

TCP/IP dan Praktek Sekuriti Jaringan

Ivan Sudirman

ivan@wiraekabhakti.co.id

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

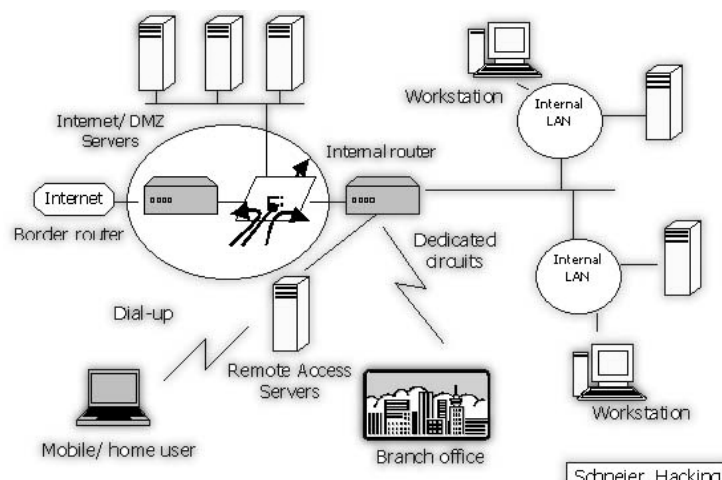
Seluruh dokumen di **IlmuKomputer.Com** dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari **IlmuKomputer.Com**.

Bagian 1: Port Scanning dan Blocking

Pendahuluan

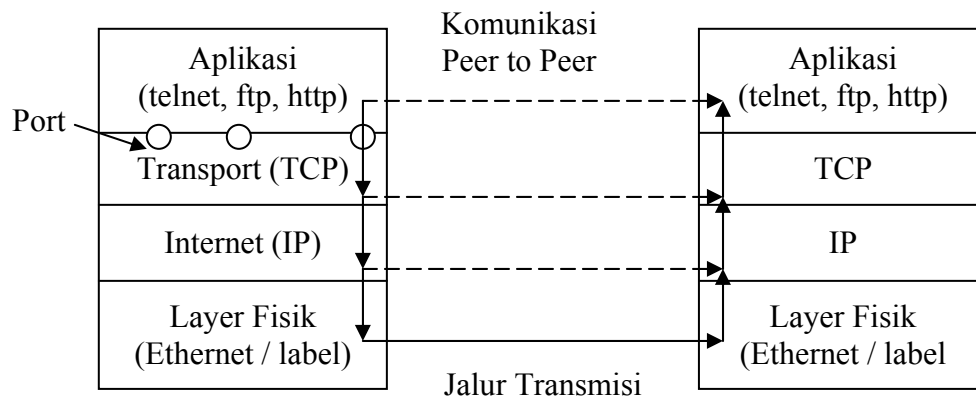
Jaringan komputer LAN pada suatu organisasi yang membentuk intranet, seperti pada gambar di bawah, memiliki 1 buah atau lebih server.

Server-server saling berkomunikasi menggunakan suatu aturan yang di sebut **protokol**.



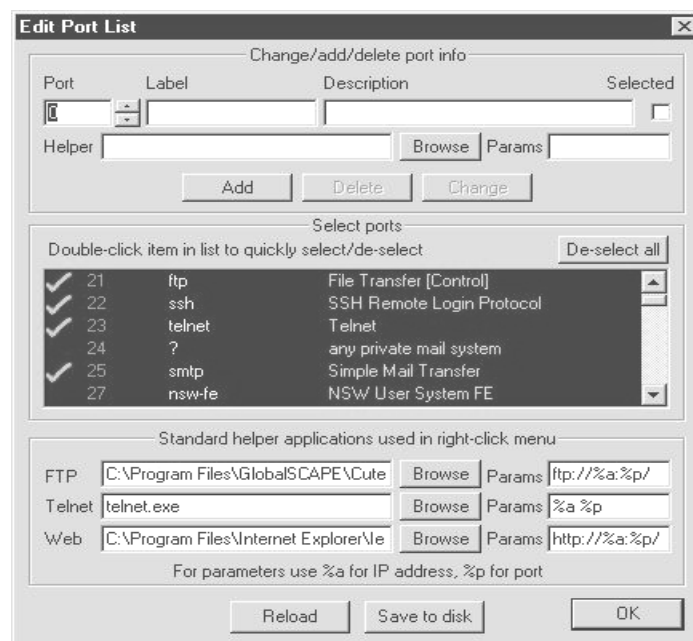
Gambar 1 Jaringan Komputer (Intranet)

Protokol komunikasi di Internet dikenal dengan nama **TCP/IP**. TCP/IP bekerja secara bertingkat atau memiliki layer-layer komunikasi. Protokol TCP/IP merupakan sekumpulan protokol dengan 2 protokol utamanya adalah TCP dan IP.



Gambar 2 Model Komunikasi TCP/IP

Analogi komunikasi di atas seperti 2 orang kepala negara yang saling berkomunikasi dengan surat. Permintaan aplikasi, misalnya **http**, sebagai kepala negara A, yang di tuju ke server aplikasi (*web server*), sebagai kepala negara B. Surat dari kepala negara A (dengan bahasa A) di terjemahkan dulu ke bahasa Inggris oleh seorang penterjemah, kemudian di ketik dan di edit oleh sekretaris. Lalu dikirim melalui seorang kurir (jalur transmisi). Di sisi penerima (negara B), surat tadi akan masuk ke sekretaris dan di proses adminitrasinya, kemudian di terjemahkan ke bahasa B. Kedua kepala negara tidak seolah-olah saling berkomunikasi (komunikasi *peer to peer*) dengan bahasanya masing-masing.

