RANCANG BANGUN APLIKASI MONITORING PAKET DATA PADA JARINGAN

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana Teknik Informatika



disusun oleh:

Nama : Pradana Wirabadi Rochman

No. Mahasiswa : 02 523 026

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

YOGYAKARTA

2007

LEMBAR PENGESAHAN PEMBIMBING

RANCANG BANGUN APLIKASI MONITORING PAKET DATA PADA JARINGAN

Tugas Akhir

Oleh :

Nama : **Pradana Wirabadi Rochman**

No. Mahasiswa : **02 523 026**

Yogyakarta, 29 Maret 2007

Pembimbing



(Taufiq Hidayat, ST. MCS)

LEMBAR PENGESAHAN PENGUJI

RANCANG BANGUN APLIKASI MONITORING PAKET DATA PADA JARINGAN

TUGAS AKHIR

Oleh:

Nama : Pradana Wirabadi Rochman

NIM : 02 523 026

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia
Yogyakarta, 29 Maret 2007

Tim Penguji

Taufiq Hidayat, ST. MCS
Ketua

Sri Kusuma Dewi S.SI MT
Anggota I

Syarif Hidayat S. Kom
Anggota II

Mengetahui,
Ketua Jurusan Teknik Informatika
Universitas Islam Indonesia

Yudi Prayudi, SSi.,M.Kom

PERSEMBAHAN

Rasa puji syukur kupanjatkan kepada Allah SWT atas karunia dan rahmat-Nya sehingga bisa menyelesaikan tugas akhir ini dengan baik dan semoga akan dapat bermanfaat dikemudian hari atau bagi orang lain.

Ucapan terima kasih juga tak lupa saya ucapkan pada semua pihak yang telah membantu selesainya TA ini....

Buat Papa Mama...Terima kasih atas kasih sayang dan doanya
maafkan kalau anakmu ini selalu menyusahkan sepanjang hidupnya
Semoga pahala yang berlipat yang mereka dapatkan karena ketidakmungkinanku untuk membalasnya untuk selamanya....

Buat Adikku Sari,Tita, dan Amik...Terima kasih atas semua canda tawa dan tangis nya...

Buat Ai'..Terima kasih atas paksaannya untuk segera menyelesaikan kuliahku
Thank's for all and just can say that I really love you...

Buat Gultom,Ambon,Mas Ade,Malun,Hanif dan Yudhi...Terima kasih atas tenaga dan pikirannya
Tanpa kalian entah apa jadinya Power Computer...

Buat anak-anak corps 69.. Terima kasih atas pengalaman barunya
Tanpa kalian Power Computer tidak akan menjadi seperti sekarang...

Buat semua teman-teman yang tak bisa kusebutkan satu-satu...Terima kasih atas semua dukungan kalian

MOTTO

"... Allah akan meninggikan orang beriman diantaramu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat... ..."

(QS.Al-Mujaadilah ayat 11)

" Sesungguhnya sesudah kesulitan itu ada kemudahan ; Maka apabila kamu telah selesai (dari suatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain ".

(Q.S. Asy Syarh ayat 6 dan 7)

" Jadilah sabar dan sholat sebagai penolongmu, sesungguhnya Allah beserta orang-orang yang sabar ".

(Q.S. Al Baqarah ayat 153)

KATA PENGANTAR

بِسْمِ اللهِ الرَّحْمٰنِ الرَّحِيْمِ

Assalamu'alaikum Wr. Wb

Alhamdulillah, segala puji syukur hanyalah kepada Allah SWT dan semoga sholawat serta salam dilimpahkan kepada junjungan kita Nabi Muhammad SAW, keluarga, sahabat, dan pengikut-pengikut beliau (amin). Sehingga penulisan laporan tugas akhir yang berjudul: **RANCANG BANGUN APLIKASI MONITORING PAKET DATA PADA JARINGAN** dapat penulis selesaikan dengan baik.

Laporan tugas akhir ini disusun untuk melengkapi salah satu syarat guna memperoleh gelar Sarjana Teknik Informatika pada Universitas Islam Indonesia dan atas apa yang telah diajarkan selama perkuliahan baik teori maupun praktek, disamping laporan itu sendiri yang merupakan rangkaian kegiatan yang harus dilakukan setelah tugas akhir ini selesai.

Untuk itu saya menyampaikan ucapan terimakasih dan penghargaan yang setinggi-tingginya kepada:

1. Ayah, Ibu dan keluarga tercinta, atas kasih sayang, do'a dan dukungannya.
2. Bapak Taufiq Hidayat, ST. MCS. selaku dosen pembimbing. Terimakasih atas segala bantuan, dukungan, semangat, dan pengetahuannya, serta kemudahan yang telah diberikan.
3. Bapak Fathul Wahid, ST. M.Sc. selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.

4. Bapak Yudi Prayudi, S.Si.,M.kom., selaku Ketua Jurusan Teknik Informatika.
5. Teman-teman yang telah membantu memberikan ide, kritik dan masukan dalam pembuatan tugas akhir ini.
6. Teman-teman Informatika 2002 dan seluruh pihak yang tidak dapat saya sebutkan satu persatu.

Semoga apa yang telah mereka berikan dengan keikhlasan, mendapat pahala yang setimpal dari Allah SWT. Penulis menyadari dalam penulisan laporan tugas akhir ini masih jauh dari sempurna, karena keterbatasan kemampuan dan pengalaman. Penulis mengharapkan saran dan kritik yang bersifat membangun untuk memperbaiki tugas akhir ini semoga dapat bermanfaat bagi penulis khususnya dan pembaca pada umumnya.

Wassalamu'alaikum Wr. Wb

Yogyakarta, 29 Maret 2007

Pradana Wirabadi Rochman

ABSTRAKSI

Penggunaan jaringan lokal atau yang lebih dikenal dengan istilah LAN (*Local Area Network*) menjadi sangat diperlukan dalam sebuah ruang lingkup kerja. Kestabilan sistem sangat tergantung dari arus paket data yang masuk dan keluar, maka diperlukan aplikasi untuk mengawasi paket data yang keluar dan masuk pada sebuah komputer supaya penggunaan koneksi internet dapat dipantau dari penggunaan yang menyimpang dari kebutuhan.
Penelitian ini menghasilkan sebuah aplikasi yang membantu pengguna jaringan jaringan untuk melakukan pengawasan seperti jumlah paket data yang masuk, jumlah paket data yang keluar, serta IP Address nya dengan memanfaatkan informasi pada TCP/IP.

Kata kunci: LAN, Jaringan

DAFTAR ISI

HALAMAN JUDUL i
LEMBAR PENGESAHAN PEMBIMBING ii
LEMBAR PENGESAHAN PENGUJI iii
PERSEMBAHAN iv
MOTTO v
KATA PENGANTAR vi
ABSTRAKSI viii
DAFTAR ISI iix
DAFTAR GAMBAR xii
BAB I PENDAHULUAN 1
1.1 Latar belakang 1
1.2 Rumusan Masalah 2
1.3 Batasan masalah 3
1.4 Tujuan penelitian 3
1.5 Manfaat penelitian 3
1.6 Metodologi penelitian 4
1.7 Sistematika penulisan 5
BAB II LANDASAN TEORI 7
2.1 Protokol TCP/IP 7
2.2 Socket 8
2.3 Windows API 9
2.4 Registry 10
2.5 Struktur registry 10
2.7 Tipe data registry 11
2.8 Client – Server 12

BAB III ANALISIS KEBUTUHAN 13
3.1 Metode analisis 13
3.2 Hasil analisis 13
3.2.1 Analisis umum aplikasi 13
3.2.2 Analisis kode program 14
3.2.3 Analisis sistem 16
3.2.4 Data yang diperlukan 17
BAB IV PERANCANGAN PERANGKAT LUNAK 18
4.1 Metode Perancangan 18
4.2 Hasil Perancangan 18
4.2.1 Rancangan Umum Aplikasi 18
4.2.2 Rancangan Antarmuka 27
BAB V IMPLEMENTASI 31
5.1 Batasan implementasi 31
5.1.1 Bahasa pemrograman 31
5.1.2 Lingkungan pengembangan 31
5.1.3 Asumsi-asumsi yang digunakan 32
5.1.4 Batasan pengembangan aplikasi 32
5.2 Implementasi 33
5.2.1 Antarmuka perangkat lunak 33
5.3 Implementasi Prosedural 35
5.3.1 Prosedur Destroy 35
BAB VI ANALISIS KINERJA PERANGKAT LUNAK 36
6.1 Analisis Pencarian IP 40
6.1.1 Pengujian dengan 1 komputer 40
6.1.2 Pengujian dengan 2 komputer 41
6.1.3 Pengujian dengan 5 komputer 42
6.1.3 Pengujian dengan 2 komputer dan 1 koneksi internet 45
6.1.4 Pengujian dengan 2 komputer dan 2 koneksi internet 47
6.1.5 Analisis Aplikasi 49
6.2 Analisis Proses *Monitoring* Data 50

6.2.1 Pengujian Proses *Uploading* 51
6.2.2 Pengujian Proses *Downloading* 52
6.2.3 Analisis Aplikasi 53
BAB VII SIMPULAN DAN SARAN 55
7.1 Simpulan 55
7.2 Saran 55
DAFTAR PUSTAKA 57

DAFTAR GAMBAR

Gambar 2.1 Registry Editor 10
Gambar 4.1 *Use Case* Diagram 19
Gambar 4.2 *Class* Diagram 20
Gambar 4.3 *Sequence Diagram* untuk Pencarian IP 23
Gambar 4.4 *Sequence Diagram* untuk Menghentikan *Monitoring* 23
Gambar 4.5 *Sequence Diagram* untuk Mengaktifkan Lagi Monitor 24
Gambar 4.6 *Sequence Diagram* untuk Me-*Reset Counter* 24
Gambar 4.7 *Sequence Diagram* untuk Meletakkan Aplikasi Selalu Di atas 25
Gambar 4.8 *Sequence Diagram* untuk Memilih IP yang Ditampilkan 25
Gambar 4.9 *Sequence Diagram* untuk Keluar Program 26
Gambar 4.10 Rancangan Tampilan Program Utama 27
Gambar 5.1 Form Utama 34
Gambar 5.2 Form About 35
Gambar 6.1 Aplikasi Hanya Menemukan IP *loopback.* 41
Gambar 6.2 IP Address di set ke 192.168.0.1 41
Gambar 6.3 Aplikasi menemukan 1 Ip dan aktif 42
Gambar 6.4 IP Address komputer pertama 42
Gambar 6.5 IP Address komputer kedua 43
Gambar 6.6 IP Address komputer ketiga 43
Gambar 6.7 IP Address komputer keempat 43
Gambar 6.8 IP Address komputer kelima 44
Gambar 6.9 Aplikasi tetap hanya menemukan 1 Ip dan aktif 44
Gambar 6.10 Koneksi internet XL GPRS 45
Gambar 6.11 Proses mengkoneksikan dengan "XL" 46
Gambar 6.12 Proses koneksi internet sukses 46
Gambar 6.13 Aplikasi menemukan 2 Ip dan aktif 46
Gambar 6.14 Koneksi internet TELKOMNET 47
Gambar 6.15 Proses mengkoneksikan dengan TELKOMNET 48

Gambar 6.16 Proses koneksi internet sukses 48
Gambar 6.17 Aplikasi menemukan 3 Ip dan aktif..... 48
Gambar 6.18 *Monitoring* data 50
Gambar 6.19 Menunjukkan besar file yang akan di upload 51
Gambar 6.20 Proses *Uploading*..... 51
Gambar 6.21 Proses *Uploading* selesai 52
Gambar 6.22 Menunjukkan besar file yang akan di *download*..... 52
Gambar 6.23 Proses *Downloading* selesai..... 52

BAB I

PENDAHULUAN

1.1 Latar belakang

Pada jaman sekarang ini teknologi jaringan dan teknologi internet dalam dunia komputer telah mengalami perkembangan yang sangat pesat dan telah mengubah fungsi dari penggunaan komputer yang semula hanya digunakan sebagai mesin hitung dan pengolah data, menjadi pusat informasi digital dan memungkinkan terbentuknya sebuah sistem jaringan komunikasi data secara masal.

Penggunaan jaringan lokal atau yang lebih dikenal dengan istilah LAN (*Local Area Network*) menjadi sangat diperlukan dalam sebuah ruang lingkup kerja. Misalnya dalam perkantoran. Kebutuhan pertukaran data secara cepat dan simultan dalam jumlah besar adalah salah satu faktor yang menyebabkan sangat pentingnya peran jaringan komputer dalam sebuah ruang lingkup kerja.

Tidak jarang dalam sebuah jaringan komputer, suatu sumber daya dipakai secara bersama-sama untuk menghemat biaya dan sumber daya yang terbatas. Misalnya sumber daya *proxy server* atau *gateway* yang digunakan bersama-sama untuk menghemat *IP Public* agar dapat melakukan *browsing* di internet. Dari contoh tersebut, dapat diambil kesimpulan bahwa setiap kegiatan yang dilakukan oleh setiap komputer di kantor tersebut di dunia internet akan dikenali sebagai komputer *gateway*, bukan komputer yang melakukan koneksi internet tersebut.

Oleh karena sifatnya yang berbagi itulah maka sebuah sumber daya yang dipakai bersama-sama seperti *gateway* tidak boleh dipakai secara sembarangan.

Contoh lainnya adalah seringnya sebuah komputer di sebuah ruang lingkup kerja seperti kantor digunakan oleh karyawan bukan untuk kepentingan kantor, melainkan untuk kepentingan pribadi karyawan tersebut.

