

BAB II

LANDASAN TEORI

Bab ini akan membahas tentang teori-teori dan aplikasi–aplikasi pendukung dalam menghitung dan menganalisis kinerja level komponen dan protokol jaringan yaitu : *Performance Modelling Concepts*, *Microsoft Management Console* (MMC), *Ethereal*, *MySQL*, *PHP* dan *IIS*. *Performance Modelling Concepts* merupakan standar yang digunakan untuk menghitung kinerja sistem.

2.1 Performance Modelling Concepts

Metric Analisis Kinerja secara umum digolongkan menjadi 2 bagian besar yaitu : berdasarkan Response Time dan Throughput (Menasce, 1994). Hasil perhitungan Response Time berdasarkan pada waktu penyelesaian task mulai dari awal sampai akhir event, mulai task datang, dilayani sampai task meninggalkan server. Hasil perhitungan Throughput berdasarkan pada jumlah task yang dapat dilayani dalam waktu tertentu, selama waktu tertentu server dapat melayani berapa task.

2.2.1. Performance Modelling Concepts untuk Physical Disk dan Processor

Tiga hukum yang digunakan untuk mengevaluasi *performance* atau kinerja sistem adalah: Utilization Law, Little's Law dan Forced Flow Law (Menasce, 1994). Hukum-hukum tersebut akan menghitung *Utilization*, *Response Time*, *Service Time* dan *Throughput* sistem serta server.

a. Utilization Law

Selama dilakukan pengamatan dalam periode waktu T , *task* datang ke server, server memproses *task* sampai selesai dan meninggalkan server. Selama periode waktu T pada server, dapat diketahui:

- a. Jumlah *task* yang datang ke server selama waktu T , yang diidentifikasi dengan A .
- b. Total waktu dimana server sibuk melakukan proses terhadap *task* yang datang selama waktu T , diidentifikasi dengan B .
- c. Jumlah keseluruhan *task* yang dapat diselesaikan oleh server diidentifikasi dengan C .

Ukuran diatas dapat digunakan untuk menghitung :

- a. Rata-rata *service time* untuk *task* yang dapat diselesaikan dalam *unit of time* : $S = B/C$ (2.1)

- b. *Output rate of Throughput of server*, dalam satuan *task per unit of time* : $X = C/T$ (2.2)

- c. Prosentase dari periode pengamatan dan *server* melakukan proses *task*, yang disebut *Utilization of Server* : $U = B/T$ (2.3)

Perkalian antara *Service time* dengan *Throughput* menghasilkan *Utilization*.

$$B/C \times C/T = B/T \quad (2.4)$$

Hukum *Utilization* :

$$U_{(i)} = X_{(i)} \times S_{(i)} \quad (2.5)$$

dimana i mengidentifikasi server.

b. Little's Law

Jumlah *task* yang diproses oleh server pada masing-masing selang waktu t selama periode pengamatan menghasilkan jumlah total *waiting* dan *process time* untuk semua *task* selama waktu T . Pengukuran dari *task* yang selesai C , dalam periode pengamatan T sama dengan akumulasi *waiting time* W selama T , dapat digunakan untuk menghitung :

- a. Rata-rata *response time task* terselesaikan, dengan satuan *unit of time* :

$$R = W/C \quad (2.6)$$

- b. Rata-rata jumlah *task* pada server : $N = W/T$ (2.7)

- c. *Throughput Server* : $X = C/T$ (2.8)

Perkalian antara *response time* dan *Throughput* sama dengan rata-rata jumlah *task* pada server, atau rata-rata panjang antrian pada server :

$$W/C \times C/T = W/T \quad (2.9)$$

Little's Law adalah relasi :

$$N_{(i)} = R_{(i)} \times X_{(i)} \quad (2.10)$$

dimana i adalah server.

c. Forced Flow Law

Sebuah transaksi akan mengalir sampai sistem yang terdapat pada server. Sebuah transaksi mungkin mempunyai beberapa *task* yang harus selesai sebelum meninggalkan server. Selama waktu pengamatan T , dapat dihitung :

- a. Jumlah transaksi yang dapat diselesaikan oleh server $C(0)$.
 b. Jumlah *task* yang dapat diselesaikan oleh server i $C(i)$.

