BAB III
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Pada analisis sistem akan dibahas mengenai analisis aplikasi sejenis dan analisis spesifikasi sistem yang akan dirancang. Pada analisis aplikasi sejenis akan dilakukan observasi terhadap beberapa aplikasi sejenis yang populer dan diminati pengguna.

3.1.1 Analisis Packet Sniffer Sejenis

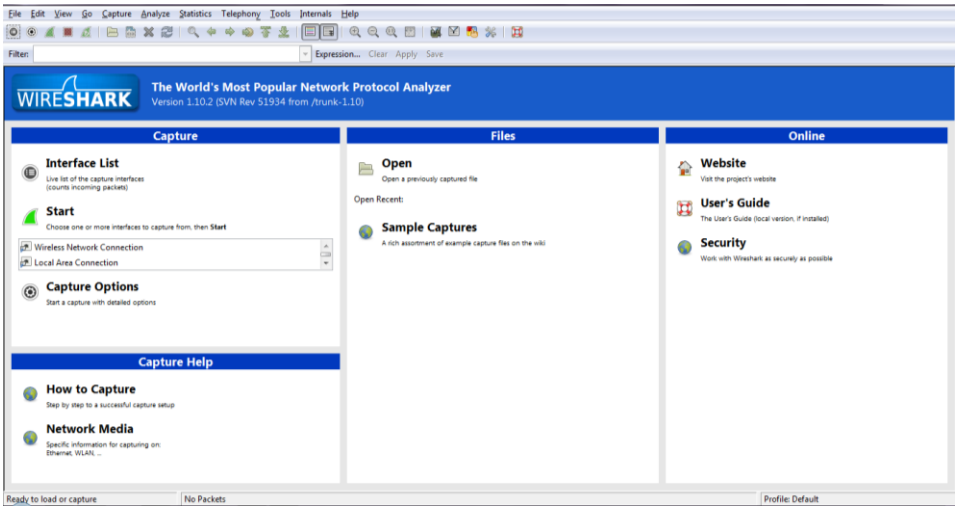
Packet Sniffer yang dikenal sebagai *network analyzer* merupakan sebuah aplikasi yang dapat melihat lalu lintas data pada jaringan. Berikut adalah aplikasi *packet sniffer* yang sering digunakan oleh *user*.

3.1.1.1 Wireshark

Wireshark merupakan *tool* yang ditujukan untuk pengalisisan paket data jaringan. Wireshark juga melakukan pengawasan paket secara *real time* dan kemudian menangkap data dan menampilkannya selengkap mungkin. Wireshark memiliki seperangkat fitur yang meliputi sebagai berikut :

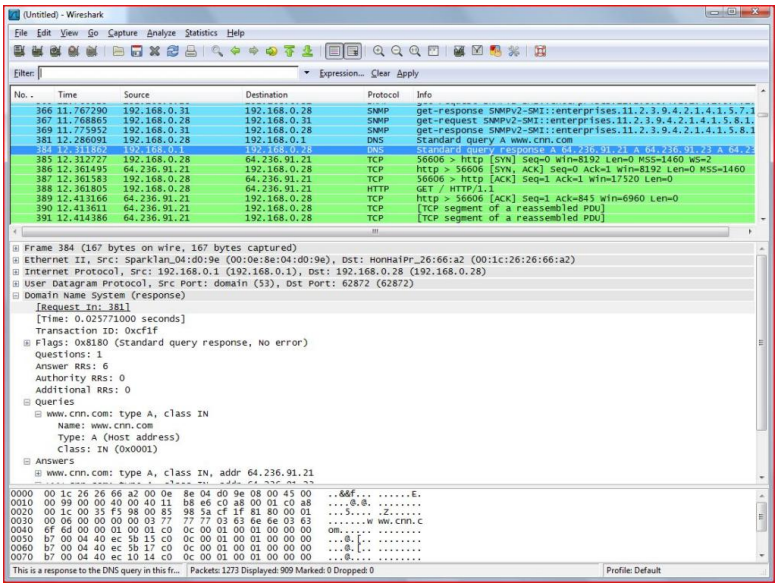
1. Tersedia untuk sistem operasi Unix, Linux dan Windows.
2. Menangkap paket data dari antarmuka jaringan.
3. Wireshark dapat menangkap lalu lintas dari banyak jenis media jaringan yang berbeda.
4. Antarmuka aplikasi dengan informasi protokol yang sangat rinci.

Berikut antarmuka *main* program Wireshark yang dapat dilihat pada Gambar 3.1.



Gambar 3.1 Antarmuka Wireshark

Menampilkan daftar semua paket yang diambil pada waktu Wireshark aktif. Berikut tampilan daftar paket yang ditunjuk pada Gambar 3.2.

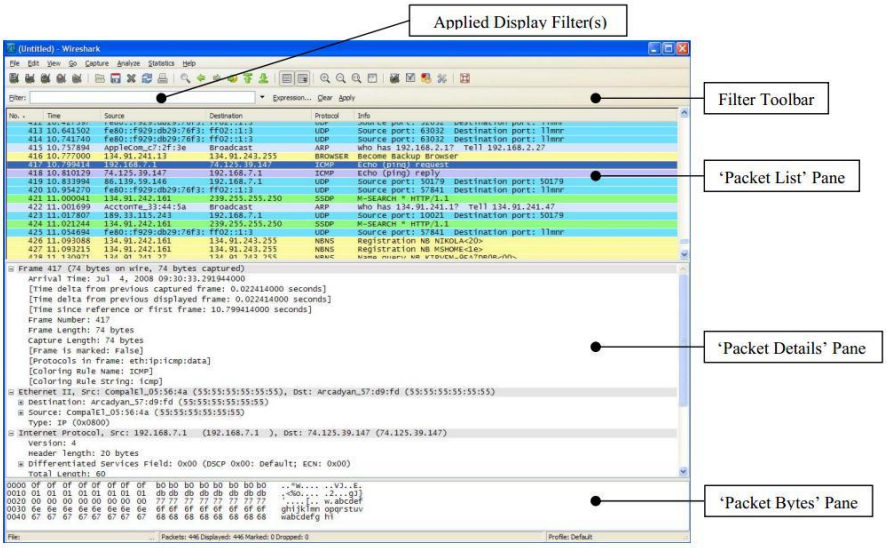


Gambar 3.2 Daftar Paket *Capture*

Setiap baris pada paket sesuai dengan satu paket pada saat Wireshark melakukan *capture* data. Jika memilih salah satu baris pada deretan paket-paket, maka rincian paket dan besar paket dapat dilihat pada kolom *default*.

1. No : Urutan paket pada saat Wireshark melakukan *capture*.
2. Time : Lama waktu pada saat Wireshark melakukan *capture*.
3. Source : IP *address* sumber.
4. Destination : IP *address* tujuan.
5. Info : Informasi tentang isi paket.

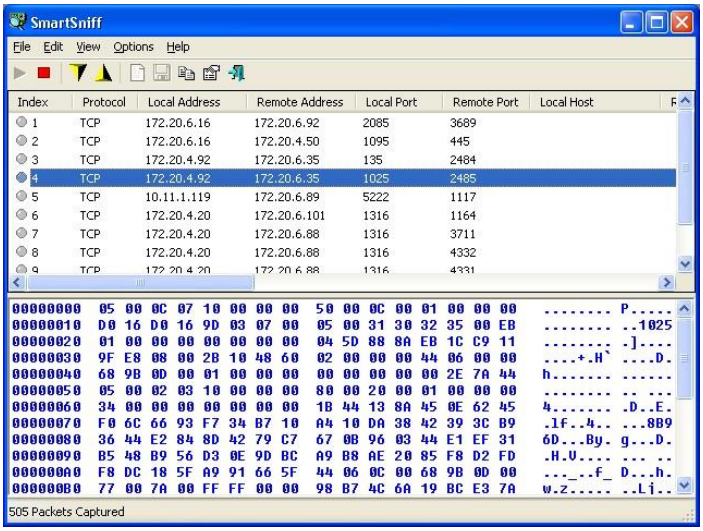
Rincian paket dapat dilihat pada *panel* "*Packet List*", berikut tampilan detail paket *pane* pada Gambar 3.3.



Gambar 3.3 Rincian *Packet Pane*

3.1.1.2 Smart Sniff

Smart Sniff adalah perangkat *software* untuk memonitor jaringan utilitas yang memungkinkan anda untuk menangkap Paket TCP/IP yang melewati *adapter* jaringan anda, dan melihat data yang diambil sebagai urutan dari percakapan antara *client* dan *server*. Percakapan TCP/IP dapat dilihat dalam kode ASCII (*American Standard Code for Information Interchange*) untuk protokol berbasis teks, seperti HTTP, SMTP, POP3 dan FTP atau sebagai Hex *dump*. Berikut antarmuka Smart Sniff yang dapat dilihat pada Gambar 3.4.



