

## BAB II

### LANDASAN TEORI

#### 2.1. *Security*

Keamanan berbanding terbalik dengan kenyamanan (*convenience*). Jika kita ingin membuat sebuah sistem informasi yang sangat aman, maka dia akan sulit digunakan dan bahkan menjadikannya tidak berfungsi. Sebagai contoh, jika kita ingin mengamankan sebuah kumpulan data dalam bentuk elektronik maka kumpulan data tersebut dapat penulis simpan dalam sebuah *notebook* (atau laptop) yang kemudian penulis simpan di dalam peti besi. Peti besi ini dapat disimpan di ruangan yang dijaga secara fisik oleh Polisi. Data-data tersebut mungkin aman, akan tetapi dapat dipastikan bahwa untuk mengakses data tersebut sangat tidak nyaman.

Untuk meningkatkan kenyamanan, data dapat disimpan dalam komputer yang terhubung dengan LAN (*Local Area Network*). Kenyamanan meningkat karena data-data dapat diakses dari *workstation* lain yang terhubung dengan LAN tersebut. Akan tetapi nilai keamanannya menurun karena pengguna lain di LAN yang sama memiliki potensi untuk mengakses data tersebut. Pengguna lain ini belum tentu boleh mengakses data tersebut. Untuk meningkatkan kenyamanan, maka data-data dapat dipasang pada sebuah server yang terhubung dengan Internet sehingga data-data tersebut dapat diakses darimana saja. (Tentunya dengan menggunakan *password*.)

Untuk yang terakhir ini, potensi celah keamanan menjadi lebih besar. Dari uraian di atas dapat kita lihat bahwa keamanan berbading terbalik dengan kenyamanan.

Pemilihan keamanan (atau kenyamanan) menjadi bergantung kepada pengguna.

### **2.1.1. Definisi *Security***

Melihat uraian pada bagian sebelumnya, definisi dari keamanan menjadi tidak baku.

Garfinkel dan Spafford mendefinisikan *komputer security* sebagai berikut:

“A *komputer is secure if you can depend on it and its software to behave as you expect.*”

Definisi di atas dapat diperluas untuk jaringan (*network*), aplikasi, dan sistem. Definisi tersebut juga memberikan kelonggaran atas apa yang disebut aman (*secure*). Aman menurut seseorang belum tentu aman untuk orang lain. Hal lain yang perlu mendapat perhatian adalah keamanan bergantung kepada tujuan dan nantinya terkait dengan resiko yang dapat diterima (mau diambil) oleh pengguna dan pengelola sistem informasi itu sendiri. [Wikipedia,2007]

### **2.2. Definisi *Hardening***

Proses *Hardening* adalah suatu proses *security* dalam sebuah sistem. *Hardening* dilakukan khususnya untuk memproteksi sistem dari attackers. *Hardening* bisa dilakukan secara fisik ataupun secara logik. *Hardening* juga meminimalisasikan celah-celah yang ada dalam sebuah sistem operasi, dengan

tujuan untuk memperkeras keamanan. Dengan dilakukannya *Hardening* pada sistem operasi maka akan memperkecil kemungkinan terjadinya *attack* dari luar maupun dari pengguna itu sendiri. [Wikipedia,2007]

### 2.3. Sistem Operasi

Sistem operasi merupakan sebuah penghubung antara pengguna dari komputer dengan perangkat keras komputer. Sebelum ada sistem operasi, orang hanya menggunakan komputer dengan menggunakan sinyal analog dan sinyal digital. Seiring dengan berkembangnya pengetahuan dan teknologi, pada saat ini terdapat berbagai sistem operasi dengan keunggulan masing-masing. Untuk lebih memahami sistem operasi maka sebaiknya perlu diketahui terlebih dahulu beberapa konsep dasar mengenai sistem operasi itu sendiri.

Pengertian sistem operasi secara umum ialah pengelola seluruh sumber-daya yang terdapat pada sistem komputer dan menyediakan sekumpulan layanan (*system calls*) ke pemakai sehingga memudahkan dan menyamankan penggunaan serta pemanfaatan sumber-daya sistem komputer.

Sistem komputer pada dasarnya terdiri dari empat komponen utama, yaitu perangkat-keras, program aplikasi, sistem-operasi, dan para pengguna. Sistem operasi berfungsi untuk mengatur dan mengawasi penggunaan perangkat keras oleh berbagai program aplikasi serta para pengguna.