Dari dua contoh yang telah dikemukakan diatas, maka diperlukan sebuah aplikasi yang dapat digunakan untuk mengontrol atau mengawasi paket data yang masuk maupun yang keluar dari sebuah komputer ke komputer yang lain maupun sebaliknya di dalam sebuah ruang lingkup jaringan lokal agar tidak terjadi penyalahgunaan komputer dan sumber daya kantor lainnya.

Penelitian ini mencoba mengembangkan solusi yang lebih mudah dan sederhana dalam melakukan pengawasan yang dilakukan seorang *administrator* dalam sebuah LAN perkantoran untuk melakukan pengawasan dan pengontrolan terhadap komputer lain yang terdapat di jaringannya.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dikemukakan sebelumnya, dapat dibuat rumusan masalah sebagai berikut:

Bagaimana merancang dan mengimplementasikan suatu aplikasi perangkat lunak yang dapat memantau dan mengawasi paket data yang masuk maupun yang keluar dari sebuah komputer ke komputer yang lain maupun sebaliknya di dalam sebuah ruang lingkup jaringan lokal.

1.3 Batasan masalah

Untuk menghindari meluasnya ruang lingkup yang dibahas pada penelitian ini, maka diberikan batasan masalah sebagai berikut:

1. Percobaan terhadap aplikasi ini hanya akan dilakukan dalam sebuah jaringan LAN.
2. Aplikasi ini hanya memantau arus paket data.
3. Aplikasi ini dijalankan dalam sistem operasi Microsoft Windows.
4. Antarmuka perangkat lunak akan dikembangkan dalam bahasa Inggris karena alasan banyaknya istilah asing yang bila dialih-bahasakan ke bahasa Indonesia akan menimbulkan kerancuan.
5. Ketidakstabilan dan kegagalan aplikasi yang disebabkan karena cacat ataupun bug yang terdapat pada perangkat keras, *compiler*, dan sistem operasi merupakan kesalahan yang terlepas dari pengembangan aplikasi.

1.4 Tujuan penelitian

Membangun suatu perangkat lunak yang dapat mengawasi paket data yang masuk maupun yang keluar dari sebuah komputer ke komputer yang lain maupun sebaliknya di dalam sebuah ruang lingkup jaringan lokal.

1.5 Manfaat penelitian

Membantu administrator dalam sebuah LAN untuk melakukan administrasi komputer-komputer yang terhubung dalam jaringannya, sehingga meminimalisir terjadinya penyalahgunaan komputer-komputer tersebut maupun penyalahgunaan sumberdaya lain yang dipakai secara bersama-sama.

1.6 Metodologi penelitian

Beberapa metode yang digunakan untuk menyelesaikan penelitian ini adalah sebagai berikut:

1.6.1 Metode pengumpulan data

Metode pengumpulan data yang digunakan adalah dengan studi literatur dari berbagai sumber yang layak digunakan sebagai landasan dengan mempelajari dan mendalami secara mendalam dan seimbang materi-materi yang bersifat konseptual dengan materi-materi yang bersifat terapan.

1.6.2 Pengembangan

Tahap-tahap pengembangan yang dilakukan adalah sebagai berikut:

a. Observasi dan pengumpulan data. Tahap ini diselesaikan dengan mengumpulkan buku-buku dan materi-materi dari internet yang dapat digunakan sebagai acuan dalam melakukan penelitian.
b. Analisis kebutuhan dan perancangan. Pada tahap ini dilakukan proses analisis terhadap berbagai kebutuhan yang mungkin akan diperlukan dan dilanjutkan dengan perancangan perangkat lunak yang akan dikembangkan.
c. Implementasi dan pengujian. Implementasi dilakukan setelah proses perancangan selesai dilakukan dan dianggap layak untuk dilanjutkan. Sedangkan pengujian dilakukan setahap demi setahap pada saat proses implementasi dilakukan agar setiap bagian mengalami pengujian sendiri-sendiri.

d. Analisis kinerja. Dan pada akhirnya pada tahap ini dilakukan pengujian akhir dari perangkat lunak yang telah selesai dikembangkan, untuk dapat dinilai kesesuaian antara perancangan dan hasil akhir yang didapatkan.

1.7 Sistematika penulisan

Penulisan laporan ini disajikan dalam tujuh bab dengan maksud untuk mempermudah pembacaan agar lebih jelas dan akurat.

Bab 1 berisi Pendahuluan yang menjelaskan tentang latar belakang, rumusan, dan batasan masalah yang diangkat menjadi materi penulisan laporan tugas akhir.

Bab 2 berisi Landasan Teori yang berisi tentang dasar-dasar teori yang digunakan untuk menyelesaikan penulisan laporan tugas akhir ini .

Bab 3 berisi Analisis Kebutuhan Perangkat Lunak yang berisi tentang uraian metode analisis kebutuhan perangkat lunak yang digunakan untuk membantu penyelesaian tugas akhir.

Bab 4 berisi Perancangan Perangkat Lunak yang memuat metode perancangan dan hasil perancangan perangkat lunak yang akan dibuat yang meliputi kebutuhan, arsitektur, rancangan antar muka perangkat lunak yang akan dikembangkan, dan lain sebagainya.

Bab 5 berisi Implementasi yang meliputi batasan implementasi dan implementasi perangkat lunak itu sendiri.

Bab 6 berisi Analisis Kinerja Perangkat Lunak yang berisi dokumentasi hasil pengujian perangkat lunak yang dikembangkan yang dibandingkan

kebenaran dan kesesuaiannya dengan teori-teori yang telah dikemukakan sebelumnya.

Bab 7 berisi Simpulan dan Saran yang memuat rangkuman dari hasil penelitian dan saran-saran yang sebaiknya dilakukan.

BAB II

LANDASAN TEORI

2.1 Protokol TCP/IP

Seperti halnya dalam dunia manusia, dalam dunia komputer, komunikasi antar komputer juga memerlukan 'bahasa' sebagai syarat utama terjadinya komunikasi dua arah yang baik. Dalam dunia komputer, 'bahasa' ini disebut sebagai protokol. Komputer-komputer yang akan saling berkomunikasi memerlukan protokol yang sama dalam rangka berkomunikasi dengan baik.

Protokol mengatur bagaimana sebuah komputer dapat berkomunikasi dengan baik dengan komputer lain. Oleh karena itu, komunikasi tidak tergantung dari sistem operasi, melainkan tergantung pada protokol.

Protokol adalah kumpulan aturan yang dipakai oleh client dan server untuk berkomunikasi [MAR02].

TCP/IP (*Transmission Control Protocol/Internet Protocol*) adalah protokol standar yang digunakan di dunia internet. Dalam protokol TCP/IP, terdapat tiga protokol di level yang lebih tinggi lagi, yaitu TCP (*Transmission Control Protocol*), UDP (*User Datagram Protocol*), dan ICMP (*Internet Control Message Protocol*).

TCP adalah protokol dalam dunia jaringan komputer yang digunakan untuk melakukan komunikasi pertukaran data secara *reliable*. Metode komunikasi TCP mengutamakan terbentuknya sebuah ikatan antara dua buah komputer terlebih dahulu, baru setelah itu komunikasi data dapat dilakukan. Hal ini lebih

dikenal dengan istilah *connection oriented.* Sebuah komputer yang hendak mengirim data ke komputer lain melalui protokol TCP harus memastikan terlebih dahulu bahwa komputer tersebut dapat dijangkau dan tidak terdapat masalah dalam koneksi tersebut, baru kemudian data dikirmkan.

Hal ini tidak terdapat dalam protokol UDP, dimana data yang dikirim tidak tergantung dari konektifitas antara kedua komputer. Istilah yang sering digunakan adalah *connectionless.*

Sedangkan ICMP biasanya digunakan untuk melakukan pemeriksaan koneksi antar dua komputer dengan mengirim pesan ke komputer tersebut, dan menunggu balasan dari komputer tersebut. Pesan yang dikirim disebut *echo* dan balasannya disebut *echo-reply.* Contoh dari penggunaan protokol ICMP ini adalah aplikasi ping.

2.2 Socket

Pemrograman socket adalah bagian dari pemrograman tingkat rendah. Socket adalah sebuah *resource* yang disediakan untuk melewatkan pengiriman dan penerimaan data pada komunikasi data. Pada sistem operasi Microsoft Windows, socket tersedia dalam *library* winsock.dll.

Terdapat tiga jenis koneksi socket:

1. Koneksi client

Dimulai oleh *client* dan menghubungkan socket *client* lokal dengan sebuah socket *server remote.* Socket *client* harus menyatakan *server* yang hendak dihubunginya dengan memberikan nama host komputer *server* atau alamat IPnya, dan *port* yang digunakan.

2. Koneksi *listen*

Socket *server* bersifat pasif dan menunggu *client*. Hal ini biasa disebut dengan istilah *listen*. Jika *client* memberikan permintaan, *server* membuat socket baru khusus untuk melayani permintaan tersebut, lalu mendengarkan lagi.

3. Koneksi server

Koneksi diaktifkan oleh *server* pada saat menerima permintaan dari *client*.

2.3 Windows API

Windows API (*Application Programming Interface*) adalah sekumpulan fungsi-fungsi yang membangun sistem operasi Microsoft Windows [NUG02]. Fungsi-fungsi ini tersimpan dalam file-file DLL (*Dynamic Link Library*) yang berlokasi di direktori sistem Windows, yang biasanya terdapat di alamat C:\Windows. Dengan memanfaatkan fungsi-fungsi tersebut, sebuah aplikasi dapat mengeksploitasi kemampuan sistem operasi Windows.

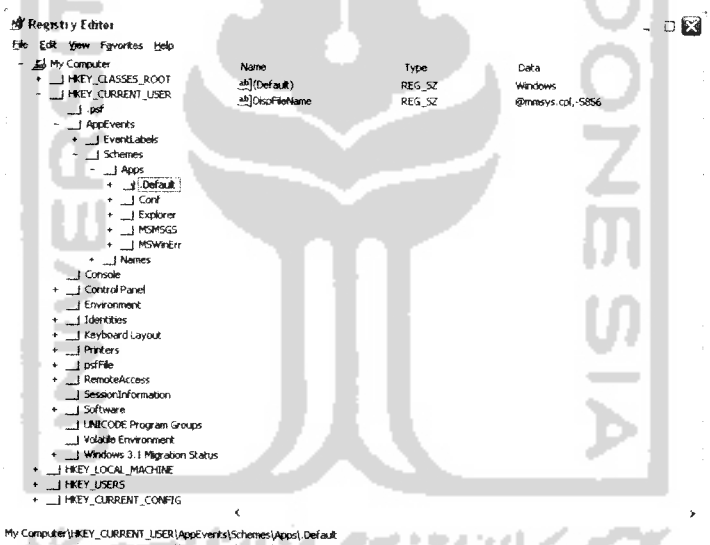
Beberapa file library yang sering digunakan antara lain:

1. user32.dll, berisi fungsi-fungsi untuk manipulasi *user interface* dari Windows.
2. gdi32.dll, berisi fungsi-fungsi GDI (*Graphical Device Interface*).
3. kernel32.dll, berisi fungsi-fungsi untuk operasi level rendah.
4. mmsystem.dll, berisi fungsi-fungsi multimedia.
5. netapi32.dll, berisi fungsi-fungsi untuk jaringan.
6. comdlg32.dll, berisi fungsi-fungsi pemanggilan kotak dialog standar pada Windows.

2.4 Registry

Registry merupakan sebuah *database* sistem, dimana aplikasi dan komponen sistem dapat menyimpan dan memperoleh kembali data konfigurasi. Registry menyimpan data dalam bentuk file biner. Untuk memanipulasi data registry, harus melalui fungsi-fungsi registry. Atau menggunakan aplikasi Registry Editor yang terdapat di direktory sistem Windows.

Registry menyimpan data dalam hirarki pohon yang terstruktur. Tiap *node* disebut *key*, dan tiap *key* memiliki *sub-key*. Data didalamnya disebut *value*.



Gambar 2.1 Registry Editor

2.5 Struktur registry

RootKey (*key* level tertinggi) dikelompokkan dalam beberapa kategori:

1. HKEY_CLASSES_ROOT, berisi dokumen tipe (kelas) untuk tiap dokumen beserta properti yang sesuai.

2. HKEY_CURRENT_CONFIG, berisi informasi profil *hardware* komputer saat ini.
3. HKEY_CURRENT_USER, berisi definisi-definisi *setting user* saat ini, seperti warna, *font* yang digunakan, dan lain sebagainya.
4. HKEY_DYN_DATA, terdapat pada Windows 95/98/Me, digunakan untuk mendapatkan data *performance*.
5. HKEY_LOCAL_MACHINE, berisi definisi fisik komputer, misalnya tipe *bus* data, sistem memori, *hardware* dan *software* yang terinstal.
6. HKEY_PERFORMANCE_DATA, terdapat pada Windows NT/2000/XP, digunakan untuk mendapatkan data *performance*.
7. HKEY_USERS, berisi standar data konfigurasi untuk *user* baru pada komputer lokal.