Pengukuran diatas mendukung untuk perhitungan :

- a. Rata-rata jumlah *task* per transaksi dari server(i), yang disebut sebagai *Visit Ratio* dari server :

$$V_{(i)} = C_{(i)} / C_{(0)} \quad (2.11)$$

- b. Rata-rata jumlah transaksi yang dapat diselesaikan server selama waktu T , yang disebut *System Throughput* :

$$X_{(0)} = C_{(0)} / T \quad (2.12)$$

- c. Rata-rata jumlah dari *task* yang dapat diselesaikan server i selama waktu T , disebut *Server Throughput* :

$$X_{(i)} = C_{(i)} / T \quad (2.13)$$

Perkalian *System Throughput* dengan *Visit Ratio* sama dengan *Server Throughput* seperti pada persamaan 2.14.

$$C_{(i)} / C_{(0)} \times C_{(0)} / T = C_{(i)} / T \quad (2.14)$$

The Forced Flow Law :

$$X_{(i)} = V_{(i)} \times X_{(0)} \quad (2.15)$$

Sistem Throughput dapat dihitung dengan *Visit Ratio* dan *Throughput* pada suatu server. Dengan mengetahui *Visit Ratio* dari semua server dan *Throughput* hanya satu server, memungkinkan dilakukannya perhitungan *Throughput* pada semua sistem dalam server.

2.2.2. Performance Modelling Concepts untuk Incoming/Outgoing Link

Untuk menghitung analisis kinerja level komponen pada Incoming dan Outgoing Link digunakan Performance Modelling Concepts yang berbeda dengan pada komponen Physical disk dan processor (Menasce, 1999). Persamaan 2.16

dan 2.17 adalah rumus yang digunakan untuk menghitung *Residence Time* pada level komponen incoming dan outgoing link.

$$R_{\text{InLink}} = \frac{D_{\text{InLink}}}{1 - \rho_{\text{InLink}}} \quad (2.16)$$

$$R_{\text{OutLink}} = \frac{D_{\text{OutLink}}}{1 - \rho_{\text{OutLink}}} \quad (2.17)$$

Dari persamaan 2.16 dan 2.17, *Service Demand* yang diidentifikasi dengan D didapatkan dengan menggunakan persamaan 2.18 dan 2.19.

$$D_{\text{InLink}} = \frac{\text{Avg}(\text{size=frame})_{\text{InLink}}}{\text{Bandwidth}} \quad (2.18)$$

$$D_{\text{OutLink}} = \frac{\text{Avg}(\text{size=frame})_{\text{OutLink}}}{\text{Bandwidth}} \quad (2.19)$$

2.2 Mean Value Analysis

Mean Value Analysis digunakan untuk menyelesaikan analisis dari *closed queueing networks* (Bobbin, 2004). Tiga Operational Laws dan rangkaian iterasi perhitungan menghasilkan *Throughput* dan *Response Time* dari suatu sistem, seperti halnya *utilization* dan *degree of multiprogramming* dari suatu sistem adalah ρ_i ($\rho_i = \rho$). Jika tidak ada *multiprogramming*, panjang antrian pada masing-masing *server* 0. Jika model menyajikan *closed system*, *degree of multiprogramming* adalah *load* pada sistem. *Visit Ratio*, (v dengan subscript i ,

yang berarti server i) dan rata - rata *Service Time* (S_i , dengan subscript i , yang berarti server ke i) dapat diketahui untuk setiap server.

Untuk menghitung Mean Value Analysis :

- a. Rata-rata *Residence Time* pada tiap server adalah waktu yang dihabiskan di server oleh suatu transaksi selama kunjungan ke server (Bobbin, 2004) seperti ditunjukkan pada persamaan 2.20.

$$R_i(n) = V_i R_i = V_i S_i(n) [1 \leq i \leq n] \quad (2.20)$$

$V_{(i)} \times S_{(i)} = D_{(i)}$ disebut sebagai *Service Demand* pada server.

