SKRIPSI

PENGEMBANGAN VISUALISASI PETA JARINGAN KOMPUTER DENGAN MEMANFAATKAN PROTOKOL JARINGAN MODEL REFERENSI OSI PADA LAPISAN KE 3

Disusun Oleh :

WIWIT AGIT WIBAKSONI

04.12.651

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG

2009

LEMBAR PERSETUJUAN

PENGEMBANGAN VISUALISASI PETA JARINGAN KOMPUTER DENGAN MEMANFAATKAN PROTOKOL JARINGAN MODEL REFERENSI OSI PADA LAPISAN KE 3

SKRIPSI

Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Teknik Komputer Dan Informatika Strata Satu (S-1)

Disusun Oleh :

WIWIT AGIT WIBAKSONI
NIM : 04.12.651

Diperiksa dan Disetujui

Dosen Pembimbing I	Dosen Pembimbing II
Dr. Cahyo Crysdian.	**Soetohadi, ST.**
NIP.Y.1030400412	**NIP.Y.1039700309**

Mengetahui
Ketua Jurusan Teknik Elektro S-1

Ir. F. Yudi Limpraptono, MT.
NIP.Y.1039500274

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2009

PENGEMBANGAN VISUALISASI PETA JARINGAN KOMPUTER DENGAN MEMANFAATKAN PROTOKOL JARINGAN MODEL REFERENSI OSI PADA LAPISAN KE 3

Wiwit Agit Wibaksoni

Konsentrasi Komputer dan Informatika, Jurusan Teknik Elektro S-1
Fakultas Teknologi Industri, Institut Teknologi Nasional Malang
Jln. Raya Karanglo Km 2 Malang
wibaksoni@yahoo.com

Dosen Pembimbing : I. Dr. Cahyo Crysdian.
II. Sotyohadi, ST.

Abstraksi

Perkembangan teknologi informasi menyebabkan semakin banyaknya penguna jaringan komputer, hal ini menuntuk para administrator jaringan untuk meningkatkan pengelolaan pada jaringan tersebut. Peta jaringan komputer merupakan salah satu faktor penting untuk mengoptimalkan pengelolaan terhadap suatu jaringan komputer yang kompleks. Dengan memanfaatkan *Internet Protocol* (IP) dan *Internet Control Message Protocol* (ICMP) dapat dihasilkan suatu mekanisme untuk mendapatkan informasi tentang keadaan sebuah jaringan komputer, sehingga dengan informasi tersebut dapat didesain sebuah aplikasi yang dapat menyajikanya dalam bentuk visual. Aplikasi **NetVis** adalah hasil dari implementasi mekanisme tersebut, aplikasi ini dapat menampilkan peta jaringan komputer secara detail, baik intranet maupun lintasan ke internet serta dilengkapi dengan aplikasi pendukung yaitu *port scanner* dan *remote command*.

Kata Kunci: Visualisasi jaringan komputer,peta jaringan, ICMP, IP, aplikasi NetVis.

KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa yang telah memberikan rahmat dan hidayah-Nya, sehingga dapat diselesaikan skripsi yang berjudul "Pengembangan Visualisasi Peta Jaringan Komputer Dengan Memanfaatkan Protokol Jaringan Model Referensi OSI Pada Lapisan Ke 3" ini dengan lancar. Skripsi ini merupakan persyaratan kelulusan Studi pada Jurusan Teknik Elektro S-1 Konsentrasi Teknik Komputer dan Informatika ITN Malang dan untuk mencapai gelar Sarjana Teknik.

Keberhasilan peyelesaian laporan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penyusun menyampaikan terima kasih kepada :

1. Prof. Dr. Ir. Abraham Lomi, MSEE, selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. F. Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro S-1.
3. Bapak Dr. Cahyo Crysdian selaku Dosen Pembimbing I.
4. Bapak Sotyohadi, ST. selaku Dosen Pembimbing II.
5. Ayah dan Ibu serta saudara-saudara kami yang telah memberikan do'a restu, dorongan, semangat, dan biaya.
6. Rekan-rekan Instruktur di Laboratorium Jaringan Komputer ITN Malang.
7. Semua yang telah membantu dalam penyelesaian penyusunan skripsi ini.

Penulis telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelasaikan laporan ini. Untuk itu penyusun mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

Harapan penyusun semoga laporan skripsi ini memberikan manfaat bagi perkembangan ilmu pengetahuan dan pembaca.

Malang, Maret 2009

penyusun

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL


INSTITUT TEKNOLOGI NASIONAL
ITN
MALANG

BAB I

PENDAHULUAN

1.1. Latar Belakang.

Pemanfaatan teknologi jaringan komputer telah berkembang dengan sangat cepat, hampir semua instansi di dunia telah memanfaatkan teknologi ini sebagai pendukung dari perkembangan teknologi informasi yang mereka gunakan. Tidak hanya untuk keperluan internet, penggunaan jaringan komputer dalam skala lokal juga telah semakin banyak, sehingga menyebabkan meningkatnya skala, jumlah *node* maupun teknologi yang digunakan pada jaringan komputer. Hal ini menunjukkan bahwa pengguna jaringan komputer semakin banyak seiring perkembangan teknologi informasi saat ini. Perkembangan ini seakan membuat dunia semakin sempit sehingga komunikasi terasa sangat cepat, dimana informasi pada suatu tempat dapat diketahui oleh seseorang yang berada pada tempat yang sangat jauh dalam waktu yang cukup singkat.

Perkembangan ini menuntut para administrator jaringan komputer untuk meningkatkan dan mengoptimalkan jaringan komputer. Dengan bertambah besarnya skala jaringan komputer maupun semakin banyaknya jumlah *node* memerlukan perhatian yang semakin besar, Luasnya skala jaringan komputer menunjukkan banyaknya *host* (komputer) yang ada pada jaringan tersebut, sedangkan banyaknya jumlah *node* memungkinkan semakin banyaknya rute yang dilewati suatu *host* untuk sampai ke tujuannya. Dengan demikian sangat dibutuhkan pengelolaan jaringan baik dan optimal, sehingga dampak negatif maupun merugikan dari berkembangnya jaringan komputer dapat dicegah.

Untuk mendukung pengelolaan jaringan komputer yang saat ini semakin meningkat, dibutuhkan suatu perangkat yang dapat membantu penanganan suatu jaringan komputer tersebut. Peta jaringan komputer adalah alat penting untuk mengoptimalkan pengelolaan terhadap suatu jaringan komputer. Namun demikian untuk mendapatkan peta jaringan komputer bukanlah hal yang mudah, terutama bila jaringan yang akan dipetakan sudah terimplementasi secara luas. Oleh karena itu perlu adanya mekanisme untuk dapat memetakan suatu jaringan komputer. Adanya mekanisme visualisasi dari rute yang terbangun juga sangat diperlukan untuk keperluan ini.

1.2. Rumusan Masalah.

Berdasarkan latar belakang diatas, permasalahan yang akan dibahas adalah sebagai berikut :

1. Bagaimana membangun sebuah aplikasi yang dapat memberikan informasi secara visual tentang peta jaringan komputer yang sedang digunakan pada suatu tempat secara otomatis.
2. Bagaimana *remote* akses sebuah *node* diterapkan dalam visualisasi untuk mendapatkan informasi secara lengkap dan mudah.

1.3. Tujuan Penelitian.

Adapun tujuan yang diharapkan adalah untuk mendesain sebuah aplikasi untuk mandapatkan informasi peta jaringan komputer dalam bentuk visual, serta menerapkan aplikasi telnet dan *port sacanner* dalam upaya untuk mendukung pengelolahan sebuah jaringan komputer.

1.4. Batasan Masalah.

Agar permasalahan mengarah sesuai dengan dengan tujuan yang diharapkan, maka pembahasan dibatasi oleh hal – hal sebagai berikut:

1. Protokol yang digunakan pada model referensi *Open System Interconnection* (OSI) lapisan ke 3 adalah *Internet Protocol* (IP) dan *Internet Control Message Protokol* (ICMP).
2. Lingkup dari pengembangan visualisasi peta jaringan komputer ini dibatasi pada visualisasi peta jaringan intranet sampai pada hop ke 2.
3. Remote akses yang diimplementasikan adalah bebasis *command dos* yang dapat digunakan pada setiap komputer maupun *node* yang berada dalam jangkauan administrator.
4. Tidak membahas teknologi jaringan dan sistem keamanan pada masing-masing *node* yang didapatkan.
5. Pengembangan Visualisasi Peta Jaringan Komputer dibuat dengan menggunakan bahasa pemrograman Borland Delphi 7.

1.5. Metode Penelitian.

Adapun metode penelitian yang digunakan adalah sebagai berikut:

1. Studi literatur

   Pengumpulan data yang dilakukan dengan mencari bahan-bahan kepustakaan dan referensi dari berbagai sumber sebagai landasan teori yang ada hubunganya dengan permasalahan yang dijadikan objek penelitian.

2. Analisa Kebutuhan Sistem

Data dan informasi yang telah diperoleh akan dianalisa agar didapatkan kerangka global yang bertujuan untuk mendefinisikan kebutuhan sitem baik hardware maupun software, di mana nantinya akan digunakan sebagai acuan perancangan sistem.

3. Perancangan sistem

Berdasarkan data dan informasi yang telah diperoleh serta analisa kebutuhan untuk membangun sistem ini, akan dibuat rancangan kerangka global yang menggambarkan mekanisme dari sistem yang akan dibuat.

4. Coding

Tahapan ini menerjemahkan hasil perancangan spesifikasi program dari tahapan sebelumnya kedalam baris–baris kode program yang dapat dimengerti oleh komputer.

5. Eksperimen dan Evaluasi

Pada tahap ini, sistem yang telah selesai dibuat akan diuji coba, yaitu pengujian berdasarkan fungsionalitas program, dan akan dilakukan koreksi dan penyempurnaan program jika diperlukan.

1.6. Sistematika Penulisan.

Untuk mempermudah dan memahami pembahasan penulisan skripsi ini, maka sistematika penulisan disusun sebagai berikut :

Bab I : Pendahuluan

Berisi Latar Belakang, Rumusan Masalah, Tujuan Penelitian, Pembatasan Permasalahan, Metode Penelitian dan Sistematika Penulisan.

Bab II : Tinjauan Pustaka

Berisi tentang landasan teori mengenai permasalahan yang berhubungan dengan penelitian yang dilakukan.

Bab III : Perancangan dan Analisa Sistem

Dalam bab ini berisi mengenai analisa kebutuhan sistem baik software maupun hardware yang diperlukan untuk membuat kerangka global yang menggambarkan mekanisme dari sistem yang akan dibuat. .

Bab IV : Pembuatan dan Pengujian Sistem

Berisi tentang implementasi dari perancangan sistem yang telah dibuat serta pengujian terhadap sistem tersebut.

Bab V : Penutup

Merupakan bab terakhir yang memuat intisari dari hasil pembahasan yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya


INSTITUT TEKNOLOGI NASIONAL
ITN
MALANG

BAB II

LANDASAN TEORI

2.1. Jaringan Komputer.

Jaringan komputer adalah sebuah kumpulan komputer yang saling berhubungan satu sama lain dengan menggunakan suatu protokol komunikasi melalui media komunikasi sehingga dapat saling berbagi informasi, aplikasi, *file*, serta penggunaan perangkat keras secara bersama seperti, hardisk, printer, scanner dan lain-lain. Untuk menghubungkan komputer - komputer tersebut dapat menggunakan berbagai macam media komunikasi sepert, kabel, gelombang radio, saluran telepon, satelit, maupun serat optik[1].

2.1.1. Jenis Jaringan Komputer.

Jenis - jenis jaringan komputer bardasarkan cakupan areanya adalah sebagai berikut [1]:

a. LAN (*Local Area Network*).

LAN merupakan jaringan dengan skala yang kecil LAN (dengan jarak antara 100m - 1km) sering kali digunakan untuk menghubungkan komputer-komputer pribadi dan *Workstation* dalam kantor atau pabrik schingga meraka dapat bertukar informasi dengan cepat dan saling *Sharing Resource* (printer, scanner dll). Keuntungan menggunakan LAN adalah:

- Akses Data relatif lebih cepat.
- Proses *Back Up* data dapat dilakukan dengan mudah.
- Dapat menghubungkan banyak komputer sekaligus ke internet.

b. MAN (*Metropolitan Area Network*).

Bila komputer yang saling berhubungan tidak dalam satu lokasi bahkan lokasinya sampai antar kota, tipe jaringan tersebut adalah *Metropolitan Area Network*. Jaringan komputer MAN merupakan jaringan komputer yang lebih besar dari LAN (rentang jarak ± 10 km).

c. WAN (*Wide Area Network*).

Wide Area Network adalah sebuah jaringan yang memiliki jaringan yang memiliki jarak yang sangat luas, karena radiusnya mencakup sebuah negara dan benua.

d. Internet

Internet adalah kumpulan jaringan yang memiliki jarak yang sangat luas dan tidak harus terikat dalam suatu organisasi namun *internet* adalah jaringan skala internasional yang dapat digunakan oleh semua orang dari berbagai penjuru dunia.

2.1.2. Protokol Jaringan Komputer.

Protokol pada jaringan komputer adalah aturan main yang mengatur komunikasi antara beberapa komputer yang terhubung di dalam sebuah jaringan. Pada aturan tersebut termasuk didalamnya adalah metode atau cara dalam mengakses jaringan, topologi fisik, tipe-tipe kabel dan kecepatan transfer data[2]. Berikut ini adalah beberapa jenis protokol pada jaringan komputer :

Tabel 2.1. Jenis protokol pada jaringan komputer[2].

NO	Nama Protokol	Topologi Fisik	Tipe Kabel	Kec Transfer
1	Ethernet	Linear Bus, Star, Tree	Twisted Pair, Coaxial, Fiber	10 Mbps
2	Fast Ethernet	Star	Twisted Pair, Fiber	100 Mbps
3	Local Talk	Linear Bus or Star	Twisted Pair	0.23 Mbps
4	Token Ring	Star-Wired Ring	Twisted Pair	4 Mbps-16Mbps
5	ATM	Linear Bus, Star, Tree	Twisted Pair, Fiber	100 Mbps
6	FDDI	Dual ring	Fiber	155-2488 Mbps

2.1.3. Topologi Jaringan Komputer.

Arsitektur fisik jaringan identik dengan toplogi yang akan digunakan dalam jaringan tersebut. Topologi adalah istilah yang digunakan untuk menguraikan cara bagaimana komputer terhubung dalam suatu jaringan[1]. Hal tersebut bertujuan agar apabila suatu saat jaringan tersebut ingin kita kembangkan menjadi suatu jaringan dengan skala lebih besar dan luas maka pemasangan maupun perawatan jaringan menjadi lebih mudah. Dengan adanya arsitektur fisik jaringan, pengguna jaringan dapat, menentukan topologi mana saja yang sesuai untuk digunakan dalam jaringannya. Adapun macam-macam dalam topologi tersebut adalah:

- Topologi BUS.
- Topologi Token Ring.
- Topologi Star.
- Topologi Mesh.
- Topologi tree/hybrid.

2.1.4. TCP/IP dan OSI Layer.

Agar dapat berkomunikasi antar berbagai macam vendor komputer diperlukan sebuah aturan baku yang standar dan disetujui berbagai pihak. Seperti halnya dua orang yang berlainan bangsa, maka untuk berkomunikasi memerlukan penerjemah/interpreter atau satu bahasa yang dimengerti kedua belah pihak. Dalam dunia komputer dan telekomunikasi interpreter identik dengan protokol. Untuk itu maka badan dunia yang menangani masalah standarisasi ISO (*International Standardization Organization*) membuat aturan baku yang dikenal dengan nama model referensi OSI (*Open System Interconnection*)[3].

Seiring dengan perkembangan teknologi *Departement of Defense* (DoD) membuat suatu standarisasi yang dikenal dengan nama *Transmission Control Protocol / Internet Protocol* (TCP/IP), TCP/IP merupakan versi pemadatan dari model referensi OSI yang hanya terdiri dari empat lapisan[4]. Namun demikian pada dasarnya kedua standarisasi tersebut memiliki konsep yang sama, namun jumlah dan nama tiap lapisan yang membedakan kedua standarisasi tersebut. Berikut ini adalah hubungan antara model referensi OSI dengan model TCP/IP :

Tabel 2.2. Hubungan antara model OSI dengan model TCP/IP[5].

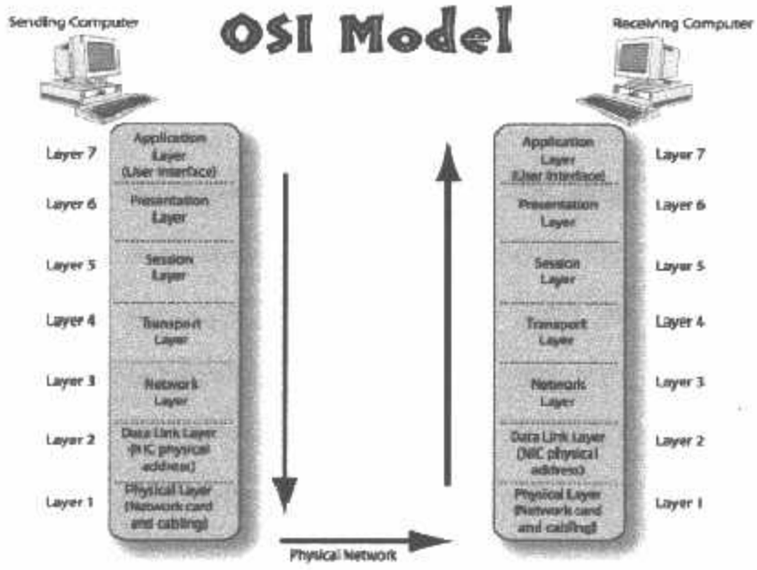
Model OSI		**TCP/IP**	**Nama Protokol**
No	**Lapisan**		
7	Aplikasi	Aplikasi	DHCP (*Dynamic Host Configuration Protocol*)
			DNS (*Domain Name Server*)
			FTP (*File Transfer Protocol*)
			HTTP (*Hyper Text Transfer Protocol*)
			MIME (*Multipurpose Internet Mail Extention*)
			NNTP (*Network News Transfer Protocol*)
			POP (*Post Office Protocol*)
			SMB (*Server Message Block*)

6	Presentasi		SMTP (*Simple Mail Transfer Protocol*)
			SNMP (*Simple Network Management Protocol*)
			Telnet
			TFTP (*Trivial FTP*)
5	Session		NETBIOS (*Network Basic Input Output System*)
			RPC (*Remote Procedure Call*)
			SOCKET
4	Transport	Transport	TCP (*Transmission Control Protocol*)
			UDP (*User Datagram Protocol*)
3	Network	Internet	**IP (*Internet Protocol*)**
			RIP (*Routing Information Protocol*)
			ICMP (*Internet Control Message Protocol*)
			ARP (*Address Resolution Protocol*)
			RARP (*Reverse ARP*)
2	Data link LLC	Network interface	PPP (*Point to Point Protocol*)
	Data Link MAC		SLIP (*Serial Line Internet Protocol*)
1	Fisik		Ethernet, FDDI, ISDN, ATM

2.2. Open System Interconnection (OSI) Layer.

Model referensi OSI dibuat pada akhit tahun 1970, model ini dibuat sebagai solusi untuk mengatasi masalah kompatibilitas antar vendor komputer maupun jaringan. Sehingga dalam membuat *hardware* atau *software* yang bisa saling kerja sama, dalam bentuk protokol-protokol sehingga *hardware* maupun *software* yang dibuat oleh vendor yang berbeda bisa saling kerja sama[3].