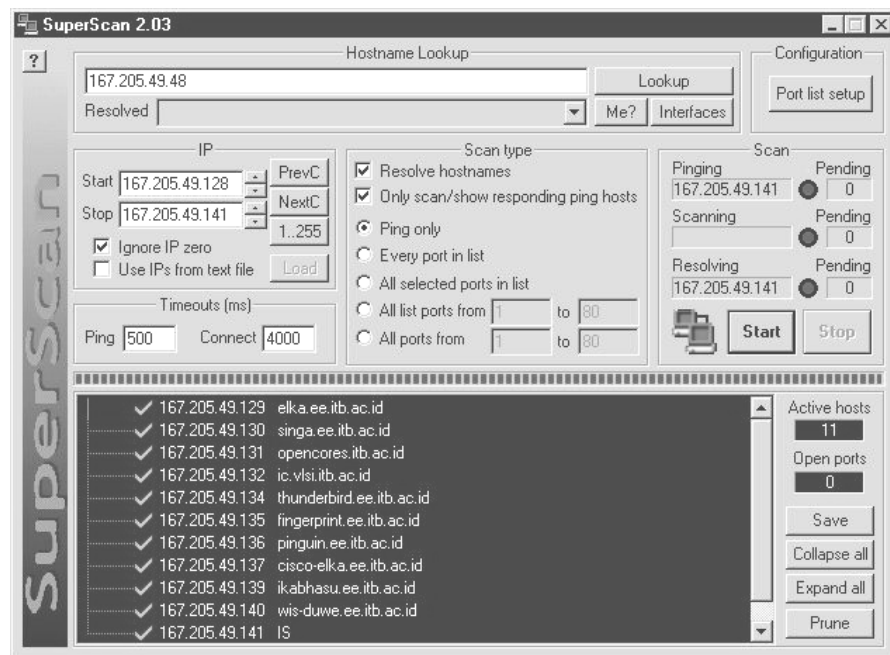


Gambar 3 Nomor Port dan Aplikasi yang

Scanning Port

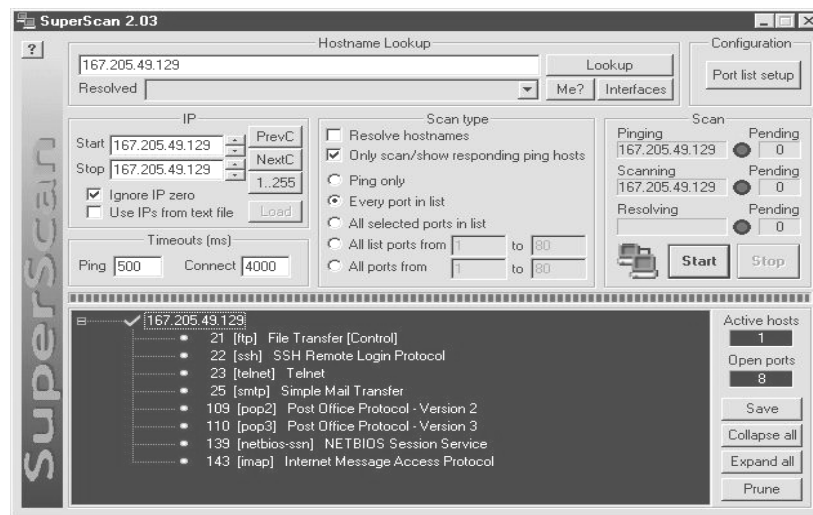
Komunikasi antara layer aplikasi dengan TCP menggunakan **port**. Setiap port di identifikasikan dengan suatu aplikasi. Gambar 3 memperlihatkan nomor port dan aplikasi yang bersesuaian. Misalnya **ftp** biasa menggunakan port 21, **ssh** (*Secure Shell*) menggunakan port 22, dan seterusnya.

Pekerjaan yang di lakukan oleh protokol TCP/IP dapat digambarkan sebagai berikut. Sebuah *web server* misalnya <http://www.company.com/> menangani permintaan dari sebuah *browser* misalnya *Internet Explorer*. *File index.htm*, hendak di kirim ke *browser*. Di TCP, *file index.htm*, akan di pecah-pecah menjadi paket yang lebih kecil dan di beri tanda urutan paketnya, agar penerima dapat menggabungkan paket tersebut kembali. Tanda urutan paket disimpan dalam *header* TCP. Kemudian pada layer IP, di buat *header* lagi yang berisi alamat IP asal dan port asal, serta alamat IP dan port tujuan. Dari penjelasan ini dapat diambil kesimpulan bahwa layer transport dan layer internet, hanya mengolah *header*, pengolah data atau menjalankan program (data yang dimaksud dapat berupa data teks, gambar ataupun sebuah script/program) di lakukan di tingkat aplikasi. Dan port menjadi penting, karena port merupakan jalan masuk ke aplikasi tertentu. Port pada suatu server dapat di buka atau di tutup seperlunya sesuai dengan aplikasi yang dilayani server tersebut, atau jenis komunikasi yang diperbolehkan melewati server tersebut.



Gambar 4 Super Scan : Scanning IP

Pada percobaan pertama berkaitan dengan keamanan jaringan ini, akan di cobakan 2 hal. Dari sisi seseorang yang akan mencoba membobol keamanan suatu server (hacker) yaitu dengan melakukan pendeteksian port, dan dari sisi orang yang akan mengamankan komputernya dengan memblok port yang tidak diperlukan. Pendeteksi port yang digunakan adalah *Freeware* (tersedia gratis), **SuperScan** yang dibuat oleh Robin Keir (<http://members.home.com/rkeir/software.html>).



Gambar 5 SuperScan : Scanning Port

Gambar 4 memperlihatkan SuperScan yang dijalankan untuk mendeteksi IP (*host*) aktif pada *range* tertentu. Sementara gambar 5 memeriksa port-port yang terbuka pada suatu server.

Memblok IP

Untuk memblok IP pada PC Windows dapat digunakan **ZoneAlarm** (<http://www.zonealarm.com>) lisensi gratis untuk pemakaian gratis.

Gambar 6 memperlihatkan pemberitahuan (*alert*) ZoneAlarm dari usaha memeriksa IP (ping) oleh program scanning port SuperScan. Dan gambar 7 memperlihatkan SuperScan yang tidak mendapatkan apa-apa dari usahanya mendeteksi port yang terbuka.



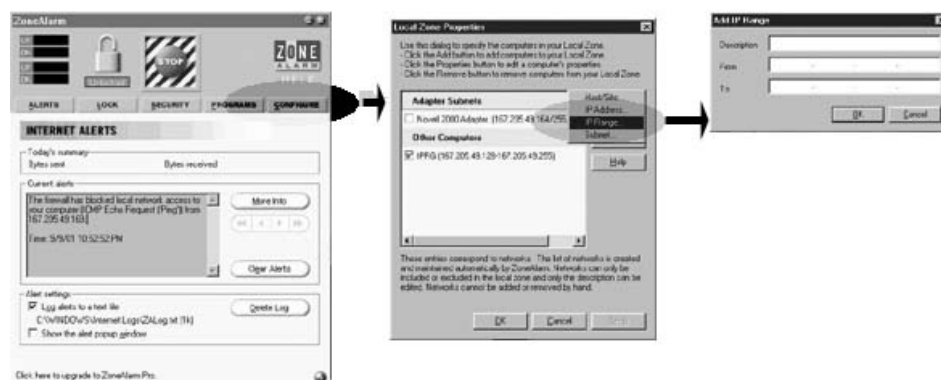
Gambar 6 ZoneAlarm : Blocking Port



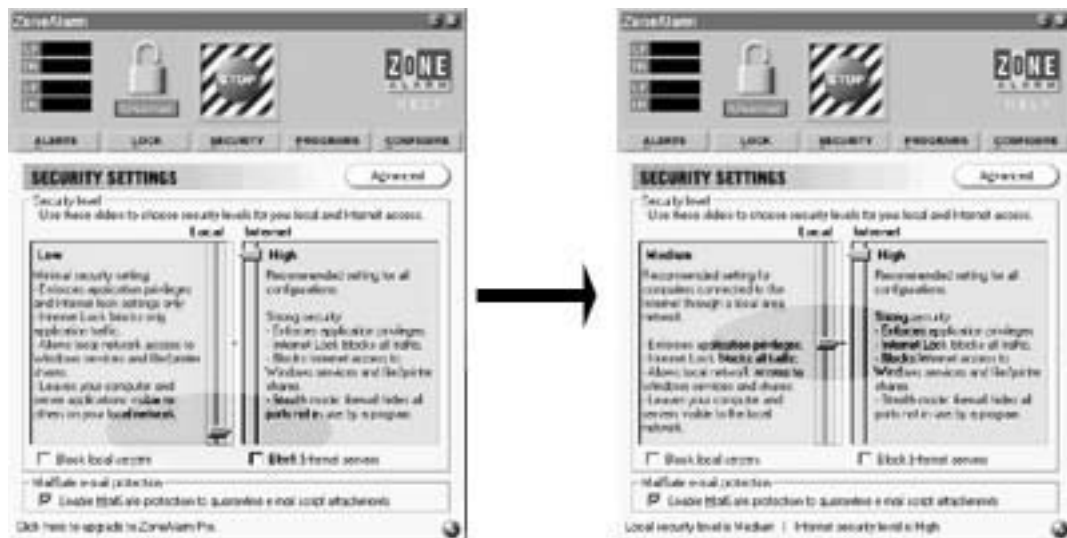
Gambar 7 SuperScan : Tidak Mendapatkan Port Yang Terbuka

Untuk mengkonfigurasi ZoneAlarm, pertama kali adalah membedakan LAN dan Internet. Gambar 8 memperlihatkan konfigurasi IP lokal.