Gambar 3.4 Program Smart Sniff

Smart Sniff menyediakan 3 metode untuk menangkap TCP/IP paket :

a. *Raw Sockets* memungkinkan menangkap paket TCP/IP pada jaringan anda tanpa menginstal *driver*. Metode ini memiliki beberapa keterbatasan dan masalah.

b. WinPcap *Capture Driver* memungkinkan menangkap paket TCP/IP pada semua sistem operasi Windows. (Windows 98/ME/NT/2000/XP/2003/Vista). Dalam menggunakannya, harus men-*download* dan menginstal *driver* WinPcap. Metode ini umumnya cara yang lebih disukai untuk menangkap paket TCP/IP dengan Smart Sniff, dan bekerja lebih baik dari pada metode *Raw Sockets*.

c. *Microsoft Network Monitor Driver* hanya dapat digunakan pada operasi sistem Windows 2000/XP/2003. Microsoft menyediakan *driver* bebas di bawah Windows 2000/XP/2003 yang dapat digunakan oleh Smart Sniff, tapi *driver* ini tidak diinstal secara *default*, tetapi harus menginstal secara manual.

3.1.2 Perbandingan fitur antara Wireshark dan Smart Sniff

Pada aplikasi Wireshark dan memiliki beberapa perbedaan fitur, berikut perbandingan fitur pada Wireshark dan pada tabel 3.1.

Tabel 3.1 Perbandingan fitur Wireshark dan Smart Sniff

NO	FITUR	Wireshark	Smart Sniff
1	Model program	*Installer* dan *Portabel*.	*Installer*
2	Antarmuka (*User Interface*)	GUI	GUI
3	*Platform*	Windows, Linux, Unix	Windows
4	*Distribution Source Code*	*Open Source*	*Freeware*
5	*Scan Port*	YA	YA
6	Dukungan Mode *Script*	YA	
7	*Scaning Packet*	YA	YA
8	*Network Interface*	*Ethernet*, IEEE 802.11, *Bluetooth*, USB, *Token Ring*	*Ethernet*, IEEE 802.11
9	*Service Detection*	YA	
10	*Save Capture*	YA	YA
11	Grafik Laporan	YA	YA
12	Mode *Capture*	*Capture* seluruh aktifitas jaringan	*Capture* seluruh aktifitas jaringan
13	*Filter Capture*	YA	YA

3.1.3 Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional menggambarkan kebutuhan sistem yang menitik-beratkan pada properti perilaku yang dimiliki oleh sistem, diantaranya kebutuhan perangkat keras, perangkat lunak, serta *user* sebagai bahan

analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan sistem yang akan diterapkan.

3.1.3.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan untuk membangun dan menjalankan aplikasi adalah sebagai berikut :

Tabel 3.2 Spesifikasi Perangkat Keras

Perangkat Keras	Spesifikasi
Intel® *Core*™ i3-2310M	CPU @2.10GHz (4CPUs)
Harddisk	500 GB
RAM	4 GB
LCD *Monitor*	1366x768
Modem	1. CDMA 2000 1x EVDO Rev. B 800/1900 Mhz 2. *Maximum Download* 14.7 Mbps 3. *Maximum Upload* 5.6 Mbps

Sedangkan perangkat keras minimum yang dibutuhkan untuk menjalankan aplikasi yang dirancang adalah sebagai berikut :

Tabel 3.3 Spesifikasi Minimum Perangkat Keras

Perangkat Keras	Spesifikasi
Processor Dual Core	CPU @2.10GHz (4CPUs)
RAM	1024 MB
Hardisk	40GB
LCD *Monitor*	Tidak ada resolusi minimum
Modem	4. CDMA 2000 1x EVDO Rev. B 800/1900 Mhz 5. *Maximum Download* 14.7 Mbps 6. *Maximum Upload* 5.6 Mbps

3.1.3.2 Spesifikasi Perangkat Lunak

Berikut ini adalah spesifikasi perangkat lunak yang digunakan dalam membangun dan menjalankan aplikasi.

Tabel 3.4 Spesifikasi Perangkat Lunak

Perangkat Lunak	Spesifikasi
Sistem operasi Windows XP	32-*bit*.
Delphi	*Version* 7
Source code Paket Sniffer	OverbyteICS : 1. OverbyteIcsDefs.inc 2. OverbyteIcsLibrary.pas 3. OverbyteIcsLogger.pas 4. OverbyteIcsThreadTimer.pas 5. OverbyteIcsTypes.pas 6. OverbyteIcsUtils.pas 7. OverbyteIcsWinsock.pas 8. OverbyteIcsWndControl.pas 9. OverbyteIcsWSockBuf.pas 10. OverbyteIcsWsocket.pas Magenta : 1. MagClasses.pas 2. MagentaBpf.pas 3. MagentaMonpcap.pas 4. MagentaMonsock.pas 5. MagentaNdis_def.pas 6. MagentaPacket32.pas 7. MagentaPackhdrs.pas 8. MagentaPcap.pas
Source code Network Traffic	1. IPHelper.pas 2. IPHLPAPI.pas

1. *Source Code* Paket Sniffer

Pada *packet sniffer source code* yang digunakan adalah ICS. ICS (*Internet Component Suite*) merupakan *library* komponen gratis yang dapat berjalan pada Delphi, C++ Builder, BDS dan RAD Studio versi (Win32). Di dalam *library* komponnen termasuk TCP, UDP, *Raw Sockets*, *client*, *server*, serta terdapat protokol seperti FTP, SMTP, POP3, NNTP, HTTP dan banyak lagi. ICS juga mendukung SSL dan TLS dengan bantuan OpenSSL. komponen ICS yang digunakan antara lain.

Tabel 3.5 Komponen ICS

Komponen	Fungsi
OverbyteIcsDefs.inc	Meng-*compile* setiap paket ICS.
OverbyteIcsLibrary.pas	Tempat menyimpan *function*, setiap fungsi-fungsi tertentu dipanggil melalui *source code library*.
OverbyteIcsLogger.pas	Pencatatan setiap *event*
OverbyteIcsThreadTimer.pas	Penerapan *custom timer class*, tidak menggunakan Windows API *timer* tapi menggunakan *timer* yang sudah disediakan.
OverbyteIcsTypes.pas	Deklarasi tipe data
OverbyteIcsUtils.pas	Tempat utilitas secara umum
OverbyteIcsWinsock.pas	Untuk mendukung Microsoft NET *framework*.
OverbyteIcsWndControl.pas	Memberi kendali pada TIcsWndHandlerPool untuk mendaftarkan pesan yang dibutuhkan.
OverbyteIcsWSockBuf.pas	Tempat antrian dengan menggunakan metode antrian FIFO (*First in First Out*)

OverbyteIcsWsocket.pas	Mendukung TCP, UDP, SOCKS5 dan dapat digunakan untuk membangun program klien dan *server*.

2. *Source Code* Network Traffic

Pada *network traffic source code* yang digunakan adalah IPHelper.pas dan IPHLPAPI.pas. IP *Helper* (*Internet Protocol Helper*) membantu administrasi jaringan komputer lokal dengan memungkinkan aplikasi untuk melakukan pembacaan informasi tentang *protocol*, *adapter info*, IP *address* dan MAC *address*. Untuk IPHLPAPI.pas berfungsi pembacaan IP *address structures*, *interface structures*, *adapter* info *structures*, TCP *structure*s, UDP *structures*, IP *structures*, dan masukan informasi *function* dari IPHLPAPI.dll.

3.1.3.3 Spesifikasi User

User dibutuhkan sebagai pengguna atau menjalankan aplikasi, adapun spesifikasi *user*, yaitu:

1. *User* mampu mengoperasikan sistem operasi *platform* Windows.
2. Mengerti informasi *traffic* pada jaringan seperti *transfer rate*, *peak rate*, *average rate*.

3.1.4 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar dapat berjalan dengan baik dan sesuai dengan kebutuhan.

3.1.4.1 Analisis Sistem

Adapun analisis sistem yang dibutuhkan antara lain :

1. Sistem yang dirancang dengan menggunakan metode *client* atau *frontend*.
2. Sistem operasi yang digunakan adalah Windows XP.
3. *Compiler* sistem yang digunakan adalah Delphi 7.

4. *Network traffic* dan *packet sniffer* menggunakan *source code* Magenta, OverByteICS, IPHelper dan IPHLPAPI.
5. Protokol yang digunakan adalah TCP dan UDP.
6. Spesifikasi sistem pada sisi *client* atau *frontend* yaitu perangkat yang digunakan bersifat *mobile*, dapat diakses kapan dan dimanapun dan terhubung dengan layanan *broadband* pada teknologi EVDO.
7. *Download* dan *upload* memanfaatkan service HTTP *monitoring traffic* yang melewati *network interface*.

3.2 Perancangan Sistem

3.2.1 Pemodelan Sistem

Tahap pemodelan dalam perancangan sistem yang digunakan antara lain *Use Case Diagram* dan *Class Diagram*.