Model OSI adalah model atau acuan arsitektural utama untuk jaringan yang mendeskripsikan bagaimana data dan informasi jaringan dikomunikasikan dari sebuah aplikasi di sebuah komputer ke sebuah aplikasi di komputer lain melalui media jaringan. Model OSI melakukan semua ini dengan menggunakan pendekatan lapisan, yaitu terdiri dari 7 lapisan. Berikut ini adalah gambaran komunikasi yang dilakukan oleh setiap lapisan beserta penjelasanya[3]:



Gambar 2.1. Lapisan pada model referensi OSI [3].

a. Lapisan Aplikasi (*Aplication Layer*).

Lapisan aplikasi merupakan lapisan tertinggi pada model referensi OSI yang merupakan tempat dimana user atau pengguna berinteraksi dengan komputer. Biasanya berupa program atau aplikasi pada tingkatan layanan informasi yang membutuhkan akses ke jaringan.

b. Lapisan Presentasi (*Presentation Layer*).

Fungsi dari lapisan ini sesuai dengan namanya, presentasi atau menyajikan data ke lapisan aplikasi dan bertanggung jawab pada penerjemahan data dan format kode (Program) yang diterima.

c. Lapisan Sesi (*Session Layer*).

Lapisan sesi bertanggung jawab untuk membentuk, mengelola, dan kemudian memutuskan sesi-sesi antarlayer-layer presentasi. Lapisan ini juga menyediakan control dialog antarperalatan atau titik jaringan (*node*).

Jadi pada dasarnya layer ini menjaga terpisahnya data dari aplikasi yang satu dengan data dari aplikasi yang lainya.

d. Lapisan Tansport (*Transport Layer*) .

Lapisan transport melakukan segmentasi dan menyatukan kembali data yang tersegmentasi tadi menjadi sebuah arus data. Lapisan ini juga bertanggung jawab menyediakan koneksi yang bebas dari gangguan (*Connection Oriented & Connectionless*).

e. Lapisan Jaringan (*Network Layer*).

Lapisan jaringan bertanggung-jawab untuk mengelola pengalamatan peralatan, melacak lokasi peralatan, dan menentukan cara terbaik untuk memindahkan data. Jadi lapisan ini yang menentukan lalulintas antar peralatan yang tidak terhubung secara lokal.

f. Lapisan Data Link (*Data Link Layer*).

Lapisan Data Link akan memastikan bahwa pesan-pesan akan terkirim melalui perangkat yang sesuai pada jaringan tersebut menggunakan alamat perangkat keras (*Hardware Addresses*), dan menerjemahkan pesan dari lapisan jaringan menjadi bit-bit untuk dipindahkan ke lapisan fisik.

g. Lapisan fisik (*Physical Layer*).

Lapisan fisik merupakan lapisan yang paling rendah dari model OSI. Lapisan fisik berhubungan dengan kabel atau media jaringan lainya yang menghubungkan satu peralatan jaringan komputer dengan peralatan jaringan lainya secara fisik. Lapisan ini bertugas menerima dan

mengirimkan bit-bit yang diterima dari lapisan data link ke media jaringan dan sebaliknya.

Model OSI bersifat hierarkis dan memiliki keuntungan seperti model *layer* lainya. Tujuan utama model OSI adalah untuk memungkinkan bisa saling bekerja samanya jaringan-jaringan yang menggunakan alat-alat dari vendor yang berbeda. Berikut ini beberapa keunggulan menggunakan model OSI :

- Memungkinkan para vendor membuat alat-alat jaringan yang standart.
- Memungkinkan bermacam-macam perangkat keras dan perangkat lunak bisa saling berkomunikasi.
- Mencegah perubahan di satu lapisan mempengaruhi lapisan lainya sehingga permasalahan seperti ini tidak menghambat masalah *development.*

2.2.1. Internet Protocol (IP).

Internet Protocol (IP) berfungsi menyampaikan paket data ke alamat yang tepat. Oleh karena itu Internet Protokol memegang peranan yang sangat penting dari jaringan komputer. Karena semua aplikasi yang memanfaatkan jaringan pasti bertumpu kepada Internet Protocol agar dapat berjalan dengan baik. IP merupakan protokol pada *network layer* yang bersifat [3]:

- ***Connectionless*** adalah kemampuan dimana setiap paket data yang dikirim pada suatu saat akan melalui rute secara *independen*. Paket IP (datagram) akan melalui rute yang ditentukan oleh setiap router yang dilalui oleh datagram tersebut. Hal ini memungkinkan keseluruhan datagram tiba di

tempat tujuan dalam urutan yang berbeda karena menempuh rute yang berbeda pula.

- ***Unreliable*** atau ketidakandalan yakni Protokol IP tidak menjamin datagram yang dikirim pasti sampai ke tempat tujuan. Ia hanya akan melakukan *best effort delivery* yakni melakukan usaha sebaik-baiknya agar paket yang dikirim tersebut sampai ke tujuan.

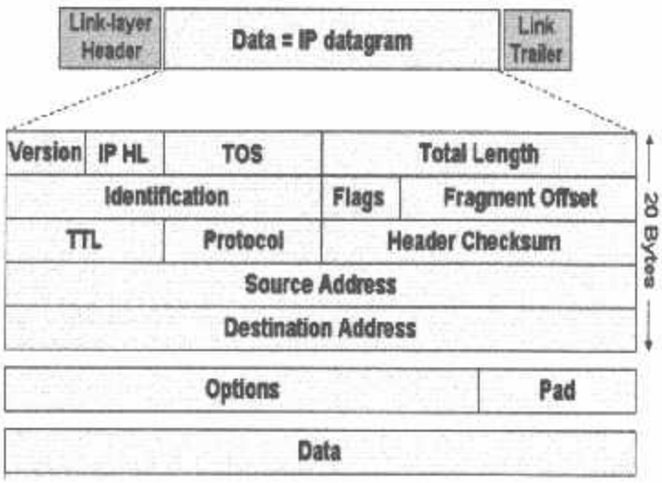
Suatu datagram bisa saja tidak sampai dengan selamat ke tujuan karena beberapa hal berikut:

- Adanya *bit error* pada saat pentransmisian datagram pada suatu medium.
- *Router* yang dilewati men*discard* datagram karena terjadinya kongesti dan kekurangan ruang memori *buffer*.
- Putusnya rute ke tujuan untuk sementara waktu akibat adanya *router* yang *down*.
- Terjadinya kekacauan *routing*, sehingga datagram mengalami *looping*.

IP juga didesain untuk dapat melewati berbagai media komunikasi yang memiliki karakteristik dan kecepatan yang berbeda-beda. Pada jaringan *Ethernet*, panjang satu datagram akan lebih besar dari panjang datagram pada jaringan publik yang menggunakan media jaringan telepon, atau pada jaringan *wireless*. Pada umumnya, semakin cepat kemampuan transfer data pada media tersebut, semakin besar panjang datagram maksimum yang digunakan.

Keunggulan protokol IP adalah kemampuan menggabungkan berbagai media komunikasi dengan karakteristik yang berbeda-beda, fleksibel dengan perkembangan jaringan, dapat merubah *routing* secara otomatis jika suatu rute mengalami kegagalan. Untuk dapat merubah *routing* secara dinamis, dipilih mekanisme *routing* yang ditentukan oleh kondisi jaringan dan elemen-elemen jaringan (*router*). Selain itu, proses *routing* juga harus dilakukan untuk setiap datagram, tidak hanya pada permulaan hubungan.

Agar datagram IP dapat menemukan tujuannya, diperlukan informasi tambahan yang harus dicantumkan pada *header* ini. struktur *header* dari protokol IP beserta fungsinya masing-masing dapat dilihat pada gambar berikut.



Gambar 2.2. Format datagram IP [5].

Setiap paket IP membawa data yang terdiri atas :

- *Version*, yaitu versi dari protokol IP yang dipakai.
- *Header Length*, berisi panjang dari header paket IP dalam hitungan 32 bit word.

- *Type of Service*, berisi kualitas service yang dapat mempengaruhi cara penanganan paket IP.
- *Total length Of Datagram*, panjang IP datagram total dalam ukuran byte.
- *Identification, Flags, dan Fragment Offset*, berisi data yang berhubungan fragmentasi paket.
- *Time to Live*, berisi jumlah router/hop maksimal yang dilewati paket IP (datagram). Nilai maksimum field ini adalah 255. Setiap kali paket IP lewat satu router, isi dari field ini dikurangi satu. Jika TTL telah habis dan paket tetap belum sampai ke tujuan, paket ini akan dibuang dan router terakhir akan mengirimkan paket ICMP *time exceeded*. Hal ini dilakukan untuk mencegah paket IP terus menerus berada dalam jaringan.
- *Protocol*, mengandung angka yang mengidentifikasikan lapisan protokol atas pengguna isi data dari paket IP ini.
- *Header Checksum*, berisi nilai checksum yang dihitung dari jumlah seluruh field dari header paket IP. Sebelum dikirimkan, protokol IP terlebih dahulu menghitung checksum dari header paket IP tersebut untuk nantinya dihitung kembali di sisi penerima. Jika terjadi perbedaan, maka paket ini dianggap rusak dan dibuang.
- *Source Address* dan *Destination Address*, isi dari masing-masing *field* ini cukup jelas, yakni alamat pengirim dan alamat penerima dari datagram. Masing-masing *field* terdiri dari 32 bit, sesuai panjang IP Address yang digunakan dalam Internet. Destination address merupakan field yang akan dibaca oleh setiap router untuk menentukan kemana paket IP tersebut akan diteruskan untuk mencapai destination address tersebut.

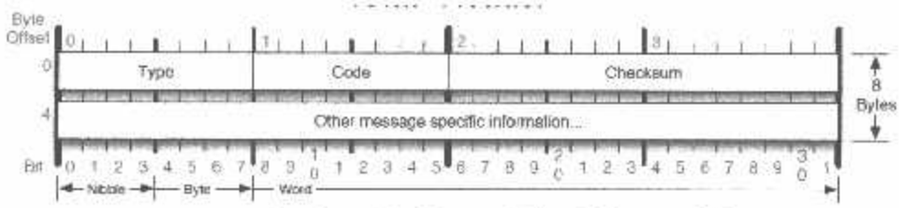
2.2.2. Internet Control Message Protocol (ICMP).

ICMP (*Internet Control Message Protocol*) adalah protokol yang bertugas mengirimkan pesan-pesan kesalahan dan kondisi lain yang memerlukan perhatian khusus. Pesan / paket ICMP dikirim jika terjadi masalah pada lapisan IP dan lapisan diatasnya (TCP/UDP) [2].

Karakteristik dari ICMP adalah :

- ICMP merupakan bagian internal dari IP dan diimplementasikan disetiap modul IP.
- ICMP digunakan untuk menyediakan *feedback* tentang beberapa eror pada sebuah proses datagram.
- Tidak mendukung kehandalan pengiriman paket IP. Datagram / paket bisa tidak terkirim dan tidak ada laporan pemberitahuan tentang kehilangan datagram. Jika diperlukan adanya kehandalan maka harus diimplementasikan pada layer transport.
- Tidak ada respon ICMP yang dikirimkan untuk menghindari adanya perulangan tak terbatas, kecuali respon dari *query message* (ICMP *type* 0, 8-10, 13-18).
- ICMP *error message* tidak pernah dikirimkan sebagai respon sebuah datagram untuk tujuan *broadcast* atau *multicast*.

ICMP *message header* dikirimkan didalam IP datagram, format ICMP *message header* dapat dilihat pada gambar 2.3.



Gambar 2.3. Format ICMP *Message*[2].

Type	Name	Code
0	Echo Reply	0 No Code
1	Unassigned	
2	Unassigned	
3	Destination Unreachable[52]	
		0 Net Unreachable 1 Host Unreachable 2 Protocol Unreachable 3 Port Unreachable 4 Fragmentation Needed and Don't Fragment was Set 5 Source Route Failed 6 Destination Network Unknown 7 Destination Host Unknown 8 Source Host Isolated[53] 9 Communication with Destination Network is Administratively Prohibited[54] 10 Communication with Destination Host is Administratively Prohibited[55] 11 Destination Network Unreachable for Type of Service. 12 Destination Host Unreachable for Type of Service. 13 Communication Administratively Prohibited. 14 Host Precedence Violation 15 Precedence cutoff in effect
4	Source Quench	0 No Code
5	Redirect	
		0 Redirect Datagram for the Network (or subnet) 1 Redirect Datagram for the Host 2 Redirect Datagram for the Type of Service and Network 3 Redirect Datagram for the Type of Service and Host
6	Alternate Host Address	0 Alternate Address for Host
7	Unassigned	
8	Echo Request	0 No Code
9	Router Advertisement	0 No Code
10	Router Selection	0 No Code
11	Time Exceeded	
		0 Time to Live exceeded in Transit 1 Fragment Reassembly Time Exceeded
12	Parameter Problem	
		0 Pointer indicates the error 1 Missing a Required Option 2 Bad Length
13	Timestamp	0 No Code
14	Timestamp Reply	0 No Code

Gambar 2.4. ICMP *Type* dan *Code*[2].

Ada dua tipe pesan yang dapat dihasilkan oleh ICMP *message* yaitu ICMP *Error Message* dan ICMP *Query Message*. ICMP *Error Message* sesuai namanya dihasilkan jika terjadi kesalahan pada jaringan, sedangkan ICMP *Query Message* adalah jenis pesan yang dihasilkan oleh protokol ICMP jika pengirim paket menginginkan informasi tertentu yang berkaitan dengan kondisi jaringan[2].

1. ICMP *Error Message* dibagi menjadi beberapa jenis diantaranya :

   - *Destination Unreachable*, Pesan ini dihasilkan oleh router jika pengiriman paket mengalami kegagalan akibat masalah putusnya jalur, baik secara fisik maupun secara logic. Destination Unreachable ini dibagi menjadi beberapa tipe. Beberapa tipe yang penting adalah:
     - *Network Unreachable*, jika jaringan tujuan tak dapat dihubungi.
     - *Host Unreachable*, jika *host* tujuan tak bisa dihubungi.
     - *Protocol at Destination is Unreachable*, jika ditujuan tak tersedia protokol tersebut.
     - *Port is Unreachable*, jika tidak ada port yang dimaksud pada tujuan.
     - *Destination Network is Unknown*, jika network tujuan tak diketahui.
     - *Destination Host is Unknown*, jika host tujuan tidak diketahui.
   - *Time exceeded*, Paket ICMP jenis ini dikirimkan jika isi field TTL. dalam paket IP sudah habis dan paket belum juga sampai ke tujuannya.

- *Parameter Problem*, paket ini dikirimkan jika terdapat kesalahan paameter pada *header* paket IP.
- *Source Quench*, Paket ICMP ini dikirimkan jika router atau tujuan mengalami kongesti. Sebagai respon pada paket ini, pihak pengirim paket harus memperlambat pengiriman paketnya.
- *Redirect*, paket ini dikirimkan jika router merasa host mengirimkan paket IP melalui router yang salah. Paket ini seharusnya dikirimkan melalui router lain.

2. Sedangkan ICMP *Query Messages* terdiri atas:
   - *Echo request* dan *Echo Reply*. Bertujuan untuk memeriksa apakah system tujuan dalam keadaan aktif. Program *ping* merupakan program pengiriman paket ini. Responder harus menembalikan data yang sama dengan data yang dikirimkan.
   - *Timestamp request* dan *Timestamp Reply*. Menghasilkan informasi waktu yang diperlukan system tujuan untuk memproses suatu paket.
   - *Address mask*, Untuk mengetahui berapa *netmask* yang harus digunakan oleh suatu *host* dalam suatu jaringan.

Sebagai paket pengatur kelancaran jaringan, paket ICMP tidak diperbolehkan membebani jaringan. Karenanya paket ICMP tidak boleh dikirim saat terjadi problem yang disebabkan oleh:

- Kegagalan pengiriman paket ICMP.
- Kegagalan pengiriman paket broadcast atau multicast.

2.2.3. Telnet.

Telnet adalah sebuah aplikasi *remote login* pada jaringan, baik jaringan lokal maupun jaringan internet. Telnet digunakan untuk mengakses sebuah komputer dari komputer lain pada jaringan yang terkoneksi, sehingga dengan *login* sebagai *user* pada komputer jarak jauh, berbagai *resource* maupun *service* dapat diakses tanpa harus datang pada komputer tersebut[3].