2.7 Tipe data registry

Registry mampu menyimpan data dalam tipe-tipe sebagai berikut:

1. REG_BINARY, tipe ini menyimpan nilai sebagai tipe data biner. Dapat ditampilkan dalam bilangan heksadesimal.
2. REG_DWORD, data direpresentasikan dalam empat *bit* bilangan heksadesimal.
3. REG_EXPAND_SZ, *Null terminated string* yang berisi referensi ke variabel *environment*, misalnya "%PATH%".
4. REG_MULTI_SZ, merupakan multiple *string* yang digunakan untuk merepresentasikan nilai yang berisi daftar nilai. Tiap *entry* dipisahkan oleh karakter *null*, dan keseluruhan data diakhiri dengan dua karakter *null*.

5. REG_SZ, merupakan tipe standar *string*, bisanya digunakan untuk menyimpan data teks yang dapat dibaca.

2.8 Client – Server

Komputer *server* adalah komputer yang menjalankan program yang menunggu permintaan dari komputer lain yang disebut sebagai *client*. Setelah mendapat permintaan dari *client*, *server* akan melakukan proses yang diminta *client*, dan memberikan hasil proses tersebut ke *client*.

Program yang digunakan untuk menunggu permintaan ini disebut sebagai daemon, dan biasanya berjalan di *background*. Salah satu contoh *server* adalah *web server* yang menggunakan daemon httpd.

Komputer *client* adalah komputer yang memberikan permintaan pada *server* untuk melakukan proses yang dimintanya, dan menunggu hasil prosesnya dari *server*.

BAB III

ANALISIS KEBUTUHAN

3.1 Metode analisis

Metode analisis yg digunakan adalah metode analisis berorientasi objek dengan menggunakan alat dan teknik yang telah disesuaikan dengan kebutuhan sistem. Arsitektur sistem berusaha didefinisikan dengan baik agar pengembangan perangkat lunak dapat dijalankan dengan terstruktur dan jelas meskipun mengalami peningkatan kompleksitas yang signifikan.

Pada tahap analisis ini digunakan suatu alat untuk melakukan pemodelan agar pengembangan perangkat lunak dapat memenuhi semua kebutuhan pengguna dengan lengkap dan tepat. Pemodelan yang dilakukan menggunakan notasi-notasi DFD (*Data Flow Diagram*) yang telah menjadi standar dalam dunia industri perangkat lunak untuk melakukan perancangan, visualisasi, dan pendokumentasian agar pengembangan perangkat lunak dapat lebih terstruktur.

3.2 Hasil analisis

3.2.1 Analisis umum aplikasi

Perangkat lunak yang akan dibangun nantinya diusahakan untuk menangani fitur-fitur standar dan lazim pada perangkat lunak pemantau lalu lintas data pada sebuah jaringan. Beberapa hal yang harus dapat ditangani adalah sebagai berikut:

1. Melakukan pencarian MAC Address yang digunakan.
2. Memberikan informasi kecepatan jaringan yang sedang berlangsung.
3. Memberikan informasi durasi waktu aplikasi dijalankan.
4. Memberikan informasi jumlah paket data yang masuk.
5. Memberikan informasi jumlah paket data yang keluar.
6. Memberikan informasi tanggal,bulan,tahun dan jam ketika aplikasi dijalankan.
7. Melakukan *Reset* ulang pada *field- field* yang ada.
8. Melakukan penghentian penghitungan jumlah paket data.

Aplikasi yang akan dibangun akan dijalankan di tiap-tiap komputer sehingga tiap komputer akan dapat memantau perputaran data yang terjadi. Pada aplikasi ini setiap komputer akan dipantau dan dikontrol sehingga semua komputer tidak akan luput dari pengawasan.

3.2.2 Analisis kode program

Pembuatan kode-kode program akan berkaitan dengan beberapa hal utama yang berhubungan dengan protokol TCP/IP, pemrograman socket, dan langkah-langkah abstraksinya dalam paradigma pemrograman berorientasi objek. Penulisan kode program pada aplikasi ini akan dilakukan dengan menggunakan bahasa pemrograman yang telah kita kenal yaitu Delphi dengan *compiler* Borland Delphi 7.

3.2.2.1 Pemrograman socket

Dalam TCP/IP, untuk melakukan segala koneksi dan pertukaran data dilakukan dengan mengimplementasikan pemrograman socket ke dalam aplikasi

yang dikembangkan. Socket merupakan implementasi dasar yang harus dilakukan untuk membentuk lapisan protokol diatasnya.

Pada dasarnya sebuah socket hanya akan melakukan dua hal yang utama yaitu mengirim dan menerima data. Sebuah socket melakukan koneksi TCP/IP berdasarkan alamat komputer (*IP address* atau *hostname*) dan nomor *port* yang dituju. Nomor *port* menunjukkan layanan *server* apa yang dituju. Misalnya *port* 80 untuk web server, 20 dan 21 untuk ftp server, dan lain sebagainya.

3.2.2.2 Windows API

Fungsi-fungis Windows API yang digunakan dalam aplikasi ini antara lain fungsi-fungsi standar dan dasar seperti fungsi manipulasi *file* dan *direktory*, pencarian alamat IP, dan lain sebagainya.

Fungsi-fungsi tersebut sudah disertakan dalam *compiler* Borland Delphi 7 yang digunakan untuk pengembangan aplikasi, dan disimpan dalam *unit* ShellAPI yang disertakan dalam aplikasi yang akan dibangun.

3.2.2.3 Struktur aplikasi

Untuk mempermudah pembuatan aplikasi, maka diperlukan mekanisme pemecahan permasalahan menjadi bagian-bagian yang lebih kecil sehingga mempermudah dalam pengembangan dan pengecekan kesalahan.

Pembuatan aplikasi ini dibagi dalam beberapa bagian sesuai tugas dan peranan komponen-komponen tersebut.

1. Komponen untuk menangani pencarian *port* yang digunakan oleh tiap komputer.
2. Komponen untuk menangani koneksi utama .
3. Komponen untuk menerima dan mengirim *file*.
4. Komponen untuk menerima hasil.
5. Komponen untuk menangani tampilan.

3.2.3 Analisis sistem

3.2.3.1 Masukan

Masukan dari aplikasi ini adalah alamat IP atau *hostname* komputer yang akan dikontrol dan dipantau aktifitasnya. Sedangkan untuk proses-proses setelah koneksi terbentuk, masukan yang dibutuhkan adalah sebagai berikut:

1. Masukan untuk membaca kecepatan koneksi.
2. Masukan untuk melakukan penghitungan durasi waktu ketika aplikasi dijalankan.
3. Masukan untuk menampilkan jumlah paket data yang masuk maupun yang keluar.

3.2.3.2 Proses

Proses yang dilakukan adalah pengidentifikasian komputer yang meliputi kecepatan jaringan, durasi waktu, jumlah paket data yang masuk maupun yang keluar.

3.2.3.3 Keluaran

Keluaran dari aplikasi ini adalah:

1. Informasi, yaitu berisi informasi kecepatan jaringan, durasi waktu ketika aplikasi dijalankan, jumlah paket data yang masuk dan keluar serta informasi tanggal,bulan,tahun dan jam ketika aplikasi dijalankan.
2. Kontrol, yaitu melakukan reset ulang pada field-field yang ada dan melakukan penghitungan paket data.

3.2.4 Data yang diperlukan

Data yang diperlukan dalam penelitian antara lain:

1. Fungsi-fungsi Windows API.
2. Kegunaan dari beberapa *key* di Registry Windows.
3. Pemahaman tentang pemrograman socket.
4. Pemahaman tentang protokol TCP/IP.

BAB IV

PERANCANGAN PERANGKAT LUNAK

4.1 Metode Perancangan

Perancangan sistem merupakan salah satu tahap yang diperlukan untuk membangun sebuah sistem. Perancangan tersebut diantaranya perancangan proses sistem. Penggunaan UML sebagai metode perancangan perangkat lunak akan membantu ketika sistem nanti dibangun menggunakan program Borland Delphi yang mempunyai metode *object oriented* dalam pemrogramannya. UML dipilih karena akan lebih jelas dalam dukungannya membangun sebuah perancangan sistem yang berorientasi obyek.

4.2 Hasil Perancangan

4.2.1 Rancangan Umum Aplikasi

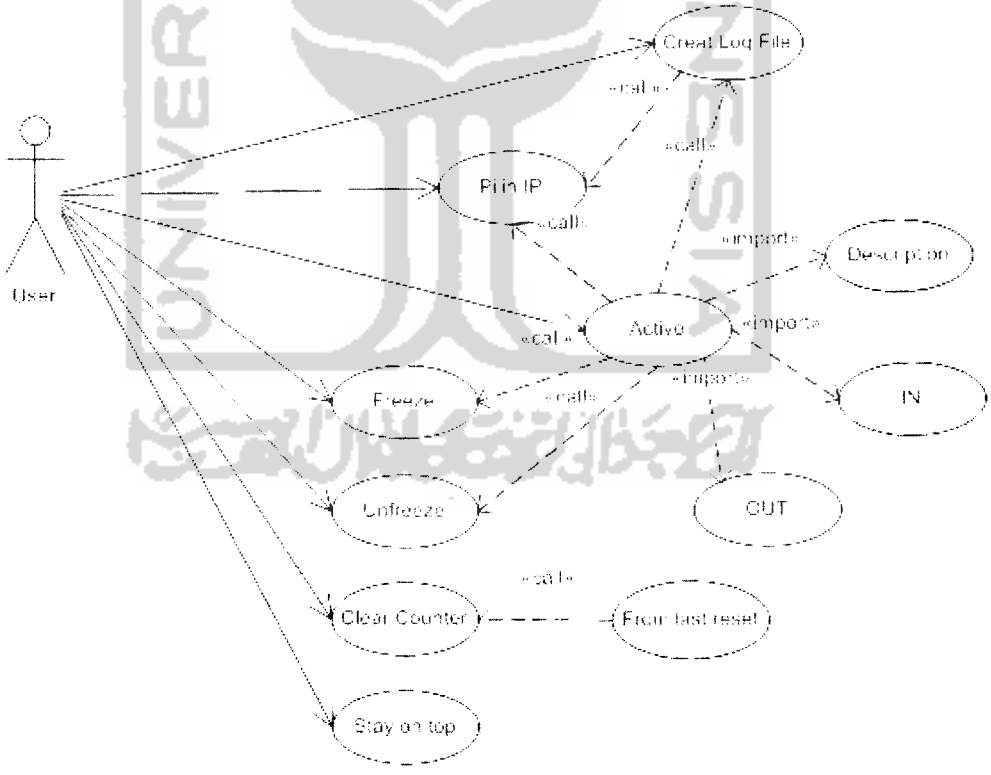
Pembuatan diagram merupakan hal penting dalam UML sehingga sesuai dengan analisis dan pengembangan sistem. Dalam tahap ini akan dibatasi pada tiga diagram saja, yaitu *use case diagram*, *class diagram*, dan *squence diagram*. UML memiliki beberapa konsep dasar yang diabstraksikan dalam bentuk *structural classification*, *dynamic behavior*, dan *model management*.

4.2.1.1 *Use Case* Diagram

Use case merupakan sebuah pekerjaan tertentu, misalnya melihat properti dari IP, menghentikan sistem dalam memonitor, dan sebagainya. *Use case*

diagram berisi gambaran fungsionalitas yang diharapkan dari sebuah sistem dengan fokus penekanan pada apa yang dilakukan oleh sistem, bukan bagaimana sistem melakukan sesuatu. Sebuah *use case* mempresentasikan sebuah interaksi antara aktor dan sistem.

Terdapat aktor *user* untuk aplikasi yang akan dibuat dan menjalankan program *network data monitoring* ini. Kemudian *user* tersebut memiliki beberapa *use case* yang di definisikan pada gambar 4.1. Secara garis besar user pertama kali menjalankan aplikasi yang kemudian terdapat *timer* yang mencari IP kemudian pengendalian dapat dilakukan *user* sesuai dengan IP yang dipilih.

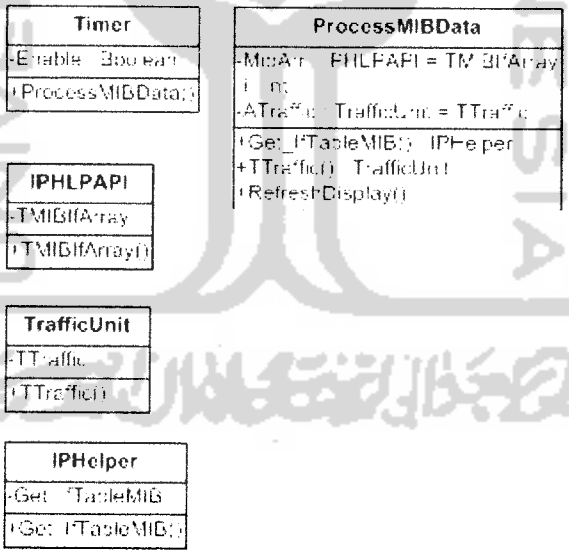


Gambar 4.1 *Use Case* Diagram

4.2.1.2 *Class* Diagram

Penggambaran struktur dan deskripsi *class*, *package*, dan obyek beserta hubungannya satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain bisa disebut dengan *Class diagram*. *Class* adalah sebuah spesifikasi yang jika diinstanisasi akan menghasilkan sebauh obyek dan merupakan inti dari pengembangan dan desain berorientasi obyek. *Class* menggambarkan keadaan atribut atau properti suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut.

Gambar 4.2 berikut ini menunjukkan beberapa *class diagram* untuk aplikasi yang akan dibuat.