Gambar 9 memperlihatkan *policy* baik terhadap LAN (lokal) maupun Internet. Terdapat 3 macam *policy* : **low**, **medium** dan **high**.



Gambar 8 Konfigurasi ZoneAlarm



Gambar 9 Konfigurasi Policy ZoneAlarm

Port Scanning Lanjut (Advanced Mode)

Pada percobaan di atas, *port scanning* di atas menggunakan jenis **scan** TCP yang terhubung (*TCP connect scan*). Selain *TCP connect scan* masih ada **scan** jenis lain, seperti yang dapat dilihat pada tabel 1.

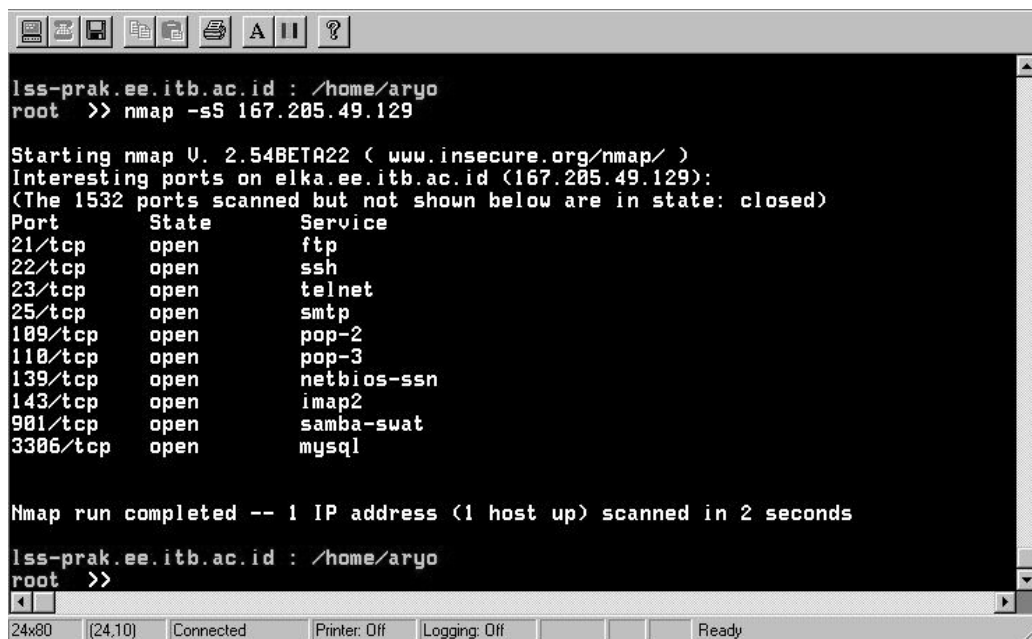
Tabel 1. Jenis scan port

Jenis	Koneksi
TCP Connect scan	SYN → SYN/ACK → ACK
TCP SYN scan	SYN → SYN/ACK → RST/ACK (terbuka) SYN → SYN/RST → RST/ACK (tertutup)
TCP FIN scan	FIN → RST (Closed port)
TCP Xmas Tress scan	FIN / URG / PUSH → RST (Closed port)
TCP null scan	Turn off all flag → RST (Closed port)
TCP ACK scan	Firewall rule set (ACK bit set)
TCP Windows scan	TCP windows size reported (di AIX, FreeBSD)
TCP RPC scan	Detect and indentify RPC (di UNIX)
UDP scan	UDP → tidak ada respon (kemungkinan terbuka) UDP → ICMP port unreachable (di tutup)

Pada **TCP connect scan** *client* akan mengirim sinyal SYN, kemudian akan di balas oleh server dengan sinyal SYN/ACK, untuk mengakhiri hubungan *client* mengirim ACK sebagai konfirmasi. Cara ini di sebut *3-way handshake* dari TCP. Server akan mencatat (terdeteksi oleh ZoneAlarm), bila ACK dari *client* diterima oleh server pada hubungan yang penuh. Sebenarnya saat *client* mengirim SYN, server akan menjawab dengan 2 kemungkinan SYN/ACK bila port terbuka dan RST/ACK bila port tertutup. Dengan informasi ini, *client*

telah dapat menentukan status port, apapun hasilnya bila *client* mengirim sinyal RST (bukan ACK), maka aktifitas *client* tidak akan tercatat (tidak dapat terdeteksi). Cara ini dikenal dengan hubungan *half-open scanning* atau mode *stealth*. *Tool* yang dapat melakukan *scanning* ini adalah **nmap** yang bekerja pada sistem operasi UNIX (gambar 10).

Tingkah laku respon server sewaktu scanning dari *client* berbeda untuk setiap sistem operasi. Tabel 2 memperlihatkan sinyal dan respon server yang membuatnya dapat dikenali jenis sistem operasinya. Perlu di catat, parameter pada server dengan mudah diubah, sehingga prediksi jenis sistem operasi dengan cara di atas bisa tidak tepat.