1. *Use Case Diagram*

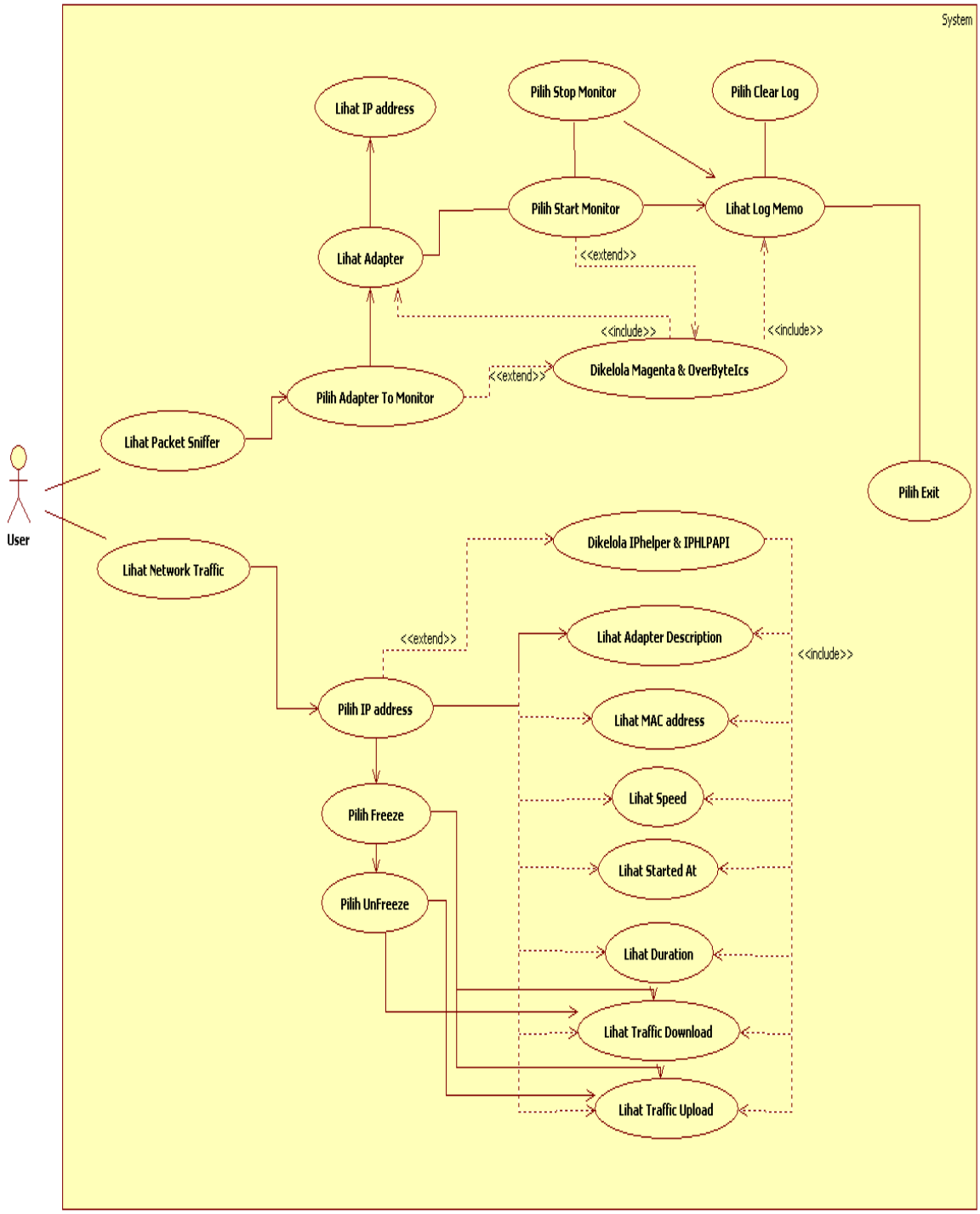
Pada tahap *use case diagram* akan mendiskripsikan fungsi dari sebuah sistem dari perspektif pengguna, *use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara pengguna sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sejauh sistem dipakai. *Use case* terbagi dalam tiga bagian yaitu definisi *actor*, definisi *use case*, dan *scenario use case*.

a. Definisi Aktor

Definisi Aktor berfungsi untuk mendeskripsikan peran aktor dalam perangkat lunak ini. Definisi tersebut dapat dilihat pada tabel 3.3.

Tabel 3.6 Definisi Aktor

Aktor	Deskripsi
User	Pengguna yang berperan menjalankan aplikasi.



Gambar 3.5 *Use Case Diagram*

b. Definisi *Use Case*

Definisi *use case* berfungsi sebagai penjelasan mengenai proses yang terdapat pada setiap *use case*. Dapat dilihat definisi tersebut pada Tabel 3.7.

Tabel 3.7 Definisi *Use Case*

Nama Use Case	Deskripsi
Lihat Packet Sniffer	*Form* yang menampilkan *adapter to monitor*, IP *address, start monitor/stop monitor, log capture, clear*.
Pilih Adapter To Monitor	Fungsional yang menampilkan *list* NIC yang tersedia.
Lihat Adapter	Fungsional NIC yang digunakan.
Lihat IP Address	Fungsional yang menampilkan alamat IP sumber.
Start Monitor	Fungsional untuk menjalankan *capture packet sniffer*.
Lihat Log Capture	Fungsional untuk menampilkan hasil *capture* data pada jaringan.
Stop Monitor	Fungsional untuk menghentikan proses *capture* data.
Pilih Clear Log	Fungsional untuk menghapus hasil *capture* data pada jaringan.
Dikelola OverByteICS	Fungsional yang dilakukan oleh sistem untuk memberi informasi yang diminta *user* seperti informasi *list adapter*, IP *address, capture* data.
Lihat Traffic	*Form* yang menampilkan *adapter description*, IP *address*, MAC *address, download traffic, upload traffic, freeze dan unfreeze*
Pilih IP Address	Fungsional kategori untuk memilih alamat IP yang akan di monitoring. Fungsional ini secara otomatis akan

	menampilkan *adapter description*, MAC *address* dan *speed adapter*.
Lihat Adapter Description	Fungsional untuk menampilkan *adapter* yang digunakan.
Lihat MAC Address	Fungsional untuk menampilkan nomor fisik perangkat keras.
Lihat Speed Adapter	Fungsional untuk menampilkan kecepatan maksimum NIC.
Lihat Traffic Download	Fungsional untuk menampilkan *traffic download* pada NIC.
Lihat Traffic Upload	Fungsional untuk menampilkan *traffic upload* pada NIC.
Pilih Freeze/UnFreeze	Fungsional untuk menghentikan sementara monitoring yang berlangsung dan menjalankan kembali monitoring.
Mengelola IPHelper & IPHLPAPI	Fungsional yang dilakukan oleh sistem untuk memberi informasi yang diminta *user* seperti informasi IP *address*, *adapter description* dan *traffic*.
Pilih Exit	Fungsional untuk menutup aplikasi

c. Skenario *Use Case*

Skenario setiap bagian pada *use case* menunjukkan penjelasan setiap bagian-bagian di dalam *use case* tersebut. Di dalam skenario *use case* akan tergambar jelas, dimana *user* memberikan perintah pada setiap bagian dan respon apa yang diberikan oleh sistem kepada *user* setelah *user* memberikan perintah pada setiap bagian-bagian *use case*.

Tabel 3.8 Skenario *Use Case* Lihat Packet Sniffer

Identifikasi	
Nama Use Case	Lihat Packet Sniffer
Aktor	*User*
Tujuan	Melihat aplikasi *packet capture*
Skenario	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan aplikasi *Packet Sniffer*.
2. Melihat fungsional pada *Packet Sniffer*.	
Skenario Alternatif	
Aktor	Reaksi Sistem
-	-

Tabel 3.9 Skenario *Use Case* Pilih *Adapter To Monitor*

Identifikasi	
Nama Use Case	Pilih *Adapter To Monitor*
Aktor	*User*
Tujuan	Melihat *list adapter*
Skenario	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan *listing* a*dapter*.
2. Pilih *adapter* yang akan di-*monitoring*.	
	3. Adapter yang dipilih ditetapkan sebagai *adapter* yang akan di *monitoring*
Skenario Alternatif	
Aktor	Reaksi Sistem

-	-

Tabel 3.10 Skenario *Use Case* Lihat *Adapter*

Identifikasi	
Nama Use Case	Lihat *Adapter*
Aktor	Sistem
Tujuan	Memastikan *adapter* yang di *monitoring*
Skenario	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan *adapter*
-	
Skenario Alternatif	
Aktor	Reaksi Sistem
-	-

Tabel 3.11 Skenario *Use Case* Lihat IP *Address*

Identifikasi	
Nama Use Case	Lihat IP *Address*
Aktor	*User*
Tujuan	Melihat IP *Address* yang di *monitoring*
Skenario	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan IP *Address* yang di *monitoring*
2. Melihat IP *Address*.	
Skenario Alternatif	
Aktor	Reaksi Sistem
-	-

Tabel 3.12 Skenario *Use Case* Pilih *Start Monitor*

Identifikasi	
Nama Use Case	Pilih *Start Monitor*
Aktor	*User*
Tujuan	Memulai aplikasi untuk *capture* data
Skenario	
Aksi Aktor	Reaksi Sistem
1. Tekan tombol *Start Monitor*	
	2. Mengambil paket data yang melewati *adapter* yang sudah ditentukan.
Skenario Alternatif	
Aktor	Reaksi Sistem
-	-

Tabel 3.13 Skenario *Use Case* Lihat *Log Capture*

Identifikasi	
Nama Use Case	Lihat *Log Capture*
Aktor	*User*
Tujuan	Melihat hasil *capture*
Skenario	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan hasil *capture* pada *log*.
2. Lihat hasil *log capture*.	
Skenario Alternatif	
Aktor	Reaksi Sistem
-	-

