phpMan, <http://www.chedong.com/phpMan.php/man/SETSOCKOPT/2>, didownload tanggal 26 Maret 2008.
- [9] Siege, an HTTP regression tester & benchmarking utility, <http://www.joedog.org/JoeDog/Siege>, didownload pada tanggal 26 Maret 2008.
- [10] SJF (Shortest Job First), <http://bebas.vlsm.org/v06/Kuliah/SistemOperasi/BUKU/SistemOperasi-4.X-1/ch14s03.html>, didownload tanggal 25 Maret 2008.
- [11] tc (Linux Reviews), <http://linuxreviews.org/man/tc/>, didownload tanggal 26 September 2008.
- [12] tc-prio(8) – Online Manual Page Of Unix/Linux, <http://www.phpman.info/index.php/man/tc-prio/8>, didownload tanggal 26 September 2007.
- [13] Wahyudi. A, “*Size-based Scheduling pada Web server Untuk Meningkatkan Response time*”, 2006, Institut Teknologi Telkom, Bandung.
- [14] 1998 World Cup Web Site Access Logs, <http://ita.ee.lbl.gov/html/contrib/WorldCup.html>, didownload tanggal 22 Mei 2008.