Sistem operasi berfungsi ibarat pemerintah dalam suatu negara, dalam arti membuat kondisi komputer agar dapat menjalankan program secara benar. Untuk menghindari konflik yang terjadi pada saat pengguna menggunakan sumber-daya yang sama, sistem operasi mengatur pengguna mana yang dapat mengakses suatu sumber-daya. Sistem operasi juga sering disebut *resource allocator*. Satu lagi fungsi penting sistem operasi ialah sebagai program pengendali yang bertujuan untuk menghindari kekeliruan (*error*) dan penggunaan komputer yang tidak perlu.

### 2.3.1. Sejarah Sistem Operasi

Menurut Tanenbaum, sistem operasi mengalami perkembangan yang sangat pesat, yang dapat dibagi kedalam empat generasi:

- Generasi Pertama (1945-1955)

Generasi pertama merupakan awal perkembangan sistem komputasi elektronik sebagai pengganti sistem komputasi mekanik, hal itu disebabkan kecepatan manusia untuk menghitung terbatas dan manusia sangat mudah untuk membuat kecerobohan, kekeliruan bahkan kesalahan. Pada generasi ini belum ada sistem operasi, maka sistem komputer diberi instruksi yang harus dikerjakan secara langsung.

- Generasi Kedua (1955-1965)

Generasi kedua memperkenalkan *Batch Processing System*, yaitu Job yang dikerjakan dalam satu rangkaian, lalu dieksekusi secara berurutan. Pada

generasi ini sistem komputer belum dilengkapi sistem operasi, tetapi beberapa fungsi sistem operasi telah ada, contohnya fungsi sistem operasi ialah FMS dan IBSYS.

- Generasi Ketiga (1965-1980)

Pada generasi ini perkembangan sistem operasi dikembangkan untuk melayani banyak pemakai sekaligus, dimana para pemakai interaktif berkomunikasi lewat terminal secara on-line ke komputer, maka sistem operasi menjadi *multi-pengguna* (di gunakan banyak pengguna sekali gus) dan *multi-programming* (melayani banyak program sekaligus).

- Generasi Keempat (Pasca 1980an)

Dewasa ini, sistem operasi dipergunakan untuk jaringan komputer dimana pemakai menyadari keberadaan komputer-komputer yang saling terhubung satu sama lainnya. Pada masa ini para pengguna juga telah dinyamankan dengan *Graphical Pengguna Interface* yaitu antar-muka komputer yang berbasis grafis yang sangat nyaman, pada masa ini juga dimulai era komputasi tersebar dimana komputasi-komputasi tidak lagi berpusat di satu titik, tetapi dipecah dibanyak komputer sehingga tercapai kinerja yang lebih baik.

Sebuah sistem operasi yang baik menurut Tanenbaum harus memiliki layanan sebagai berikut: pembuatan program, eksekusi program, pengaksesan *I/O*

*Device*, pengaksesan terkendali terhadap berkas pengaksesan sistem, deteksi dan pemberian tanggapan pada kesalahan, serta akunting.

Pembuatan program yaitu sistem operasi menyediakan fasilitas dan layanan untuk membantu para pemrogram untuk menulis program; Eksekusi Program yang berarti Instruksi-instruksi dan data-data harus dimuat ke memori utama, perangkat-perangkat masukan/ keluaran dan berkas harus di-inisialisasi, serta sumber-daya yang ada harus disiapkan, semua itu harus di tangani oleh sistem operasi; Pengaksesan *I/O Device*, artinya Sistem Operasi harus mengambil alih sejumlah instruksi yang rumit dan sinyal kendali menjengkelkan agar pemrogram dapat berfikir sederhana dan perangkat pun dapat beroperasi; Pengaksesan terkendali terhadap berkas yang artinya disediakan mekanisme proteksi terhadap berkas untuk mengendalikan pengaksesan terhadap berkas; Pengaksesan sistem artinya pada pengaksesan digunakan bersama (*shared system*); Fungsi pengaksesan harus menyediakan proteksi terhadap sejumlah sumber-daya dan data dari pemakai tak terdistorsi serta menyelesaikan konflik-konflik dalam perebutan sumber-daya; Deteksi dan Pemberian tanggapan pada kesalahan, yaitu jika muncul permasalahan muncul pada sistem komputer maka sistem operasi harus memberikan tanggapan yang menjelaskan kesalahan yang terjadi serta dampaknya terhadap aplikasi yang sedang berjalan; dan Akunting yang artinya Sistem Operasi yang bagus mengumpulkan data statistik penggunaan beragam sumber-daya dan memonitor parameter kinerja.