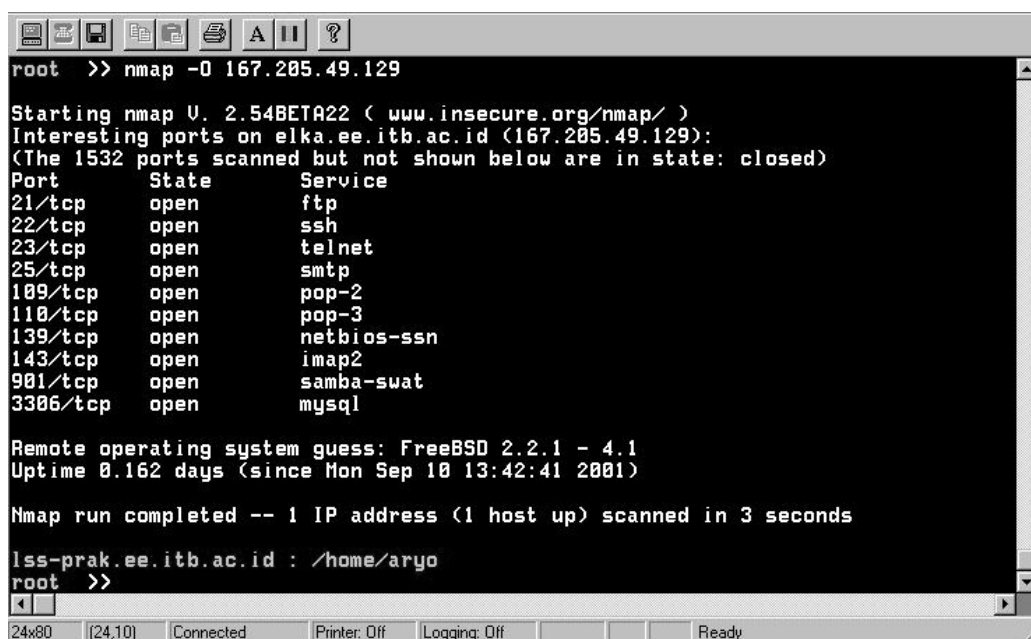
```
lss-prak.ee.itb.ac.id : /home/aryo
root >> nmap -sS 167.205.49.129

Starting nmap U. 2.54BETA22 ( www.insecure.org/nmap/ )
Interesting ports on elka.ee.itb.ac.id (167.205.49.129):
(The 1532 ports scanned but not shown below are in state: closed)
Port      State      Service
21/tcp    open       ftp
22/tcp    open       ssh
23/tcp    open       telnet
25/tcp    open       smtp
109/tcp   open       pop-2
110/tcp   open       pop-3
139/tcp   open       netbios-ssn
143/tcp   open       imap2
901/tcp   open       samba-swat
3306/tcp  open       mysql

Nmap run completed -- 1 IP address (1 host up) scanned in 2 seconds

lss-prak.ee.itb.ac.id : /home/aryo
root >>
```

Gambar 10 NMAP mode Stealth



```
root >> nmap -O 167.205.49.129

Starting nmap U. 2.54BETA22 ( www.insecure.org/nmap/ )
Interesting ports on elka.ee.itb.ac.id (167.205.49.129):
(The 1532 ports scanned but not shown below are in state: closed)
Port      State      Service
21/tcp    open       ftp
22/tcp    open       ssh
23/tcp    open       telnet
25/tcp    open       smtp
109/tcp   open       pop-2
110/tcp   open       pop-3
139/tcp   open       netbios-ssn
143/tcp   open       imap2
901/tcp   open       samba-swat
3306/tcp  open       mysql

Remote operating system guess: FreeBSD 2.2.1 - 4.1
Uptime 0.162 days (since Mon Sep 10 13:42:41 2001)

Nmap run completed -- 1 IP address (1 host up) scanned in 3 seconds

lss-prak.ee.itb.ac.id : /home/aryo
root >>
```

Gambar 11 NMAP Mendeteksi Sistem Operasi

Tabel 2 Sinyal Penguji (*Probe*) dan Respon Server Spesifik untuk Dapat Membedakan (*Distinguish*)

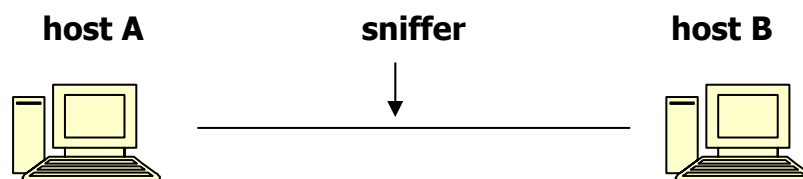
Jenis Probe	Respon Server	Sistem Operasi
FIN probe	FIN/ACK	Windows NT
Bogus Flag probe (undefined flag)	Flag set	Linux
ISN sampling (Initial Sequence Number)		Asumsi OS
“Don’t fragment bit” monitoring	Set “Don’t fragment bit”	Some OS
TCP initial window size		Asumsi OS
ACK value	Send back sequence number - number sent - number sent - 1	
ICMP error message Quenching	UDP scan pada selang waktu tertentu : Jumlah pesan unreachable	Asumsi OS
ICMP message quoting		Asumsi OS
ICMP error message – Echoing integrity	Alter the IP headers	Asumsi OS
TOS (Type of Service)	Pesan “ICMP port unreachable”	Banyak menggunakan 0
Fragmentation handling	Noting how probe packets are reassembled	Asumsi OS
TCP options	Multiple options sets	Asumsi OS

Bagian 2: Sniffer dan Enkripsi

Pendahuluan

Ide internet berawal saat Amerika Serikat melakukan perang dingin dengan Uni Sovyet. Sebuah bom atom mampu menghancurkan suatu area yang luas. Ide Internet sangat sederhana, yaitu bagaimana komputer dapat berkomunikasi tanpa melewati suatu jalur yang permanen (*dedicated*), sehingga bila pusat komando dihancurkan, koordinasi (komunikasi) dapat dilakukan di mana saja. Implikasinya adalah setiap komputer yang terhubung ke internet, datanya akan melewati banyak server (maksimal 30 *hops*), sebelum mencapai tujuannya. Setiap orang yang berada pada jalur data tersebut, dimungkinkan untuk membaca data tersebut. Pembacaan data yang bukan tujuannya ini dikenal sebagai **sniff**.