Tabel 3.14 Skenario *Use Case* Pilih *Stop Monitor*

Identifikasi	
Nama Use Case	Pilih *Stop Monitor*
Aktor	*User*
Tujuan	Menghentikan aplikasi untuk *capture* data
Skenario	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan tombol *Stop Monitor*
2. Tekan tombol *Stop Monitor*	
	3. *Capture* dihentikan
Skenario Alternatif	
Aktor	Reaksi Sistem
-	-

Tabel 3.15 Skenario *Use Case* Pilih *Clear log*

Identifikasi	
Nama Use Case	Pilih *Clear log*
Aktor	*User*
Tujuan	Menghapus hasil *capture* pada *log*
Skenario	
Aksi Aktor	Reaksi Sistem
1. Tekan Tombol *Clear log*	
	2. Isi *log* dihapus.
Skenario Alternatif	
Aktor	Reaksi Sistem
-	-

Tabel 3.16 Skenario *Use Case* Mengelola OverByteICS

Identifikasi	
Nama Use Case	Dikelola OverByteICS
Aktor	Sistem
Tujuan	Memberi informasi yang dibutuhkan *user* pada aplikasi
Skenario	
Aksi Aktor	Reaksi Sistem
	1. Memberi setiap informasi dan menjalankan kebutuhan yang diperintahkan oleh *user* terhadap aplikasi.
-	-
Skenario Alternatif	
Aktor	Reaksi Sistem
-	-

Tabel 3.17 Skenario *Use Case* Lihat *Traffic*

Identifikasi	
Nama Use Case	Lihat *Traffic*
Aktor	*User*
Tujuan	Melihat aplikasi *traffic*
Skenario	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan *form network traffic*
2. Melihat *form network traffic*	
Skenario Alternatif	
Aktor	Reaksi Sistem
-	-

Tabel 3.18 Skenario *Use Case* Pilih IP *Address*

Identifikasi	
Nama Use Case	Pilih IP *Address*
Aktor	*User*
Tujuan	Memilih IP *Address* untuk *monitoring traffic*
Skenario	
Aksi Aktor	Reaksi Sistem
1. Memilih IP *address*.	
	2. Menetapkan IP *address*.
Skenario Alternatif	
Aktor	Reaksi Sistem
-	-

Tabel 3.19 Skenario *Use Case* Lihat *Adapter Description*

Identifikasi	
Nama Use Case	Lihat *Adapter Description*
Aktor	Sistem
Tujuan	Melihat *Adapter* yang akan digunakan untuk *monitoring traffic*
Skenario	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan *adapter description*
2. Melihat *adapter* yang akan digunakan untuk *monitoring traffic*	
Skenario Alternatif	
Aktor	Reaksi Sistem
-	-

Tabel 3.20 Skenario *Use Case* Lihat *Speed*

Identifikasi	
Nama Use Case	Lihat *Speed*
Aktor	Sistem
Tujuan	Menampilkan *Speed* Maksimum *Adapter*.
Skenario	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan deskripsi *speed* maksimum pada *adapter*.
2. Melihat keterangan *speed adapter*.	
Skenario Alternatif	
Aktor	Reaksi Sistem
-	-

Tabel 3.21 Skenario *Use Case* Lihat *Traffic Download*

Identifikasi	
Nama Use Case	Lihat *Traffic Download*
Aktor	Sistem
Tujuan	Menampilkan hasil *download traffic, peak rate, average* dan total.
Skenario	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan hasil *download traffic*.
2. Melihat hasil *download transfer rate, peak rate, avarege* dan total.	
Skenario Alternatif	
Aktor	Reaksi Sistem
-	-

Tabel 3.22 Skenario *Use Case* Lihat *Traffic Upload*

Identifikasi	
Nama Use Case	Lihat Traffic Upload
Aktor	Sistem
Tujuan	Menampilkan hasil *upload traffic*, *peak rate*, *average* dan total.
Skenario	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan hasil *upload traffic*.
2. Melihat hasil *upload transfer rate*, *peak rate*, *avarege* dan total.	
Skenario Alternatif	
Aktor	Reaksi Sistem
-	-

Tabel 3.23 Skenario *Use Case* Pilih *Freeze*

Identifikasi	
Nama Use Case	Pilih *Freeze*
Aktor	*User*
Tujuan	Menghentikan sementara *monitoring traffic*.
Skenario	
Aksi Aktor	Reaksi Sistem
1. Menekan tombol *Freeze*	
	2. Menghentikan sementara monitoring *traffic*
Skenario Alternatif	
Aktor	Reaksi Sistem
-	-

Tabel 3.24 Skenario *Use Case* Pilih *UnFreeze*

Identifikasi	
Nama Use Case	Pilih *UnFreeze*
Aktor	*User*
Tujuan	Menjalankan kembali *monitoring network traffic*
Skenario	
Aksi Aktor	Reaksi Sistem
1. Menekan tombol *UnFreeze*	
	2. Menjalankan kembali *monitoring traffic*
Skenario Alternatif	
Aktor	Reaksi Sistem
-	-

Tabel 3.25 Skenario *Use Case* Dikelola IPHelper & IPHLPAPI

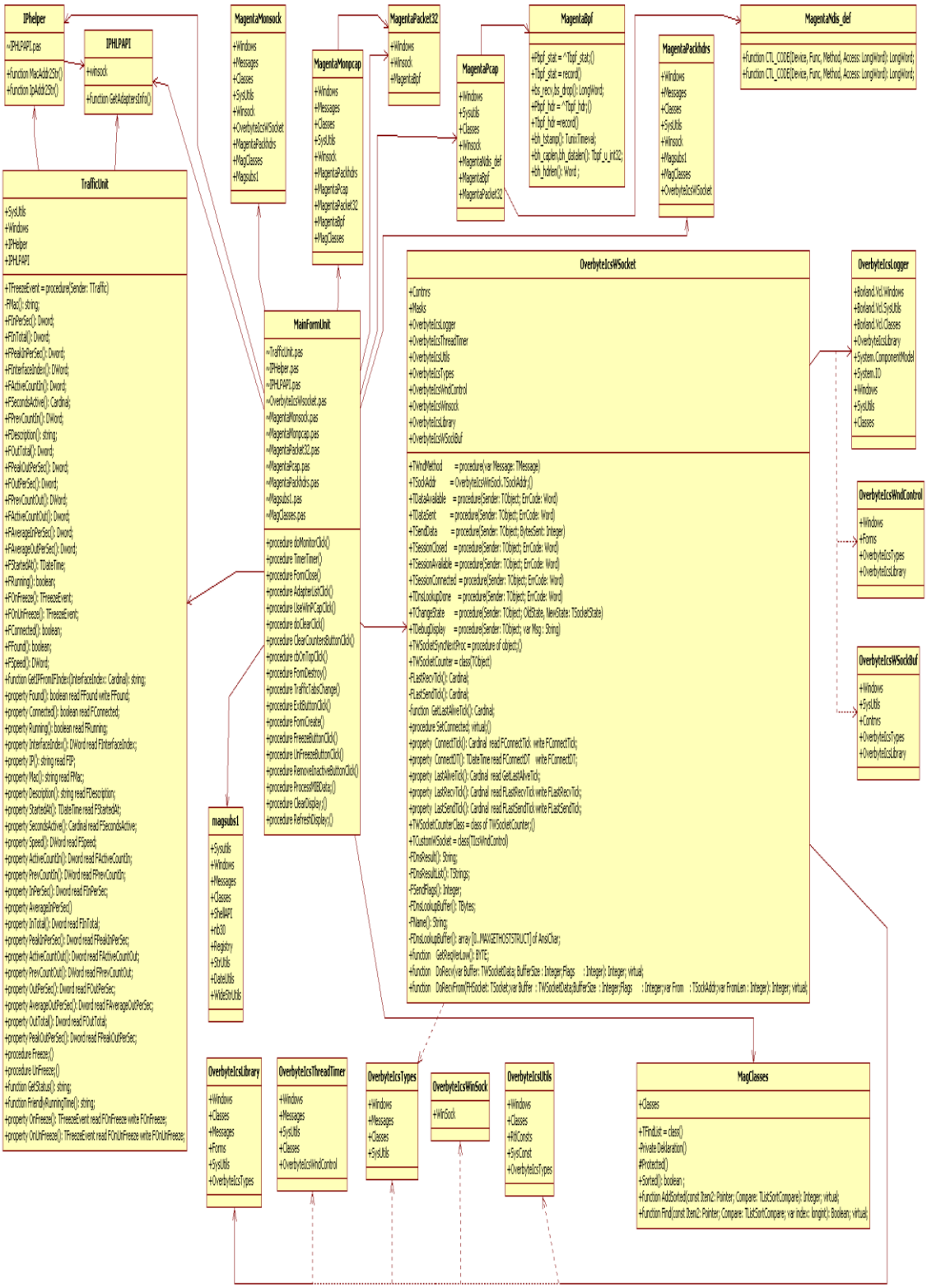
Identifikasi	
Nama Use Case	Dikelola IPHelper & IPHLPAPI
Aktor	Sistem
Tujuan	Memberi informasi yang dibutuhkan *user* pada aplikasi
Skenario	
Aksi Aktor	Reaksi Sistem
	1. Memberi setiap informasi dan menjalankan kebutuhan yang diperintahkan oleh *user* terhadap aplikasi.