### 2.3.2. Struktur Sistem Operasi

Pada kenyataannya tidak semua sistem operasi mempunyai struktur yang sama. Namun menurut Avi Silberschatz, Peter Galvin, dan Greg Gagne, umumnya sebuah sistem operasi modern mempunyai komponen sebagai berikut:

#### ✦ **Manajemen Proses.**

Proses adalah keadaan ketika sebuah program sedang di eksekusi. Sebuah proses membutuhkan beberapa sumber daya untuk menyelesaikan tugasnya. sumber daya tersebut dapat berupa *CPU time*, memori, berkas-berkas, dan perangkat-perangkat I/O.

Sistem operasi bertanggung jawab atas aktivitas-aktivitas yang berkaitan dengan manajemen proses seperti:

- Pembuatan dan penghapusan proses pengguna dan sistem proses.
- Menunda atau melanjutkan proses.
- Menyediakan mekanisme untuk proses sinkronisasi.
- Menyediakan mekanisme untuk proses komunikasi.
- Menyediakan mekanisme untuk penanganan *deadlock*.

#### ✦ **Manajemen Memori Utama.**

Memori utama atau lebih dikenal sebagai memori adalah sebuah *array* yang besar dari *word* atau *byte*, yang ukurannya mencapai ratusan, ribuan, atau bahkan jutaan. Setiap *word* atau *byte* mempunyai alamat tersendiri.

Memori Utama berfungsi sebagai tempat penyimpanan yang akses datanya digunakan oleh CPU atau perangkat I/O. Memori utama termasuk tempat penyimpanan data yang sementara (*volatile*), artinya data dapat hilang begitu sistem dimatikan.

Sistem operasi bertanggung jawab atas aktivitas-aktivitas yang berkaitan dengan manajemen memori seperti:

- Menjaga *track* dari memori yang sedang digunakan dan siapa yang menggunakannya.
- Memilih program yang akan di-*load* ke memori.
- Mengalokasikan dan meng-dealokasikan ruang memori sesuai kebutuhan.

#### 🔧 **Manajemen *Secondary-Storage*.**

Data yang disimpan dalam memori utama bersifat sementara dan jumlahnya sangat kecil. Oleh karena itu, untuk menyimpan keseluruhan data dan program komputer dibutuhkan *secondary-storage* yang bersifat permanen dan mampu menampung banyak data. Contoh dari *secondary-storage* adalah *harddisk*, disket, dll.

Sistem operasi bertanggung-jawab atas aktivitas-aktivitas yang berkaitan dengan *disk-manajemen* seperti: *free-space manajemen*, alokasi penyimpanan, penjadualan disk.



#### 🔥 **Manajemen Sistem I/O.**

Sering disebut *device manager*. Menyediakan "*device driver*" yang umum sehingga operasi I/O dapat seragam (membuka, membaca, menulis, menutup). Contoh: pengguna menggunakan operasi yang sama untuk membaca berkas pada *hard-disk*, CD-ROM dan *floppy disk*.

Komponen Sistem Operasi untuk sistem I/O:

- *Buffer*: menampung sementara data dari/ ke perangkat I/O.
- *Spooling*: melakukan penjadwalan pemakaian I/O sistem supaya lebih efisien (antrian dsb.).
- Menyediakan *driver* untuk dapat melakukan operasi "rinci" untuk perangkat keras I/O tertentu.

#### 🔥 **Manajemen Berkas.**

Berkas adalah kumpulan informasi yang berhubungan sesuai dengan tujuan pembuat berkas tersebut. Berkas dapat mempunyai struktur yang bersifat hirarkis (direktori, volume, dll.). Sistem operasi bertanggung-jawab:

- Pembuatan dan penghapusan berkas.
- Pembuatan dan penghapusan direktori.
- Mendukung manipulasi berkas dan direktori.
- Memetakan berkas ke *secondary storage*.

### 🚩 **Command-Interpreter system.**

Sistem Operasi menunggu instruksi dari pengguna (*command driven*). Program yang membaca instruksi dan mengartikan *control statements* umumnya disebut: *control-card interpreter*, *command-line interpreter*, dan *UNIX shell*. *Command-Interpreter System* sangat bervariasi dari satu sistem operasi ke sistem operasi yang lain dan disesuaikan dengan tujuan dan teknologi *I/O devices* yang ada. Contohnya: *CLI*, *Windows*, *Pen-based (touch)*, dan lain-lain.