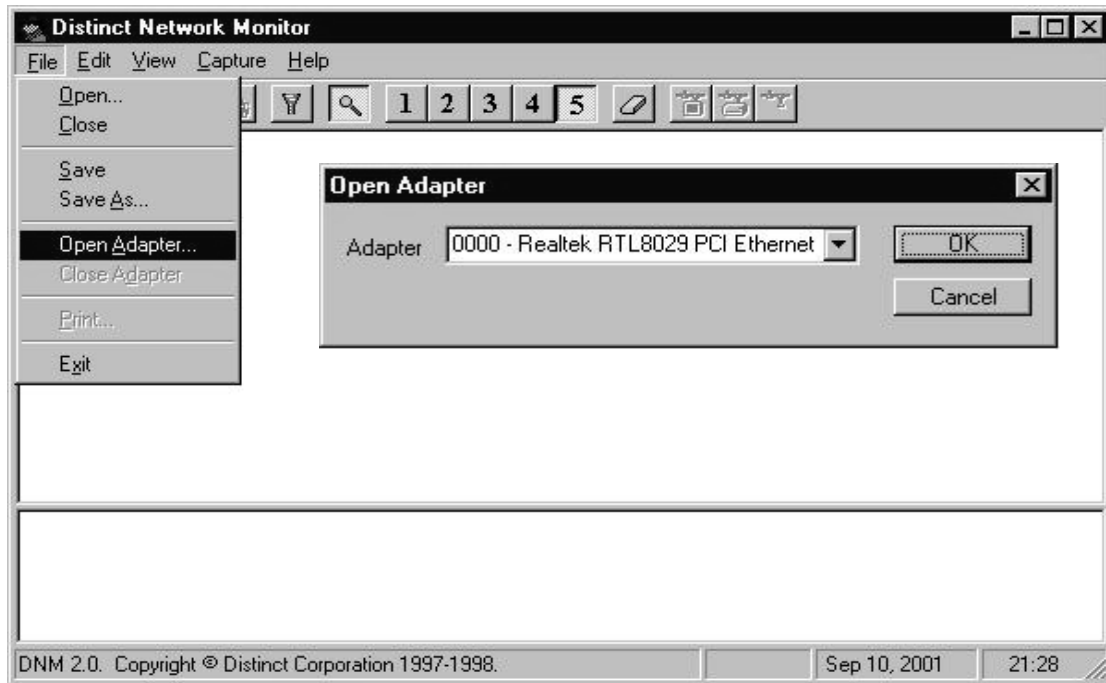
Sniffer



Gambar 1 : Sniff Dalam Komunikasi Komputer

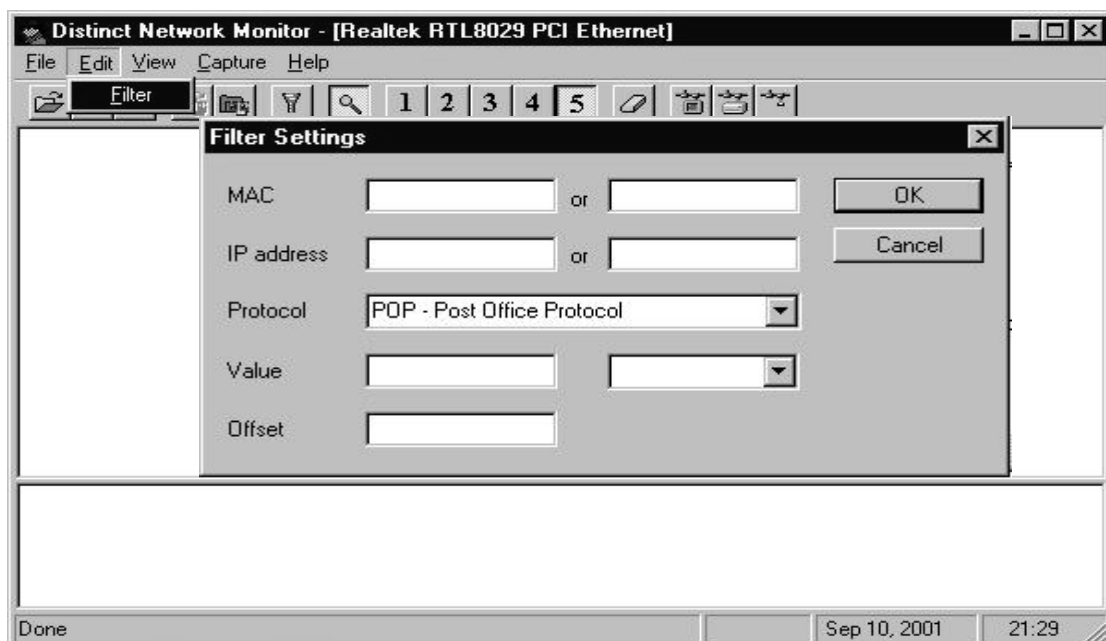
Program Sniffer yang digunakan adalah **Network Monitor** dari Distinct Corporation (<http://www.distinct.com>). Program ini merupakan versi trial yang berumur 10 hari. Di dalam komunikasi TCP/IP atau yang menggunakan model komunikasi 7 layer OSI, sebuah komputer akan mengirim data dengan alamat komputer tujuan. Pada sebuah LAN dengan topologi *bus* atau *star* dengan menggunakan **hub** yang tidak dapat melakukan **switch** (hub tersebut melakukan *broadcast*), setiap komputer dalam jaringan tersebut menerima data tersebut. Standarnya hanya komputer dengan alamat yang bersesuaian dengan alamat tujuanlah yang akan mengambil data tersebut. Tetapi pada saat sniff, komputer dengan alamat bukan alamat tujuan tetap mengambil data tersebut.

Sebelum melakukan sniff, pertama kali adalah membuka adapter (**ethernet card**), agar mengambil semua data yang melewatinya, sekalipun bukan sebagai alamat tujuan. (Gambar 2)



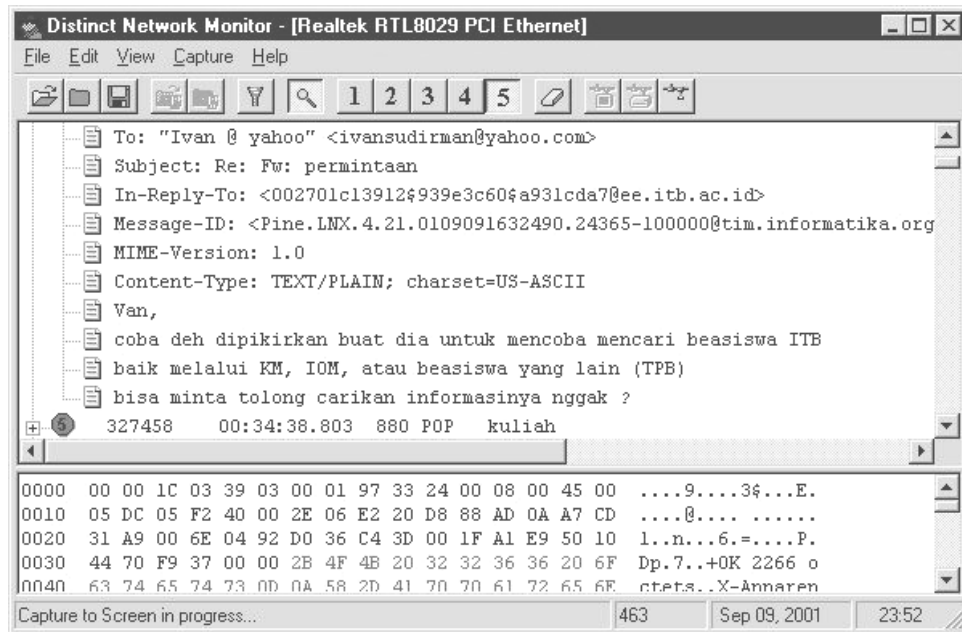
Gambar 2 Sniffer : Membuka Adapter

Biasanya data yang melewati Adapter akan sangat banyak, untuk mempermudah encarian, unakan fasilitas filter seperti terlihat pada gambar 3.



Gambar 3 : Sniffer : Filter

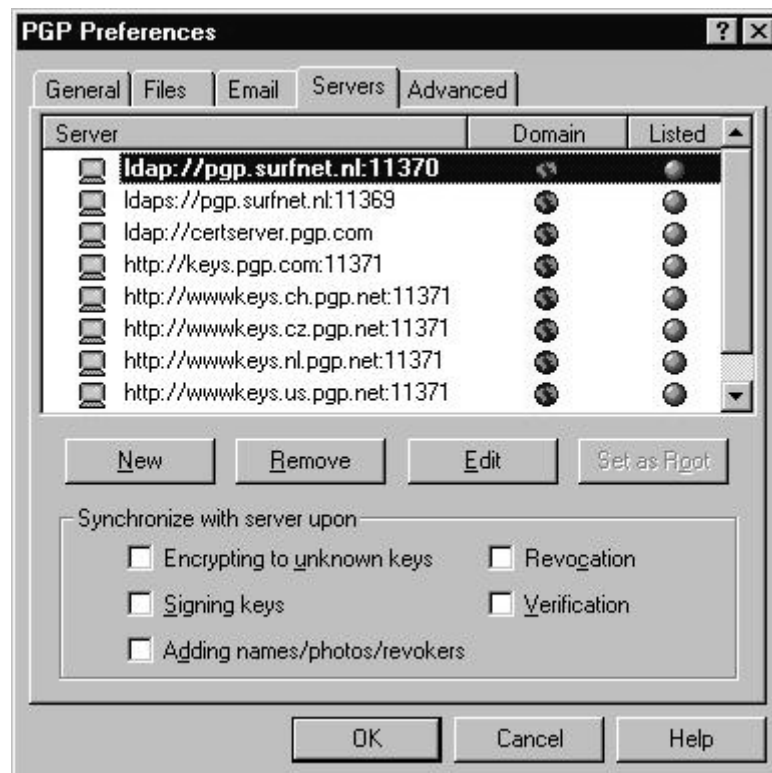
Hasil dari sniff pada protokol POP (Post Office Propotocol), dapat dilihat pada gambar 4.