Telnet terdiri dari 2 aplikasi, yaitu telnet *client (telnet)* dan telnet *server (telnetd),* telnet *client* digunakan pada komputer yang akan meminta pelayanan, sedangkan telnet *server* adalah komputer yang memberikan pelayanan. Pada hampir semua sistem operasi telah menyertakan aplikasi telnet didalamnya dan secara *default* telnet digunakan pada port 23 untuk saling berkomunikasi. Namun kekurangan dari aplikasi ini adalah tampilannya yang berupa *text-mode,* sehingga pengguna harus mengetahui dasar-dasar perintah *DOS* untuk dapat lebih jauh mendapatkan *resource* maupun *service* dari server telnet.

2.3. Bahasa Pemrograman Borland Delphi 7.

Delphi adalah sebuah perangkat lunak (bahasa pemrograman) untuk membuat program / aplikasi komputer berbasis windows. Delphi merupakan bahasa pemograman berbasis objek, artinya semua komponen yang ada merupakan objek-objek. Ciri sebuah objek adalah memiliki nama, properti dan method/procedure. Delphi disebut juga *visual programming* artinya komponen-komponen yang ada tidak hanya berupa teks (yang sebenarnya program kecil) tetapi muncul dalam bentuk visual[6].

Umumnya delphi lebih banyak digunakan untuk pengembangan aplikasi desktop dan enterprise berbasis database, tapi sebagai perangkat pengembangan yang bersifat general-purpose delphi juga mampu dan digunakan dalam berbagai jenis proyek pengembangan software. Borland Delphi 7 merupakan versi Delphi yang diluncurkan pada tahun 2002 dan didukung oleh banyak perusahaan pembuat komponen atau tools pemrograman membuat Borland Delphi kaya akan fitur pemrograman dibanding IDE-IDE lainnya. Kompatibilitasnya yang sangat tinggi dengan teknologi Microsoft (COM, ActiveX, .NET, Web Services, dan lain-lain)[7].

Delphi 7 telah dilengkapi dengan komponen *Indy* (*Internet Direct*) yang memungkinkan untuk dapat membuat sebuah aplikasi yang memanfaatkan protokol-protokol internet. Sehingga dengan komponen *indy* pengembangan software dibidang jaringan seperti *web server*, *chating*, *remote monitor*, dan aplikasi lainya menjadi semakin mudah.

2.4. Internet Direct (Indy).

Indy merupakan salah satu komponen pada Delphi yang menawarkan protokol-protokol untuk jaringan, baik itu jaringan *Local Area Network* (LAN), *Wide Area Network* (WAN) atau *Metropolitan Area Network* (*internet*). Secara global, Indy memiliki dua kelompok komponen yang dapat dipergunakan untuk membangun aplikasi *client server*. Masing-masing komponen tersebut adalah *Indy Client* dan *Indy Server*[9].

Indy Client menyediakan komponen yang dibutuhkan untuk membangun aplikasi bagi komputer *client* dan *Indy Server* menyediakan komponen untuk membangun aplikasi bagi komputer *server*. Keunggulan dari Indy terletak pada sisi

open source - nya, yang mampu mendukung banyak bahasa pemrograman, seperti Delphi, C++, Kylix dan bahasa pemrograman lain yang berbasiskan pada pemrograman *socket*.


INSTITUT TEKNOLOGI NASIONAL
ITN
MALANG

BAB III

ANALISA DAN DESAIN SISTEM

3.1. Analisa Sistem.

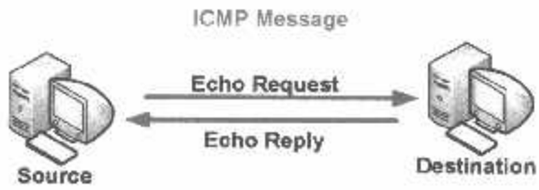
Untuk mendesain sebuah sistem pemetaan jaringan komputer dibutuhkan suatu mekanisme yang dapat memberikan informasi - informasi yang dapat menggambarkan keadaan suatu jaringan komputer. Informasi keadaan jaringan lokal serta rute yang dilalui untuk sampai ke tujuan pada jaringan internet adalah dua faktor penting yang dibutuhkan dalam membangun sebuah sistem pemetaan jaringan komputer.

Mekanisme untuk mendapatkan informasi-informasi tersebut dapat dibuat dengan memanfaatkan protokol ICMP dan IP, ICMP sebagai protokol yang mengirimkan informasi yang dibutuhkan, sedangkan IP adalah protokol yang membawa informasi tersebut dari mana dan harus kemana harus disampaikan. Dengan memanfaatkan dua protokol tersebut dapat diperoleh informasi – informasi tentang keadaan suatu jaringan yang dapat mendukng proses pemetaan jaringan komputer.

3.1.1. Pemetaan Jaringan Intranet.

Pemetaan jaringan intranet adalah pemetaan terhadap komputer – komputer yang sedang aktif dan terhubung pada jaringan yang sama atau jaringan lokal. Untuk memetakan jaringan tersebut dibutuhkan suatu mekanisme yang dapat mengetahui keadaan suatu komputer, apakah komputer tersebut sedang aktif atau tidak.

Mekanisme tersebut dapat didapatkan dengan memanfaatkan protokol ICMP *Query Messages*, *Echo request* dan *Echo Reply* adalah tipe pesan yang digunakan untuk mengetahui keadaan komputer yang dituju.



Gambar 3.1. Mekanisme ICMP *Message echo request* dan *echo reply*.

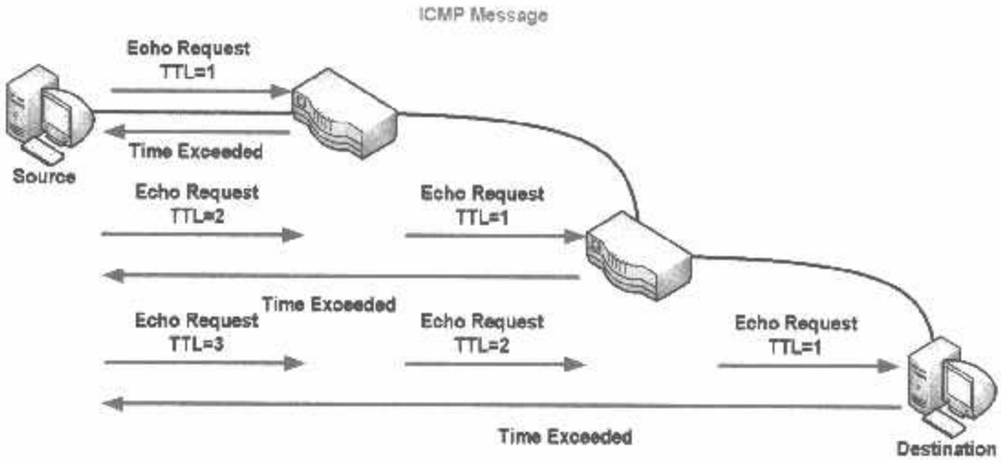
Dengan mengirimkan *echo request* ke komputer tujuan dapat diketahui keadaan dari komputer tersebut apakah sedang aktif atau tidak, hal tersebut dapat diketahui dari *echo reply* yang diterima. Apabila ICMP *reply* yang diterima adalah *echo*, dapat disimpulkan bahwa komputer tujuan sedang aktif, dan sebaliknya apabila *reply* yang diterima adalah kondisi lainya seperti *Error*, *Timeout*, *ErrorUnreachable dan ErrorTTLExceeded* berarti komputer yang dituju tidak ditemukan. Pesan – pesan yang menunjukkan bahwa komputer tujuan tidak ditemukan adalah pesan yang disediakan protokol ICMP *error messange*.

Pesan ICMP merupakan bagian internal dari IP dan diimplementasikan disetiap modul IP, hal ini yang menyebabkan setiap pesan ICMP dapat disampaikan ke tujuan yang tepat dalam jaringan komputer.

3.1.2. Penelusuran Rute Internet.

Rute internet adalah jalur yang dilalui sebuah paket untuk sampai pada tujuan yang berada pada jaringan yang berbeda, untuk dapat memetakan hal

tersebut dibutuhkan suatu mekanisme yang dapat mengetahui rute mana yang dilewati sebuah paket untuk sampai pada tujuanya. Dengan memanfaatkan protokol ICMP *Query Messages*, dengan mengirimkan sebuah pesan *echo request*, dapat diketahui router – router mana saja yang telah dilewati sebuah pesan tersebut untuk sampai pada tujuanya.



Gambar 3.2. Mekanisme ICMP *Message echo request* dan *time exceeded.*

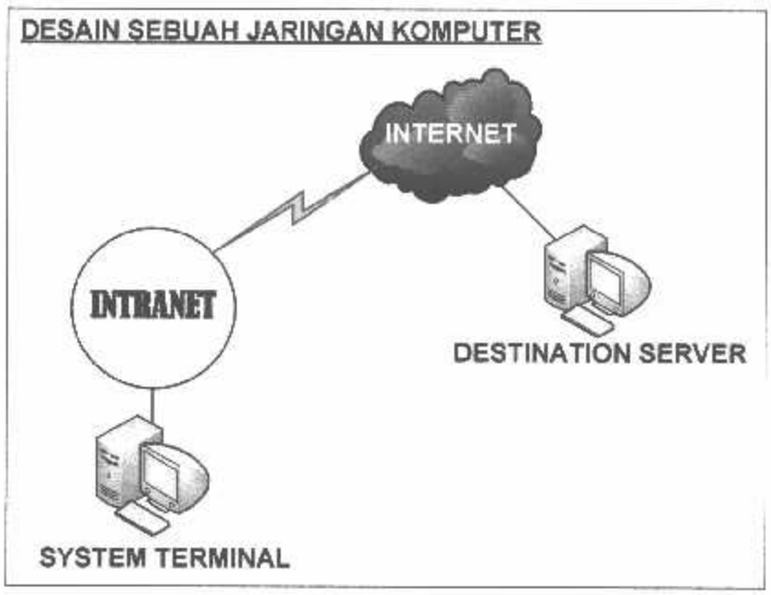
Dengan mengirimkan *echo request* ke komputer tujuan dapat diketahui rute mana yang dilewati dengan cara memberikan nilai TTL (Time to live) sama dengan 1, hal ini ditujukan agar *reply* yang diterima adalah *time exceeded. Reply time exceeded* adalah pesan yang dikirimkan oleh router apabila TTL yang diterima = 1, hal ini dilakukan oleh router karena untuk meneruskan paket ke router selanjutnya nilai TTL tesebut akan dikurangi 1, sehingga untuk dapat melanjutkan paket yang dikirim ke router berikutnya, *Reply time exceeded* akan dikirimkan rouer tersebut ke pengirim paket. Dari *Reply time exceeded* yang diterima nilai TTL akan ditambah 1 oleh pengirim dan dilakukan proses echo request kedua agar didapatkan *Reply time exceeded*

dari router berikutnya, proses ini terus berulang sampai paket diterima oleh tujuan.

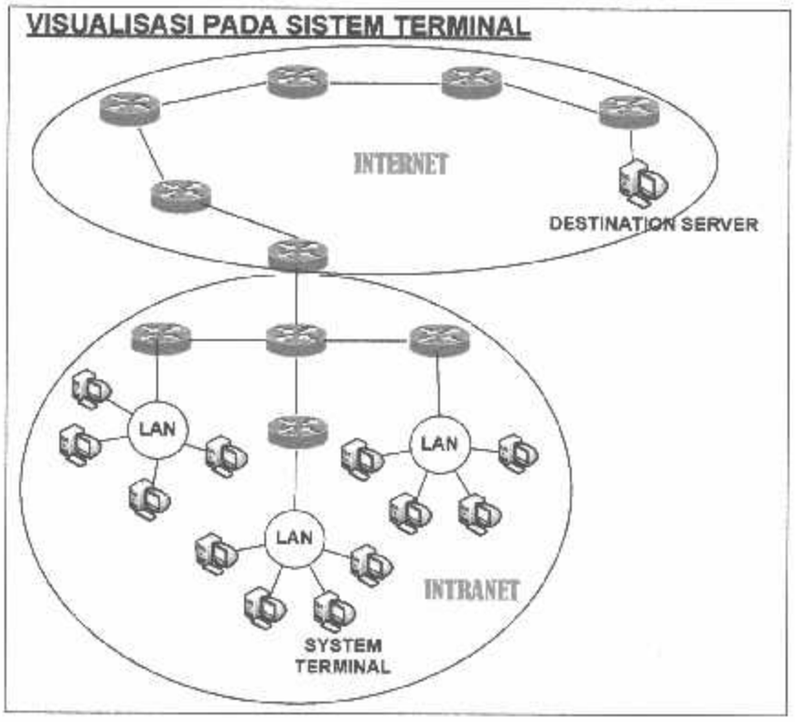
Berbeda dengan router, *host* atau komputer apabila menerima paket dengan TTL=1 tetap akan diproses sesuai dengan permintaan yang diinginkan. Dari beberapa *reply time exceded* dari router yang dilewati dan *echo reply* dari komputer tujuan, dapat diperoleh IP dari masing-masing router maupun komputer yang mengirimkan *reply* tersebut, sehingga rute yang telah dilewati dapat dipetakan dengan mudah.

3.2. Desain Sistem.

Dengan memanfaatkan informasi yang diperoleh dari pemetaan jaringan intranet maupun rute untuk sampai pada tujuan di internet, dapat didesain sebuah aplikasi yang dapat menampilkan hasil pemetaan jaringan yang telah diperoleh secara visual.



Gambar 3.3. Desain jaringan komputer.



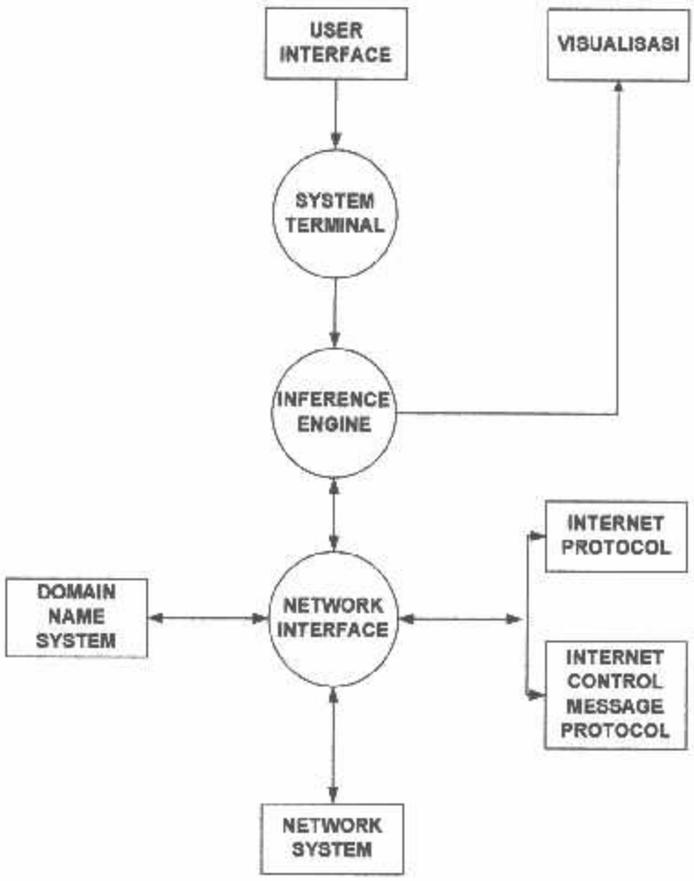
Gambar 3.4. Visualisasi jaringan komputer yang akan dihasilkan.

Dari gambar 3.3 dapat dijelaskan bahwa pada desain sebuah jaringan komputer tidak bisa digambarkan bagaimana peta sebenarnya jaringan tersebut, diharapkan dengan manggunakan aplikasi yang akan dibuat, peta jaringan komputer tersebut dapat divisualkan seperti pada gambar 3.4.

3.2.1. Diagram Blok.

Untuk membuat sebuah aplikasi yang dapat memberikan informasi secara visual sebuah peta jaringan komputer, dibutuhkan beberapa komponen yang membantu baik dalam proses mendapatkan informasi maupun proses visualisasi sebuah jaringan komputer. Berikut ini adalah diagram blok dari

sistem yang akan dibuat, diagram blok ini menggambarkan interaksi antar komponen-komponen yang terlibat dalam pembuatan sistem.



Gambar 3.5. Diagram blok sistem.

Keterangan dari masing-masing komponen diatas adalah sebagai berikut :

1. User Interface.

   User interface merupakan komponen yang menghubungkan *user* dengan mesin untuk menjalankan sistem, dalam hal ini dilakukan oleh tampilan yang dihasilkan dari *coding* yang akan dilakukan pada Delphi 7.

2. System Terminal.

System terminal adalah sebuah mesin yang menjalankan sistem, pada mesin inilah aplikasi ini akan dibuat dan dijalankan, mesin ini diwakili oleh sebuah komputer.

3. Visualisasi

Visualisasi adalah rekayasa dalam pembuatan gambar atau peta untuk penampilan suatu informasi yang diterima, visualisasi ini adalah hasil akhir yang diharapkan.

4. Inference Engine

Inference engine adalah komponen mengontrol kerja sistem dari mana dan kemana paket akan dilewatkan sesuai dengan informasi yang diterima, kontrol pada *inference engine* ini dilakukan dalam prosedur - prosedur *coding*.

5. Network Interface

Network Interface merupakan komponen-komponen fisik pada jaringan komputer, komponen ini dibutuhkan unuk membentuk jaringan komputer yang akan diambil informasinya untuk divisualisasikan.

6. Network System

Network System merupakn komponen-komponen *logic* yang mendukung kerja jaringan komputer, komponen inilah yan akan

menangani bagaimana dan kemana informasi pada jaringan itu disampaikan.

7. Domain Name System

Domain Name System adalah sebuah mesin yang berfungsi untuk mengubah nama *Domain* ke *IP address* dan sebaliknya, sistem ini dibutuhkan apabila inputan yang diberikan user adalah sebuah nama domain, sehangkan yang dibutuhkan sistem adalah alamat IP.

8. Internet Protocol

Internet Protocol (IP) berfungsi untuk menyampaikan paket-paket yang dikirimkan melalui jaringan dari satu titik ke titik lainnya.