### 🚩 **Layanan Sistem Operasi**

Eksekusi program adalah kemampuan sistem untuk "*load*" program ke memori dan menjalankan program. Operasi I/O: pengguna tidak dapat secara langsung mengakses sumber daya perangkat keras, sistem operasi harus menyediakan mekanisme untuk melakukan operasi I/O atas nama pengguna. Sistem manipulasi berkas adalah kemampuan program untuk operasi pada berkas (membaca, menulis, membuat, and menghapus berkas). Komunikasi adalah pertukaran data/ informasi antar dua atau lebih proses yang berada pada satu komputer (atau lebih). Deteksi *error* adalah menjaga kestabilan sistem dengan mendeteksi "*error*", perangkat keras mau pun operasi. Efisiensi penggunaan sistem:

- *Resource allocator* adalah mengalokasikan sumber-daya ke beberapa pengguna atau *job* yang jalan pada saat yang bersamaan.

- Proteksi menjamin akses ke sistem sumber daya dikendalikan (pengguna dikontrol aksesnya ke sistem).
- *Accounting* adalah merekam kegiatan pengguna, jatah pemakaian sumber daya (keadilan atau kebijaksanaan).

### 📌 **System Calls**

*System call* menyediakan interface antara program (program pengguna yang berjalan) dan bagian OS. *System call* menjadi jembatan antara proses dan sistem operasi. *System call* ditulis dalam bahasa *assembly* atau bahasa tingkat tinggi yang dapat mengendalikan mesin (C). Contoh: UNIX menyediakan *system call*: *read*, *write* => operasi I/O untuk berkas.

Sering pengguna program harus memberikan data (parameter) ke OS yang akan dipanggil. Contoh pada UNIX: *read (buffer, max\_size, file\_id)*; Tiga cara memberikan parameter dari program ke sistem operasi:

- Melalui registers (sumber daya di CPU).
- Menyimpan parameter pada data struktur (table) di memori, dan alamat table tsb ditunjuk oleh *pointer* yang disimpan di register.
- *Push (store)* melalui "*stack*" pada memori dan OS mengambilnya melalui *pop* pada *stack* tsb.

## ✦ Mesin Virtual

Sebuah mesin virtual (*Virtual Machine*) menggunakan misalkan terdapat sistem program  $\Rightarrow$  control program yang mengatur pemakaian sumber daya perangkat keras. Control program = trap *System call* + akses ke perangkat keras. Control program memberikan fasilitas ke proses pengguna. Mendapatkan jatah CPU dan memori. Menyediakan *interface* "identik" dengan apa yang disediakan oleh perangkat keras  $\Rightarrow$  *sharing devices* untuk berbagai proses.

Mesin Virtual (MV) (MV)  $\Rightarrow$  control program yang minimal MV memberikan ilusi *multitasking*: seolah-olah terdapat prosesor dan memori eksklusif digunakan MV. MV memilah fungsi *multitasking* dan implementasi *extended machine* (tergantung proses pengguna)  $\Rightarrow$  flexible dan lebih mudah untuk pengaturan. Jika setiap pengguna diberikan satu MV  $\Rightarrow$  bebas untuk menjalankan OS (kernel) yang diinginkan pada MV tersebut. Potensi lebih dari satu OS dalam satu komputer. Contoh: IBM VM370: menyediakan MV untuk berbagai OS: CMS (interaktif), MVS, CICS, dll. Masalah: Sharing disk  $\Rightarrow$  OS mempunyai sistem berkas yang mungkin berbeda. IBM: *virtual disk* (minidisk) yang dialokasikan untuk pengguna melalui MV.

Konsep MV menyediakan proteksi yang lengkap untuk sumberdaya sistem, dikarenakan tiap MV terpisah dari MV yang lain. Namun, hal tersebut menyebabkan tidak adanya *sharing* sumberdaya secara langsung. MV merupakan alat yang tepat untuk penelitian dan pengembangan sistem operasi.