- b. Gunakan total dari rata-rata *Residence Time* semua server untuk menghitung *System Throughput* pada derajat dari *multiprogramming level* = n , pada persamaan 2.21.

$$X_0(n) = \frac{n}{\sum_{k=1}^n R_k(n)} \quad (2.21)$$

- c. Dengan menerapkan Little's Law, panjang antrian arus *multiprogramming level* dapat diperoleh untuk tiap-tiap server adalah seperti persamaan 2.22.

$$n_i(n) = X_0(n) R_i(n) \quad (2.22)$$

- d. Kenaikan n dan pengulangan langkah 1-3 sampai $n > \text{actual degree of programming}$. *System Throughput* telah dihitung pada langkah b. Maka *System Response Time* adalah tampak pada persamaan 2.23.

$$R(n) = \frac{\sum_{k=1}^n R_k(n)}{V_k} \quad (2.23)$$

Utilization Law dan Forced Flow Law dapat diterapkan untuk memperoleh informasi dari Mean Value Analysis untuk menentukan ukuran unjuk kerja sistem (Bobbin, 2004).

2.3 Protokol Jaringan

Protokol jaringan adalah standar atau konvensi yang digunakan untuk komunikasi antar perangkat jaringan. Protokol terdiri dari format rule yang mendefinisikan bagaimana data dikirimkan dari suatu *Source Address* ke *Destination Address*.

Dalam membahas protokol jaringan, tidak dapat dipisahkan dari keberadaan OSI Layer Model, yang dikembangkan oleh International Organization for Standardization (OSI). OSI Layer merupakan standar yang digunakan untuk komunikasi data antar perangkat jaringan yang bervariasi dan dari vendor yang berbeda.

OSI Layer terdiri 7 layer, masing-masing layer bisa mempunyai sub layer (Odom, 2000) yaitu :

a. Layer 7 (Application Layer)

Suatu aplikasi yang berkomunikasi dengan komputer lain akan mengimplementasikan OSI pada layer application ini. Contoh protokol yang ada pada layer ini adalah HTTP, FTP, WWW Browser, SMTP gateways (Eudora, CC:mail), SNMP.

b. Layer 6 (Presentation Layer)

Layer ini bertujuan untuk mendefinisikan format suatu data yang akan dikirimkan dari application layer. Contoh dari presentation layer adalah JPEG, ASCII, EBCDIC, TIFF, GIF, MPEG, MIDI dan proses enkripsi.

c. Layer 5 (Session Layer)

Layer ini mendefinisikan bagaimana proses komunikasi antar komputer dikendalikan mulai dari mulai, selama proses komunikasi, sampai sesi

komunikasi berakhir. Contoh dari layer ini adalah SQL, NFS, Netbios Name, AppleTalk.

d. Layer 4 (Transport Layer)

Layer ini bertanggung jawab terhadap recovery error dalam pengiriman data, bagaimana jika pengiriman data tidak sampai pada tujuan atau ada kerusakan data dalam proses pengirimannya. Contoh protokol pada layer ini adalah TCP, UDP, SPX.

e. Layer 3 (Network Layer)

Layer ini mendefinisikan end to end pengiriman packet (data yang berada pada layer 3 disebut packet). Untuk mendefinisikan end to end ini diperlukan alamat logical yang disebut IP Address. Dalam layer ini didefinisikan bagaimana proses routing bekerja dan bagaimana jalur dipelajari oleh packet yang akan dikirimkan ke tujuan/Destination IP. Contoh protokol pada layer ini adalah IP, IPX, AppleTalk DDP.

f. Layer 2 (Data Link Layer)

Data Link Layer mendefinisikan bagaimana data dikirimkan melalui media pengiriman yang bervariasi. Contoh protokol pada layer ini adalah IEEE 802.3/802.2, HDLC, Frame Relay, PPP, FDDI, ATM.

g. Layer 1 (Physical Layer)

Layer ini akan memberikan karakteristik fisik media jaringan komputer termasuk connector, pin dan lain-lain. Contoh protokol pada layer ini adalah EIA/TIA-232, V.35, Ethernet, RJ45, FDDI.