Tabel 3.26 Skenario *Use Case* Pilih *Exit*

Identifikasi	
Nama Use Case	Pilih *Exit*
Aktor	*User*
Tujuan	Menutup aplikasi
Skenario	
Aksi Aktor	Reaksi Sistem
1. Menekan tombol *Exit*	
	2. Menutup aplikasi
Skenario Alternatif	
Aktor	Reaksi Sistem
-	-

2. *Class Diagram*

Class Diagram menggambarkan keadaan suatu sistem (*atribut*), dan memberikan pelayanan untuk menyelesaikan keadaan tersebut. Pada gambar 3.6 digambarkan *class* diagram dari sistem aplikasi *network traffic*.



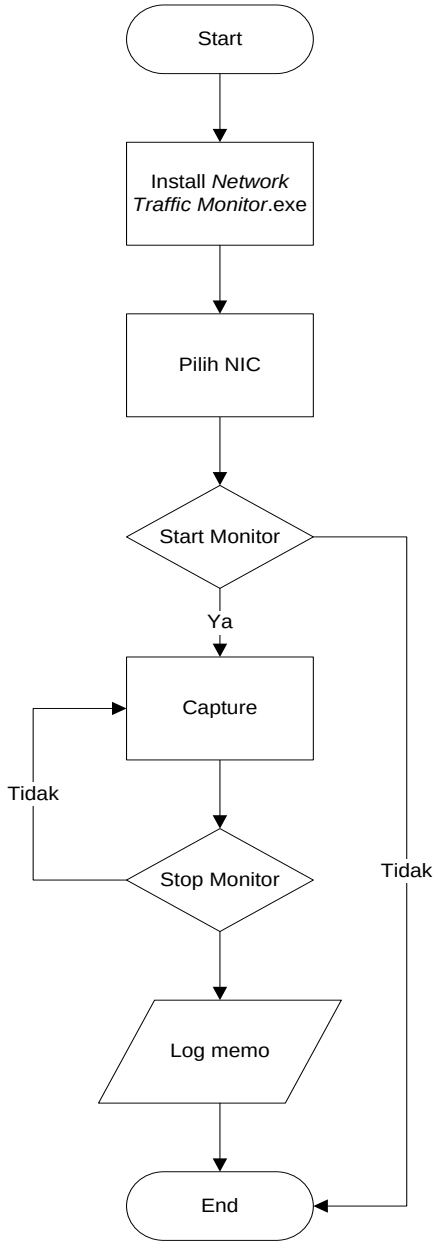
Gambar 3.6 *Class* Diagram *Network Traffic* dan *Packet Sniffer*

Tabel 3.27 Deskripsi *class* diagram *Network Traffic* dan *Packet Sniffer*

Class	Jenis Kelas	Deskripsi
MainFormUnit	*Interface*	*Class* Utama yang menampilkan *Network Traffic* dan *Packet Sniffer*
IPHelper	*Control Network Traffic*	*Class* yang bertugas untuk pengenalan alamat IP dan MAC.
IPHLPAPI	*Control Network Traffic*	*Class* yang bertugas untuk pengenalan struktur alamat IP dan MAC.
TrafficUnit	*Control Network Traffic*	*Class* yang berisikan perumusan perhitungan *traffic*.
OverbyteIcsLibrary	*Control Packet Sniffer*	*Class* yang bertugas atas fungsional mengaktifkan pengenalan perintah dalam pemrograman.
OverbyteIcsLogger	*Control Packet Sniffer*	*Class* yang bertugas melakukan pencatatan hasil *capture*.
OverbyteIcsThreadTimer	*Control Packet Sniffer*	*Class* yang berisikan pengaturan waktu dan *trigger* saat *capture* data.
OverbyteIcsTypes	*Control Packet Sniffer*	*Class* yang berisikan pendeklarasian pada tipe data.
OverbyteIcsUtils	*Control Packet Sniffer*	*Class* yang berisikan perhitungan konversi

		setiap tipe data.
OverbyteIcsWinsock	*Control Packet Sniffer*	*Class* yang berisikan pemanggilan fungsional yang berkaitan dengan aktivitas jaringan atau untuk mendukung Microsoft NET *framework*.
OverbyteIcsWndControl	*Control Packet Sniffer*	*Class* yang berisikan kontrol pada TIcsWndHandlerPool untuk mendaftarkan pesan yang dibutuhkan.
OverbyteIcsWSockBuf	*Control Packet Sniffer*	*Class* yang bertugas sebagai tempat antrian dengan menggunakan metode antrian FIFO (*First in First Out*)
OverbyteIcsWsocket	*Control Packet Sniffer*	*Class* yang bertugas mendukung TCP, UDP, SOCKS5 dan dapat digunakan untuk membangun program klien dan *server*.

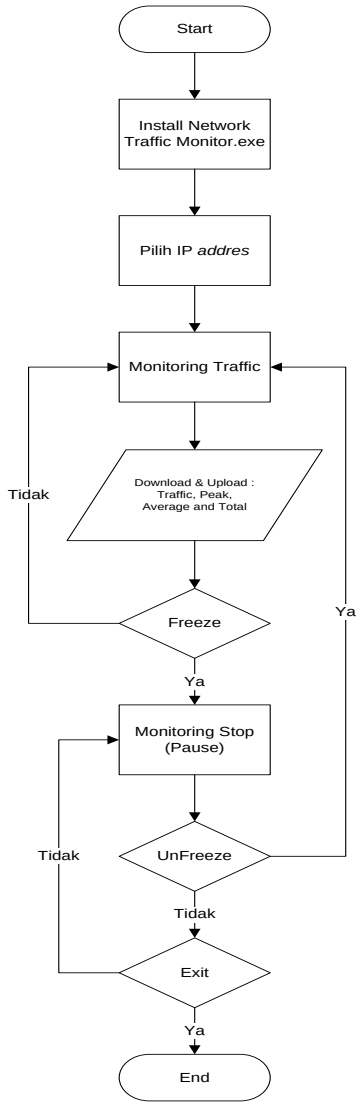
3.2.2 *Flowchat Packet Sniffer*



Gambar 3.7 *Flowchart Packet Sniffer*

Berdasarkan gambar 3.7, ketika aplikasi dijalankan list NIC akan ditampilkan, pilih salah satu *network interface* yang akan di-*capture*, tekan tombol *Start Monitor* untuk memulai *capture* data. Jika ingin menghentikan *capture* tekan tombol *Stop Monitor*, untuk *log* dari setiap *capture* data dilihat pada *log memo*.

3.2.3 *Flowchart Network Traffic*



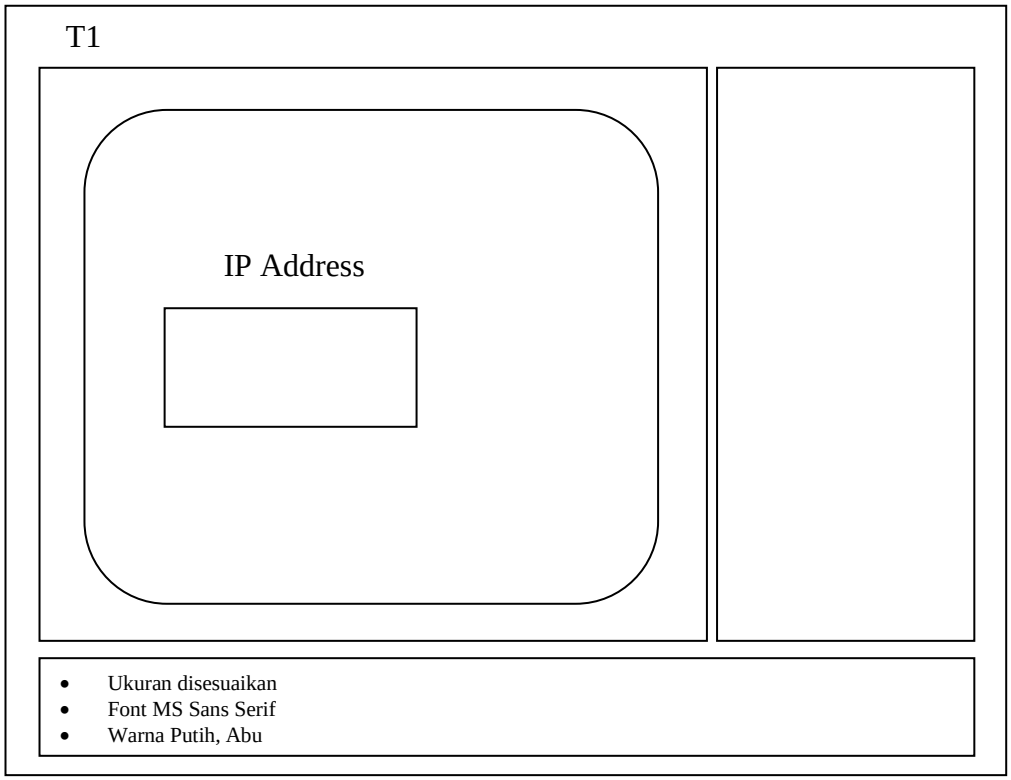
Gambar 3.8 *Flowchart Network Traffic Monitor*

Berdasarkan gambar 3.8, ketika aplikasi dijalankan pada tab akan ditampilkan list alamat IP, pilih satu alamat IP maka secara otomatis informasi *traffic* akan ditampilkan. Jika ingin menghentikan informasi *traffic* sementara tekan tombol *Freeze* dan tekan tombol *UnFreeze* untuk menjalankan kembali informasi *traffic*. Jika informasi *traffic* sudah cukup atau ingin keluar dari aplikasi maka tekan tombol *exit* sebagai perintah menutup aplikasi.