Konsep MV susah untuk diimplementasi sehubungan dengan usaha yang diperlukan untuk menyediakan duplikasi dari mesin utama. [Tenenbaum,2005]

## **2.4. Program BIOS**

Program BIOS adalah program kontroler standard yang terpasang dalam *motherboard*. Program ini disimpan dalam chip IC yang disebut ROM (*read only memory*) dan selalu bekerja mengontrol *hardware* yang terpasang pada *motherboard* meskipun komputer tidak digunakan (dimatikan). Hal ini dimungkinkan oleh adanya batere kecil yang terpasang pada *motherboard*. Orang sering menyebut chip IC ini dengan sebutan IC ROM BIOS. Pada saat selesai merakit komputer, program ini harus diatur sesuai dengan jenis-jenis peralatan elektronik yang akan dihubungkan dengan *motherboard*. Dengan demikian selanjutnya setiap saat program BIOS selalu mengontrol perangkat keras yang terpasang meskipun komputer anda dimatikan. Kontrol setiap saat ini memungkinkan komputer selalu siap pakai pada saat komputer anda dihidupkan. Pada kondisi batere kecil lemah, biasanya pengaturan pada program BIOS ini hilang, sehingga anda sulit mempergunakan komputer. Anda harus kembali mengatur ulang program BIOS seperti yang anda lakukan pada waktu merakit komputer.

### **2.4.1. Utility Setup BIOS**

Memiliki *Setting hardware* yang berlimpah dimana berfungsi untuk kinerja dari *hardware* tersebut, diantaranya:

- **Optimalisasi dan kompatibilitas setting:** *Utility Setup* sering berisi *setting* untuk *kinerja hardware*. Kadang-kadang ini tidak diatur secara optimal di pabrik, utamanya pada PC yang dibuat berdasarkan pesanan atau generik. *Scan* seluruh program *Setup*. *Setting* modus AGP dan *setting* DMA merupakan kandidat utama untuk dioptimalisasi. *Setting-setting* ini juga berguna untuk *troubleshooting hardware* yang baru diinstall: Menonaktifkan atau menurunkan *setting* yang diberikan bisa memicu ketidak-kompatibelan kartu grafis, *CD-ROM drive*, atau peranti lain yang terjadi sebelumnya.
- **Setting harddisk:** Anda akan menemukan tabel, biasanya pada halaman kedua atau *Advanced* dan dengan judul "IDE", yang mendaftarkan semua parameter konfigurasi untuk EIDE *harddisk* yang secara langsung terkoneksi ke *motherboard*. (SCSI *harddisk* dan EIDE *harddisk* yang tidak punya kartu ekspansi akan menjalankan program konfigurasinya sendiri.) Sementara kebanyakan PC yang dibuat selama beberapa tahun terakhir dapat dengan mulus mendeteksi dan mengkonfigurasi *harddisk* baru, sebagian memerlukan instalasi manual. Baca your dokumentasi disk baru Anda mengenai prosedur tersebut dan gunakan opsi-opsi *on-screen* untuk membuat perubahan di *table* ini.
- **Floppy disk:** Opsi ini memungkinkan Anda menyetel tipe *floppy drive* (3,5 inci, 1,44MB, sebagai contoh) yang telah Anda tetapkan sebagai drive A: atau B: Anda. Ini merupakan *setting* yang perlu diperiksa jika

Anda mengalami problem *floppy-drive*. Sebagian utiliti Setup mempunyai *setting security 'Floppy Read only'* tersendiri yang mencegah data Anda dituliskan ke *floppy disk* dan dihapus.

- **Urutan booting:** *Setting* ini menentukan mana yang pertama kali dilihat PC ketika instruksi *boot-up*. Contoh: `A: kemudian C:`, `C: kemudian A:`, atau `C:, Zip drive`. Untuk *booting* dari *CD-ROM drive*, Zip, atau LS-120, Anda sepertinya harus mengubah *setting* ini.
- **Proteksi password:** Bila ini diaktifkan, BIOS akan menanyakan *password* sebelum *booting up*. Sangat berhati-hatilah dengan yang satu ini: Bila melupakan *password*, Anda terpaksa mereset *jumper motherboard* atau mencopot batere CMOS, yang akan menyebabkan Anda kehilangan semua *setting*, atau Anda bahkan harus membeli *motherboard* baru.
- **Setting IRQ:** Bila Anda butuh *setting* IRQ ekstra untuk *hardware* baru, Anda perlu membebaskan satu IRQ dengan menonaktifkan *feature* yang tidak digunakan, seperti *port serial*, *port paralel*, atau *port* USB.
- **Setting port paralel:** Pilih modus yang paling cocok untuk *hardware*. Modus ECP atau EPP dapat sangat mempercepat *printer* dan peranti lain.
- **Kipas RPM dan temperatur CPU:** Dua parameter kritis ini sebaiknya dicentang secara periodik untuk memastikan mereka berfungsi dengan benar. [JPN.Sumarno,2003]