Dari penjabaran OSI layer diatas, dapat diketahui bahwa protokol tidak dapat dipisahkan dari OSI Layer Model karena tiap protokol bekerja pada layer OSI. Contoh protokol dalam layer OSI antara lain :

1. Addressing Resolution Protocol (ARP)

ARP akan dikirimkan ketika suatu *client* akan mengirimkan data ke *client* lain tetapi hanya mempunyai alamat MAC saja, tidak diketahui IP Addressnya. Untuk mengirimkan data diperlukan MAC Address dan IP Address, sehingga ARP akan mengirimkan frame ke seluruh *client* yang berada dalam 1 network untuk meminta alamat IP suatu *client* yang hanya diketahui MAC Addressnya.

2. User Datagram Protocol (UDP)

UDP adalah protokol process-to-process yang menambahkan alamat port dalam proses pengirimannya. UDP juga melakukan *chek-sum error control* yang berada pada Transport Layer OSI. Protokol ini bersifat *connection-less*, artinya tidak diperlukan proses *handshaking* sebelum memulai komunikasi data.

3. Transmission Control Protocol (TCP)

TCP menyediakan layanan penuh pada lapisan transport untuk aplikasi. TCP merupakan protokol transport yang *reliable/connection-oriented*, artinya koneksi antar *source* dan *destination* harus dibangun terlebih dahulu sebelum *source* mengirimkan data ke *destination*.

4. Hypertext Transfer Protocol (HTTP)

HTTP adalah standar rule yang digunakan untuk transfer file berupa text, grafik, image, sound, video dan file multimedia lainnya dengan menggunakan

World Wide Web (www). Untuk menampilkan data dari protokol HTTP digunakan tool standar yang dikenal dengan nama browser.

5. Spanning Tree Protocol (STP)

STP adalah protokol manajemen yang merupakan standar IEEE 802.1 untuk media akses bridge. STP merupakan algoritma dalam mengurangi redundancy pada jaringan.

6. Enhanced Interior Gateway Routing Protocol (EIGRP)

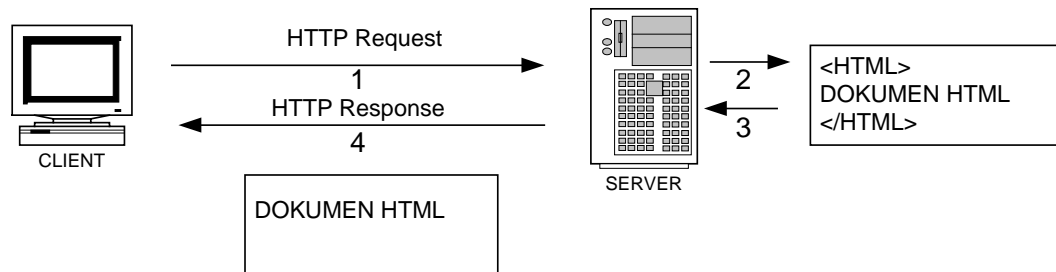
EIGRP merupakan pengembangan dari IGRP yang digunakan pada skala network yang besar daripada pada IGRP. Protokol EIGRP digunakan oleh router untuk berkomunikasi dengan router lain.

2.4 PHP

PHP (*Hypertext PreProcessor*) merupakan *script* yang membuat sebuah halaman web menjadi dinamis, yang berarti halaman web menjadi lebih interaktif. Halaman yang akan ditampilkan dibuat saat *client* melakukan *request* halaman tersebut, sehingga informasi yang diterima oleh *client* adalah selalu informasi yang terbaru. *Script* PHP dieksekusi pada server dimana *script* tersebut dijalankan (*serverside*), jadi semua informasi yang ingin ditampilkan di halaman *web* bisa dilihat dengan baik oleh semua jenis *browser client*. PHP termasuk dalam HTML-*embedded*, oleh karena itu *script* PHP bisa disisipkan pada sebuah halaman HTML.