3.2.4 Perancangan Antarmuka

Perancangan antarmuka bertujuan memberikan gambaran tentang isi pada aplikasi yang akan dirancang, sehingga mudah dalam mengimplementasikan aplikasi serta mudah dalam melakukan perancangan. Berikut merupakan perancangan antarmuka pada aplikasi *network traffic*:

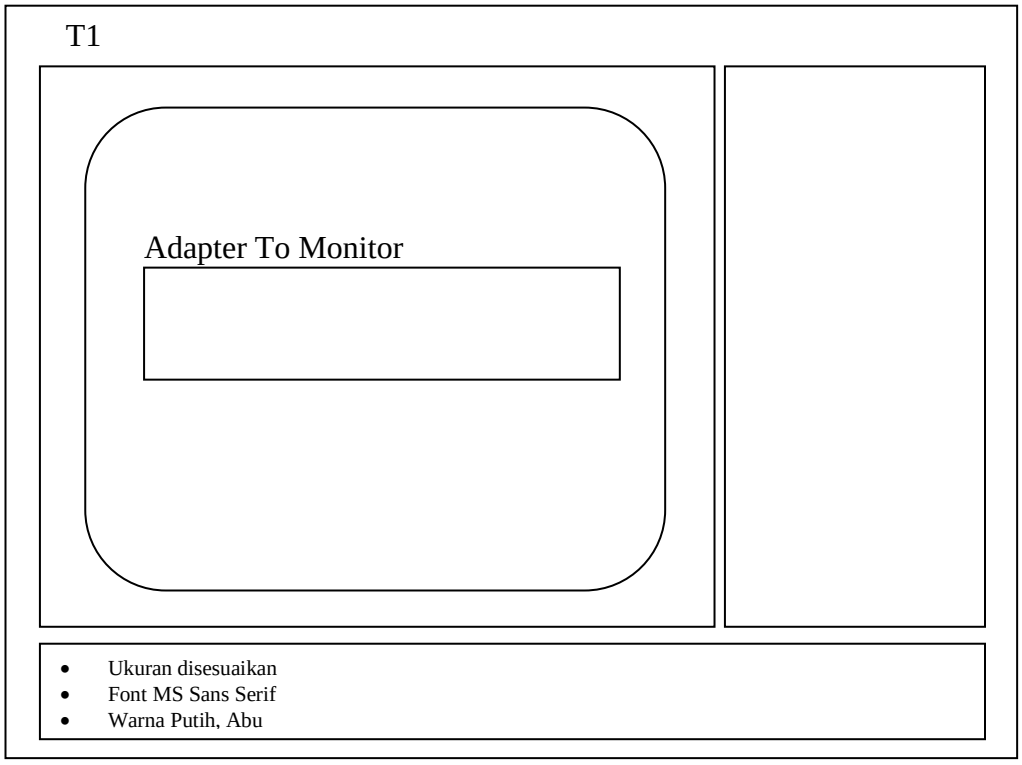
1. Antarmuka IP *Address*



Gambar 3.9 Antarmuka IP *Address*

IP *address list box* berfungsi menampilkan IP *address* yang terbaca secara *default* yang merupakan IP *address network interface card* yang digunakan.

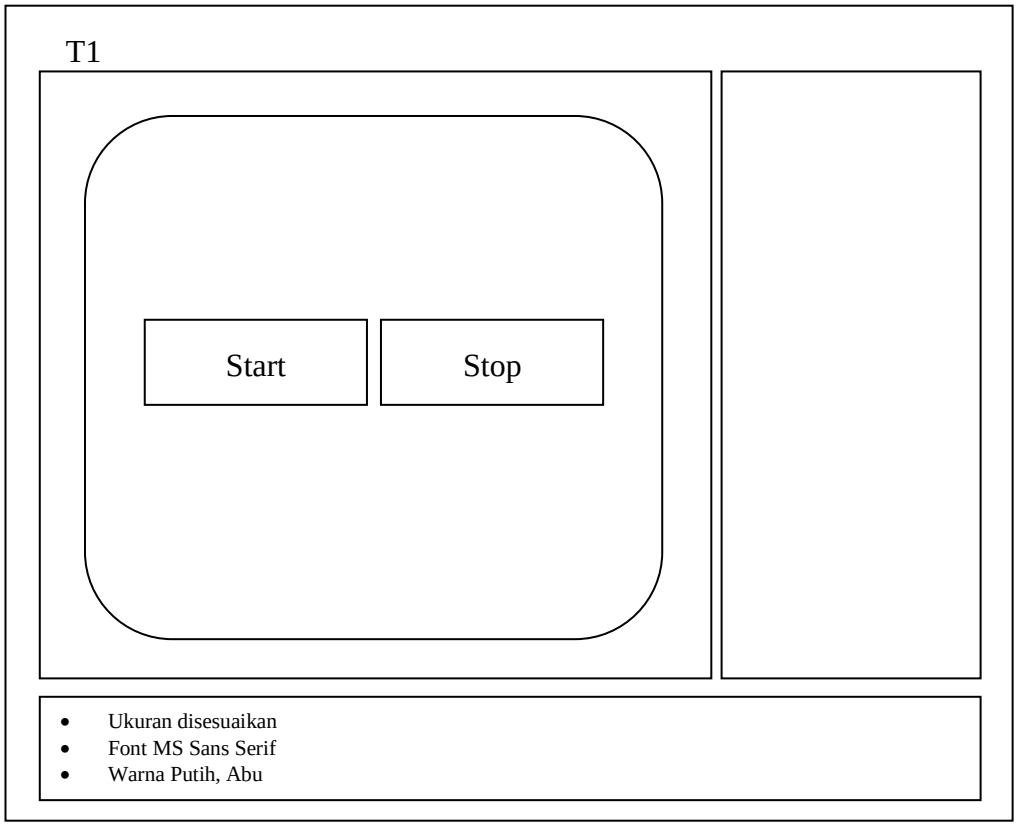
2. Antarmuka NIC



Gambar 3.10 Antarmuka Network Interface Card (NIC)

Adapter to monitor List Box berfungsi untuk memilih NIC yang akan dilakukan *monitoring* dan menampilkan setiap *network interface card* yang di *install*.

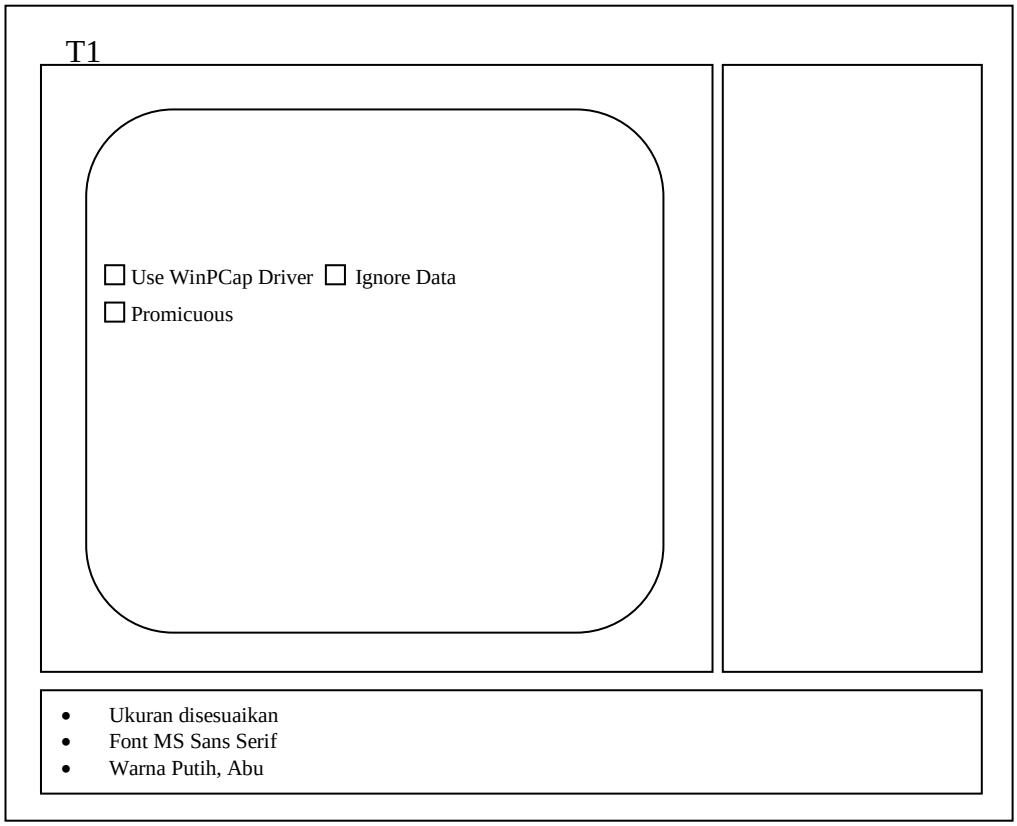
3. Antarmuka *Start Monitor* dan *Stop Monitor*



Gambar 3.11 Antarmuka *Start/Stop*

Start Monitor Button berfungsi untuk memulai *capture packet* yang sudah terhubung dalam jaringan. *Stop Monitor Button* akan muncul ketika *Start Monitor Button* ditekan dan berfungsi untuk menghentikan *capture packet* yang berlangsung.