## 2.5. TCP/IP

TCP/IP merupakan protokol jaringan komputer terbuka dan bisa terhubung dengan berbagai jenis perangkat keras dan lunak. TCP terdiri beberapa layer atau lapisan yang memiliki fungsi tertentu dalam komunikasi data. Setiap fungsi dari layer selain dapat bekerjasama dengan layer pada tingkat lebih rendah atau lebih tinggi, juga bisa berkomunikasi dengan layer sejenis pada *remote host* (peering). IP adalah jantung TCP/IP memiki peran sebagai pembawa data yang independen. Semua dokumen TCP/IP dalam bentuk *public document* IEN dan RFC. IP dibagi atas kelas network A,B, dan C. Sedangkan kelas D untuk keperluan *reverse IP* yang boleh diabaikan. IP ditulis dalam bilangan desimal dari 0 sampai 255. Data yang mengalir antar *layer* atau antar *host* dienkapsulasi dan diberi *header* agar tiap layer bisa memprosesnya. Sebuah *host* tidak tahu alamat IP *gateway* di *network* lain, tetapi data mengalir ke *host* tujuan di *network* lain melalui *gateway network*-nya setelah diberi penentuan ruting alamat IP.

TCP/IP merupakan jenis protokol yang memungkinkan komputer berkomunikasi dengan komputer lain melalui suatu jaringan yang sering disebut internet. TCP/ IP bekerja untuk mengatur bagaimana komputer terhubung ke internet dan mengatur pertukaran data yang terjadi pada komputer tersebut.

TCP /IP terdapat beberapa protokol kecil yang menangani masalah komunikasi antar komputer, diantaranya adalah :

- ✚ TCP ( *Transmission Control Protocol* ) mengatur komunikasi antar aplikasi.



- ✚ **UDP ( *Pengguna Datagram Protokol* )** mengatur komunikasi antar aplikasi.
- ✚ **IP ( *Internet Protokol* )** mengatur komunikasi antar komputer.
- ✚ **ICMP ( *Internet Control Message Protokol* )** untuk mengatasi kesalahan dan statistik.
- ✚ **DHCP ( *Dynamic Host Configuration Protokol* )** untuk pengalamatan dinamik.

Sebelum TCP/IP digunakan, tentukan lebih dahulu *IP-address* dan *subnet* yang akan digunakan. Jika pada suatu jaringan disertakan dengan *IP-address* yang berbeda jaringan, maka komputer tersebut tidak dapat berkomunikasi dengan jaringan yang sudah ada. Untuk menghubungkan 2 jaringan yang berbeda inilah dibutuhkan alat bantu yang disebut dengan *ROUTER*. Router dapat berbentuk mesin *Windows NT*, *UNIX*, atau real router seperti *CISCO*. Router ini mempunyai 2 IP yang berbeda atau lebih, sesuai dengan jaringan yang dihubungkannya. Router pada konfigurasi TCP/IP disebut sebagai *GATEWAY*, yaitu pintu untuk mencapai jaringan diluar jaringan local. Testing bisa dilakukan dengan instruksi *IPconfig*.

```
Command Prompt
C:\>ipconfig /all

Windows IP Configuration

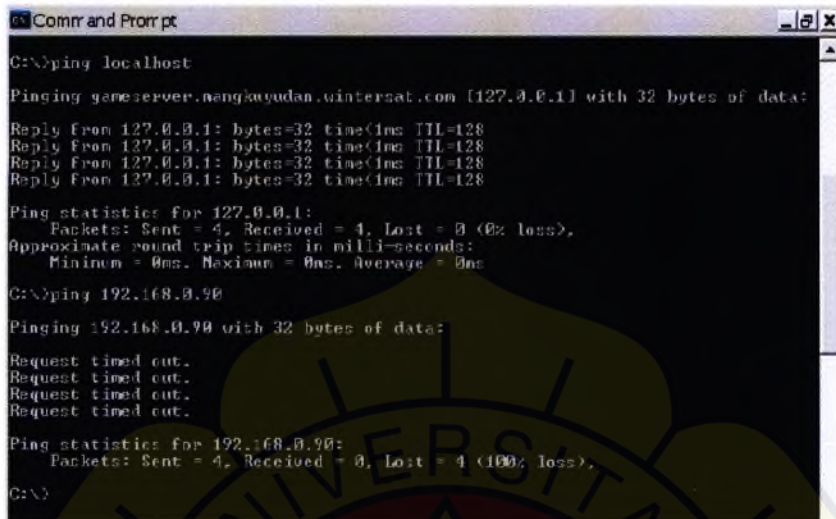
Host Name . . . . . : game server
Primary Dns Suffix . . . . . : mangkuyudan.wintersat.com
Node Type . . . . . : Unknown
IP Routing Enabled. . . . . : No
DNS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : mangkuyudan.wintersat.com

Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix . . . . . :
Description . . . . . : Realtek RTL8139 Family PCI Fast Eth
enet NIC
Physical Address . . . . . : 00-C0-26-80-FB-C0
Dhcp Enabled. . . . . : No
IP Address. . . . . : 192.168.0.80
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.0.254
DNS Servers . . . . . : 192.168.0.254
                          64.110.163.249
```