9. Internet Control Message Protocol

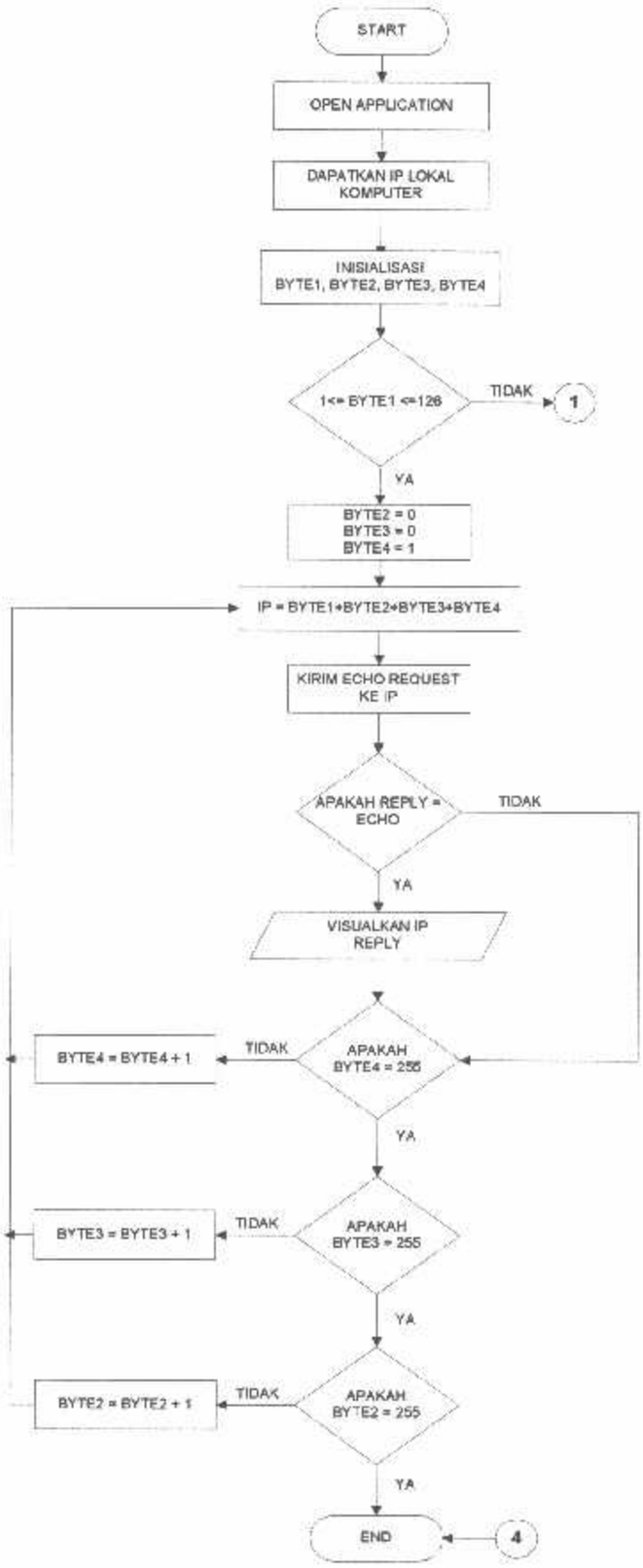
Internet Control Message Protocol (ICMP) adalah protokol yang bertugas mengirimkan pesan-pesan kesalahan dan kondisi lain yang memerlukan perhatian khusus.

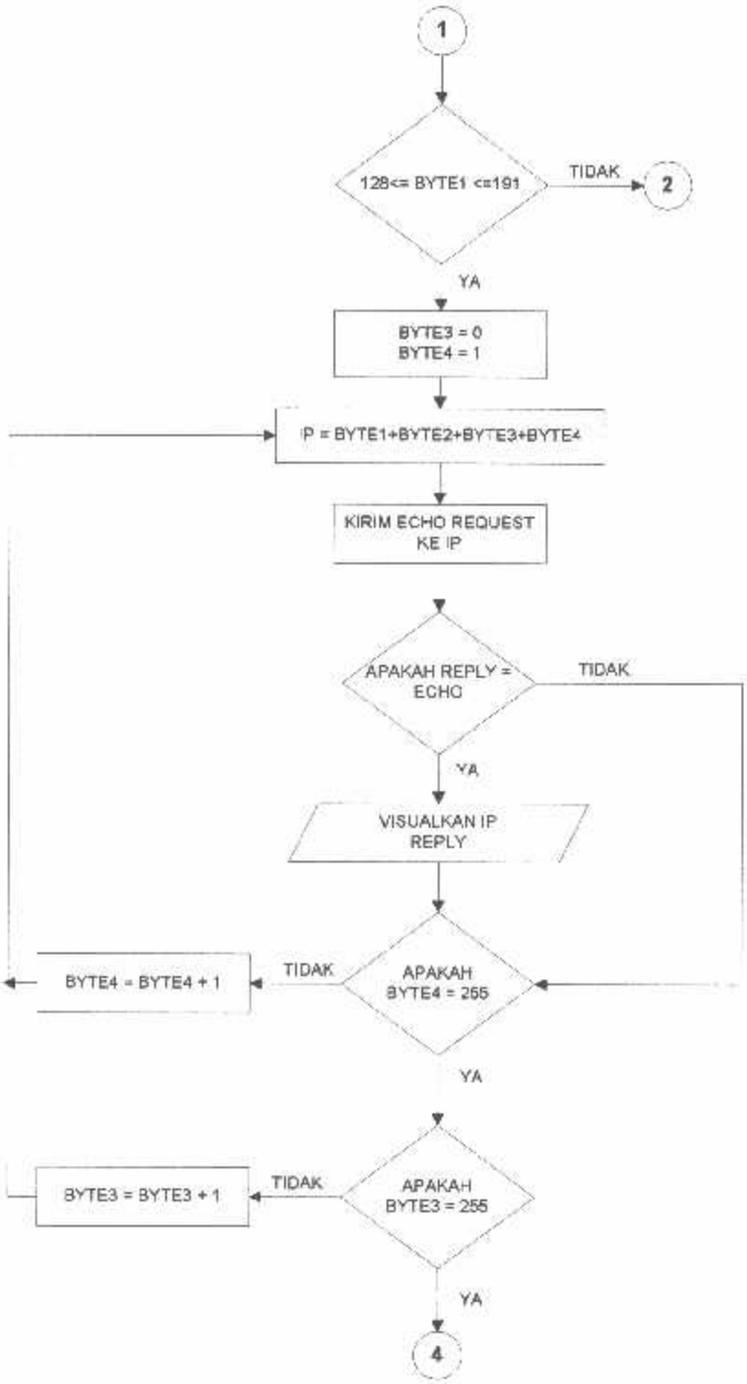
3.2.3. Flowchart.

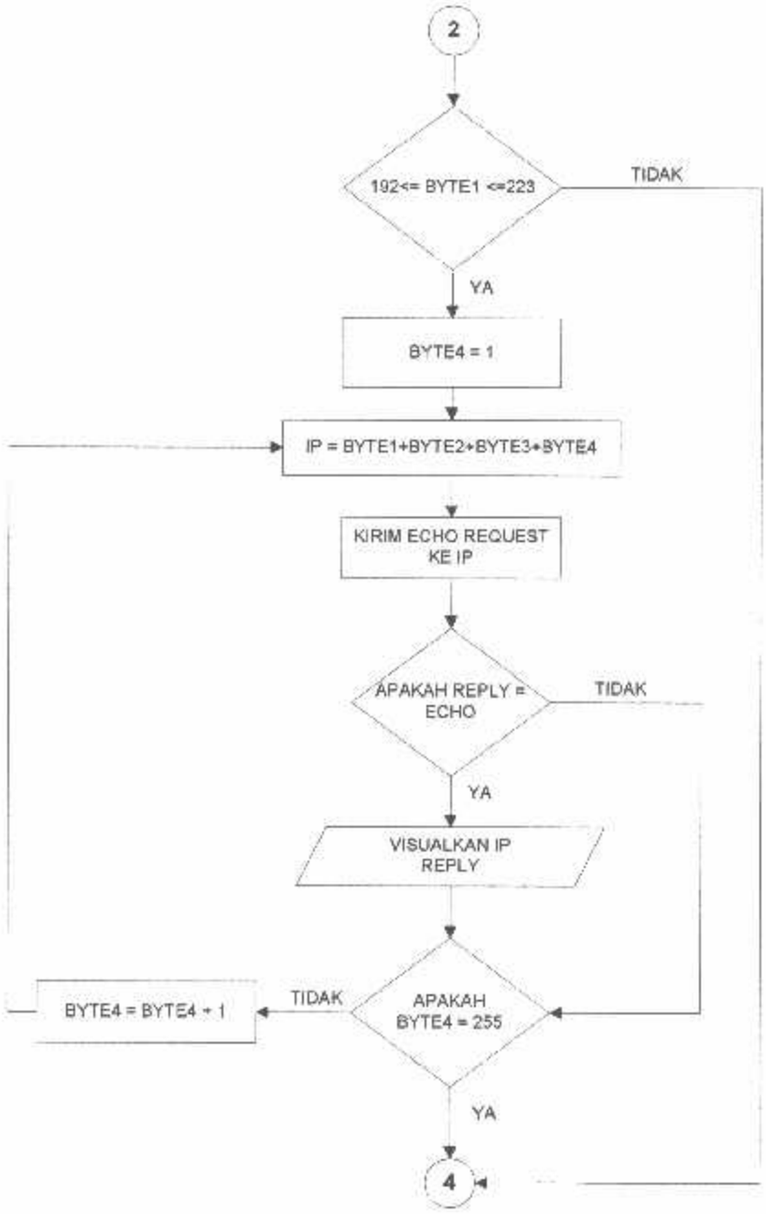
Sesuai dengan desain sistem yang dibuat, dapat dibuat diagram alir dari proses visualisasi peta jaringan komputer sebagai berikut :

1. *Flowchart* visualisasi jaringan intranet.

Flowchart ini menggambarkan proses yang dilakukan untuk visualisasi jaringan intranet dimana akan digambarkan komputer mana saja yang sedang aktif pada saat proses visualisasi dilakukan.


START
OPEN APPLICATION
DAPATKAN IP LOKAL KOMPUTER
INISIALISASI BYTE1, BYTE2, BYTE3, BYTE4
1<= BYTE1 <=126
TIDAK
1
YA
BYTE2 = 0
BYTE3 = 0
BYTE4 = 1
IP = BYTE1+BYTE2+BYTE3+BYTE4
KIRIM ECHO REQUEST KE IP
APAKAH REPLY = ECHO
TIDAK
YA
VISUALKAN IP REPLY
APAKAH BYTE4 = 255
TIDAK
BYTE4 = BYTE4 + 1
YA
APAKAH BYTE3 = 255
TIDAK
BYTE3 = BYTE3 + 1
YA
APAKAH BYTE2 = 255
TIDAK
BYTE2 = BYTE2 + 1
YA
END
4


1
128<= BYTE1 <=191
TIDAK
2
YA
BYTE3 = 0
BYTE4 = 1
IP = BYTE1+BYTE2+BYTE3+BYTE4
KIRIM ECHO REQUEST KE IP
APAKAH REPLY = ECHO
TIDAK
YA
VISUALKAN IP REPLY
BYTE4 = BYTE4 + 1
TIDAK
APAKAH BYTE4 = 255
YA
BYTE3 = BYTE3 + 1
TIDAK
APAKAH BYTE3 = 255
YA
4



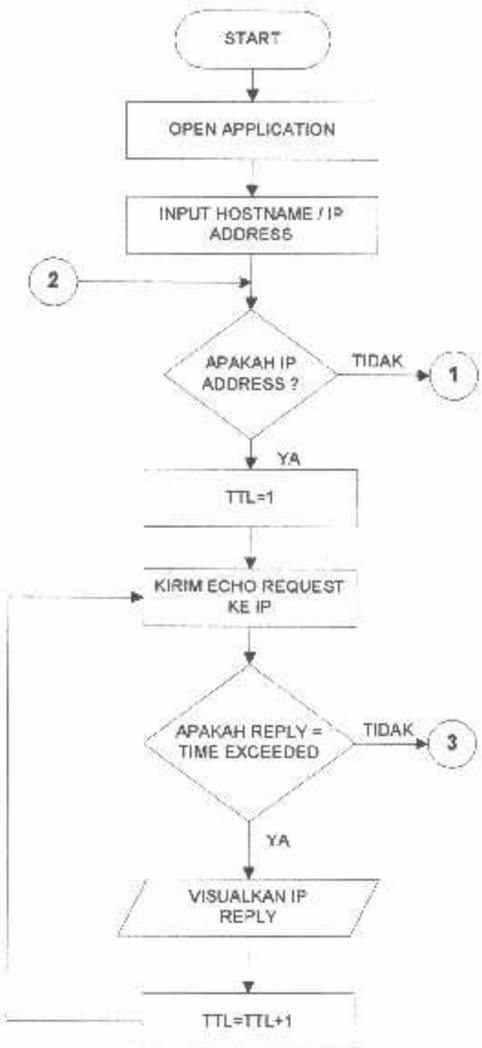
Gambar 3.6. *Flowchart* visualisasi jaringan intranet.

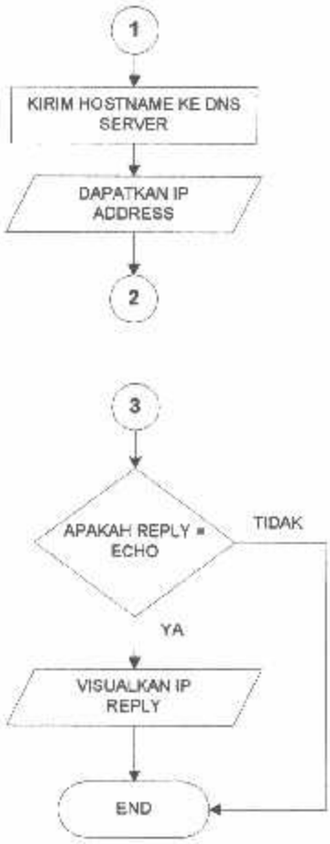
Proses visualisasi jaringan intranet dapat dilakukan apabila alamat jaringan yang digunakan sudah diketahui, dengan demikian dapat dilakukan proses pengiriman pesan ICMP *echo request* ke seluruh alamat yang memungkinkan pada jaringan tersebut. Dengan seleksi terhadap

pesan balasan dari IP yang dituju yaitu apabila balasan berupa *echo reply* maka IP tersebut akan divisualisasikan.

2. *Flowchart* visualisasi rute internet.

Proses visualisasi rute yang dilalui untuk sampai pada tujuan di jaringan internet dapat digambarkan dalam diagrm alir sebagai berikut.





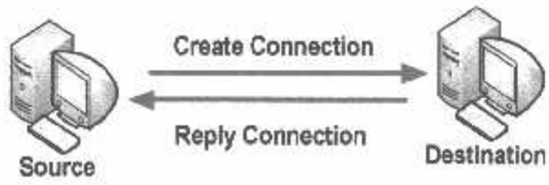
Gambar 3.7. *Flowchart* visualisasi rute internet.

3.3. Port Scanner.

Sebagai pendukung pengelolaan jaringan komputer, *port scanner* digunakan untuk mengetahui *port – port* apa saja yang sedang aktif pada sebuah komputer, sehingga segala aktifitas komunikasi komputer tersebut dapat terkontrol.

Untuk dapat melakukan *scaning* terhadap *port* pada sebuah komputer, dibutuhkan suatu mekanisme yang dapat memberikan informasi tentang *port* yang sedang aktif pada sebuah komputer. Informasi ini dapat diperoleh apabila dilakukan komunikasi pada *port* tertentu dengan komputer tujuan, sehingga

apabila komunikasi tersebut diterima berarti *port* tersebut pada komputer tujuan sedang aktif, berikut ini adalah mekanisme dari *port scanner*.



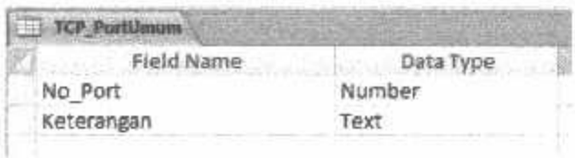
Gambar 3.8. Mekanisme *Port Scanner*.

Untuk mengetahui apakah *port* tertentu pada sebuah komputer sedang aktif atau tidak, dilakukan dengan cara membuat koneksi pada port tertentu dengan memanfaatkan protokol transport TCP sehingga dapat diketahui ststus dari port tersebut. Sebuah *port* dinyatakan sedang aktif apabila koneksi yang dibuat pada port tersebut mendapatkan balasan dari server atau komputer tujuan, namun apabila koneksi gagal dilakukan berarti *port* tersebut sedang tidak aktif, mekanisme ini dilakukan berulang sesuai banyak *port* yang ingin diketahui statusnya.

Untuk mendukung manajemen port pada setiap komputer yang berada pada sebuah jaringan komputer, dibuat sebuah *database* yang menyimpan deskripsi berbagai *port* yang ditemukan, baik itu port yang didefinisikan secara umum, terdaftar, privat, maupun port yang tidak diinginkan dapat didefinisikan sesuai keinginan pengguna. Berikut ini adalah desain *database* yang digunakan untuk menyimpan deskripsi dari beberapa karakteristik *port*.

- *Well known port* (Port Umum).

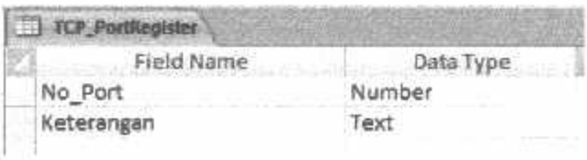
  Port umum yaitu port 0-1023 digunakan untuk protokol komunikasi secara umum misalnya port 80 untuk HTTP, 21 FTP, 23 Telnet dan sebagainya.



Gambar 3.9. Tabel *port* umum.

- *Registered Port* 1024-49151.