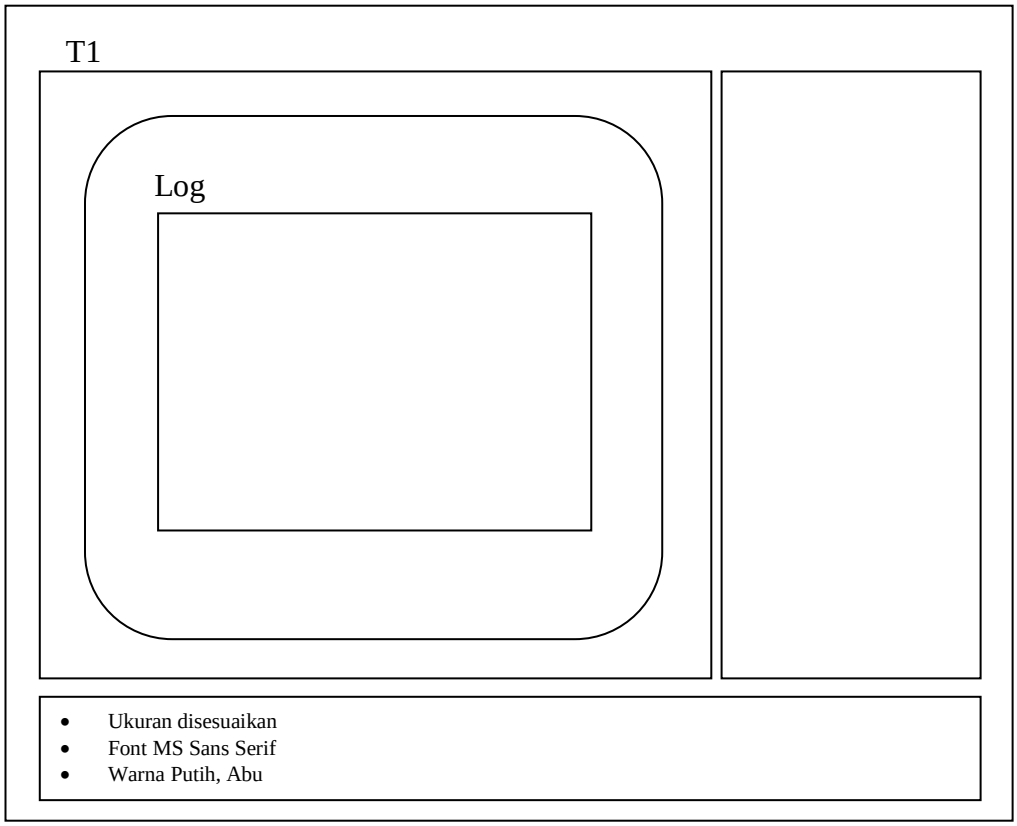
4. Antarmuka WinPcap *Driver*, *Promicuous*, dan *Ignore Data*.



Gambar 3.12 Antarmuka *chekbox*

WinPCap Driver checkbox berfungsi untuk mengintegrasikan NIC driver yang sudah ter-*install* pada operasi sistem dengan program yang dirancang agar dapat dilakukan pengontrolan terhadap NIC. *Promicouos checkbox* berfungsi sebagai mode *listening*. *Ignore data* berfungsi untuk menghiraukan data yang di-*capture* untuk tidak ditampilkan.

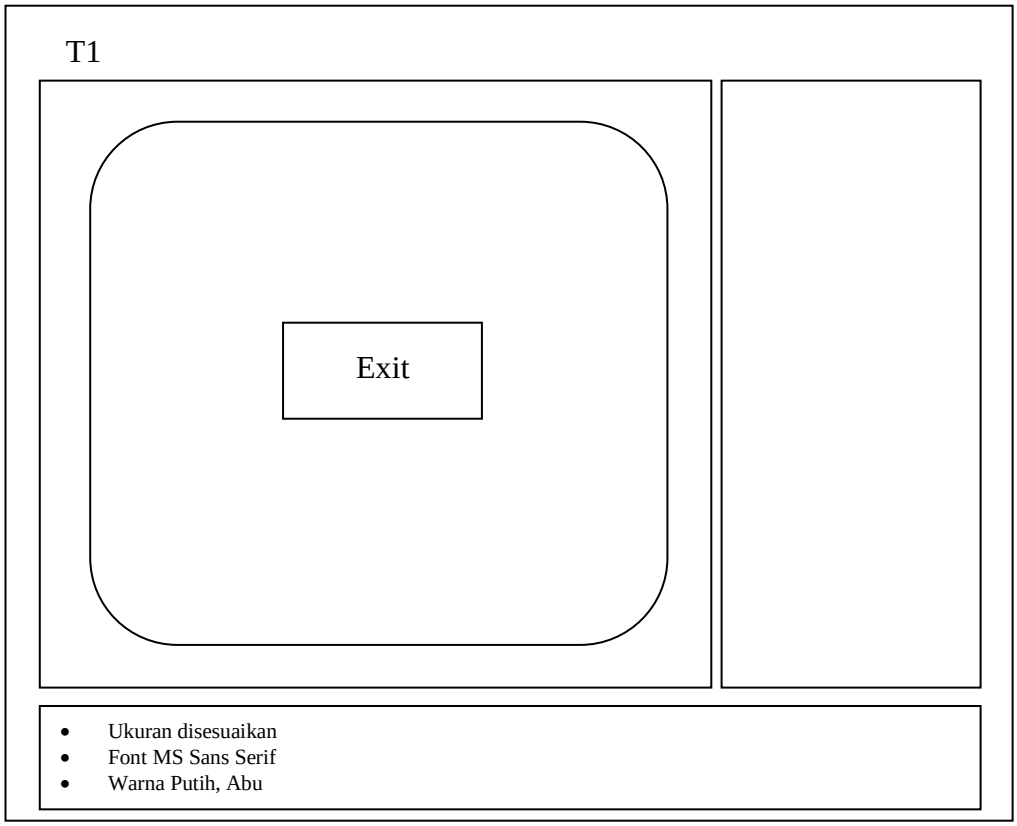
5. Antarmuka *Report*



Gambar 3.13 Antarmuka *Memo log*

Memo log berfungsi menampilkan seluruh aktivitas jaringan pada saat *capture packet* yang berlangsung. Menampilkan IP *address* sumber dan tujuan, protokol, besar paket, besar data dan data.

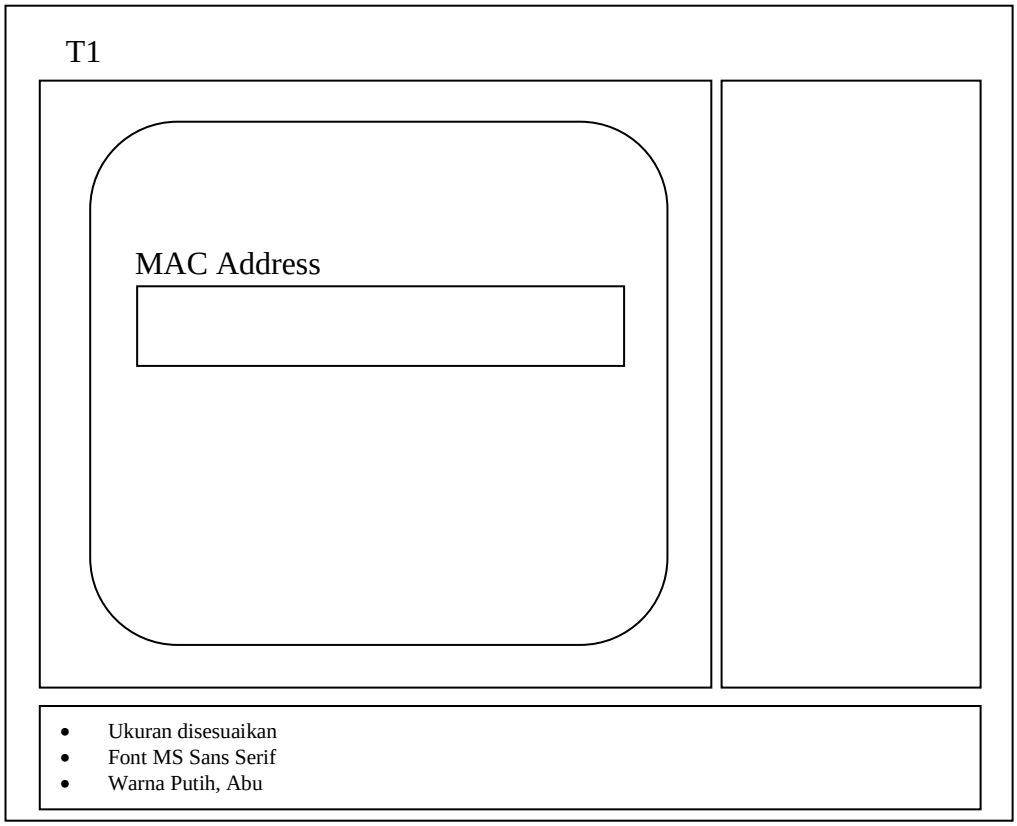
6. Antarmuka *Exit*



Gambar 3.14 Antarmuka *Exit*

Exit Button berfungsi menghentikan kerja program secara keseluruhan dan menghilangkan GUI dari tampilan operasi sistem.