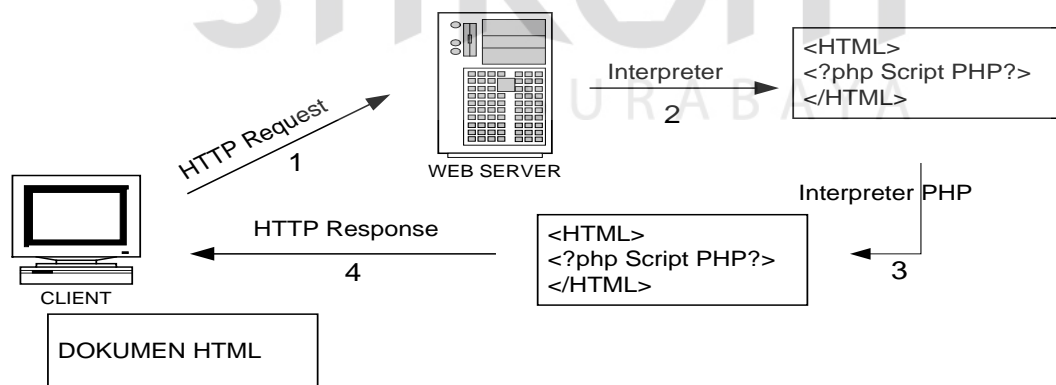
Perbedaan utama antara *script* PHP dengan HTML adalah HTML murni sebuah dokumen teks sedangkan *script* PHP di dalamnya terdapat program yang akan diproses oleh *web server* dan hasil pemrosesannya adalah sebuah dokumen

teks. Proses *request* pada HTML murni dan PHP dapat dilihat pada Gambar 2.1 dan Gambar 2.2.



Gambar 2.1 Proses pada *request* HTML murni

Pada Gambar 2.1 dan Gambar 2.2 tersebut terlihat bahwa perbedaan dari keduanya adalah adanya *script* PHP yang akan diproses/diinterpretasikan terlebih dahulu oleh PHP *interpreter* dalam *web server* sebelum dikirimkan ke *browser* pada *client*. Proses inilah yang menyebabkan PHP mampu membuat sebuah halaman *web* menjadi dinamis.

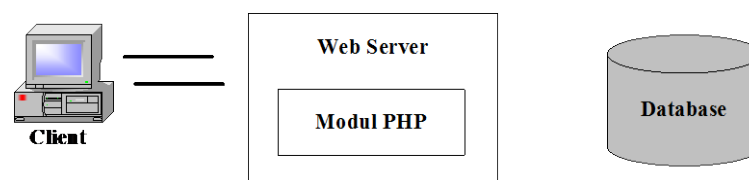


Gambar 2.2 Proses pada *request* PHP

Pada awalnya PHP adalah sebuah proyek pribadi dari Rasmus Lerdorf yang membuat PHP versi pertama untuk *homepage* pribadinya, versi ini masih

berupa kumpulan *script* Perl. Kemudian Rasmus membuat versi kedua dari PHP dengan cara menulis ulang *script-script* Perl menggunakan bahasa C. Pada versi kedua ini ditambahkan dua fasilitas yang penting yaitu *Form* HTML dan koneksi dengan *database* MySQL. PHP versi ketiga dikembangkan oleh Rasmus dan suatu kelompok *open source*, dimana pada versi ini PHP mulai menampakkan keunggulannya sebagai sebuah bahasa *server scripting* yang handal (Choi, 2000). Melalui perkembangan yang pesat ini banyak fasilitas yang ditambahkan dan oleh kelompok ini PHP disebut sebagai "PHP: *Hypertext Preprocessor*". Sintak yang digunakan berasal dari bahasa C, Java dan Perl. Sampai dengan tulisan ini dibuat, versi terakhir dari PHP adalah 4.3.10. PHP juga mendukung beberapa layanan yang menggunakan protokol seperti IMAP, SNMP, NNTP, POP3, HTTP, dan protokol-protokol lainnya. Beberapa *database* yang didukung oleh PHP diantaranya adalah Adabas D, Ingres, Oracle, dBase, InterBase, PostgreSQL, mSQL, MS-SQL, Sybase, IBM DB2, MySQL, Informix, ODBC (Choi, 2000).