Gambar 4.2 *Class* Diagram

Timer class berjalan ketika aplikasi berhasil menginisialisasi dari kebutuhan sistem. Ketika sistem telah siap untuk memonitor lalu lintas data pada suatu IP, *timer* akan berjalan dengan *interval* 1000 atau setiap detiknya akan mengeksekusi

syntaks di dalam timer tersebut. Di dalam obyek *timer* ini terdapat properti *enable* yang bertipe *boolean*. Properti ini berfungsi untuk mengetahui atau mengatur apakah *timer* sedang aktif atau tidak. Secara *default*, timer ini akan diatur *true* atau sedang aktif ketika aplikasi menjalankan fungsi *create form* atau aplikasi dibentuk. Dan ketika *timer* berjalan, di dalamnya terdapat fungsi *true* sebelum menjalankan obyek pencarian IP serta data-datanya dan bernilai *false* setelah proses pancarian IP dan adta-datanya. Sehingga setiap detiknya, data yang ditampilkan akan selalu yang terbaru hasil dari pencarian data-data tersebut. Selain properti *enable*, terdapat *procedure ProcessMIBData* yang berfungsi untuk mencari IP dan data-datanya. Proses ini akan terus berjalan selama *timer* memanggil atau bernilai *true*.

ProcessMIBData Class berjalan ketika dipanggil oleh *procedure timer*. Properti dan variabel yang terdapat pada *class* ini yaitu *MibArr*, *i* dan *Atraffic*. *MibArr* mempunyai tipe data *TMIBIfArray* yang terdapat pada *class IpHlpAPI*. Variabel ini berfungsi untuk menyimpan data-data dari hasil pencarian IP dan data-data yang masuk maupun keluar melalui IP yang sedang aktif. Variabel *i* bertipe *integer* yang berfungsi untuk menyimpan suatu angka *integer* yang digunakan untuk melakukan perulangan dalam pencarian banyaknya IP yang ada dan perulangan dalam pencarian data. *Atraffic* mempunyai tipe data *TTraffic* yang terdapat pada *class TrafficUnit*. Variabel ini berfungsi untuk memperoleh dan menyimpan data-data yang lebih spesifik seperti data masuk, keluar, rata-rata dan lain-lainnya. Variabel ini sekaligus juga meng-*handle* status dari *buttons* pada aplikasi apakah sedang aktif atau tidak.

IPHLPAPI class berjalan ketika dipanggil oleh *procedure ProcessMIBData*. Variabel atau *procedure* yang terdapat pada *class* ini diantaranya yaitu *TMIBIfArray*. Dalam *procedure* ini berfungsi untuk berhubungan keluar, dalam hal ini berhubungan dengan kartu jaringan, yang juga sering disebut dengan *Application Programming Interface*.

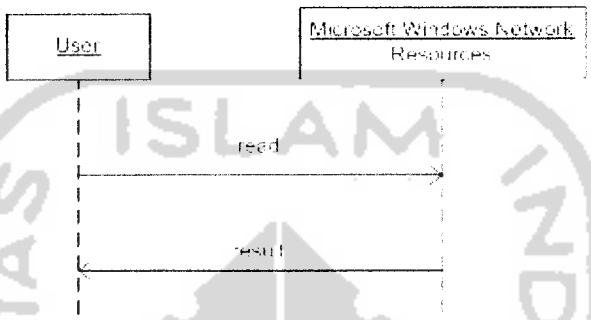
TrafficUnit class berjalan ketika dipanggil oleh *procedure ProcessMIBData*. Variabel ini berfungsi untuk menyimpan data-data hasil *monitoring* dari suatu IP yang aktif. Mulai dari data masuk, keluar, rata-rata dan total data disimpan dalam *procedure-procedure* di dalamnya. Selain itu, ketika *procedure* ini berjalan akan mengetahui jika ada *event* pada *button*. Misalnya *event onClick*, maka akan ada *procedure* yang menangani *event* tersebut. Misalnya ketika ditekan tombol *Freeze*, maka data-data dalam variabel tersebut juga tidak akan diperbaharui.

IPHelper class berjalan ketika dipanggil oleh *procedure ProcessMIBData*. Yaitu melakukan proses pencarian data-data yang masuk maupun keluar melalui IP yang aktif dan menyimpannya dalam variabel *MibArr*. *Procedure* untuk melakukan proses ini diantaranya yaitu *Get_IfTableMIB*.

4.2.1.3 *Squence* Diagram

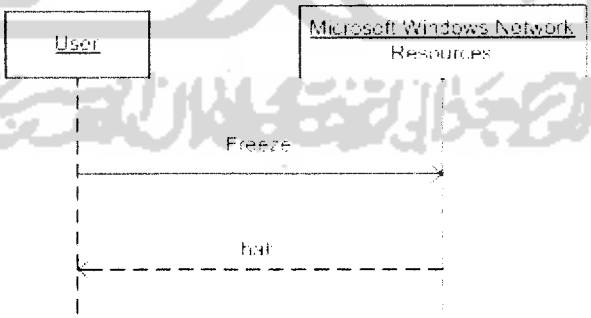
Squence diagram terdiri dari dimensi vertikal atau waktu dan dimensi horizontal atau obyek-obyek yang terkait. *Squence diagram* menggambarkan interaksi antar obyek di dalam dan di sekitar sistem berupa pesan yang digambarkan terhadap waktu. Beberapa *squence diagram* untuk aplikasi yang akan dibuat ditunjukkan dalam Gambar 4.3 yang menunjukkan *sequence diagram* untuk proses pencarian IP pada komputer lokal. Proses ini berjalan tidak hanya

ketika aplikasi pertama kali dijalakan, akan tetapi terus berjalan sampai aplikasi dihentikan.



Gambar 4.3 *Sequence Diagram* untuk Pencarian IP

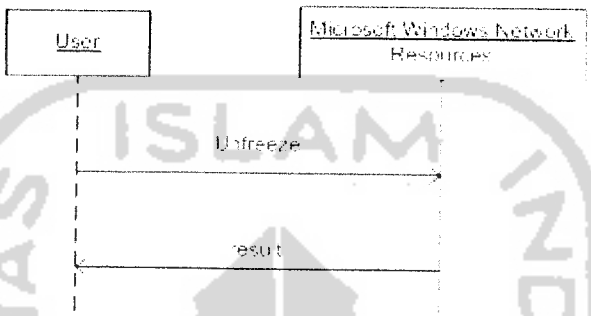
Gambar 4.4 adalah *sequence diagram* untuk proses penghentian *monitoring* lalu lintas data melalui IP yang aktif. Proses ini hanya menghentikan proses secara sementara selama proses belum dikembalikan menjadi aktif memonitor lalu lintas data dengan cara menekan tombol Unfreeze.



Gambar 4.4 *Sequence Diagram* untuk Menghentikan *Monitoring*

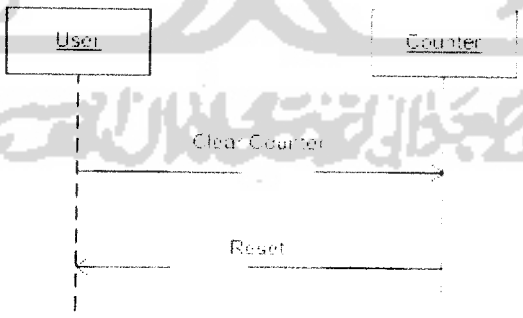
Gambar 4.5 adalah *sequence diagram* untuk proses pengaktifan kembali aplikasi dalam memonitor lalu lintas data yang melalui IP aktif. Proses ini bisa dilakukan ketika *user* menghentikan sementara dengan menekan tombol *Freeze*.

Proses *Freeze* maupun *Unfreeze* mempengaruhi dari besarnya data yang ditampilkan aplikasi dan tidak mempengaruhi proses yang lainnya seperti memonitor IP atau menjalankan *Counter*.



Gambar 4.5 *Sequence Diagram* untuk Mengaktifkan Lagi Monitor

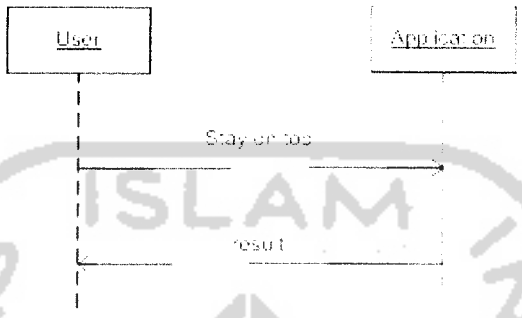
Gambar 4.6 adalah *sequence diagram* untuk proses menghentikan *Counter* atau lebih tepatnya mengembalikan ke awal dari posisi *Counter*. Sehingga penghitungan akan dimulai lagi dari awal ketika tombol *Clear Counter* ditekan. Proses ini tidak mempengaruhi proses yang lain seperti *monitoring* IP.



Gambar 4.6 *Sequence Diagram* untuk Me-*Reset Counter*

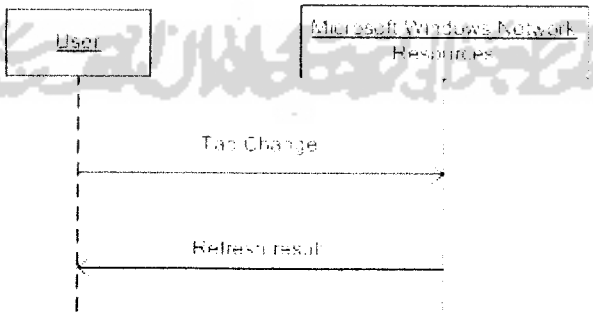
Gambar 4.7 adalah *sequence diagram* untuk proses meletakan aplikasi selalu di atas jika ada lebih dari satu aplikasi yang aktif maupun tidak. Sehingga

walaupun *window* aplikasi sedang tidak aktif, tampilan aplikasi akan selalu kelihatan selama aplikasi tidak di-*minimize*.



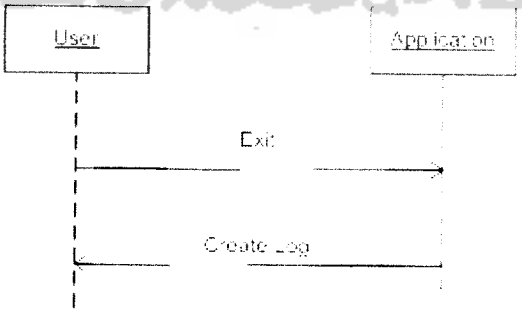
Gambar 4.7 *Sequence Diagram* untuk Meletakkan Aplikasi Selalu Di atas

Gambar 4.8 adalah *sequence diagram* untuk proses memilih IP yang ditampilkan. Proses ini akan bisa dilakukan ketika IP yang berhasil ditemukan pada komputer lebih dari satu. Jika proses ini dijalankan, maka tampilan akan menyesuaikan dari IP yang sedang aktif tersebut. Selain data yang dimonitor, tampilan deskripsi dari perangkat kartu jaringan juga akan disesuaikan dengan IP yang sedang aktif.



Gambar 4.8 *Sequence Diagram* untuk Memilih IP yang Ditampilkan

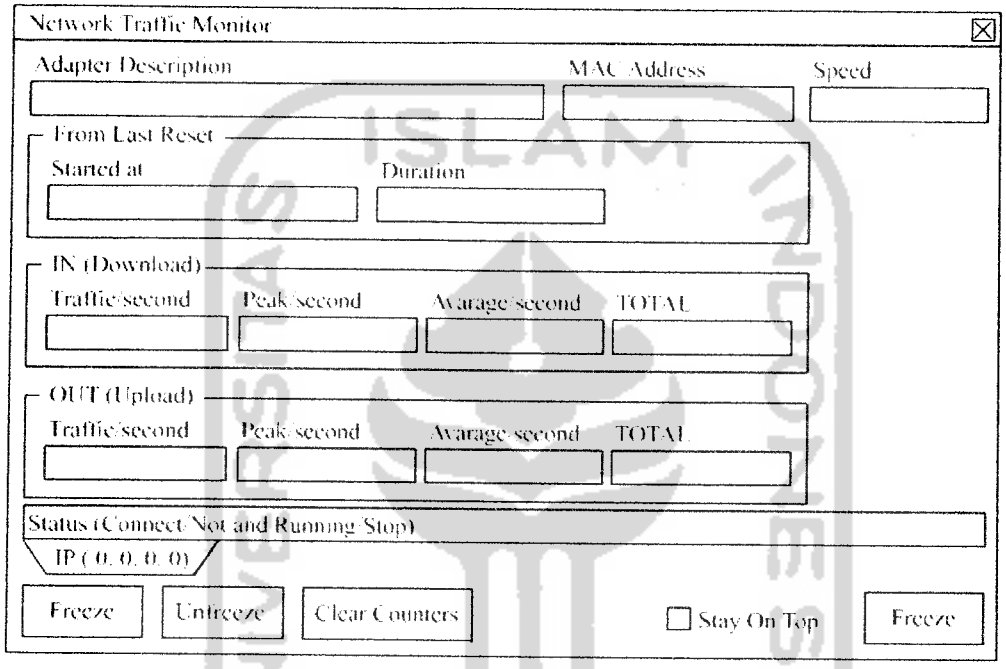
Gambar 4.9 adalah *sequence diagram* untuk proses keluar dari aplikasi. Ketika aplikasi dihentikan, maka proses akan menghentikan semua proses seperti pencarian IP, *monitoring* lalu lintas data dan penghitungan *counter*. Akan tetapi ketika semua telah dihentikan, masih terdapat satu proses lagi yaitu pembuatan *file log* yang berisi data-data dari aplikasi mulai dari aplikasi pertama kali dijalankan. Ketika aplikasi dijalankan pertama kali, *file log* akan mencatat tanggal dan jam pertama kali aplikasi dijalankan yang diperoleh dari waktu sistem. Kemudian deskripsi dari IP seperti *Adapter Description*, *MAC Address* dan *Speed* dari kartu jaringan dari IP tersebut juga akan dicatat. Setelah proses *monitoring* berjalan, data tersebut tidak lagi dilakukan kecuali terdapat IP baru yang aktif. Selama proses *monitoring* berjalan, semua data dari proses *menitoring* itu juga akan dicatat tiap detiknya, baik itu data yang masuk maupun data yang keluar melalui IP yang aktif tersebut. Selain awal dari isi *file log*, akhir *file log* juga akan dicatat tanggal dan jam aplikasi dihentikan, sehingga paling akhir dari *file log* ini dicatat durasi sistem memonitor data. Sedangkan nama *file* diambil dari tanggal dan jam aplikasi dihentikan dan ber-ekstensi *log*.