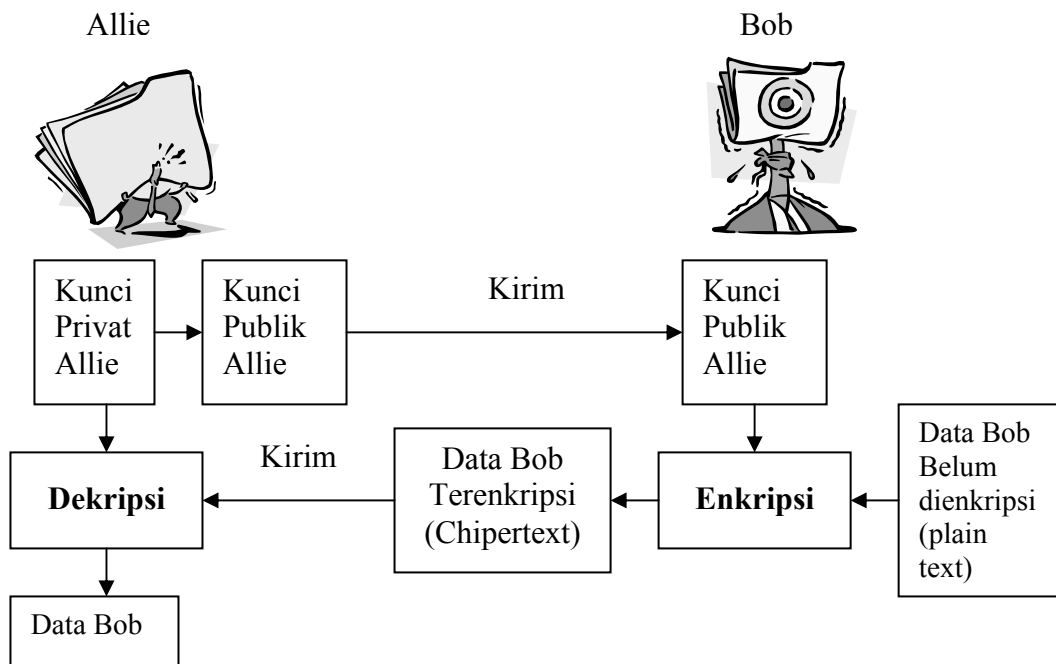
Gambar 4 : Sniffer : Data Result

Pengacakan Data

Agar data tidak dapat disadap oleh orang lain, maka data yang hendak dikirim di acak (enkripsi) terlebih dahulu. Mekanisme Enkripsi yang akan digunakan pada percobaan ini adalah mekanisme yang menggunakan kunci publik. Pada mekanisme kunci publik, terdapat 2 macam kunci yaitu kunci privat dan kunci publik. Kunci publik di hasilkan (*generate*) oleh kunci privat milik kita. Untuk selanjutnya kunci publik di sebar ke setiap orang yang akan berkomunikasi dengan kita. Di internet terdapat beberapa server kunci publik (gambar 5), yang menyimpan kunci publik dari orang yang terdaftar di server tersebut. Publik key berfungsi untuk meng-enkripsi, dan data hasil enkripsi ini hanya bisa dibuka oleh kunci privat yang menghasilkan kunci publik peng-enkripsi tadi. (Gambar 6). Cara ini juga merupakan Sistem Kriptografi Asimetris. (Enkripsi dan dekripsi menggunakan kunci yang berbeda).



Gambar 5 : Server Penyimpan Kunci Publik



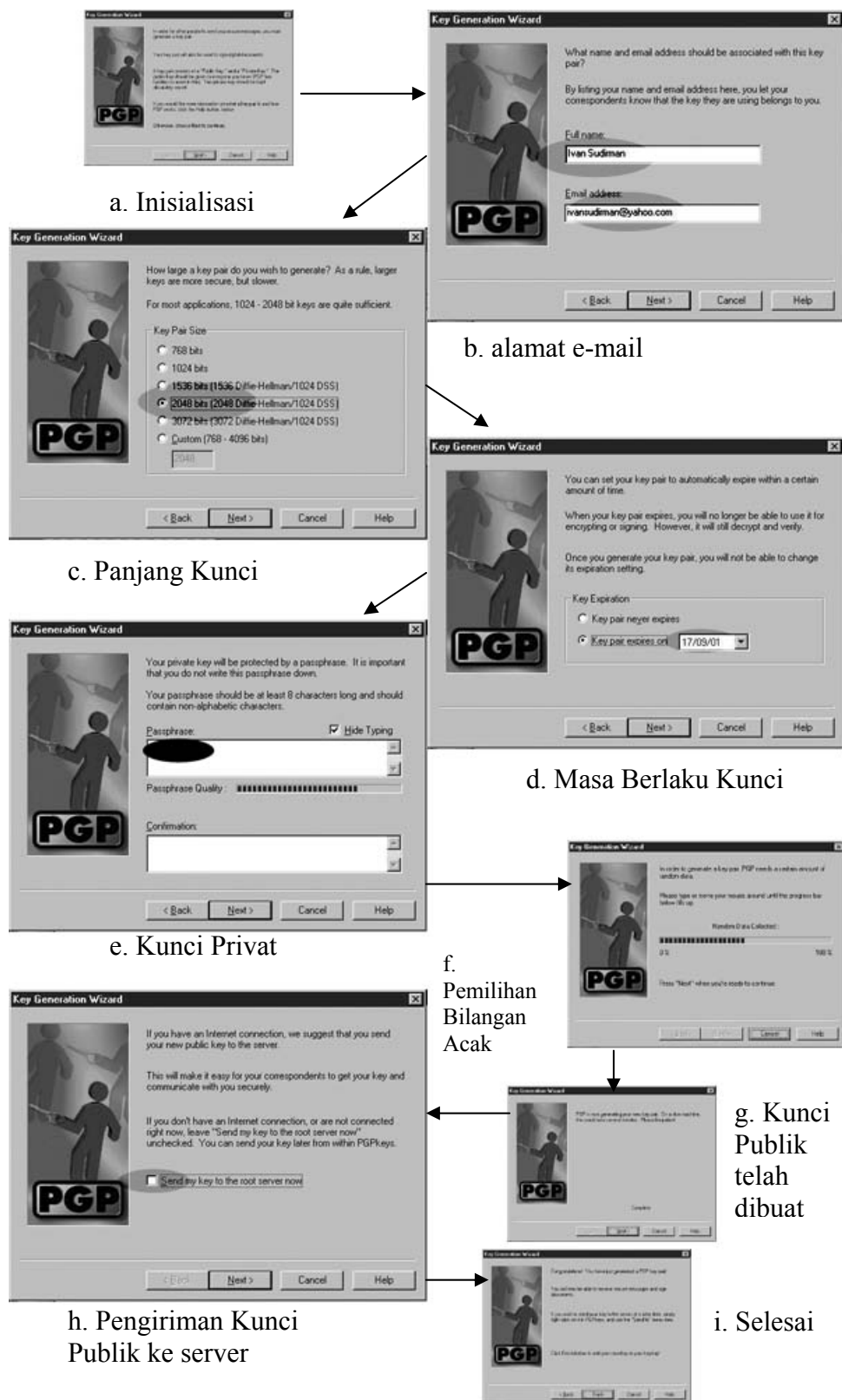
Gambar 6 : Pengiriman Data dengan Mekanisme Kunci Publik