Mekanisme kerja sebuah *web server* yang memanfaatkan PHP dan *database* ditunjukkan pada Gambar 2.3. Web Server PHP akan mengakses *database* server untuk melayani *request* dari *client*.



Gambar 2.3. Mekanisme kerja PHP mendukung *database*

2.5 MySQL

MySQL adalah sebuah sistem manajemen *database open source* yang populer dan gratis untuk *platform* Unix dan Windows. Sistem manajemen database MySQL menggunakan kumpulan perintah sederhana untuk memasukkan, memanggil, menghapus dan memperbarui data, maka dapat dikembangkan database yang kompleks.

Beberapa kemampuan MySQL (MySQL, 2004) adalah sebagai berikut :

- a. MySQL bisa diakses dan dimanipulasi dari sejumlah bahasa pemrograman terkenal, diantaranya adalah C, C++, Java, Perl, Python dan PHP.
- b. MySQL ditulis dalam C/C++ dan dioptimasi untuk *platform* Unix dan Win32.
- c. MySQL mendukung tipe data yang umum digunakan, termasuk FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, SET dan ENUM.
- d. MySQL mendukung subset fungsi *query* dan pengelompokan lanjut, termasuk diantaranya GROUP BY dan ORDER BY.
- e. MySQL memungkinkan alokasi *password* tiap *server*. *Password* yang melalui MySQL diperlukan untuk *encrypted authentication*.
- f. MySQL mendukung berbagai macam metode koneksi, seperti TCP/IP, soket Unix, dan koneksi untuk Windows NT/2000.
- g. MySQL bisa diperoleh secara gratis termasuk aplikasi-aplikasi lain yang diperlukan dalam memakai MySQL, seperti misalnya MySQLFront sebagai tool front end MySQL.

MySQL juga merupakan salah satu sistem manajemen database yang stabil di pasaran. Ketika MySQL diluncurkan pertama kali pada pertengahan 1996, beberapa *bug* dengan cepat dapat diketahui dan diperbaiki. Sekarang MySQL sudah menjadi sangat stabil dan banyak dipercaya oleh korporasi-korporasi di dunia untuk menyimpan data bisnis penting. Data ini biasanya membutuhkan media penyimpanan yang besar dan hal ini bukan menjadi masalah bagi MySQL, karena tabel MySQL dapat menampung data lebih dari 4 Gigabytes. MySQL 3.23 berisi jenis tabel baru yaitu MyISAM yang sanggup menampung 8 juta Terabytes (MySQL, 2004).

2.6 IIS

Internet Information Service (IIS) merupakan web server berkecepatan tinggi yang digunakan untuk mempublikasikan dan mendistribusikan content *www-based* pada standar browser. IIS hanya ada pada platform Windows. IIS 5.0 secara otomatis ada pada Windows 2000 Server atau Windows 2000 Advanced Server serta tidak disediakan untuk Sistem Operasi sebelumnya dari Microsoft. IIS 5.0 dapat diinstal pada Windows 2000 Professional Edition tetapi ada beberapa fungsi yang tidak dapat digunakan yaitu : *hosting multiple web sites*, *logging* pada ODBC Database serta tidak dapat membatasi IP Address yang melakukan akses. *Service* protokol yang disediakan oleh IIS 5.0 (Walker, 2003) antara lain :

a) Hypertext Transfer Protocol (HTTP)

HTTP adalah sekumpulan aturan untuk memindahkan file berupa text, grafik, image, sound, video dan file multimedia lainnya melalui World Wide Web (WWW). Untuk mengakses WWW diperlukan software yang biasa

dikenal dengan nama browser pada sisi *client*. Jenis browser yang dapat digunakan untuk mengakses WWW tersebut antara lain : Internet Explorer (IE), Netscape, Opera dan lain-lain.