Gambar 4.9 *Sequence Diagram* untuk Keluar Program

4.2.2 Rancangan Antarmuka

Rancangan tampilan program utama akan tampak seperti pada Gambar 4.10 berikut:



Gambar 4.10 Rancangan Tampilan Program Utama

Komponen yang ada adalah sebagai berikut :

1. *NIC Description*

   Menampilkan Deskripsi dari *Network Interface Card* yang sedang aktif, yaitu:

   a. *Adapter Description*: Menampilkan nama dari kartu jaringan yang sedang aktif, misalnya *Realtek RTL8139 Family PCI Fast Ethernet NIC* dan *MS TCP*

   b. *Loopback interface* untuk nama dari IP *Loopback* atau untuk *localhost*.

c. *MAC Address*: Alamat fisik dari kartu jaringan. Semua kartu memiliki serial yang tidak sama antara satu dengan yang lain. Contoh dari alamat ini adalah 00-E0-06-09-55-66. Jika itu merupakan loopback, maka alamat fisiknya tidak ada atau 00-00-00-00-00-00.

d. *Speed* : Kecepatan maksimal dari kartu jaringan yang sedang aktif tersebut. Untuk loopback, mempunyai kecepatan default 10,00 Mbps yang berarti bahwa data dapat melalui kartu jaringan ini sebesar 10 Mega Byte per detiknya. Sedangkan untuk kartu jaringan yang aktif memiliki kecepatan bervariasi antara 10 sampai 100 Mbps.

2. *From Last Reset*

Memberikan keterangan mengenai tanggal dan jam mulainya program dijalankan serta memberikan keterangan lamanya durasi waktu untuk memonitor lalu lintas jaringan.

a. *Started at* : Menampilkan Tanggal dan Jam ketika program dijalankan. Contohnya yaitu 17/02/2007 12:00:00.

b. *Duration* : Lamanya program dalam memonitor lalu lintas jaringan akan ditampilkan di sini. Contoh tampilannya yaitu 00:02:01 yang berarti bahwa program telah memonitor lalu lintas jaringan selama dua menit satu detik. Durasi ini dapat di*reset* menjadi 00:00:00 dengan menekan tombol *Clear counters*.

3. *IN (download)*

a. *Traffic / second* : Menampilkan banyaknya data dalam Byte yang masuk melalui IP yang sedang aktif setiap detiknya.

b. *Peak / second* : Menampilkan banyaknya data maksimal dalam Byte yang masuk melalui IP yang sedang aktif setiap detiknya.

c. *Average / second* : Menampilkan rata-rata banyaknya data dalam Byte yang masuk melalui IP yang sedang aktif setiap detiknya.

d. *Total* : Menampilkan jumlah data dalam Byte yang masuk melalui IP yang sedang aktif.

4. *OUT (upload)*

a. *Traffic / second* : Menampilkan banyaknya data dalam Byte yang keluar melalui IP yang sedang aktif setiap detiknya.

b. *Peak / second* : Menampilkan banyaknya data maksimal dalam Byte yang keluar melalui IP yang sedang aktif setiap detiknya.

c. *Average / second* : Menampilkan rata-rata banyaknya data dalam Byte yang keluar melalui IP yang sedang aktif setiap detiknya.

d. *Total* : Menampilkan jumlah data dalam Byte yang keluar melalui IP yang sedang aktif.

5. *Status*

Menampilkan status dari aktifitas program dalam memonitor lalu lintas jaringan. Apakah itu sedang terkoneksi dengan IP yang sedang aktif tersebut yang diterarangkan dengan *Connected* jika terkoneksi dan menerangkan apakah program sedang memmonitor dengan keterangan *Running* jika sedang jalan dan *NOT Running* jika berhenti.

6. *IP*

Menampilkan seluruh IP yang ada pada komputer, apakah IP tersebut sedang aktif maupun tidak. Jika tidak terdapat kartu jaringan pada komputer, maka akan ditampilkan IP *Loopback* yaitu 127.0.0.1. Dan jika ditemukan IP satu atau lebih dalam komputer, IP *Loopback* juga akan tetap ditampilkan karena IP *Loopback* ini akan ada secara *default*.

7. *Buttons*

a. *Freeze* : Tombol untuk menghentikan program memonitor lalu lintas jaringan. Maka, jika tombol ini ditekan akan menghentikan program dalam memberikan ketarangan banyaknya data yang masuk dan keluar melalui IP yang aktif tersebut. Selain itu juga akan memberikan status menjadi *NOT Running* dan tombol ini menjadi tidak aktif dan mengaktifkan tombol *Unfreeze*.

b. *Unfreeze* : Mengembalikan kondisi program yang sedang tidak memonitor menjadi aktif memonitor lagi. Hal ini dapat diketahui dengan melihat status menjadi *Running* dan tombol ini tidak aktif, sedangkan tombol *Freeze* menjadi aktif kembali.

c. *Clear counters* : Tombol untuk membersihkan durasi program dalam memonitor lalu lintas jaringan. Tidak menghentikan durasi waktunya akan tetapi mengembalikan nilainya menjadi 00:00:00 dan kemudian akan berjalan lagi.

d. *Exit* : Tombol untuk keluar dari program monitor lalu lintas jaringa.

e. *Stay On Top* : *Checkbox* untuk meletakkan *form* selau di atas atau tidak.

BAB V

IMPLEMENTASI

5.1 Batasan implementasi

Pada bagian batasan implementasi ini akan menjelaskan apa yang menjadi batasan implementasi perangkat lunak yang dikembangkan, antara lain adalah bahasa pemrograman yang digunakan, lingkungan pengembangan, asumsi-asumsi yang sudah ditetapkan sebelumnya dan selama pengembangan perangkat lunak, serta batasan-batasan lain.

5.1.1 Bahasa pemrograman

Perangkat lunak ini dibuat menggunakan bahasa pemrograman delphi dan menggunakan *compiler* Borland Delphi 7 sebagai visualisasinya. Bahasa ini digunakan karena sudah teruji kemudahan penggunaannya. Bahasa ini juga paling banyak digunakan dalam pengembangan aplikasi-aplikasi di lingkungan Microsoft Windows selain Visual Basic. Sumber pemrograman diambil dari www. delphi.about.com

5.1.2 Lingkungan pengembangan

Lingkungan pengembangan dilakukan di rumah peneliti di Jl.kalisahak No.9, Jogjakarta. Pengembangan dilakukan langsung pada jaringan LAN, maupun dengan menggunakan sebuah komputer *stand alone* yang langsung terkoneksi dengan internet. Komputer yang digunakan menggunakan sistem operasi Windows XP Service Pack 2.

5.1.3 Asumsi-asumsi yang digunakan

Dalam pengembangan aplikasi ini, terdapat beberapa asumsi yang diberikan sebelumnya, antara lain:

1. Aplikasi ini akan digunakan di lingkungan LAN dalam perkantoran dan sejenisnya. Diasumsikan aplikasi telah terpasang dan berjalan di setiap komputer.
2. Aplikasi ini dibatasi hanya dapat melakukan pemantuan jumlah paket data yang masuk dan yang keluar. Artinya satu komputer hanya dapat memantau dirinya sendiri tidak bisa memantau komputer yang lain

5.1.4 Batasan pengembangan aplikasi

Batasan-batasan yang diberikan dalam pengembangan perangkat lunak ini adalah sebagai berikut:

1. Aplikasi ini berjalan pada sistem operasi Microsoft Windows.
2. Aplikasi ini hanya memantau arus paket data.
3. Aplikasi ini tidak dirancang untuk memantau komputer yang lain
4. Aplikasi ini hanya akan dilakukan dalam sebuah jaringan LAN.
5. Tidak terdapat fasilitas penggunaan *password* untuk menjalanakan, menghentikan, atau mematikan aplikasi
6. Tidak terdapat fasilitas penggunaan *password* sebagai upaya pencegahan penggunaan oleh pihak lain yang sama-sama ingin memanfaatkan aplikasi.

5.2 Implementasi

Penelitian ini akan diimplementasikan dalam sebuah pernangkat lunak yang bernama Network Traffic Management. Pemilihan nama ini tidak didasarkan pada hal apapun, hanya disesuikan dengan fungsi dari aplikasi tersebut.

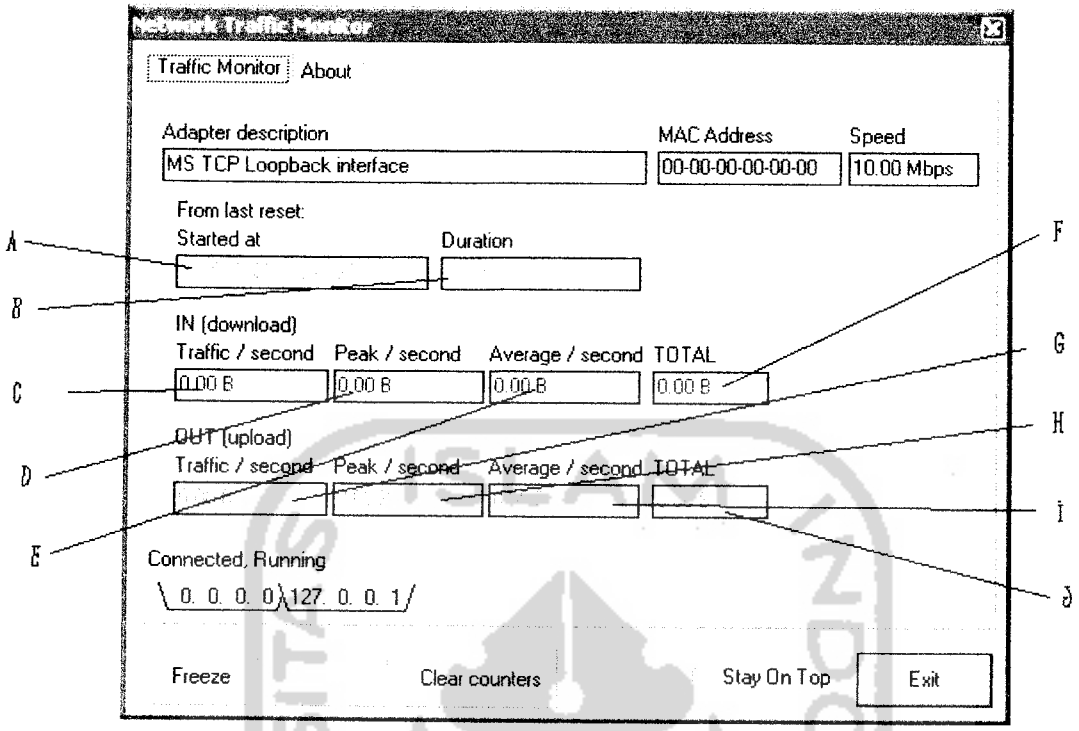
Secara umum, implementasi aplikasi ini adalah penggabungan beberapa fungsi yang dikelompokkan menjadi sebuah aplikasi baru yang dapat digunakan seorang user untuk memantau arus paket data yang masuk dan paket data yang keluar.

5.2.1 Antarmuka perangkat lunak

Aplikasi *network traffic mangement* hanya memiliki dua *form*, yaitu sebuah *form* utama dan *form about*.

1. Form utama

Form ini adalah *form* utama aplikasi *network traffic mangement*. Tampilan awal aplikasi *network traffic mangement*, sebelum terdapat aktifitas apapun termasuk koneksi, sehingga pada aplikasi ini masih kosong, ditunjukkan pada gambar 5.1.