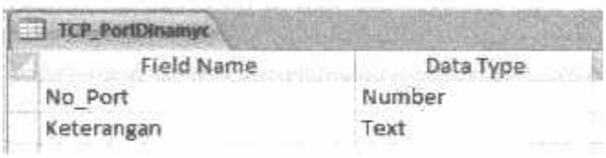
  Port terdaftar yaitu *port* 1024-49151 digunakan untuk komunikasi protokol yang telah didaftarkan penggunaanya sehingga tidak digunakan oleh orang lain dengan fungsi yang berbeda.



Gambar 3.10. Tabel *registered port*.

- *Dynamic Port or Private Port*.

  Port pribadi atau port yang dapat berubah setiap saat yaitu *port* 49152-65535 digunakan secara bebas oleh pengguna.



Gambar 3.11. Tabel *Dynamic Port or Private Port* .

- *Trojan Port*.

  Port Trojan merupakan port yang didefinisikan secara pribadi sebagai upaya untuk menentukan *port-port* yang tidak diinginkan dalam sebuah jaringan komputer.

TCP_PortTrojan

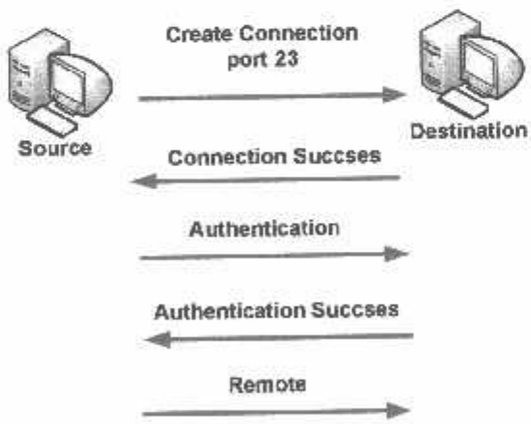
Field Name	Data Type
No_Port	Number
Keterangan	Text

Gambar 3.12. Tabel *Trojan Port*.

3.4. Remote Command.

Selain *port scanner* sebagai pendukung untuk pengelolaan jaringan komputer, juga didesain sebuah aplikasi *remote command* dalam upaya untu memprsempit ruang kerja administrator sehingga pengelolaan yang dilakukan dapat lebih maksimal. Telnet adalah sebuah aplikasi *remote login* pada jaringan, baik jaringan lokal maupun jaringan internet. Telnet digunakan untuk mengakses sebuah komputer dari komputer lain pada jaringan yang terkoncksi, sehingga dengan *login* sebagai *user* pada komputer jarak jauh, berbagai *resource*, *service* maupun aktifitas pada komputer *remote* dapat didapatkan tanpa harus datang pada komputer tersebut.

Telnet adalah aplikasi yang akan diterapkan sebagai aplikasi *remote command* dalam aplikasi ini, berikut ini adalah mekanisme dari proses koneksi aplikasi telnet.



Gambar 3.13. Mekanisme Telnet.

Aplikasi telnet secara *default* menggunakan *port* 23 sebagai pintu komunikasinya, sehingga apabila server telah menyediakan layanan pad port 23 (telnet) maka komunikasi telnet dapat terbentuk. Autentikasi dilakukan untuk dapat mengakses lebih jauh komputer server, tingkat akses ini dibedakan berdasarkan *account* yang diberikan oleh komputer server, misalnya administrator, user, guest dan sebagainya. Dengan aplikasi telnet diharapkan penenganan masalah jaringan dapat dilakukan secara jarak jauh.

BAB IV
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Implementasi Sistem.

Implementasi dilakukan dengan menerapkan hasil desain yang telah dibuat kedalam bahasa pemrograman (*Coding*) Borland Delphi7, sehingga prosedur-prosedur yang telah dibuat dapat dimengerti oleh mesin sehingga menghasilkan keluaran seperti yang diharapkan. Berikut ini adalah perlengkapan yang digunakan dalam implementasi sistem :

Tabel 4.1. Spesifikasi perlengkapan implementasi.

NO	Perlengkapan	Spesifikasi	Keterangan
1	Software	Sistem Operasi	Windows XP Service Pack 2
		Bahasa Pemrograman	Borland Delphi 7
2	Personal Komputer	Processor	P4. 3.0 GHz HT
		Memori	1 Gb DDR2
		Hardisk	80 Gb

Aplikasi network visual (selanjutnya disebut aplikasi **NetVis**) merupakan hasil implementasi yang telah dilakukan, berikut ini penjelasan bagian-bagian dari aplikasi NetVis tersebut :

4.1.1. Visualisasi Peta Jaringan Komputer.

4.1.1.1. Visualisasi Jaringan Intranet.

Visualisasi jaringan intranet dihasilkan dari implementasi mekanisme **ping** ke dalam bahasa pemrograman Delphi 7. Untuk melakukan mekanisme tersebut digunakan sebuah modul protokol

ICMP dari komponen InDy yang tersedia dalam Delphi 7, komponen tersebut adalah IdIcmpClient  . Berikut ini adalah implementasi mekanisme tersebut kedalam Delphi 7 :

1. Mendapatkan IP lokal komputer.

   Untuk mendapatkan IP pada komputer lokal digunakan sebuah modul protokol IP dari INDY yaitu IdIpWatch  , IP dan hostname komputer lokal didapatkan dengan perintah berikut :

```
// IP komputer lokal
IP :=IdIPWatch.LocalIP;
// Hostname komputer lokal
Hostname :=IdIPWatch.LocalName;
```

2. Mendapatkan kelas dan *range* dari IP lokal.

   Untuk mendapatkan kelas dan *Range* IP lokal, terlebih dahulu harus diketahui alamat jaringan dari IP tersebut, untuk mendapatkan hal tersebut dibuat prosedur sebagai berikut :

```
Var
t,jml:Integer;
a,b,c,d,HostId,NetId,ClassAddr,temp : String;
Begin
//=======Memisahkan Byte IP========//
jml:=Length(IP);
//==========byte 1============//
t:=Pos('.',IP);
a:=copy(IP,1,t-1);
temp:=Copy(IP,t+1,jml);
//==========byte 2============//
t:=Pos('.',temp);
b:=copy(temp,1,t-1);
```

```
jml:=length(temp);
temp:=Copy(temp,t+1,jml);
//==========byte 3==========//
t:=Pos('.',temp);
c:=copy(temp,1,t-1);
jml:=length(temp);
temp:=Copy(temp,t+1,jml);
//==========byte 4==========//
d:=copy(temp,1,jml);
//======Mendapatkan Kelas IP======//
case StrToInt(a) of
1..126 : begin
        ClassAddr:='Class A';
        NetID:=a;
        HostID:=b+'.'+c+'.'+d;
        NetAddr:=NetId+'.'+'0'+'.'+'0'+'.'+'0';
        end;
127    : begin
        Application.MessageBox('This Computer Not Connected To The
        Network','Warning..!');
        end;
128..191 : begin
        ClassAddr:='Class B';
        NetID:=a+'.'+b;
        HostID:=c+'.'+d;
        NetAddr:=NetId+'.'+'0'+'.'+'0';
        end;
192..223 : begin
        ClassAddr:='Class C';
        NetID:=a+'.'+b+'.'+c;
        HostID:=d;
        NetAddr:=NetId+'.'+'0';
        end;
end;
end;
```

3. **Ping** pada range IP yang didapatkan.

Ping dilakukan pada range IP yang didapatkan sesuai dengan kelas dari IP tersebut, pembagian range dari setiap kelas berdasarkan tabel berikut :

Tabel 4.2. *Range* IP jaringan lokal.

NO	KELAS	RANGE KELAS	RANGE IP
1	A	1 – 126	A.0.0.1 - A.255.255.254
2	Localhost	127	-
3	B	128 – 191	B.B.0.1 - B.B.255.254
4	C	192 - 223	C.C.C.1 - C.C.C. 254

Kelas D dan E tidak diimplementasikan dalam *coding* karena kelas tersebut tidak bisa digunakan jaringan umum (hanya digunakan untuk jaringan militer dan riset). Hasil implementasi proses ping tersebut adalah sebagai berikut :

```
//===IP tujuan ping====//
ICMP.Host :=IpHost;
//===Perintah ping====//
ICMP.Ping;
{ o> ICMP adalah nama dari komponen IdIcmpClient
  o> IpHost adalah IP komputer pada jaringan lokal }
```

4. Visualisasi komputer yang aktif.

Visualisasi dilakukan apabila dari proses *ping* yang dilakukan mendapatkan balasan (reply) dari komputer yang dituju dan berarti komputer tersebut sedang aktif. *Reply* diinisialisasikan dalam

coding adalah apabila *Byte reply* tidak Nol berarti komputer tujuan sedang aktif.

```
//===Seleksi status komputer====//
if (ICMP.ReplyStatus. ReplyStatus.ReplyStatusType=rsEcho) then
//===Jika komputer aktif====//
    GetIPVisual;
Else
Next;
```

4.1.1.2. Visualisasi Jaringan Internet.

Visualisasi jaringan internet adalah visualisasi rute yang ditempuh untuk sampai pada tujuan yang telah ditentukan. Proses visualisasi dilakukan dengan melalui beberapa tahap sebagai berikut :

1. Mendapatkan IP tujuan.

   IP atau *hostname* dari komputer tujuan adalah inputan yang harus diberikan sebelum proses visualisasi internet dilakukan. Apabila alamat komputer tujuan yang diberikan berupa IP maka proses visualisasi dapat dilanjutkan, namun apabila alamat tersebut berupa *hostname* atau alamat *domain* perlu dilakukan mekanisme untuk mengubah hostname tersebut kedalam IP sehingga proses dapat dilanjutkan. Berikut ini perintah yang digunakan untuk melakukan mekanisme tersebut.

```
//=====kirimkan ke DNS server======//
    ResolvedHost := gStack.WSGetHostByName(HostTarget);
{ o> HostTarger = Hostname komputer tujuan
  o> ResolvedHost = Ip yang didapatkan }
```

2. Penelusuran rute ke tujuan ke IP tujuan.

Penelusuran rute dilakukan dengan melakukan *ping* ke *gateway* dengan mendefinisikan nilai TTL=1 schingga informasi dari tiap node yang dilewati dapat diperoleh. Berikut ini adalah langkah-langkah yang dilakukan untuk menelusuri rute hingga sampai tujuan :

```
TTL:=0;
repeat
  inc(TTL);
  ICMP.Host := ResolvedHost;
  ICMP.TTL := TTL;
  ICMP.ReceiveTimeout := 5000;
  ICMP.Ping;
   case ICMP.ReplyStatus.ReplyStatusType of
    rsEcho:   ...
    rsError:   ...
    rsTimeOut:  ...
    rsErrorUnreachable:   ...
    rsErrorTTLExceeded:
     replyfrom:=ICMP.ReplyStatus.FromIpAddress;
     ...
    end;
  end;
 until reached or (TTL > MaxHop) or Stopped;
end;
```

3. Visualisasi *node* yang terlewati.

Visualisasi dilakukan apabila dari proses *trace* yang dilakukan mendapatkan balasan (reply) "*Time Exceeded*" dari komputer yang

dilewati. Untuk setiap *Reply* dari komputer yang dilewati tersebut didapatkan sebuah IP untuk divisualisasikan.

```
...
    rsErrorTTLExceeded:
     replyfrom:=ICMP.ReplyStatus.FromIpAddress;
     GetRouterVisual(sender);
...
```

4.1.1.3. Visualisasi Hop ke 2.

Visualisasi jaringan pada Hop ke 2 dilakukan setelah mendapatkan IP dari Hop ke 2 yang kemudian dengan IP tersebut dilakukan proses visualisasi jaringan intranet. Untuk mendapatkan IP pada hop ke 2 digunakan perintah sebagai berikut :

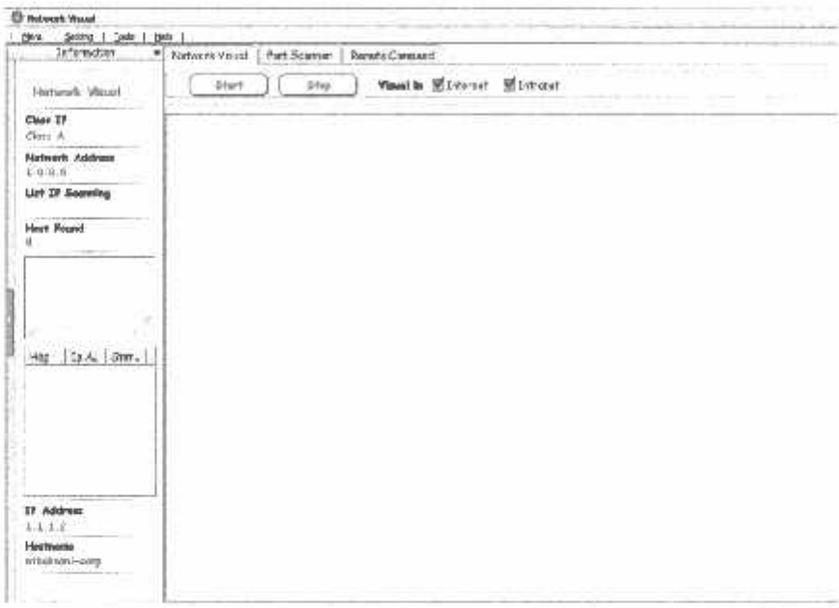
```
If TTL=2 then begin
 if replyfrom<>'  ?.?.?.?' then begin
 hopfound:=true;
 iphop2:=replyfrom;
 end else
 hopfound:=false;
 end;
// replyfrom = IP node yang dilewati
```

Dengan iphop yang didapatkan kemudian diinisialisasikan ke variaber IP dan dilakukan proses yang sapa dengan visualisasi jaringan intranet.

4.1.1.4. Tampilan Menu Visualisasi.

Proses visualisasi peta jaringan komputer dilakukan secara otomatis dengan menekan tombol START, hal ini dapat dilakukan

karena secara otomatis pula aplikasi akan mendapatkan IP lokal dari komputer yang kemudian akan diproses untuk melakukan *scanning* IP pada jaringan yang sama. Dengan IP atau *domain* dari komputer tujuan (*default* www.google.co.id), aplikasi akan melakukan penelusuran rute yang dilalui untuk sampai pada tujuan dan ditampilkan secara visual.

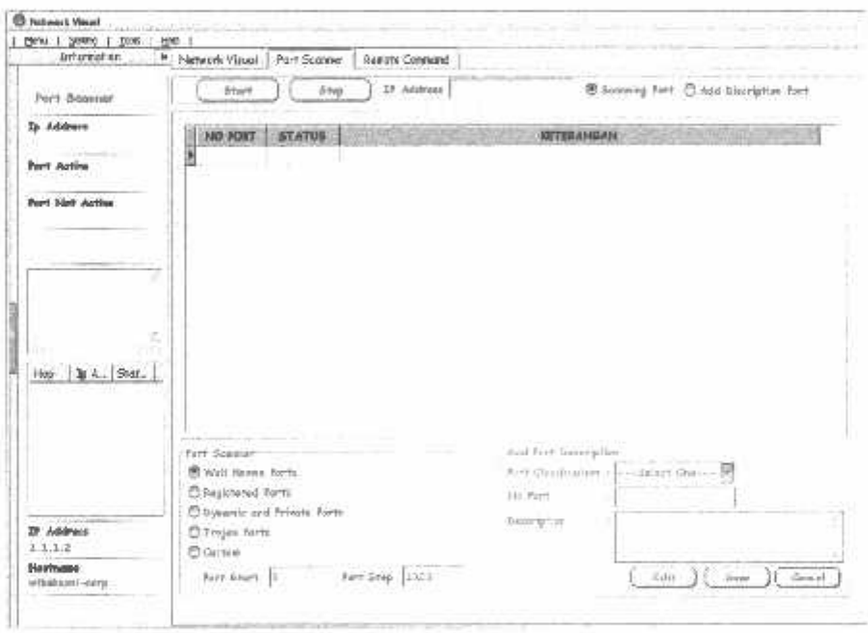


Gambar 4.1. Tampilan menu visualisasi.

Pada kolom sebelah kiri ditampilkan informasi – informasi yang didapatkan dalam proses visualisasi, diantaranya adalah *class* IP, alamat *network*, banyak komputer yang divisualkan serta detail proses *scaning* LAN dan penelusuran rute internet. Pilihan visualisasi dapat ditentukan dengan memilih tanda *check* pada panel atas untuk memilih visualisasi pada jarngan intranet, internet atau keduanya.

4.1.2. Menu Port Scanner.

Port scanner sebagai pendukung aplikasi Netvis digunakan untuk memanajemen port yang aktif pada setiap komputer di jaringan yang dikelola. Dengan *port scanner* dapat diketahui *port-port* apa saja yang sedang aktif di setiap komputer, selain itu untuk mempermudah dalam pengawasan pada port, disediakan menu untuk mengelola port sesuai dengan definisi yang diinginkan. Terdapat dua radio *button* yang digunakan untuk memilih menu yang akan dilakukan yaitu *scanning port* dan menu untuk mengelola port. Untuk menjalankan *port scanner* masukkan IP komputer target pada panel atas, kemudian pilih klasifikasi dan tekan tombol START untuk memulainya.



Gambar 4.2. Tampilan menu *port scanner*.

Gambar diatas adalah tampilan dari menu *port scanner*, pada kolom sebelah kiri ditampilkan informasi – informasi yang didapatkan dalam proses *scanning port*, diantaranya adalah jumlah port yang aktif dan jumlah port yang tidak aktif. Pada panel dibawahnya tedapat dua groupbox, yang pertama adalah

jenis – jenis port yang akan discan dapat dipilih sesuai kebutuhan, dan port yang kedua adalah menu untuk mengelola definisi port sesuai dengan keinginan administrator. Berikut ini adalah pengelompokan jenis port yang digunakan dalam aplikasi Netvis.

Tabel 4.3. Klasifikasi jenis port.