b) File Transfer Protocol (FTP)

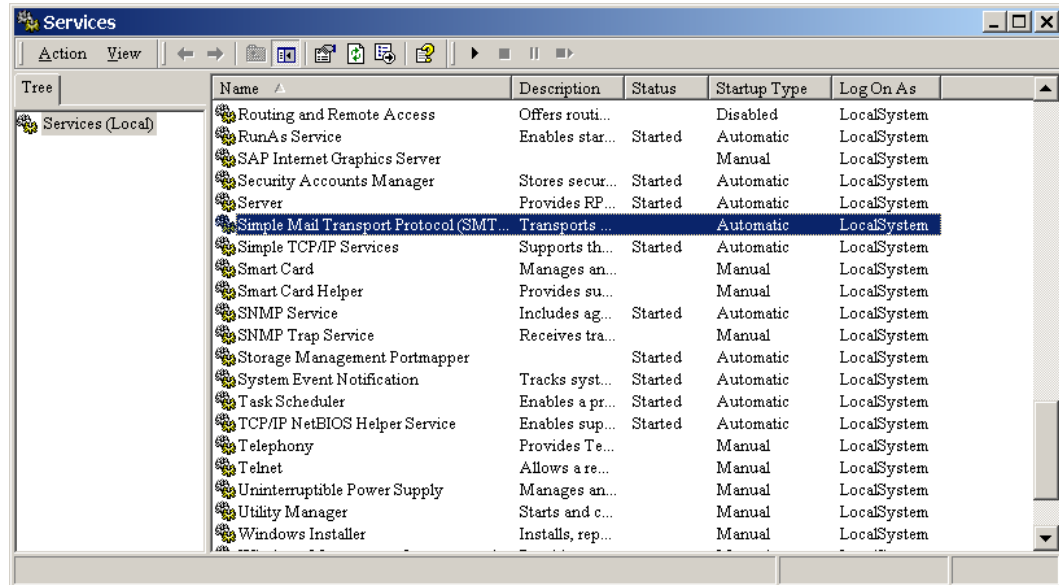
FTP merupakan standar Internet Protocol (IP) yang berfungsi untuk mempermudah pertukaran file antara suatu komputer dengan komputer lain pada jaringan internet. Umumnya FTP digunakan untuk mengirimkan file web page dari pembuat/programmer web ke server agar siapa saja dapat mengaksesnya melalui internet. User internet biasa menggunakan FTP untuk mengambil file yang ada di internet.

c) Simple Mail Transfer Protocol (SMTP)

SMTP merupakan TCP/IP protokol yang digunakan untuk mengirimkan dan menerima e mail pada jaringan komputer, internet maupun intranet. SMTP ini terbagi menjadi 2 fungsi protokol, POP3 atau IMAP yang digunakan untuk mengambil e mail dari server ke client dan SMTP digunakan untuk mengirimkan e mail dari client ke server.

d) Network News Transfer Protocol (NNTP)

NNTP adalah protokol yang digunakan oleh komputer client dan server untuk mengatur *note* dari client yang ditempatkan pada Newsgroup. NNTP mengatur global network dari kumpulan Newsgroup.



Gambar 2.4 Service tool pada Administrative Tool.

Service yang disediakan oleh IIS dapat dilihat dan disetting dengan menggunakan menu Administrative Tool, Service, yang ditunjukkan pada Gambar 2.4. IIS dapat disetting secara otomatis atau manual melalui menu Service tersebut.

2.6 Ethereal

Ethereal adalah salah satu *network packet analyzer* yang merupakan software freeware. Suatu *Network Packet Analyzer* akan melakukan *capture* terhadap frame yang ada pada jaringan dan mencoba untuk menampilkan informasi frame itu sedetail mungkin.

Beberapa keuntungan menggunakan ethereal (Sharpe, 2004), selain free adalah :

- Berjalan pada platform Windows maupun UNIX.
- Dapat melakukan *capture* secara langsung dari *Network Interface Card* (NIC).