Program yang akan digunakan pada percobaan ini adalah PGP 6.0.2 versi *freeware*, buatan Network Associates, Inc. (<http://www.nai.com>).

Pembuatan Kunci Publik

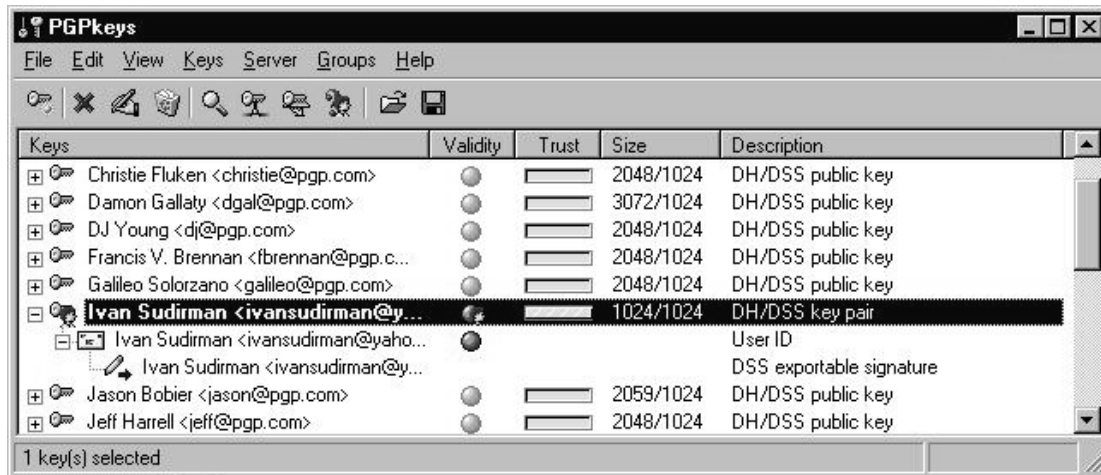
Langkah pertama sebelum dapat melakukan komunikasi dengan data yang ter-acak, adalah membuat kunci publik.

Gambar 7.1 sampai 7.9, memperlihatkan proses pembuatan kunci publik.



Gambar 7 : Pembuatan Kunci Publik

Untuk pengatur kunci gunakan tool PGPKeys (gambar 8) dan PGPTools (gambar 9) untuk meng-enkripsi serta men-dekripsi suatu *file*.



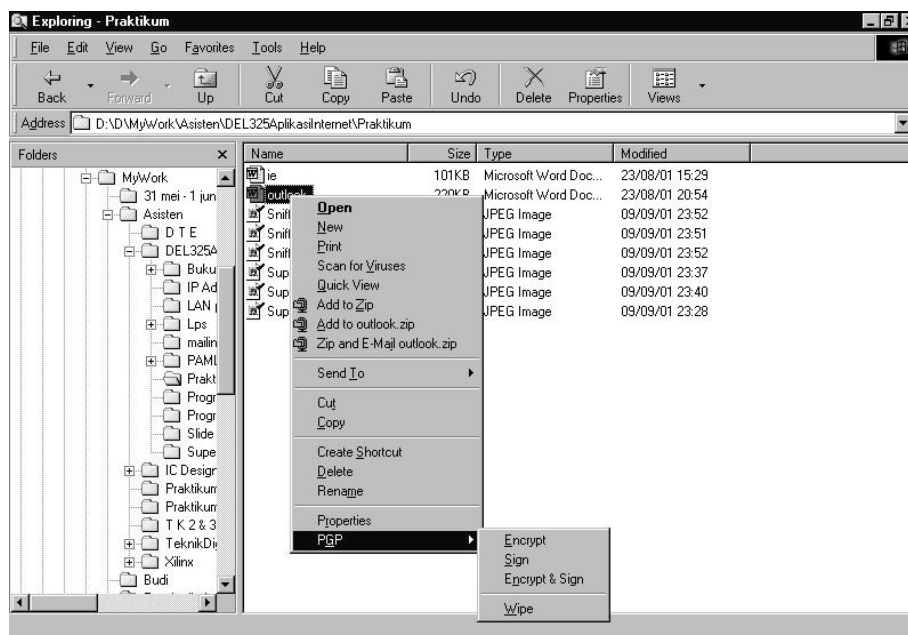
Gambar 8 . Manajemen Kunci PGP



Gambar 9. PGPTools

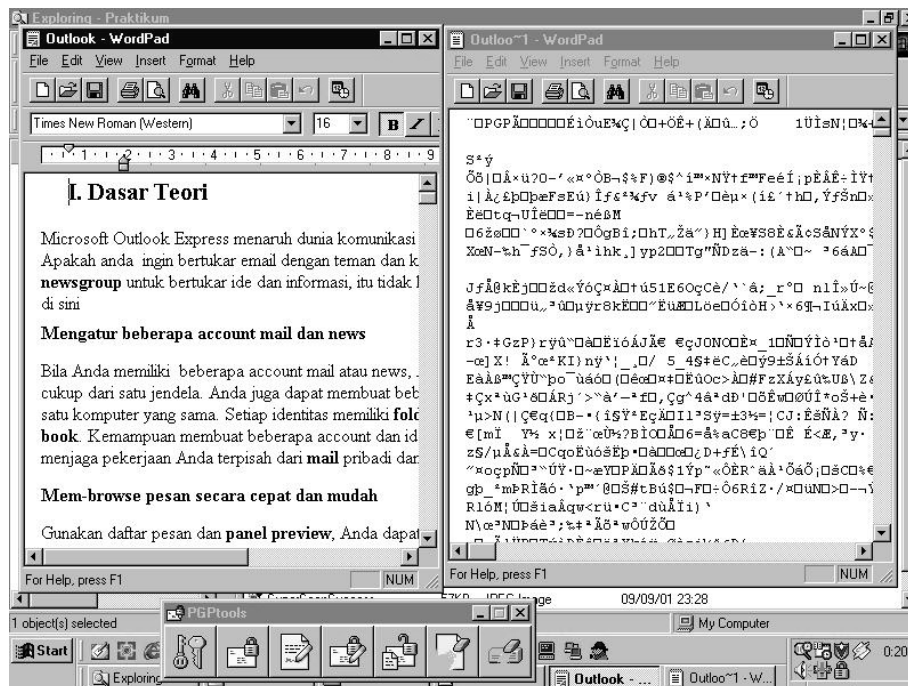
Enkripsi File

Enkripsi dapat dilakukan dengan meng-*click* kanan *file* yang akan diacak pada **Windows Explorer** (gambar 10).