Gambar 2.1 IP Config

Perintah IPconfig digunakan untuk melihat indikasi pada konfigurasi IP yang terpasang pada Komputer kita, dari gambar diatas kita dapat melihat beberapa informasi penting setelah kita menjalankan perintah *IPConfig* pada jendela command prompt dikomputer kita, misalnya adalah kita bias melihat *Host Name*, *primary* DNS jaringan, *physical Address* dan sebagainya. Harus diingat bahwa perintah ini dapat dijalankan dengan baik apabila telah terpasang *Network Card* di komputer anda. *Ipconfig* menampilkan informasi berdasarkan *Network Card* yang terpasang. Untuk mendeteksi apakah hubungan komputer dengan jaringan sudah berjalan dengan baik, utilitas ping dapat digunakan sebagai berikut:



```
Command Prompt
C:\>ping localhost
Pinging ganeserver.nengkojudan.uintersat.com [127.0.0.1] with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>ping 192.168.0.90
Pinging 192.168.0.90 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.0.90:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\>
```

Gambar 2.2 Utilitas Ping

Utilitas ping digunakan untuk mengecek apakah jaringan kita sudah bisa berfungsi dan terhubung dengan baik, misalkan pada gambar diatas terlihat perintah ping *LocalHost*, jika kita melihat ada keluar pesan *Replay form No IP* ( 127.0.0.1 ) besarnya berapa bites dan waktunya berapa detik itu menandakan bahwa perintah untuk menghubungkan ke *LocalHost* dapat berjalan dan diterima dengan baik, namun seandainya jika kita melakukan ping untuk nomor IP yang tidak dikenal seperti gambar diatas maka akan dikeluarkan pesan *Request Time Out* yang berarti nomor IP tidak dikenal dalam jaringan tersebut ( ping 192.168.0.90 ). Untuk lebih jelasnya dapat melihat hasil gambar diatas. TTL adalah *Time To Live*, yaitu batasan waktu agar paket data tersebut tidak mengambang dijaringan ( karena destinasi/tujuan tidak ditemukan ). [Onno W Purbo,2001]

## **2.6. Group Policy**

*Policy* bisa diilustrasikan sebagai peraturan – peraturan yang harus dijalankan. Peraturan dalam hal ini ditujukan untuk pengguna, walaupun ditujukan untuk pengguna tidak bisa secara langsung mengenakan peraturan tersebut kepada pengguna. Peraturan ini seperti menjalankan script tertentu setiap *logon/logoff*, *startup/shutdown*, kompleksitas pass-word yang digunakan dan lain-lain. Karena namanya adalah *Group Policy*, sehingga sebelum peraturan tersebut dikenakan kepada suatu group baru kemudian ke *pengguna*. Peraturan atau policy yang harus dijalankan tersebut dinamakan juga sebagai *Group Policy Object* (GPO).

*Group Policy* secara umum dapat dibedakan menjadi dua macam yaitu *Local Group Policy* dan *Domain Group Policy*.