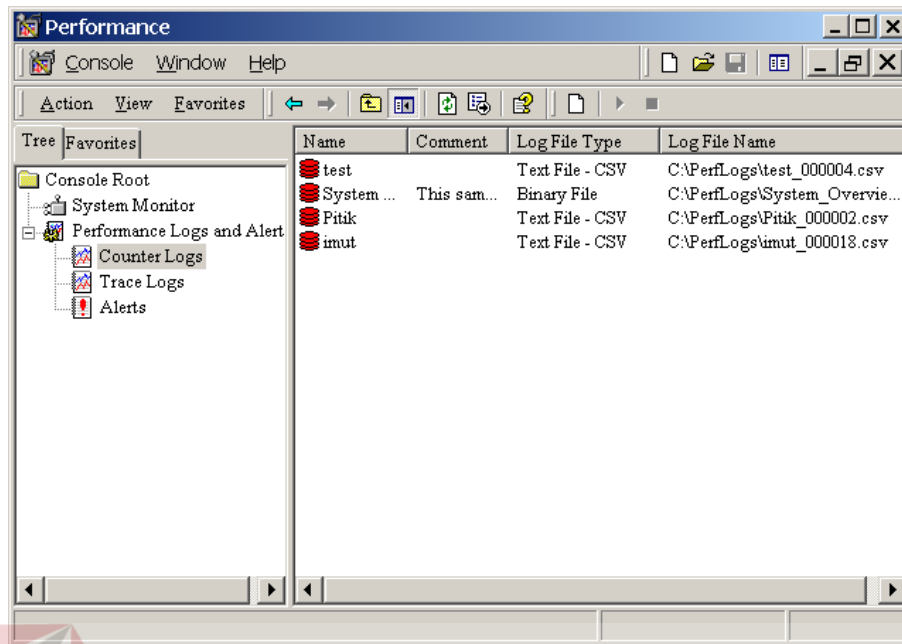
- c. Dapat membuka kembali dan menyimpan hasil dari packet capture.
- d. Dapat diimpor dan diekspor dengan format program packet capture yang lain.
- e. Dapat melakukan *filter* dengan banyak kriteria.
- f. Dapat melakukan pencarian dengan banyak kriteria.
- g. Dapat melakukan banyak analisa statistik.

Untuk melakukan install ethereal, diperlukan Windows Packet Capture (WinpCap) yang merupakan tool standar untuk pembacaan packet pada jaringan komputer. WinpCap ini dapat diperoleh di <http://netgroup-serv.polito.it/winpcap/install/> secara cuma-cuma. Winpcap ini mempunyai kesamaan format hasil packet capture dengan tcpdump. Karena menggunakan format standar tcpdump, maka file hasil log Ethereal dapat dipindahkan ke format lain yang mendukung tcpdump (Sharpe, 2004).

2.7 Microsoft Management Console

Microsoft Management Console (MMC) merupakan tool yang disediakan oleh Microsoft pada OS Windows NT, seperti Windows 2000 dan Windows XP. MMC adalah service standar untuk melakukan manajemen seluruh *resource* dan aplikasi yang berjalan di atas Sistem Operasi (Microsoft, 1999).

MMC menyediakan interface yang sangat mudah untuk dioperasikan oleh Administrator, yang dikenal dengan istilah snap-ins. Dengan snap-ins tersebut, Administrator dapat memanajemen tool dari MMC dengan banyak variasi sesuai dengan kebutuhan (Microsoft, 1999).



Gambar 2.5 Menu Tampilan MMC

Menu MMC ini berada pada Control Panel, Administrative Tools, Performance, seperti yang ditunjukkan pada Gambar 2.5. Untuk menghasilkan file log pada saat pengamatan, setting dapat dilakukan secara manual, dan dapat diatur sesuai keinginan melalui Counters Log. Pada MMC dapat dilakukan setting penulisan ke file log dengan interval waktu tertentu sesuai kebutuhan. Hasil file log dapat disimpan dengan format .csv, .txt, .tsv dan binary file.