Gambar 10 Enkripsi File

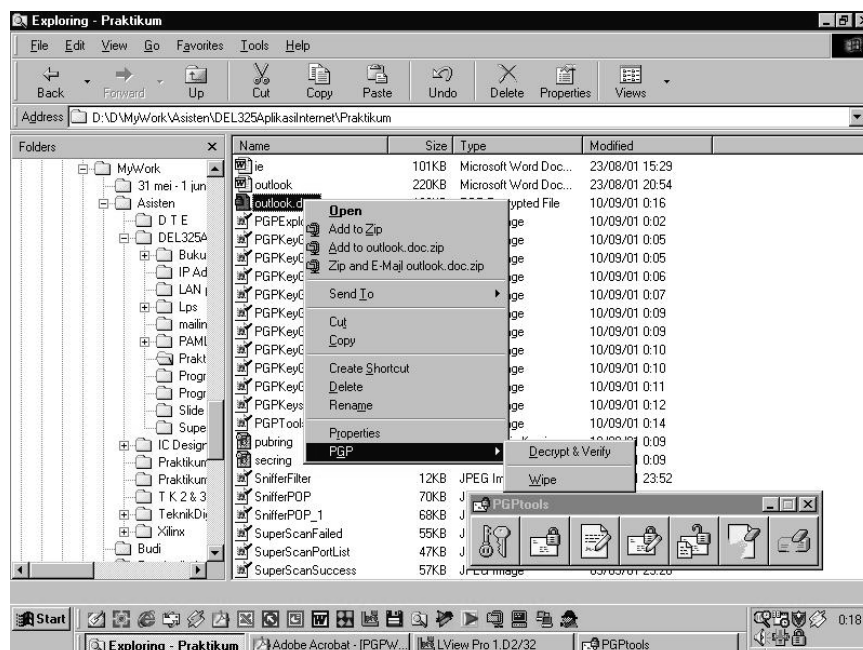
Gambar 11 memperlihatkan *file* sebelum di enkripsi dan file setelah di enkrip.



Gambar 11 Komparasi File Setelah Di-Enkripsi

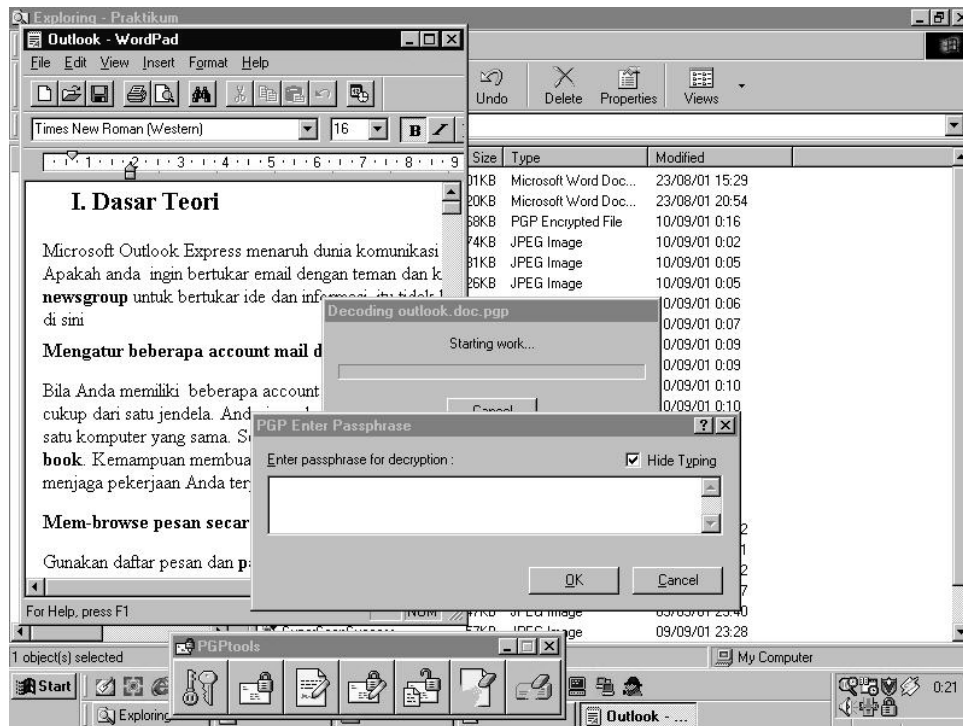
Dekripsi File

Kebalikan dari enkripsi adalah dekripsi (gambar 12).



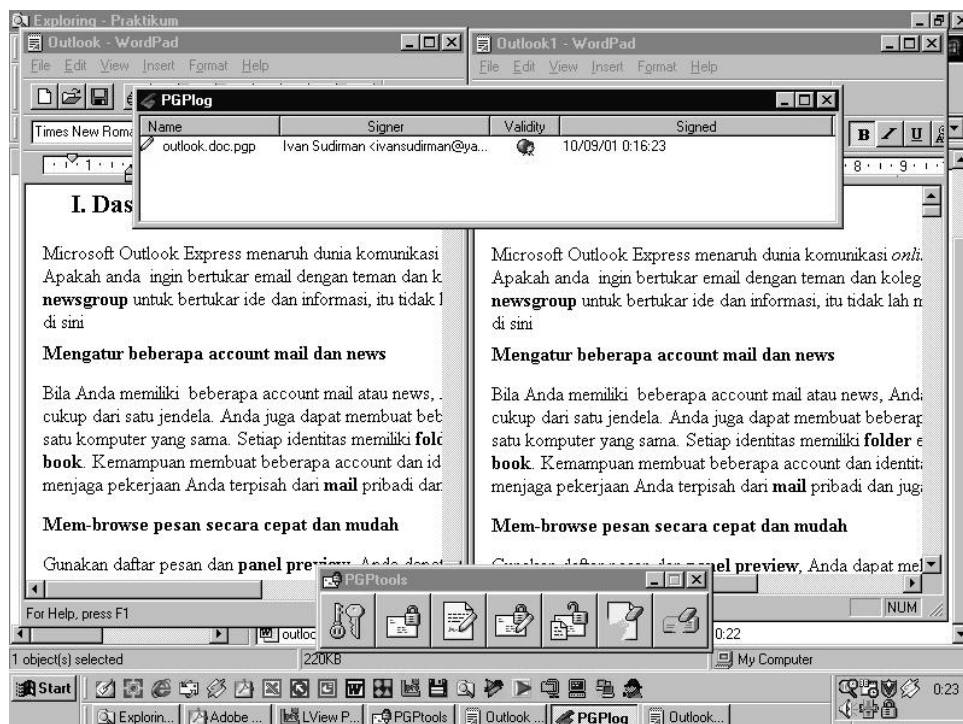
Gambar 12 Dekripsi File

Untuk mendekripsi *file* diperlukan kunci privat (gambar 13)



Gambar 13 Kunci Privat Untuk Dekripsi

Perbandingan file asli dengan file yang telah di dekripsi (gambar 14).



Gambar 14 Perbandingan File Setelah Didekripsi