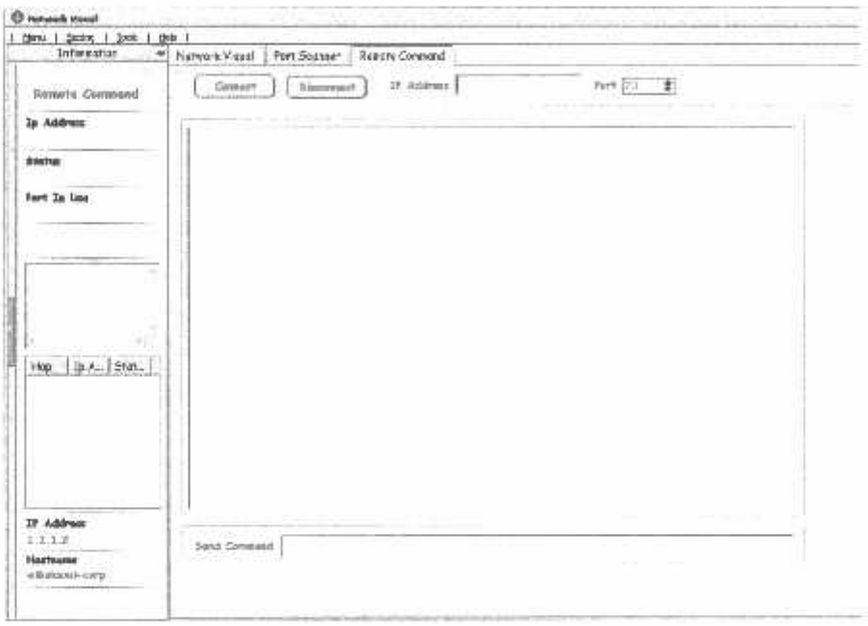
NO	JENIS PORT	NO PORT
1	Well Known Port	1 – 1023
2	Registered Port	1024 - 49151
3	Dynamic and Private Port	94152 - 65535
4	Trojan Port	Sesuai definisi

Mekanisme *port scanner* yang telah diterapkan dalam *coding* memanfaatkan komponen internet yang tersedia dalam Delphi 7 yaitu ClietSocket  , dengan komponen ini dapat dilakukan koneksi terhadap port-port komputer yang dianggap sebagai server. Apabila koneksi berhasil berarti port tersebut sedang aktif, hal ini diulang sampai batas port yang diinginkan.

```
clientsocket1.Port:=a;              // Port tujuan
clientsocket1.Address:=ipport;      // IP tujuan
clientsocket1.Active:=true;
if clientsocket1.Active=false then begin
 status:='Close';                   // Port tertutup
 end else
 status:='Open';                    // Port terbuka
 end;
end;
```

4.1.3. Menu Remote Command.

Remote command perlu diterapkan dalam sistem ini sebagai pendukung fungsi remote komputer, remote yang dipilih adalah remote berbasis *command line* yaitu dengan memanfaatkan aplikasi telnet. Telnet dipilih karena telah terintegrasi di semua sistem operasi, sehingga tidak diperlukan aplikasi tersendiri untuk membuat server remote. Dengan memasukkan IP dan port yang digunakan pada remote server (telnet *default* 23), aplikasi ini dapat dijalankan dengan melakukan autentikasi pada server tersebut.



Gambar 4.3. Tampilan menu *remote command*.

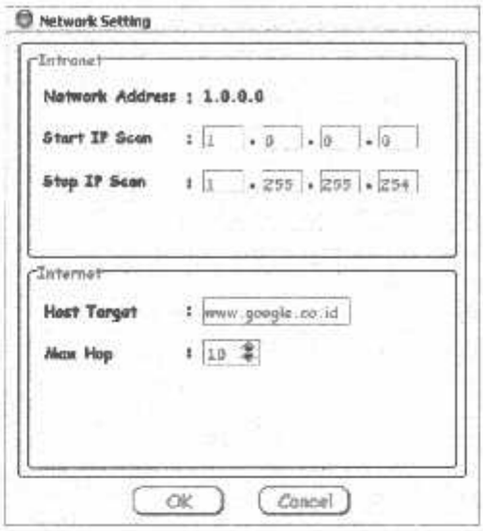
Untuk membuat komunikasi telnet, pada aplikasi Netvis dibutuhkan aplikasi telnet client. Untuk membuat aplikasi digunakan komponen INDY yaitu IdTelnetClient sehingga tidak diperlukan prosedur-prosecur manual untuk membuat koneksi, namun hanya diperlukan interface serta proses pengiriman

perintah telnet, Berikut ini adalah perintah yang digunakan dalam membuat komunikasi telnet tersebut.

```
//===Membuat koneksi telnet==========//
IdTelnet.Host := Ed_IPRemote.Text;
IdTelnet.port := StrToInt(spnedtPort.Text);
IdTelnet.Connect;
//=====Send Command telnet======//
If IdTelnet.Connected then
 if (key = #13) then
  begin
   s := ed_SendCommand.Text;
   for i := 1 to length(s) do
    IdTelnet.SendCh(s[i]);
   IdTelnet.SendCh(#13);
   ed_SendCommand.Clear;
   ed_SendCommand.SetFocus;
   ListCommand.Clear;
end;
```

4.1.4. Network Setting.

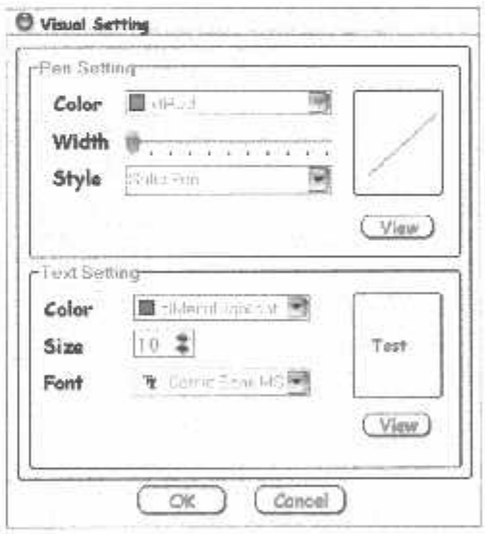
Menu *network setting* digunakan untuk melakukan konfigurasi jaringan sebelum melakukan proses visualisasi, konfigurasi yang dapat dilakukan antara lain adalah menentukan batasan IP LAN scanner, target akhir penelusuran rute internet dan jumlah hop maksimum yang diberikan untuk penelusuran rute tersebut. Berikut ini adalah tampilan menu seting jaringan tersebut.



Gambar 4.4. Tampilan menu seting jaringan.

4.1.5. Visual Setting.

Menu *visual setting* digunakan untuk melakukan konfigurasi tampilan pada hasil visualisasi sesuai dengan keinginan, konfigurasi yang dapat dilakukan antara lain adalah menentukan warna, tebal dan jenis garis serta warna, besar dan jenis *font* yang ditampilkan dalam visualisasi . Berikut ini adalah tampilan menu seting visual tersebut.

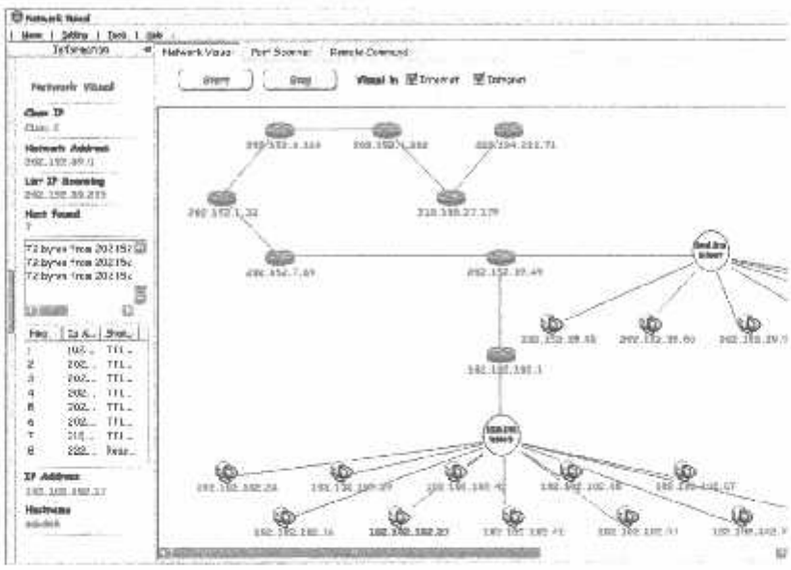


Gambar 4.5. Tampilan menu seting visual.

4.1.6. Hasil Implementasi.

Berikut ini adalah hasil implementasi aplikasi Netvis yang diterapkan pada jaringan kampus ITN Malang melalui jaringan laboratorium.

- Hasil visualisasi.

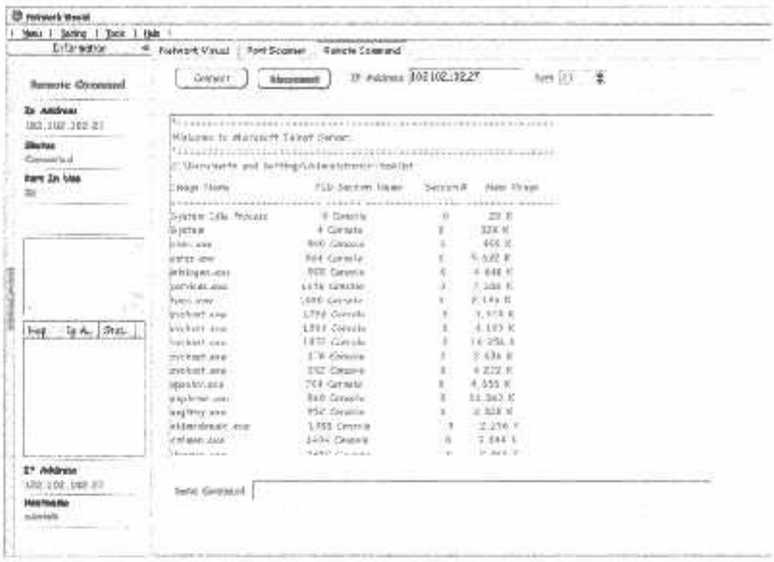


Gambar 4.6. Tampilan visualisasi.

- Hasil *port scanner*.

Gambar 4.7. *Port scanner*.

➢ Hasil *remote command.*



Gambar 4.8. *remote command.*

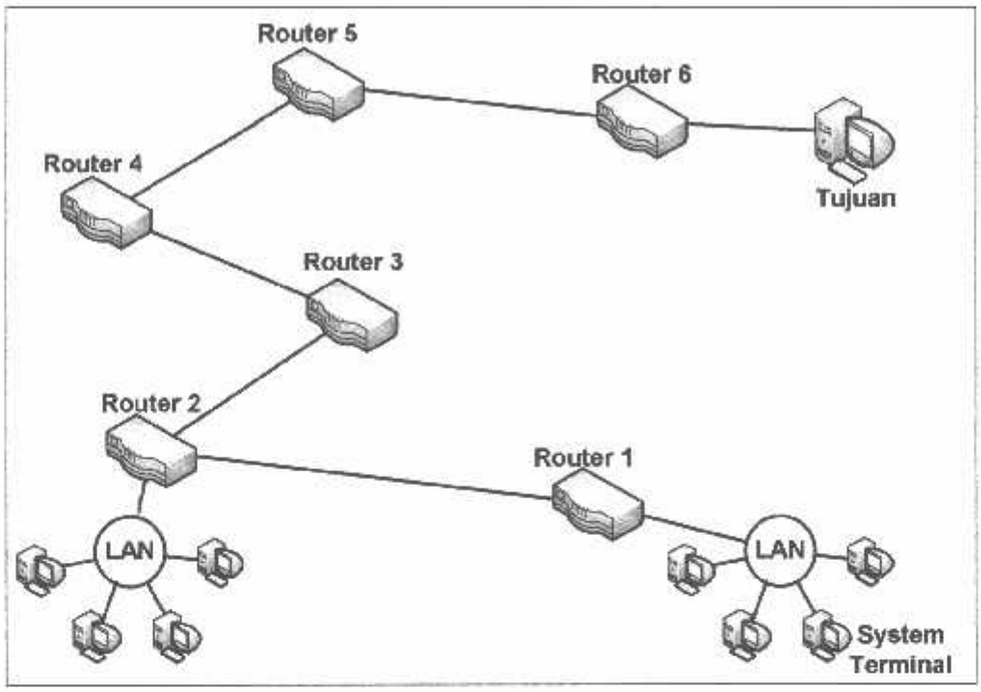
4.2. Pengujian Sistem.

Pengujian ini dilakukan untuk mengetahui kinerja aplikasi network visual pada beberapa kondisi jaringan komputer yang berbeda dengan spesifikasi hardware yang digunakan pada pengujian adalah sebagai berikut:

Tabel 4.4. Hardware pengujian.

NO	Hardware	Spesifikasi	Keterangan
1	Router	Cisco	Series 2811
2	Switch	Cisco	Catalys 2950
3	Komputer	Processor	Pentium 4 3.0 GHz
		Memori	512 Mb DDR2
		Hardisk	80 Gb SATA
		Graphic	SIS 760

Desain jaringan komputer yang digunakan saat pengujian adalah seperti pada gambar dibawah ini :



Gambar 4.9. Desain jaringan pengujian.

Berikut ini adalah hasil pengujian yang telah dilakukan pada berbagai kriteria yang telah ditentukan untuk mengetahui kinerja aplikasi Network Visual ini.

4.2.1 Berdasarkan Jumlah Hop.

Pengujian berdasarkan banyak hop perlu dilakukan untuk mengetahui dampak yang terjadi pada kinerja aplikasi dengan jumlah hop yang berbeda. Berikut ini hasil pengujian yang dilakukan dengan jumlah komputer pada intranet adalah 10 dan pada hop kedua 5 komputer.

Tabel 4.5 Pengujian berdasarkan Hop.

NO	Jumlah Hop	Waktu(Detik)
1	1	10.5
2	2	11.1
3	3	11.9
4	4	12.1
5	5	13.3
6	6	15.5
7	7	15.9

4.2.2. Berdasarkan Jumlah Host.

Sedikit atau banyak jumlah host yang akan ditampilkan pada aplikasi ini perlu dilakukan pengujian, sehingga dapat diketahui pengaruhnya terhadap kinerja aplikasi ini. Pengujian ini dilakukan dengan menggunakan 7 buah hop untuk sampai ke tujuan.

Tabel 4.6. Pengujian berdasarkan *host*.

NO	Jumlah Host		Waktu (Detik)
	Intranet	Hop Ke2	
1	0	0	13.2
2	12	3	13.8
3	10	5	13.4
4	8	7	13.0
5	3	12	13.9

4.2.3. Pada Jaringan Kampus ITN.

Pengujian juga dilakukan pada jaringan kampus ITN malang melalui dua jalur koneksi yaitu, jaringan *Hotspot* dan jaringan laboratorium.

Tabel 4.7. Pengujian pada jaringan ITN.

NO	Tujuan	Jalur Koneksi (Detik)	
		Hotspot	Lab. ITN
1	www.cisco.com	11.7	11.1
2	www.yahoo.com	11.2	11.5
3	www.google.co.id	8.9	9.1
4	www.itn.ac.id	212.0	221.5
5	www.pemkot-malang.go.id	11.1	10.5

4.2.4. Penggunaan memori dan CPU.

Untuk mengetahui pengaruh aplikasi ini terhadap kinerja komputer yang digunakan untuk manjalankanya, dilakukan pengujian terhadap penggunaan memori fisik maupun kinerja CPU.

Tabel 4.8. Pengujian kinerja aplikasi.

NO	Spesifikasi Pengujian	Penggunaan CPU (%)	Penggunaan Memori (Kb)
1	Keadaan normal.	1	10.136
2	Visualisasi intranet.	10	11.126
3	Visualisasi internet.	9	11.252
4	Visualisasi peta jaringan.	11	11.295
5	Visualisasi jaringan hotspot ITN.	12	11.324
6	Visualisasi jaringan lab. ITN.	11	11.125

7	Port scanner host.	2	11.222
8	Port scanner hop (router).	2	11.225
9	Remote host.	5	10.817
10	Remote hop (router).	5	10.912

BAB V

PENUTUP

5.1. Kesimpulan.

1. Dari pengujian yang dilakukan, hasil visualisasi peta jaringan yang didapatkan adalah sesuai dengan jaringan sebenarnya.
2. Banyak hop mempengaruhi lama proses visualisasi, hal ini dikarenakan setiap penambahan hop akan bertambah pula waktu yang dibutuhkan untuk melakukan *trace* serta visualisasi.
3. Banyak *host* tidak mempengaruhi lama proses visualisasi, karena setiap proses *scan* jaringan lokal yang dilakukan adalah berdasarkan jumlah *host* maksimum dalam kelas alamat yang didapatkan
4. Besar kecilnya bandwidth yang dimiliki pada saat melakukan visualisasi tidak mempengaruhi hasil yang didapatkan karena paket yang dikirim untuk mendapatkan informasi sangat kecil.
5. Autentikasi pada tujuan maupun hop yang dilewati seperi, bandwidth akses yang dibatasi maupun akses *ping* yang ditolak mempengaruhi hasil visualisasi peta jaringan yang dilakukan.
6. Penggunaan aplikasi Netvis tidak memiliki pengaruh yang signifikan terhadap kinerja komputer.

5.2 Saran.

1 Untuk pengembangan selanjutnya dapat dilakukan dngan mengembangkan batasan visualisasi jaringan area lokal ke semua hop sesuai dengan kebutuhan.

2 Tampilan visualisasi bisa dibuat secara dinamis sehingga dapat lebih interaktif dalam penataan peta jaringan yang diperoleh.

DAFTAR PUSTAKA

[1]. Kristanto, Andri, *Jaringan Komputer*, Graha Ilmu, 2006.

[2]. Rafiudin, Rahmat, *Protokol-Protokol Esensial Internet*, Penerbit Andi, 2006.

[3]. Lammle, Todd, *Cisco Certified Network Associate Study Guide*, Elex Media Computindo, Jakarta, 2005.

[4]. Mansfield, Niall, *Practical TCP/IP Jilid 1*, Penerbit Andi, 2004.

[5]. Robert, Dave, *Internet Protocols Handbook*, The Coriolis Group, 2006.

[6]. Wahana Komputer, *"Membuat Program Kreatif dan Profesional dengan DELPHI"*, Elex Media Computindo, Jakarta, 2005.

[7]. Raharjo, Budi, *Langkah dan Proses tercepat menjadi programmer Kylix dan Delphi*, INFORMATIKA, 2005.