Gambar 5.1 Form utama

Keterangan :

A. Menampilkan keterangan kapan aplikasi mulai dijalankan.

B. Menampilkan durasi waktu aplikasi telah dijalankan

C. Menampilkan banyaknya data yang masuk setiap detiknya.

D. Menampilkan banyaknya data maksimal yang masuk setiap detiknya.

E. Menampilkan rata-rata banyaknya data yang masuk setiap detiknya.

F. Menampilkan jumlah data yang masuk melalui IP yang sedang aktif.

G. Menampilkan banyaknya data yang keluar setiap detiknya.

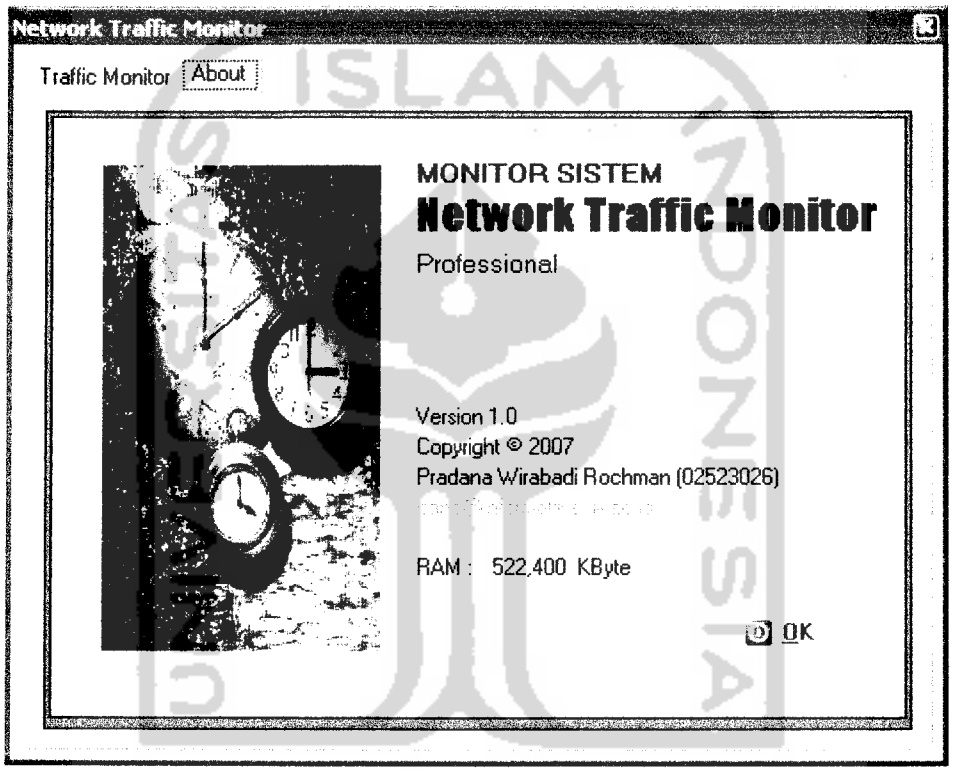
H. Menampilkan banyaknya data maksimal yang keluar setiap detiknya.

I. Menampilkan rata-rata banyaknya data yang keluar setiap detiknya.

J. Menampilkan jumlah data yang keluar melalui IP yang sedang aktif.

2. Form About

Form ini adalah *form* about aplikasi *network traffic mangement*. Tampilan about aplikasi *network traffic mangement*, ditunjukkan pada gambar 5.2.



Gambar 5.2 *Form About*

5.3 Implementasi Prosedural

5.3.1 Prosedur Koneksi

Prosedur ini berfungsi untuk mengkoneksikan aplikasi pada saat dijalankan dengan mendapatkan nilai awal dari nilai konstan pertama yaitu 0, dan setelah terkoneksi maka akan menampilkan tabs IP Address yang ada pada

komputer disertai dengan IP Address loopback. Prosedur ini dianggap penting karena prosedur inilah yang mengkoneksikan arus data pada komputer.

```
function BytesToFriendlyString(Value : DWord) : string;
const
  OneKB = 1024;
  OneMB = OneKB * 1024;
  OneGB = OneMB * 1024;
begin
  if Value < OneKB then
    Result := FormatFloat('#,##0.00 B',Value)
  else
    if Value < OneMB then
      Result := FormatFloat('#,##0.00 KB', Value / OneKB)
    else
      if Value < OneGB then
        Result := FormatFloat('#,##0.00 MB', Value / OneMB)
end; (*Akhir dari Metode BytesToFriendlyString*)

function BitsToFriendlyString(Value : DWord) : string;
const
  OneKB = 1000;
  OneMB = OneKB * 1000;
  OneGB = OneMB * 1000;
begin
  if Value < OneKB then
    Result := FormatFloat('#,##0.00 bps',Value)
  else
    if Value < OneMB then
      Result := FormatFloat('#,##0.00 Kbps', Value / OneKB)
    else
      if Value < OneGB then
        Result := FormatFloat('#,##0.00 Mbps', Value / OneMB)
end; (*Akhir dari Metode BytesToFriendlyString*)


{ Awal dari Unit TTraffic }
constructor TTraffic.Create(const AMibIfRow: TMibIfRow;
OnNewInstance : TNewInstanceEvent);
var
  Descr : string;
begin
  inherited Create;

  FRunning := True;
  FConnected := True;

  self.FInterfaceIndex := AMibIfRow.dwIndex;
  self.FIP := GetIPFromIFIndex(self.InterfaceIndex);
  self.FMac := MacAddr2Str(TMacAddress(AMibIfRow.bPhysAddr),
AMibIfRow.dwPhysAddrLen);

  SetLength(Descr, Pred(AMibIfRow.dwDescrLen));
```

```
  Move(AMibIfRow.bDescr, Descr[1], pred(AMibIfRow.dwDescrLen));
  self.FDescription := Trim(Descr);

  self.FPrevCountIn := AMibIfRow.dwInOctets;
  self.FPrevCountOut := AMibIfRow.dwOutOctets;

  self.FStartedAt := Now;


  self.FSpeed := AMibIfRow.dwSpeed;

  FActiveCountIn := 0;
  FActiveCountOut:= 0;
  FInTotal := 0;
  FOutTotal:= 0;
  FInPerSec:= 0;
  FOutPerSec:= 0;
  FPeakInPerSec := 0;
  FPeakOutPerSec:=0;

  //mengetahui kejadian ini
  if Assigned(OnNewInstance) then OnNewInstance(self);
end; (*Akhir dari Metode Create*)

procedure TTraffic.NewCycle(const InOctets, OutOctets,
TrafficSpeed: Dword);
begin
  Inc(self.FSecondsActive);

  If not Running then Exit;

  FSpeed := TrafficSpeed;

  //Masuk
  self.FInPerSec := InOctets - self.PrevCountIn;
  Inc(self.FInTotal, self.InPerSec);
  if InPerSec > 0 then Inc(FActiveCountIn);
  if InPerSec > PeakInPerSec then FPeakInPerSec := InPerSec;
  try
    self.FAverageInPerSec := InTotal div ActiveCountIn
  except
    self.FAverageInPerSec := 0
  end;
  FPrevCountIn := InOctets;

  //Keluar
  self.FOutPerSec := OutOctets - self.PrevCountOut;
  Inc(self.FOutTotal, self.OutPerSec);
  if OutPerSec > 0 then Inc(FActiveCountOut);
  if OutPerSec > PeakOutPerSec then FPeakOutPerSec := OutPerSec;
  try
    self.FAverageOutPerSec := OutTotal div ActiveCountOut
  except
    self.FAverageOutPerSec := 0
  end;
  FPrevCountOut := OutOctets;
```

```
end; (*Akhir dari Metode NewCycle*)


function TTraffic.GetIPFromIFIndex(InterfaceIndex: Cardinal):
string;
var
 i:  integer;
 IPArr : TMIBIPAddrArray;
begin
   Result := '!not_found!';  // seharusnya tidak terjadi...
   Get_IPAddrTableMIB( IpArr );  // mendapatkan tabel IP-address
   if Length(IPArr) > 0 then
     for i := low(IPArr) to High(IPArr) do  // mencari index yang
sesuai...
       if IPArr[i].dwIndex = InterfaceIndex then
       begin
         Result := IPAddr2Str(IParr[i].dwAddr);
         BREAK;
       end;
end; (*Akhir dari Metode GetIPFromIFIndex*)
```

BAB VI

ANALISIS KINERJA PERANGKAT LUNAK

Analisis diperlukan untuk mengetahui bagaimana sistem jika dijalankan, proses untuk mendapatkan data serta hasil yang dibentuk dan lain-lainnya. Analisis yang dilakukan yaitu antara lain anilisis terhadap aplikasi secara keseluruhan, pencarian IP, proses *monitoring* data, *file log* yang dihasilkan dan kesalahan-kesalahan yang mungkin terjadi.

Pengujian perangkat lunak adalah proses yang harus dilewati untuk melihat hasil kinerja aplikasi yang dibuat apakah sudah memenuhi tujuan yang telah ditentukan atau tidak. Pengujian diantaranya dilakukan dengan menggunakan komputer *desktop* yang terhubung dalam sebuah jaringan LAN dengan spesifikasi sebagai berikut:

- Komputer prosesor *AMD Turion 64*,
- *RAM* 512 MB,
- *Hard disk* 60 GB,
- Monitor dengan *VGA ATI Radeon Xpress 700 Series* 128 MB,
- *Network Interface Card* atau kartu jaringan *Realtek RTL8139 Family PCI Fast Ethernet NIC*.

Pengujian juga dapat dilakukan pada semua komputer yang ber-*Operating System Windows XP*, baik yang terhubung jaringan atau tidak, karena tanpa jaringanpun dapat dilihat lalu lintasnya dengan menggunakan IP *loopback*. Sedangkan

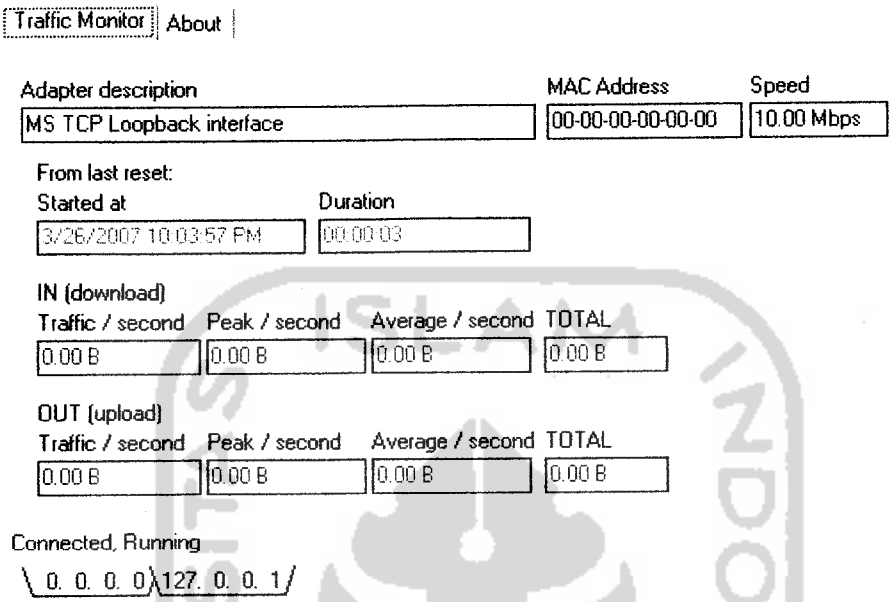
pengujian ini sendiri ditekankan kepada jumlah IP yang terdapat pada suatu komputer, status konksi pada sejumlah IP tersebut dan tentunya lalu lintas data yang melalui IP tersebut.

6.1 Analisis Pencarian IP

Pencarian IP ini dilakukan ketika aplikasi pertama kali dijalankan, yaitu ketika *timer* bernilai *true* setelah aplikasi terbentuk dan menginisialisasi semua kebutuhan sistem, sampai dengan aplikasi dihentikan. Pencarian IP dilakukan pada komputer tempat aplikasi dijalankan, yaitu *resource* pada *Microsoft Windows Network* yang diantaranya berupa IP, nama dari kartu jaringan, nomor seri dan kecepatan dari kartu jaringan, menggunakan fungsi-fungsi dari *Windows API* untuk berhubungan dengan *resource* tersebut.

6.1.1 Pengujian dengan 1 komputer

Disini akan dilakukan pengujian kinerja perangkat lunak jika aplikasi dijalankan dan tidak ada komputer lain atau koneksi lain yang terhubung. Maka tampilan pada aplikasi akan menjadi seperti gambar 6.1



Gambar 6.1 Aplikasi Hanya Menemukan IP *loopback*.