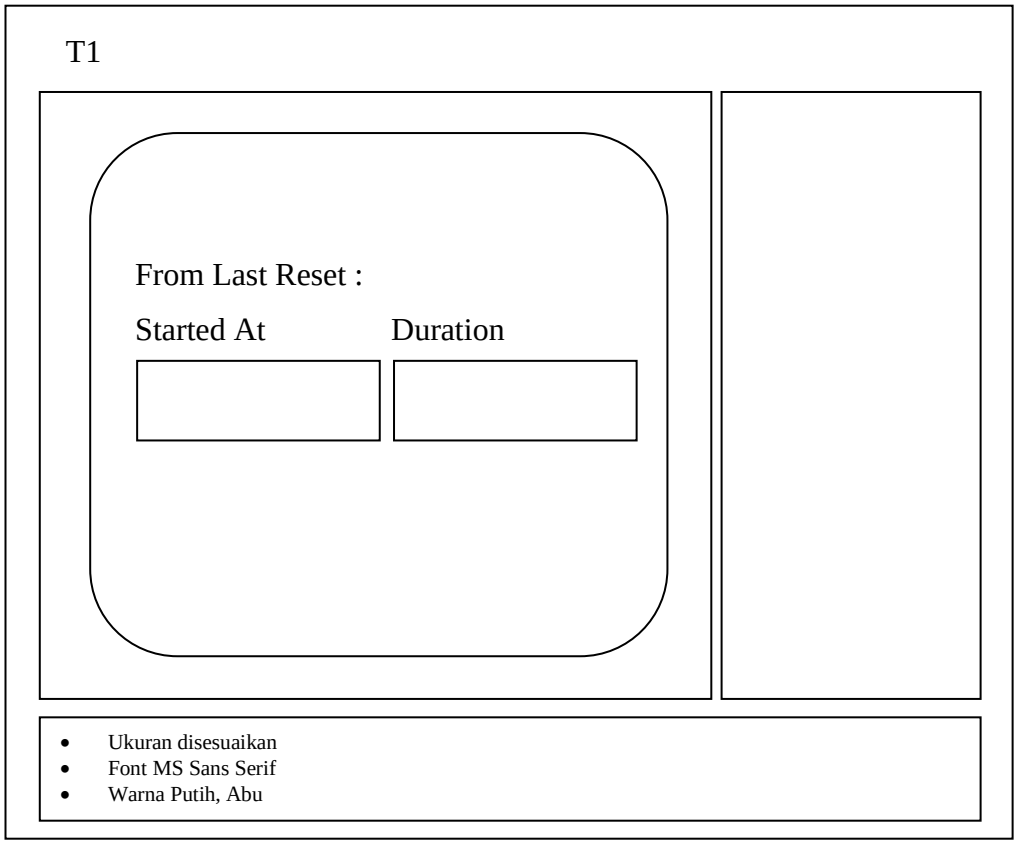
7. Antarmuka MAC *Address*



Gambar 3.15 Antarmuka MAC Address

Media Control Access Address merupakan alamat unik yang memiliki panjang 48-bit (6 byte) yang mengidentifikasikan sebuah komputer, *interface* dalam sebuah *router*, atau *node* lainnya dalam jaringan.

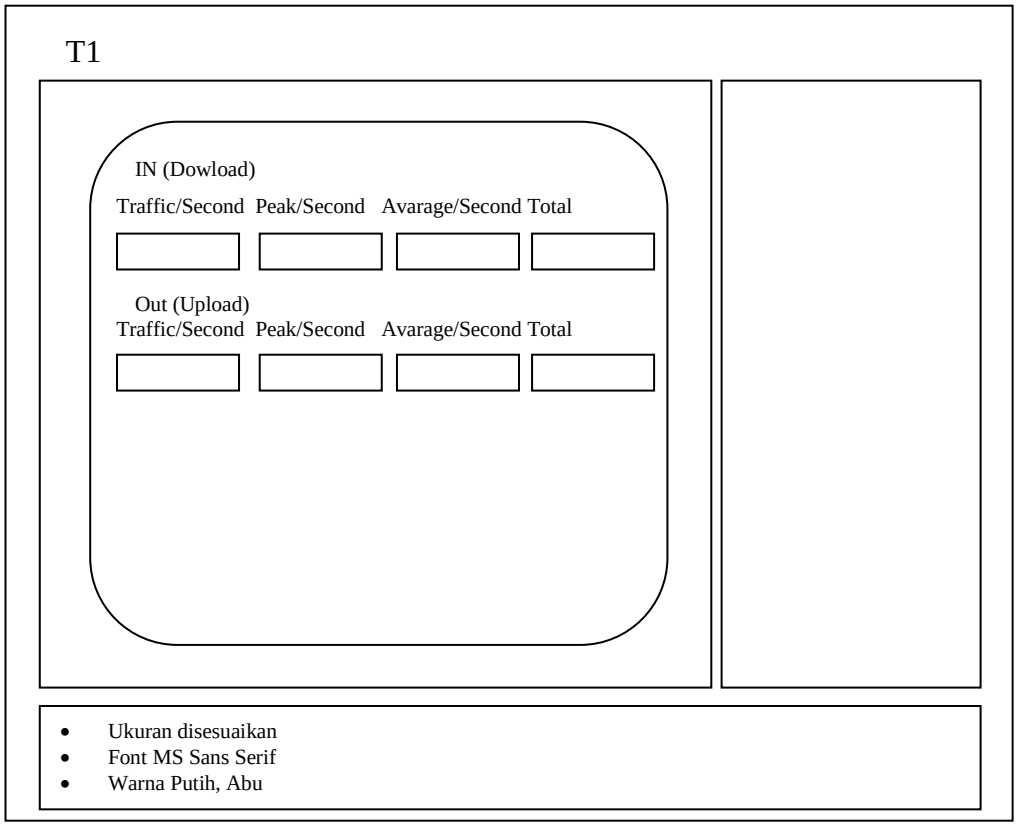
8. Antarmuka *Started* dan *Duration*



Gambar 3.16 Antarmuka *Started* dan *Duration*

Started At menampilkan waktu ketika program pertama kali berjalan dan *duration* menampilkan lama waktu program ketika berjalan.

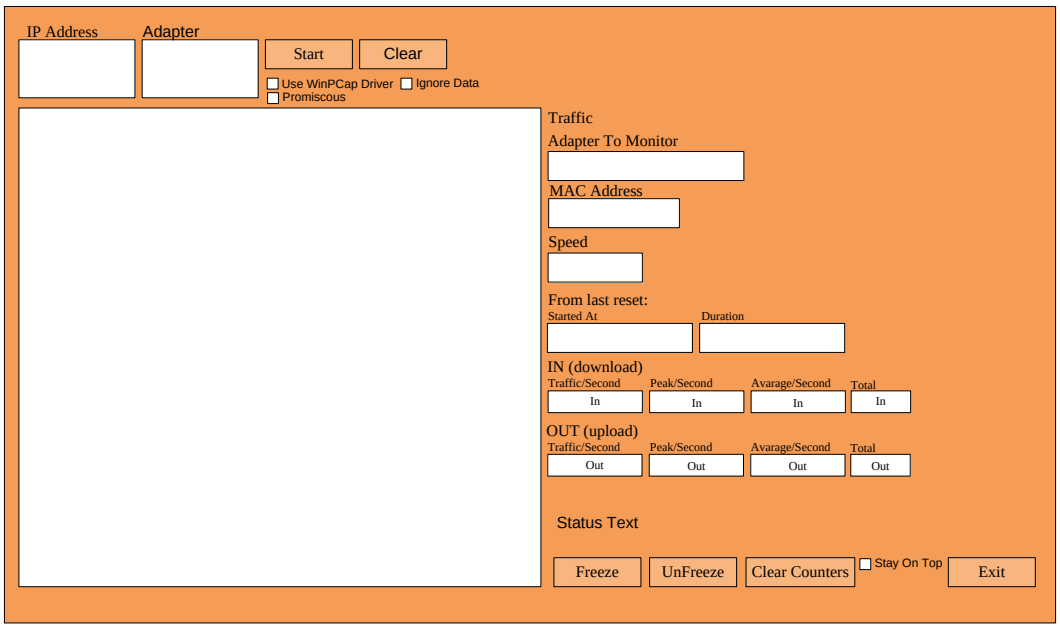
9. Antarmuka Total *traffic*



Gambar 3.17 Antarmuka Total *traffic*

Total *traffic* menampilkan banyaknya selama program sudah terhubung dalam jaringan. *Traffic* menampilkan besar data yang di-*capture* oleh NIC. *peak* menampilkan nilai puncak besar data yang masuk atau keluar, sedangkan *average* adalah besar rata-rata data yang masuk atau keluar.

10. Antarmuka T1 Keseluruhan (*main* program)



Gambar 3.18 Antarmuka *Main Program*

Main program menunjukan sebagai antarmuka keseluruhan dari fungsionalitas *network traffic* dan *packet sniffer*.