[8]. Cantu, Marco, *Mastering Delphi, Chapter 22: Graphics*, Cybex, Inc, Alameda, CA, WWW.CYBEX.COM

[9]. Website komponen *Internet Direct (Indy)* Http://www.indy-project.com

LAMPIRAN

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama Mahasiswa	: Wiwit Agit Wibaksoni
NIM	: 04.12.651
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik Komputer dan Informatika
Judul Skripsi	: Pengembangan Visualisasi Peta Jaringan Komputer dengan Memanfaatkan Protokol Jaringan Model Referensi OSI pada Lapisan ke 3.

Dipertahankan dihadapan tim penguji skripsi jenjang Strata Satu (S-1) pada:

Hari : Sabtu
Tanggal : 21 Maret 2009
Dengan Nilai : 85.45 (A)

PANITIA UJIAN SKRIPSI

KETUA

Ir. H. Sidik Noertjahjono, MT.
NIP.Y. 1028700167

SEKRETARIS

Ir. F. Yudi Limpraptono, MT
NIP.Y. 1039500274

ANGGOTA PENGUJI

PENGUJI I

Dr. Eng. Aryuanto, ST,MT.
NIP.1030800417

PENGUJI II

Sandy Nataly Mantja, SKom.

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

N (PERSERO) MALANG
NK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting) Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 09 Agt, 2008

Nomor : ITN-218/I.TA/2/08
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr. **DR. CAHYO CRYSDIAN , MSc**
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk Mahasiswa :

Nama	: WIWIT AGIT W
Nim	: 0412651
Fakultas	: Teknologi Industri
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik Komputer & Informatika

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu (enam) 6 bulan, terhitung mulai tanggal :

31 Juli 2008 s/d 31 Januari 2009

Sebagai satu syarat untuk menempuh ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih

Ketua Jurusan
Teknik Elektro S-1

Ir. F. Yudi Limpraptono, MT
Nip. Y. 1039500274

Tembusan Kepada Yth :
1. Mahasiswa Yang Bersangkutan
2. Arsip

Form. S 4a

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK
Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting) Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 09 Agt, 2008

Nor : ITN-219/I.TA/2/08
Lampiran : -
Hal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr. **SOTYOHADI, ST**
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk Mahasiswa :

Nama	: WIWIT AGIT W
Nim	: 0412651
Fakultas	: Teknologi Industri
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik Komputer & Informatika

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu (enam) 6 bulan, terhitung mulai tanggal :

31 Juli 2008 s/d 31 Januari 2009

Sebagai satu syarat untuk menempuh ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih

Ketua Jurusan
Teknik Elektro S-1

Ir. F. Yudi Limpraptono, MT
Nip. Y. 1039500274

Tembusan Kepada Yth :
1. Mahasiswa Yang Bersangkutan
2. Arsip

Form. S 4a

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

FORMULIR BIMBINGAN SKRIPSI

Nama : **WIWIT AGIT WIBAKSONI**
Nim : **04.12.651**
Masa Bimbingan : **31 JULI 2008 s/d 31 JANUARI 2009**
Judul Skripsi : **PENGEMBANGAN VISUALISASI PETA JARINGAN KOMPUTER DENGAN MEMANFAATKAN PROTOKOL JARINGAN MODEL REFERENSI OSI PADA LAPISAN KE 3**

Tanggal	Uraian	Paraf Pembimbing
15/'08 /08	Konsultasi Desain Sistem	
17/'08 /09	Konsultasi Desain Aplikasi	
8/'08 /10	Konsultasi Bab I, II & III	
10/'08 /11	Pengembangan Aplikasi	
12/'09 /01	Revisi Mekanisme Aplikasi	
23/'09 /02	Revisi Bab III & Aplikasi	
27/'09 /02	Demo Program final	
5/'09 /03	Konsultasi Bab IV & V	
14/'09 /03	Revisi Pengujian Sistem	
17/'09 /03	Acc Seminar Hasil & komprehensif	

Malang,
Dosen Pembimbing

Dr. Cahyo Crysdian
NIP : Y.103040412

Form S-4B

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

FORMULIR BIMBINGAN SKRIPSI

Nama : WIWIT AGIT WIBAKSONI
Nim : 04.12.651
Masa Bimbingan : 31 JULI 2008 s/d 31 JANUARI 2009
Judul Skripsi : PENGEMBANGAN VISUALISASI PETA JARINGAN KOMPUTER DENGAN MEMANFAATKAN PROTOKOL JARINGAN MODEL REFERENSI OSI PADA LAPISAN KE 3

Tanggal	Uraian	Paraf Pembimbing
10/8 '08	Konsultasi Desain Sistem	Jadi
17/9 '08	Konsultasi Desain Aplikasi	Jadi
8/10 '08	Konsultasi Bab I, II & III	Jadi
10/11 '08	Pengembangan Aplikasi	Jadi
12/01 '09	Revisi Mekanisme Aplikasi	Jadi
25/02 '09	Revisi Bab III & Aplikasi	Jadi
27/03 '09	Demo Program final	Jadi
5/03 '09	Konsultasi Bab IV & V	Jadi
14/03 '09	Revisi Pengujian Sistem	Jadi
17/03 '09	Acc Seminar Hasil & komprehensif	Jadi