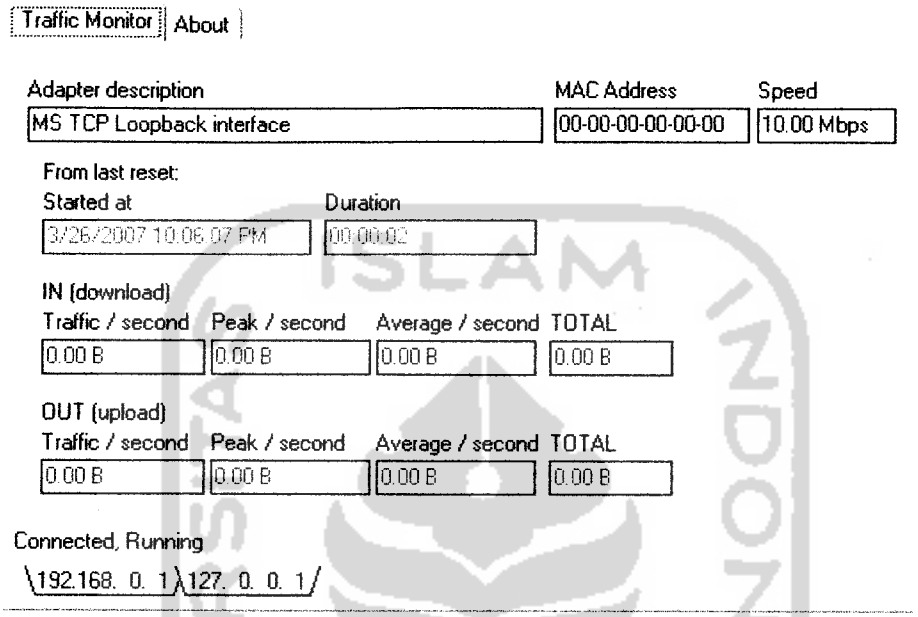
6.1.2 Pengujian dengan 2 komputer

Disini akan dilakukan pengujian kinerja perangkat lunak jika aplikasi dijalankan dan terhubung dengan hanya 1 komputer lain. IP Address komputer pertama di set sesuai dengan yang terlihat pada gambar 6.2



Gambar 6.2 IP Address di set ke 192.168.0.1

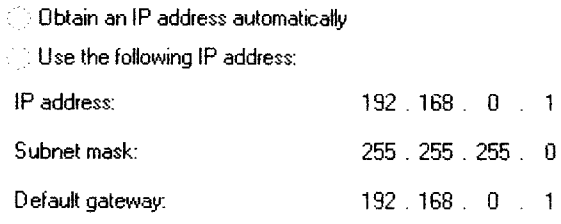
Ketika aplikasi dijalankan maka tampilan akan tampak seperti pada gambar 6.3



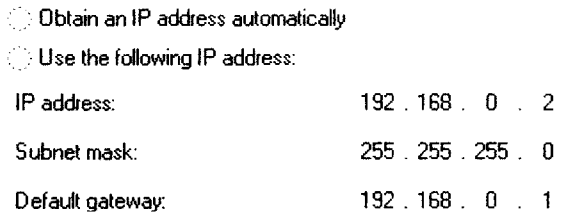
Gambar 6.3 Aplikasi menemukan 1 IP dan aktif

6.1.3 Pengujian dengan 5 komputer

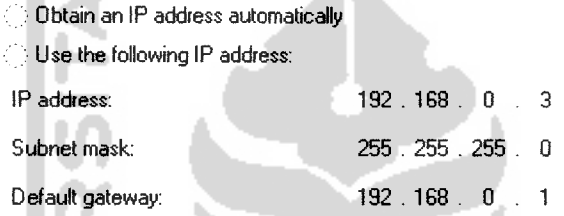
Disini akan dilakukan pengujian kinerja perangkat lunak jika aplikasi dijalankan dan terhubung dengan 5 komputer lain. IP Address komputer pertama sampai ke lima akan di set sesuai dengan yang terlihat pada gambar 6.4 sampai gambar 6.8



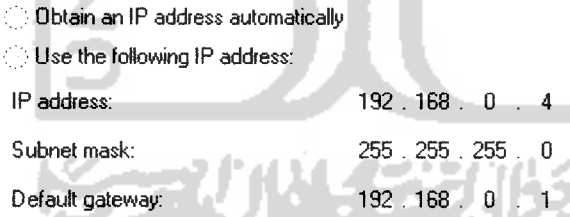
Gambar 6.4 IP Address komputer pertama



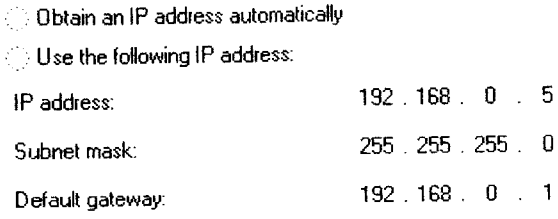
Gambar 6.5 IP Address komputer kedua



Gambar 6.6 IP Address komputer ketiga

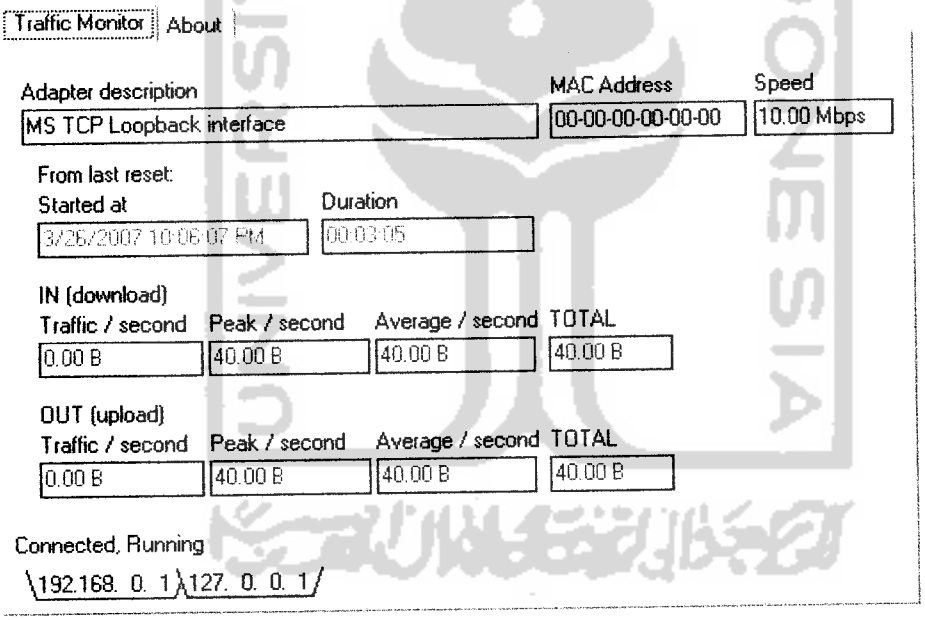


Gambar 6.7 IP Address komputer keempat



Gambar 6.8 IP Address komputer kelima

Ketika aplikasi dijalankan maka tampilan akan tampak seperti pada gambar 6.9

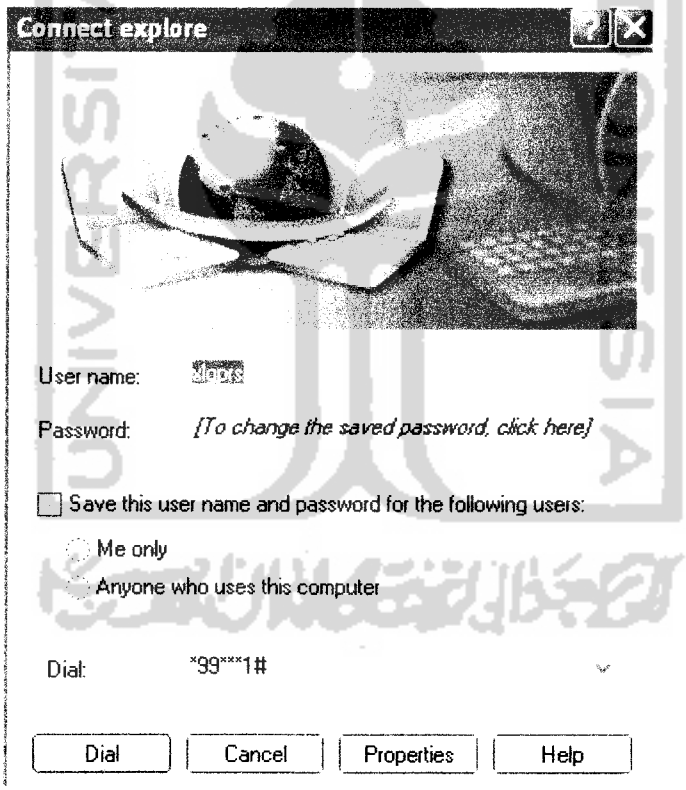


Gambar 6.9 Aplikasi tetap hanya menemukan 1 IP dan aktif

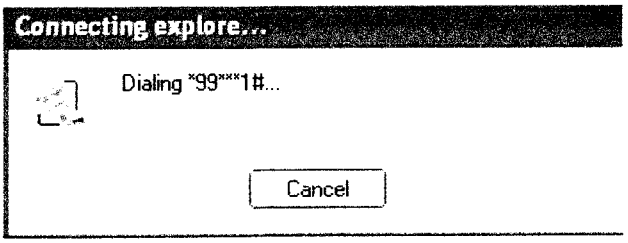
6.1.3 Pengujian dengan 2 komputer dan 1 koneksi internet

Disini akan dilakukan pengujian kinerja perangkat lunak jika aplikasi dijalankan dan terhubung dengan 1 komputer lain dan 1 koneksi internet. IP Address komputer pertama di set 192.168.0.1.

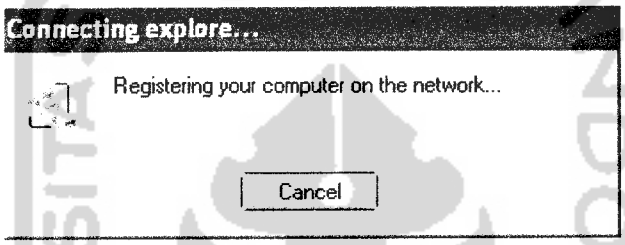
Koneksi Internet yang akan di uji adalah koneksi internet menggunakan kartu gsm "XL" yang berbasis GPRS , setting koneksi di set seperti yang terlihat pada gambar 6.10



Gambar 6.10 Koneksi internet XL GPRS

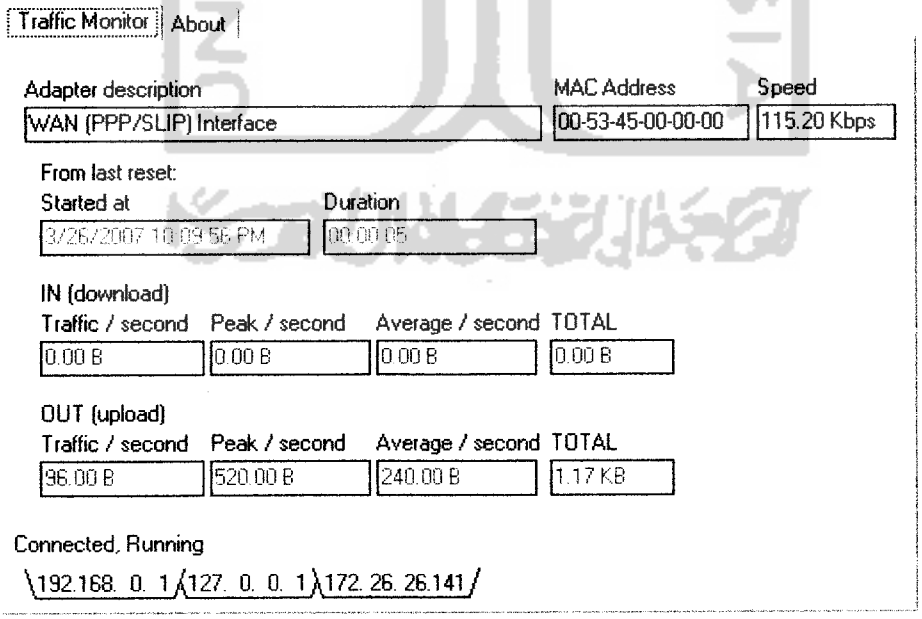


Gambar 6.11 Proses mengkoneksikan dengan "XL"



Gambar 6.12 Proses koneksi internet sukses

Ketika aplikasi dijalankan maka tampilan akan tampak seperti pada gambar 6.13

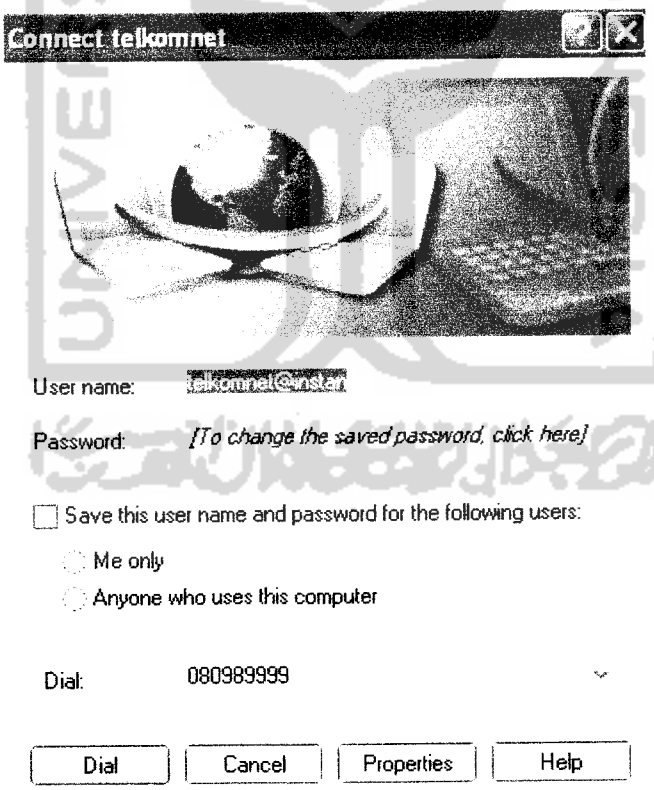


Gambar 6.13 Aplikasi menemukan 2 IP dan aktif

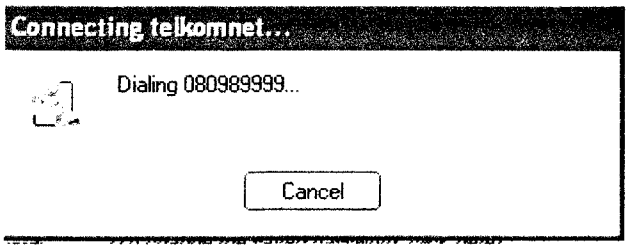
6.1.4 Pengujian dengan 2 komputer dan 2 koneksi internet

Disini akan dilakukan pengujian kinerja perangkat lunak jika aplikasi dijalankan dan terhubung dengan 1 komputer lain dan 1 koneksi internet. IP Address komputer pertama di set 192.168.0.1.