Malang,
Dosen Pembimbing

Sotyohadi, ST, MT
NIP : Y.1039700309

Form S-4B

LISTING PROGRAM

```
//========VISUALISASI INTRANET=========//
procedure TForm_Utama.GetScanIP(Sender: TObject);
var
m:String;
begin
o:=0;s:=0;u:=0;v:=0;l:= 0;
case StrToInt(a) of
1..126 :   begin                                        // Class A
           Rto:=4;
           k :=StrToInt(bbit);
            repeat
            m:=NetId+'.'+IntToStr(k);
            NetId:=m;
              j :=StrToInt(cbit);
              repeat
              stopped:=false;
              NetId:=m+'.'+IntToStr(j);
                  GetIpHost(sender);
              NetId:=a+'.'+b;
              Inc(j);
              until  (j =StrToInt(ncbit)) or Stopped;
           NetId:=a;
           Inc(k);
           until  (k =StrToInt(nbbit)) or Stopped;
           end;
127      : begin                                       // Localhost
            connect:=false;
           end;
128..191 : begin                                       // Class B
           Rto:=8;
           j :=StrToInt(cbit);
            repeat
            stopped:=false;
            NetId:=NetId+'.'+IntToStr(j);
               GetIpHost(sender);
            NetId:=a+'.'+b;
            Inc(j);
            until  (j =StrToInt(ncbit)) or Stopped;
           end;
192..223 : begin                                      // Class C
            Rto:=20;
           GetIpHost(sender);
           end;
end;
end;

Procedure TForm_Utama.GetIpHost(Sender: TObject);
begin
i :=StrToInt(dbit);
      repeat
      Inc(i);
      stopped:=false;
      ICMP.ReceiveTimeout := Rto;
      Btn_StartNetvis.Enabled:=false;
      IpHost:=NetId+'.'+IntToStr(i);
      Lb_3b.Caption:=IpHost;
      ICMP.Host :=IpHost;
      ICMP.Ping;
      Application.ProcessMessages;
        if (ICMP.ReplyStatus. ReplyStatusType = rsEcho)
       and(IpHost<>LbHop1.Caption)then
        begin
        Inc(l);
        Inc(o);
```

```
        ActNetVis.Execute;
        case scanhop of
        false : begin
          If IpHost=IP then begin

TRzLabel(FindComponent('RzLabel'+IntToStr(l))).Font.Color:=clRed;
          GetIPVisual(sender);
          LbLog.Lines.Append(Format('%d bytes from %s:
icmp_seq=%d ttl=%d time%s%d ms',
          [ICMP.ReplyStatus.BytesReceived,IpHost,

ICMP.ReplyStatus.SequenceId,ICMP.ReplyStatus.TimeToLive,
          sTime,ICMP.ReplyStatus.MsRoundTripTime]));
          end else begin
          GetIPVisual(sender);
          LbLog.Lines.Append(Format('%d bytes from %s:
icmp_seq=%d ttl=%d time%s%d ms',
          [ICMP.ReplyStatus.BytesReceived,IpHost,

ICMP.ReplyStatus.SequenceId,ICMP.ReplyStatus.TimeToLive,
          sTime,ICMP.ReplyStatus.MsRoundTripTime]));
          end
          end;
        true : begin
          if IpHost<>IdIPWatch.LocalIP then begin
          Image_LanHop.Visible:=true;
          ImageCanvas.Canvas.MoveTo(x2+25,y2);
          ImageCanvas.Canvas.LineTo(x3,y3);
          GetIPVisual(sender);
          LbLog.Lines.Append(Format('%d bytes from %s:
icmp_seq=%d ttl=%d time%s%d ms',
          [ICMP.ReplyStatus.BytesReceived,IpHost,

ICMP.ReplyStatus.SequenceId,ICMP.ReplyStatus.TimeToLive,
          sTime,ICMP.ReplyStatus.MsRoundTripTime]));
          end else
          end;
        end;
        end;
      until  (i = StrToInt(ndbit)) or Stopped;
      Btn_StartNetvis.Enabled:=true;
end;

Procedure TForm_Utama.GetIPVisual(Sender: TObject);
begin
image_Lan.Visible:=true;
x:=image_Lan.Left+10;
y:=image_Lan.Top+25;
if scanhop=true then
  st:=2
else if Odd(o)=false then
  st:=0
else begin
  st:=1
end;

case st of
0 : begin
  TImage(FindComponent('image'+IntToStr(l))).Left:=75+(150*u);
  TImage(FindComponent('image'+IntToStr(l))).Top:=470;

TRzLabel(FindComponent('RzLabel'+IntToStr(l))).Left:=50+(150*u);
  TRzLabel(FindComponent('RzLabel'+IntToStr(l))).top:=495;
  TImage(FindComponent('image'+IntToStr(l))).Visible:=true;
  TRzLabel(FindComponent('RzLabel'+IntToStr(l))).Caption:=IpHost;
  TRzLabel(FindComponent('RzLabel'+IntToStr(l))).Visible:=true;
  x1:=TImage(FindComponent('image'+IntToStr(l))).Left;
```

```
  y1:=TImage(FindComponent('image'+IntToStr(l))).Top;
  ImageCanvas.Canvas.MoveTo(x+25,y);
  Inc(u);
  ADOTableVisual.Append;

ADOTableVisual.FieldValues['Nama_Image']:=TImage(FindComponent('i
mage'+IntToStr(l))).Name;
  ADOTableVisual.FieldValues['IP']:=IpHost;
  ADOTableVisual.Post;
  end;
1 : begin
  TImage(FindComponent('image'+IntToStr(l))).Left:=150+(150*v);
  TImage(FindComponent('image'+IntToStr(l))).Top:=530;

TRzLabel(FindComponent('RzLabel'+IntToStr(l))).Left:=125+(150*v);
  TRzLabel(FindComponent('RzLabel'+IntToStr(l))).top:=555;
  TImage(FindComponent('image'+IntToStr(l))).Visible:=true;
  TRzLabel(FindComponent('RzLabel'+IntToStr(l))).Caption:=IpHost;
  TRzLabel(FindComponent('RzLabel'+IntToStr(l))).Visible:=true;
  x1:=TImage(FindComponent('image'+IntToStr(l))).Left;
  y1:=TImage(FindComponent('image'+IntToStr(l))).Top;
  ImageCanvas.Canvas.MoveTo(x+25,y);
  Inc(v);
  ADOTableVisual.Append;

ADOTableVisual.FieldValues['Nama_Image']:=TImage(FindComponent('i
mage'+IntToStr(l))).Name;
  ADOTableVisual.FieldValues['IP']:=IpHost;
  ADOTableVisual.Post
  end;
2 : begin
  TImage(FindComponent('hop'+IntToStr(l))).Left:=500+(130*s);
  TImage(FindComponent('hop'+IntToStr(l))).Top:=275;
  TRzLabel(FindComponent('lbhop'+IntToStr(l))).Left:=475+(130*s);
  TRzLabel(FindComponent('lbhop'+IntToStr(l))).top:=300;
  TImage(FindComponent('hop'+IntToStr(l))).Visible:=true;
  TRzLabel(FindComponent('lbhop'+IntToStr(l))).Caption:=IpHost;
  TRzLabel(FindComponent('lbhop'+IntToStr(l))).Visible:=true;
  x1:=TImage(FindComponent('hop'+IntToStr(l))).Left;
  y1:=TImage(FindComponent('hop'+IntToStr(l))).Top;
  ImageCanvas.Canvas.MoveTo(x3,y3);
  Inc(s);
  ADOTableVisual.Append;

ADOTableVisual.FieldValues['Nama_Image']:=TImage(FindComponent('h
op'+IntToStr(l))).Name;
  ADOTableVisual.FieldValues['IP']:=IpHost;
  ADOTableVisual.Post
  end;
end;
ImageCanvas.Canvas.LineTo(x1+10,y1+15);
end;

//========VISUALISASI RUTE INTERNET=========//
procedure TForm_Utama.acResolveExecute(Sender: TObject);
begin
  bResolved := false;
  lbLog.Lines.Append(Format('resolving %s',[HostTarget]));
  try
    Application.ProcessMessages;
    ResolvedHost := gStack.WSGetHostByName(HostTarget);
    bResolved := true;
    lbLog.Lines.Append(format('%s resolved to %s',[HostTarget,
ResolvedHost]));
  except
    on e: EIdSocketError do
```

```
      lbLog.Lines.text  := Form_Utama.lbLog.Lines.text +
e.message;
  end;
end;

procedure TForm_Utama.acGoExecute(Sender: TObject);
begin
try
      Stopped := false;
      Btn_StartNetvis.Enabled:=false;
      acGo.Enabled := false;
      acStop.enabled := true;
      acResolve.execute;
      if bResolved and not stopped then
      begin
        acPing.execute;
        if not stopped then
          acTrace.Execute;
      end;
      acGo.Enabled := true;
      acStop.enabled := false;
except
close;
end;
end;

function TForm_Utama.PingHost(Host: string; TTL: Integer):
Boolean;
begin
  result := false;
  ICMP.Host := Host;
  ICMP.TTL := TTL;
  ICMP.ReceiveTimeout := 5000;
  ICMP.Ping;
  case ICMP.ReplyStatus.ReplyStatusType of
    rsEcho:
    begin
      lbLog.Lines.Append(format('response from host %s in %d
millisec.',
                                [
                                ICMP.ReplyStatus.FromIpAddress,
                                ICMP.ReplyStatus.MsRoundTripTime
                                ]));
      result := true;
    end;
    rsError:
      lbLog.Lines.Append('Unknown error.');
    rsTimeOut:
      lbLog.Lines.Append('Timed out.');
    rsErrorUnreachable:
      lbLog.Lines.Append(format('Host %s reports destination
network unreachable.',
                                [
                                ICMP.ReplyStatus.FromIpAddress
                                ]));
    rsErrorTTLExceeded:
      lbLog.Lines.Append(format('Hope %d %s: TTL expired.',
                                [
                                ICMP.TTL,
                                ICMP.ReplyStatus.FromIpAddress
                                ]));
  end;
end;

procedure TForm_Utama.acPingExecute(Sender: TObject);
begin
  PingHost(ResolvedHost, MaxHop);
```

```
  Application.ProcessMessages;
end;

function TForm_Utama.FindItem(TTL: Integer; Add: boolean):
TListItem;
var
  p: Integer;
begin
  result := nil;
  if lvTrace.Items.Count < TTL Then
  begin
    for p := 0 to lvTrace.Items.Count - 1 do
    begin
      if StrToIntDef(Form_Utama.lvTrace.Items[p].Caption, -1) =
TTL then
      begin
        result := Form_Utama.lvTrace.Items[p];
        Break;
      end;
    end;
  end;
  if not assigned( result ) then
  begin
    result := lvTrace.Items.Add;
    result.Caption := IntToStr(TTL);
  end;
end;

procedure TForm_Utama.acTraceExecute(Sender: TObject);
var
  TTL: Integer;
  Reached: boolean;
  aItem: TListItem;
begin
  TTL := 0;
  z:=0;f:=4;g:=5;h:=6;p:=0;q:=0;r:=0;
  reached := false;
  lvTrace.Items.Clear;
  repeat
    inc(TTL);
    Inc(z);
    ICMP.Host := ResolvedHost;
    ICMP.TTL := TTL;
    ICMP.ReceiveTimeout := 5000;
    ICMP.Ping;
    aItem := FindItem(TTL, True);
    aItem.SubItems.Clear;
    case ICMP.ReplyStatus.ReplyStatusType of
      rsEcho:
      begin
        aItem.SubItems.Append(ICMP.ReplyStatus.FromIpAddress);
        aItem.SubItems.Append(format('Reached in : %d ms',
[ICMP.ReplyStatus.MsRoundTripTime]));
        reached := true;
        replyfrom:=ICMP.ReplyStatus.FromIpAddress;
        GetRouterVisual(sender);
      end;
      rsError:
      begin
        aItem.SubItems.Append(ICMP.ReplyStatus.FromIpAddress);
        aItem.SubItems.Append('Unknown error.');
        replyfrom:=ICMP.ReplyStatus.FromIpAddress;
        GetRouterVisual(sender);
      end;
      rsTimeOut:
      begin
        aItem.SubItems.Append('?.?.?.?');
```

```
        aItem.SubItems.Append('Timed out.');
        replyfrom:='   ?.?.?.?';
        GetRouterVisual(sender);
      end;
      rsErrorUnreachable:
      begin
        aItem.SubItems.Append(ICMP.ReplyStatus.FromIpAddress);
        aItem.SubItems.Append(format('Destination network
unreachable',
        [ICMP.replyStatus.MsRoundTripTime]));
        replyfrom:=ICMP.ReplyStatus.FromIpAddress;
        GetRouterVisual(sender);
        break;
      end;
      rsErrorTTLExceeded:
      begin
        aItem.SubItems.Append(ICMP.ReplyStatus.FromIpAddress);
        aItem.SubItems.Append(format('TTL=%d',
[ICMP.ReplyStatus.TimeToLive]));
        replyfrom:=ICMP.ReplyStatus.FromIpAddress;
        GetRouterVisual(sender);
      end;
    end; // case
    Application.ProcessMessages;
    if TTL=2 then begin
        if replyfrom<>'   ?.?.?.?' then begin
        hopfound:=true;
        iphop2:=replyfrom;
        end else
        hopfound:=false;
        end;
  until reached or (TTL > MaxHop) or Stopped;
end;

procedure TForm_Utama.lvTraceCompare(Sender: TObject; Item1,
  Item2: TListItem; Data: Integer; var Compare: Integer);
begin
  Compare := StrToIntDef(Item1.Caption, -1) -
StrToIntDef(Item2.Caption, -1);
end;

Procedure TForm_Utama.GetRouterVisual(Sender: TObject);
begin
image_Lan.Visible:=true;
x:=image_Lan.Left+30;y:=image_Lan.Top+25;
x3:=Image_LanHop.Left+30;y3:=Image_LanHop.Top+25;
if z=1 then begin
    TImage(FindComponent('Router'+IntToStr(z))).Left:=430;
    TImage(FindComponent('Router'+IntToStr(z))).Top:=320;
    TRzLabel(FindComponent('LbRot'+IntToStr(z))).Left:=405;
    TRzLabel(FindComponent('LbRot'+IntToStr(z))).top:=340;
    end
else if z=2 then begin
    TImage(FindComponent('Router'+IntToStr(z))).Left:=430;
    TImage(FindComponent('Router'+IntToStr(z))).Top:=190;
    TRzLabel(FindComponent('LbRot'+IntToStr(z))).Left:=405;
    TRzLabel(FindComponent('LbRot'+IntToStr(z))).top:=210;
    x2:=TImage(FindComponent('Router'+IntToStr(z))).Left;
    y2:=TImage(FindComponent('Router'+IntToStr(z))).Top;
    end
else if z=3 then begin
    TImage(FindComponent('Router'+IntToStr(z))).Left:=140;
    TImage(FindComponent('Router'+IntToStr(z))).Top:=190;
    TRzLabel(FindComponent('LbRot'+IntToStr(z))).Left:=115;
    TRzLabel(FindComponent('LbRot'+IntToStr(z))).top:=210;
    end
else if z=f then begin
```

```
    TImage(FindComponent('Router'+IntToStr(z))).Left:=65+(300*p);
    TImage(FindComponent('Router'+IntToStr(z))).Top:=110;

TRzLabel(FindComponent('LbRot'+IntToStr(z))).Left:=40+(300*p);
    TRzLabel(FindComponent('LbRot'+IntToStr(z))).top:=130;
    f:=f+3;
    Inc(p);
    end
else if z=g then begin

TImage(FindComponent('Router'+IntToStr(z))).Left:=140+(300*q);
    TImage(FindComponent('Router'+IntToStr(z))).Top:=20;

TRzLabel(FindComponent('LbRot'+IntToStr(z))).Left:=115+(300*q);
    TRzLabel(FindComponent('LbRot'+IntToStr(z))).top:=40;
    g:=g+3;
    inc(q);
    end
else if z=h then begin

TImage(FindComponent('Router'+IntToStr(z))).Left:=280+(300*r);
    TImage(FindComponent('Router'+IntToStr(z))).Top:=20;

TRzLabel(FindComponent('LbRot'+IntToStr(z))).Left:=255+(300*r);
    TRzLabel(FindComponent('LbRot'+IntToStr(z))).top:=40;
    h:=h+3;
    Inc(r);
    end;
TImage(FindComponent('Router'+IntToStr(z))).Visible:=true;
TRzLabel(FindComponent('LbRot'+IntToStr(z))).Caption:=replyfrom;
TRzLabel(FindComponent('LbRot'+IntToStr(z))).Visible:=true;
x1:=TImage(FindComponent('Router'+IntToStr(z))).Left;
y1:=TImage(FindComponent('Router'+IntToStr(z))).Top;

ADOTableVisual.Append;
ADOTableVisual.FieldValues['Nama_Image']:=TImage(FindComponent('R
outer'+IntToStr(z))).Name;
ADOTableVisual.FieldValues['IP']:=replyfrom;
ADOTableVisual.Post;
if (z>1) then begin
  x0:=TImage(FindComponent('Router'+IntToStr(z-1))).Left;
  y0:=TImage(FindComponent('Router'+IntToStr(z-1))).Top;
  ImageCanvas.Canvas.MoveTo(x0+25,y0);
  ImageCanvas.Canvas.LineTo(x1+25,y1);
  end
  else begin
  ImageCanvas.Canvas.MoveTo(x,y);
  ImageCanvas.Canvas.LineTo(x1+25,y1);
  end;
end;
//========VISUALISASI HOP KE 2=========//
procedure TForm_Utama.GetIpHop2(Sender: TObject);
begin
scanhop:=true;
IP:=iphop2;
FormSetNetw.GetNetwork(sender);
GetScanIP(sender);
end;

//===========PORT SCANNER===========//
procedure TForm_Utama.ClientSocket1Error(Sender: TObject;
  Socket: TCustomWinSocket; ErrorEvent: TErrorEvent;
  var ErrorCode: Integer);
begin
    errorcode:=0;
end;
```

```
procedure TForm_Utama.ClientSocket1Connect(Sender: TObject;
  Socket: TCustomWinSocket);
var
 ll,oo:integer;
begin
  //connectstatus:=1;
  status:='Open';
  GetPortList(sender);
  ll:=strtoint(Lb_2b.Caption);
  oo:=ll+1;
  Lb_2b.Caption:=inttostr(oo);
  clientsocket1.Active:=false;
   if portstart=portstop then
  timer1.Enabled:=false;
end;

procedure TForm_Utama.ClientSocket1Disconnect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
  timer1.Enabled:=true;
end;

procedure TForm_Utama.Timer1Timer(Sender: TObject);
var
  ii,zz,jj,kk:integer;
begin
  //connectstatus:=0;
  ii:=strtoint(portstart);
  clientsocket1.Port:=ii;
  clientsocket1.Address:=ipport;
  clientsocket1.Active:=true;
  zz:=ii+1;
  portstart:=inttostr(zz);
  Ed_StartPort.Text:=portstart;
  if clientsocket1.Active=false then begin
    status:='Close';
    kk:=strtoint(Lb_3b.Caption);
    jj:=kk+1;
    Lb_3b.Caption:=inttostr(jj);
    //GetPortList(sender);
   timer1.Enabled:=true;
  end;
  if portstart=portstop then
  timer1.Enabled:=false;
end;

procedure TForm_Utama.Bt_StartPortClick(Sender: TObject);
begin
if (Ed_IPPortScan.Text='')then begin
  Application.MessageBox('Set The IP Address Computer That Will
Scan The Port','Warning..!');
  Ed_IPPortScan.SetFocus;
  end else begin
  ipport:=Ed_IPPortScan.Text;
  portstart:=Ed_StartPort.Text;
  portstop:=Ed_stopPort.Text;
  Timer1.Enabled:=true;
  Lb_1b.Caption:=Ed_IPPortScan.Text;
  clearAdoTemp(sender);
  Ed_StartPort.Text:=StartPort;
  Lb_3b.Caption:='0';
  Lb_2b.Caption:='0';
  end;
end;

procedure TForm_Utama.Bt_StopPortClick(Sender: TObject);
begin
```

```
  timer1.Enabled:=false;
end;

procedure TForm_Utama.GetPortList(Sender: TObject);
var
ado:integer;
dapatkan:Boolean;
begin
dapatkan:=false;
Portlist:=inttostr(clientsocket1.port);
Adoselect.First;
for ado:=1 to Adoselect.RecordCount do begin
  if(Portlist=Adoselect.Fields[0].AsString)then
  dapatkan:=true
  else
  Adoselect.Next;
end;
if dapatkan=true then begin
ADOTableTemp.Append;
ADOTableTemp.FieldValues['No_port']:=portlist;
ADOTableTemp.FieldValues['Status']:=status;
ADOTableTemp.FieldValues['Keterangan']:=Adoselect.FieldValues['ke
terangan'];
ADOTableTemp.Post
end else begin
ADOTableTemp.Append;
ADOTableTemp.FieldValues['No_port']:=portlist;
ADOTableTemp.FieldValues['Status']:=status;
ADOTableTemp.FieldValues['Keterangan']:='Unassigned';
ADOTableTemp.Post;
end;

end;

//==========TELNET==========//
procedure TForm_Utama.Bt_telConnectClick(Sender: TObject);
begin
if (Ed_IPRemote.Text='')then begin
  Application.MessageBox('Set The IP Address Computer That Will
Remote','Warning..!');
  Ed_IPRemote.SetFocus;
  end else begin
  ListCommand.Clear;
    ActRemote.Execute;
    ed_SendCommand.Enabled:=true;
    Lb_3b.Caption:=spnedtPort.Text;
    Bt_telConnect.Enabled := False;
    Bt_telDisConnect.Enabled := True;
    IdTelnet.Host := Ed_IPRemote.Text;
    IdTelnet.port := StrToInt(spnedtPort.Text);
    IdTelnet.Connect;
end;
end;

procedure TForm_Utama.Bt_telDisConnectClick(Sender: TObject);
var
  i: integer;
  s : string;
begin
    s :='exit';
    for i := 1 to length(s) do begin
      IdTelnet.SendCh(s[i]);
    end;
  IdTelnet.SendCh(#13);
  ed_SendCommand.Clear;
  ed_SendCommand.Enabled:=false;
  ListCommand.Clear;
```

```
  ListCommand.Lines.Add('Disconnected....');
  statustelnet:='Disconnected';
  ActRemote.Execute;
  Bt_telConnect.Enabled := True;
  Bt_telDisConnect.Enabled := False;
end;

procedure TForm_Utama.IdTelnetConnect(Sender: TObject);
begin
statustelnet:='Connected';
ActRemote.Execute;
Lb_3b.Caption:=spnedtPort.Text;
ed_SendCommand.SetFocus;
end;

procedure TForm_Utama.IdTelnetDataAvailable(Sender: TIdTelnet;
  const Buffer: String);
const
    CR = #13;
    LF = #10;
var
    Start, Stop : Integer;
begin
  if ListCommand.Lines.Count = 0 then
      ListCommand.Lines.Add('');

  Start := 1;
  Stop  := Pos(CR, Buffer);
  if Stop = 0 then
      Stop := Length(Buffer) + 1;
  while Start <= Length(Buffer) do begin
      ListCommand.Lines.Strings[ListCommand.Lines.Count - 1] :=
          ListCommand.Lines.Strings[ListCommand.Lines.Count - 1]
+
          Copy(Buffer, Start, Stop - Start);
      if Buffer[Stop] = CR then begin
          ListCommand.Lines.Add('');
          {$IFNDEF Linux}
          SendMessage(ListCommand.Handle, WM_KEYDOWN, VK_UP, 1);
          {$ENDIF}
      end;
      Start := Stop + 1;
      if Start > Length(Buffer) then
          Break;
      if Buffer[Start] = LF then
         Start := Start + 1;
      Stop := Start;
      while (Buffer[Stop] <> CR) and (Stop <= Length(Buffer)) do
          Stop := Stop + 1;
  end;

end;

procedure TForm_Utama.ed_SendCommandKeyPress(Sender: TObject;
  var Key: Char);
var
  i : integer;
  s : string;
begin
if IdTelnet.Connected then
  if (key = #13) then
   begin
    s := ed_SendCommand.Text;
    for i := 1 to length(s) do
      IdTelnet.SendCh(s[i]);
    IdTelnet.SendCh(#13);
    ed_SendCommand.Clear;
```

```
end;

end;

procedure TFormSetNetw.FormCreate(Sender: TObject);
begin
Form_Utama.Enabled:=false;
GetNetwork(sender);
case StrToInt(a) of
1..126 : begin
         IntraNet.Caption:=NetAddr;
         IntraStartA.Text:=a;IntraStartB.Text:='0';
         IntraStartC.Text:='0';IntraStartD.Text:='0';
         IntraStopA.Text:=a;IntraStopB.Text:='255';
         IntraStopC.Text:='255';IntraStopD.Text:='254';
         IntraStartA.Enabled:=false;
         IntraStopA.Enabled:=false;
         end;
127    : begin
         IntraNet.Caption:=NetAddr;
         IntraStartA.Text:=a;IntraStartB.Text:='0';
         IntraStartC.Text:='0';IntraStartD.Text:='0';
         IntraStopA.Text:=a;IntraStopB.Text:='0';
         IntraStopC.Text:='0';IntraStopD.Text:='0';
         IntraStartA.Enabled:=false;IntraStopA.Enabled:=false;
         IntraStartB.Enabled:=false;IntraStopB.Enabled:=false;
         IntraStartC.Enabled:=false;IntraStopC.Enabled:=false;
         IntraStartD.Enabled:=false;IntraStopD.Enabled:=false;
         end;
128..191 : begin
         IntraNet.Caption:=NetAddr;
         IntraStartA.Text:=a;IntraStartB.Text:=b;
         IntraStartC.Text:='0';IntraStartD.Text:='0';
         IntraStopA.Text:=a;IntraStopB.Text:=b;
         IntraStopC.Text:='255';IntraStopD.Text:='254';
         IntraStartA.Enabled:=false;
         IntraStopA.Enabled:=false;
         IntraStartB.Enabled:=false;
         IntraStopB.Enabled:=false;
         end;
192..223 : begin
         IntraNet.Caption:=NetAddr;
         IntraStartA.Text:=a;IntraStartB.Text:=b;
         IntraStartC.Text:=c;IntraStartD.Text:='0';
         IntraStopA.Text:=a;IntraStopB.Text:=b;
         IntraStopC.Text:=c;IntraStopD.Text:='254';
         IntraStartA.Enabled:=false;
         IntraStopA.Enabled:=false;
         IntraStartB.Enabled:=false;
         IntraStopB.Enabled:=false;
         IntraStartC.Enabled:=false;
         IntraStopC.Enabled:=false;
         end;
end;
InterHostTarget.Text:=HostTarget;
InterMaxHop.Value:=MaxHop;
end;

//========SETTING VISUAL========//
procedure TFormSetVis.getvisualset(Sender: TObject);
begin
pencolor:=clRed;
penwidth:=1;
penstyle:=psSolid;
textcolor:=clMenuHighlight;
textsize:=10;
textfont:='Comic Sans MS';
```

```
    ed_SendCommand.SetFocus;
    ListCommand.Clear;
   end;
end;

//========SETTING NETWORK========//
procedure TFormSetNetw.GetNetwork(Sender: TObject);
var
t,jml:Integer;
begin
jml:=Length(IP);
//==========byte 1============//
t:=Pos('.',IP);
a:=copy(IP,1,t-1);
temp:=Copy(IP,t+1,jml);
//==========byte 2============//
t:=Pos('.',temp);
b:=copy(temp,1,t-1);
jml:=length(temp);
temp:=Copy(temp,t+1,jml);
//==========byte 3============//
t:=Pos('.',temp);
c:=copy(temp,1,t-1);
jml:=length(temp);
temp:=Copy(temp,t+1,jml);
//==========byte 4============//
d:=copy(temp,1,jml);
//-----------Detail IP---------------//
case StrToInt(a) of
1..126 : begin
         connect:=true;
         ClassAddr:='Class A';
         NetID:=a;
         HostID:=b+'.'+c+'.'+d;
         NetAddr:=NetId+'.'+'0'+'.'+'0'+'.'+'0';
         bbit:='0';cbit:='0';dbit:='0';
         nbbit:='255';ncbit:='255';ndbit:='255';
         end;
127    : begin
         connect:=false;
         ClassAddr:='IP Local';
         NetID:=a;
         HostID:=b+'.'+c+'.'+d;
         NetAddr:=NetId+'.'+'0'+'.'+'0'+'.'+'0';
         bbit:='0';cbit:='0';dbit:='0';
         nbbit:='0';ncbit:='0';ndbit:='0';
         Application.MessageBox('This Computer Not Connected To
The Network','Warning..!');
         Application.Terminate;
         end;
128..191 : begin
         connect:=true;
         ClassAddr:='Class B';
         NetID:=a+'.'+b;
         HostID:=c+'.'+d;
         NetAddr:=NetId+'.'+'0'+'.'+'0';
         cbit:='0';dbit:='0';
         ncbit:='255';ndbit:='255';
         end;
192..223 : begin
         connect:=true;
         ClassAddr:='Class C';
         NetID:=a+'.'+b+'.'+c;
         HostID:=d;
         NetAddr:=NetId+'.'+'0';
         dbit:='0';ndbit:='255';
         end;
```

```
end;

procedure TFormSetVis.ActiPenViewExecute(Sender: TObject);
begin
//============Color============//
pencolor:=PenColorBox.Selected;
ImagePen.Canvas.Pen.Color:=pencolor;
//============Width============//
penwidth:=PenTrackBar.Position;
ImagePen.Canvas.Pen.width:=penwidth;
//============Style============//
case PenComboBox.ItemIndex of
0 : begin
    penstyle:=psSolid;
    ImagePen.Canvas.Pen.Style:=psSolid;
    end;
1 : begin
    penstyle:=psDash;
    ImagePen.Canvas.Pen.Style:=psDash;
    end;
2 : begin
    penstyle:=psDot;
    ImagePen.Canvas.Pen.Style:=psDot;
    end;
3 : begin
    penstyle:=psDashDot;
    ImagePen.Canvas.Pen.Style:=psDashDot;
    end;
4 : begin
    penstyle:=psDashDotDot;
    ImagePen.Canvas.Pen.Style:=psDashDotDot;
    end;
5 : begin
    penstyle:=psClear;
    ImagePen.Canvas.Pen.Style:=psClear;
    end;
end;
end;

procedure TFormSetVis.ActTextViewExecute(Sender: TObject);
begin
textcolor:=TextColorBox.Selected;
textsize:=Textsizespin.Value;
textfont:=textFontselect.FontName;
Lbtest.Font.Color:=textcolor;
Lbtest.Font.Size:=textsize;
Lbtest.Font.Name:=textfont;
end;


//====================ANDA JUGA PASTI BISA====================//
```