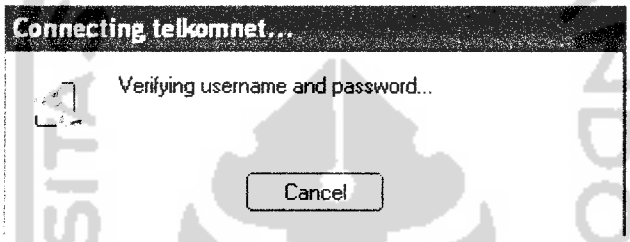
Koneksi Internet yang akan di uji adalah koneksi internet menggunakan kartu gsm "XL" yang berbasis GPRS (telah dilakukan di pengujian sebelumnya) dan koneksi internet menggunakan TELKOMNET yang berbasis dial-up, setting koneksi telkomnet di set seperti yang terlihat pada gambar 6.14



Gambar 6.14 Koneksi internet TELKOMNET

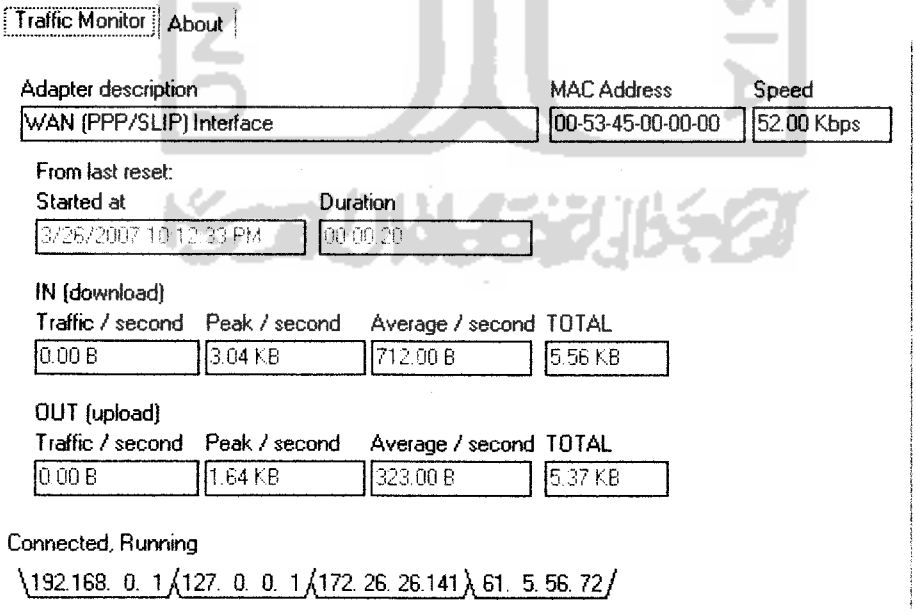


Gambar 6.15 Proses mengkoneksikan dengan TELKOMNET



Gambar 6.16 Proses koneksi internet sukses

Ketika aplikasi dijalankan maka tampilan akan tampak seperti pada gambar 6.17



Gambar 6.17 Aplikasi menemukan 3 IP dan aktif

6.1.5 Analisis Aplikasi

Secara umum, aplikasi yang dikembangkan telah berjalan baik sesuai dengan kebutuhan dan tujuan awalnya.

Hasil dari pencarian IP ini berbeda antara satu komputer dengan komputer yang lain jika aplikasi dijalankan pada berbeda-beda komputer. Perbedaan ini terletak pada jumlah IP yang ditemukan dan nomor dari masing-masing IP tersebut yang juga akan mempengaruhi deskripsi dari masing-masing IP, lalu lintas data dan tombol pengaturan yang akan disediakan. Sedangkan setiap aplikasi dijalankan pada suatu komputer, maka akan ditemui kesamaan dari salah satu nomor IP pada komputer tersebut, yaitu 127.0.0.1 yang digunakan sebagai *loopback*.

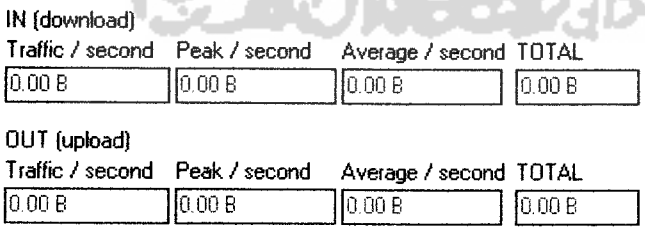
Jumlah IP yang ditemukan ini tergantung pada beberapa hal, diantaranya banyaknya kartu jaringan pada komputer, apakah IP diatur otomatis atau manual dan apakah koneksi sedang *enable*. Namun demikian, walaupun pada komputer tersebut tidak terdapat kartu jaringan atau ada kartu jaringan tetapi koneksinya dalam status *disable*, maka setidak-tidaknya terdapat satu IP yang akan ditampilkan yaitu IP *loopback* (127.0.0.1).

Selain IP *loopback*, IP yang ditampilkan tergantung pada banyaknya kartu jaringan pada komputer, banyaknya port pada kartu jaringan dan status koneksinya. Jika terdapat lebih dari satu kartu jaringan, maka IP yang ditampilkan juga akan lebih dari satu. Namun ini tidak berlaku untuk berapapun komputer yang terkoneksi, jadi sebanyak apapun komputer yang terkoneksi ke dalam jaringan, yang akan di deteksi adalah sebanyak port pada kartu jaringan komputer

itu sendiri. Hal yang sama juga akan terjadi ketika pada kartu jaringan terdapat lebih dari satu *port* walaupun hanya terdapat satu kartu jaringan. Status koneksi juga mempengaruhi hasil dari pencarian IP, walapun terdapat lebih dari satu kartu jaringan atau lebih dari satu *port*, maka tidak akam ditampilkan IP tersebut. Jika status koneksi *disable*, maka tidak akan ditampilkan nomor IP tersebut walaupun dalam properti IP sudah di-*setting* nomor IP tertentu.

6.2 Analisis Proses *Monitoring* Data

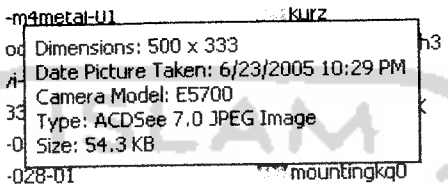
Proses *monitoring* data ini terjadi setelah IP ditemukan dan terletak pada *tab* IP ditampilkan. Proses *monitoring* data ini terdiri dari data masuk dan keluar melalui IP yang aktif pada *tab* IP. Data-data tersebut akan ditampilkan terus menerus selama tidak terjadi perubahan kondisi dari IP maupun aplikasi. Maksudnya kondisi dari IP yaitu ketika IP tersebut terus terkoneksi dan tidak terputus sebelum aplikasi berhenti. Sedangkan kondisi aplikasi yaitu ketika aplikasi berhenti karena dihentikan oleh *user* maupun berhenti karena hal-hal yang lainnya.



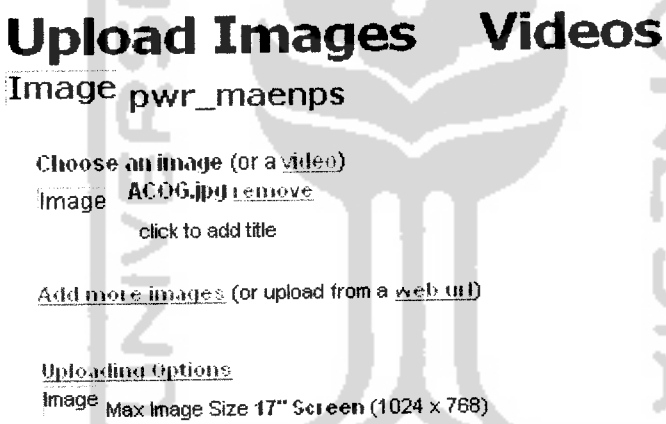
Gambar 6.18 *Monitoring* data

6.2.1 Pengujian Proses *Uploading*

Disini akan dilakukan pengujian kinerja perangkat lunak jika sebuah komputer melakukan proses *Uploading* .

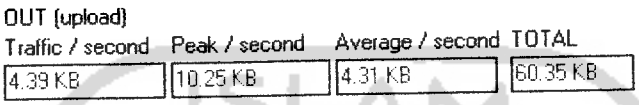


Gambar 6.19 Menunjukkan besar file yang akan di upload



Gambar 6.20 Proses *Uploading*

Ketika proses *uploading* selesai maka total data yang keluar tampak pada gambar 6.17



Gambar 6.21 Proses *Uploading* selesai

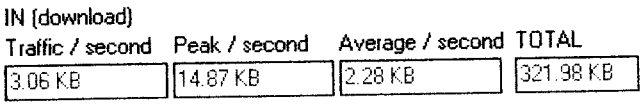
6.2.2 Pengujian Proses *Downloading*

Disini akan dilakukan pengujian kinerja perangkat lunak jika sebuah komputer melakukan proses *Downloading* .



Gambar 6.22 Menunjukkan besar file yang akan di *download*

Ketika proses *uploading* selesai maka total data yang keluar tampak pada gambar 6.19



Gambar 6.23 Proses *Downloading* selesai

6.2.3 Analisis Aplikasi

Secara umum, aplikasi yang dikembangkan telah berjalan baik sesuai dengan kebutuhan dan tujuan awalnya.

Selain data-data arus data yang masuk dan keluar, pada aplikasi juga ditampilkan data deskripsi dari masing-masing IP yang ditemukan. Data tersebut antara lain deskripsi dari kartu jaringan, nomor serial dan kecepatannya. Jika IP loopback akan mempunyai deskripsi *MS TCP Loopback Interface*, serialnya kosong atau 00-00-00-00-00-00 dan mempunyai kecepatan 10,00 Mbps. Sedangkan jika IP dari suatu kartu jaringan akan memiliki deskripsi, serial dan kecepatan yang berbeda-beda sesuai dengan kartu jaringannya. Telah dibuktikan bahwa kartu jaringan tersebut mempunyai deskripsi sebagai *Realtek RTL8139 Family PCI Fast Ethernet NIC - Packet Scheduler Miniport*, mempunyai serial 00-16-76-0C-06-52 dan kecepatannya 100,00 Mbps.

Proses *monitoring* data ini juga akan dicatat mengenai jam dan tanggal memonitor serta durasi dari *monitoring*. Baik data deskripsi maupun data waktu tersebut berpengaruh dari IP yang ditampilkan. Sehingga masing-masing IP memiliki data yang berbeda, akan tetapi tidak akan berubah sampai aplikasi dihentikan kecuali data durasi yang dapat di-*reset* ulang ke posisi 00:00:00 atau posisi semula. Sedangkan proses *monitoring* sendiri akan menampilkan *monitoring* data masuk dan data keluar melalui IP yang aktif atau jika terdapat lebih dari satu IP dapat dipilih melalui *tab* IP. Proses ini diantaranya akan mencatat lalu lintas data yang masuk maupun keluar, data tertinggi yang pernah melalui IP pada saat aplikasi tersebut dijalankan, rata-rata data yang keluar dan

masuk serta total data secara keseluruhan. Semua data tersebut ditampilkan dalam satuan *Byte* untuk lalu lintas data dan *Kilo Byte* untuk yang lainnya serta akan diperbaharui setiap detiknya.

Dari hasil analisis pengujian diatas dapat terlihat kelebihan dan kekurangan dari aplikasi *Network Traffic Monitorin.*

Kelebihan :

a. Sistem telah mampu memantau arus paket data yang masuk dan keluar.
b. Sistem telah mampu menampilkan hasil dari semua perhitungan yang dilakukan.
c. Sistem telah mampu menyimpan hasil dari semua pemantauan yang dilakukan

Kekurangan :

a. Program hanya mampu memantau arus paket data yang masuk dan keluar tanpa bisa mencegahnya.
b. Program hanya mampu memantau arus paket data yang masuk dan keluar pada komputer dimana program dijalankan.
c. Program hanya bisa dijalankan di sistem operasi Microsoft Windows.

BAB VII

SIMPULAN DAN SARAN

7.1 Simpulan

Dari hasil analisis dan perbandingan yang telah dilakukan, maka dapat disimpulkan sebagai berikut:

1. Aplikasi Network Traffic Monitoring yang telah dikembangkan memberikan kemudahan bagi pengguna internet volume based untuk memantau penggunaan internetnya.
2. Aplikasi Network Traffic Monitoring sangat menguntungkan untuk mengontrol pemakaian internet.
3. Aplikasi Network Traffic Monitoring lebih memberikan kemudahan dalam penggunaannya karena sangat *user-friendly* dan tidak memerlukan keahlian khusus untuk menjalankannya.

7.2 Saran

Untuk pengembangan aplikasi ini di kemudian hari, dan karena banyaknya kekurangan yang terdapat di sana-sini, maka berikut adalah saran-saran yang diberikan oleh penulis terhadap pengembangan berikutnya.

1. Perlunya dikembangkan aplikasi sejenis yang berjalan antar sistem operasi.
2. Fitur pemantau paket data yang masuk dan keluar secara kesuluruhan meskipun aplikasi telah dihentikan.
3. Perlunya fitur memutuskan koneksi internet.

4. Perlunya fitur aplikasi dijalankan otomatis ketika komputer tersambung dengan komputer lain secara LAN maupun komputer tersambung ke internet.

DAFTAR PUSTAKA

[MAR02] Martina, I. *Pemrograman Internet dengan Delphi*. Jakarta: PT Elex Media Komputindo, 2002

[NUG02] Nugroho, W. *Tip dan Trik Pemrograman Delphi*. Jakarta: PT Elex Media Komputindo, 2002

[WAH03] Wahana Komputer. *Tip dan Trik Pemrograman Delphi 7.0*. Yogyakarta: ANDI, 2003