### **2.6.1. Local Group Policy**

Selain terdapat pada *Windows Server 2003*, *Local Group Policy* juga terdapat pada *Windows* yang lainnya yaitu *Windows XP* dan *Windows 2000*. *Local Group Policy* atau *Local GPO* secara otomatis akan tersedia saat komputer terhubung ke domain atau tidak terhubung ke domain. *Local GPO* digunakan untuk pengaturan suatu konfigurasi dan untuk menerapkan peraturan pada komputer local.

### **2.6.2. Domain Group Policy**

Domain Group Policy ini hanya ada untuk sistem operasi di lingkungan domain, dalam hal ini tentunya *Windows Server 2003* yang telah dipasang *Active*

*Directory*. Jika pada komputer lokal namanya adalah Local GPO, pada komputer yang terhubung ke suatu domain namanya adalah Domain GPO. Pada dasarnya bila telah melakukan instalasi *Active Directory* telah memiliki Domain GPO, tetapi Domain GPO tersebut belum aktif. *Group Policy* yang ada saat menginstalasi *Active Directory* ada dua macam yaitu :

✚ *Default Domain Policy*

*Default Domain Policy* ini berlaku untuk semua pengguna dan komputer yang ada di dalam sebuah domain, dan termasuk di dalamnya adalah unit organisasi. GPO pada *Default Domain Policy* ini sifatnya *inheritance* atau biasa disebut dengan turunan.

✚ *Default Domain Controller Policy*

*Default Domain Controller Policy* ini berbeda dengan *Default Domain Policy*. Jika pada *Default Domain Policy* digunakan untuk semua pengguna dan komputer yang ada dalam domain maka pada GPO ini digunakan untuk unit organisasi dalam *Domain Controller* yang telah ada. [S'to,2004]

## 2.7. Registry

*Registry* merupakan database yang digunakan untuk menyimpan semua *setting* dan informasi *hardware*, *software* dan berbagai *preferences* untuk *Windows* 32 bit, termasuk *Window* 95, 98, NT, Milenium, 2000 dan 2003. Salah satu contohnya adalah misalnya seseorang mengganti asosiasi *file* atau menginstall program, maka perubahan *setting* tersebut akan dituliskan pada

registry. Contoh lainnya adalah *men-disable Display Properties*, menyembunyikan berbagai menu pada Menu *Start*.

Selain sebagai tempat untuk menyimpan informasi sistem operasi *Windows* sendiri, *registry* juga digunakan sebagai tempat untuk menyimpan berbagai informasi *setting* dan konfigurasi pada aplikasi atau program. Misalnya WinZip menggunakan *registry* untuk menyimpan informasi *toolbar*, aplikasi untuk membuka *file (viewer)*, *pengguna name*, *serial number*, dan lain-lain. *Registry* diletakkan pada dua buah *hidden file* yaitu *pengguna.dat* dan *system.dat* yang terletak pada directory *Windows* untuk Win 95/98/Me dan pada directory *Windows/System32/Config* untuk *Windows NT*. Selain menggunakan *registry (system.dat dan pengguna.dat)*, *Windows* juga menyimpan informasi *setting* tertentu pada *file msdos.sys, system.ini dan win.ini*.

*Registry* terdiri dari beberapa bagian yang disebut *key* atau kunci. Terdapat enam macam *key* utama pada *registry*, yaitu :

📁 *HKEY\_CLASSES\_ROOT*,

Berisi semua tipe *file* beserta assosiasinya yang masing-masing tipe *file* tersebut akan mempunyai *subkey* sendiri-sendiri.

📁 *HKEY\_CURRENT\_PENGGUNA*

Berisi informasi tentang pengguna yang sedang log in pada saat itu.

Terdiri dari *subkey*, antara lain :

- a. AppEvents, berisi *setting* untuk sound events.
- b. Control Panel, berisi *setting* control panel.

- c. *InstallLocationMRU*, berisi path/lokasi master *Windows* (waktu pertama kali menginstall *Windows*).
- d. *Network*, berisi informasi yang berhubungan dengan *Network* atau jaringan.
- e. *Software*, berisi tentang *setting* atau konfigurasi software untuk pemakai yang logon pada saat itu.
- f. *RemoteAccess*, berisi informasi tentang *Dial up Networking*.

#### 🚩 *HKEY\_LOCAL\_MACHINE*

Berisi informasi tentang *hardware* dan *setting software* yang berlaku untuk semua pengguna. Terdiri dari *subkey*, antara lain :

- a. *Enum*, berisi informasi tentang *hardware*, misalnya tentang monitor.
- b. *Hardware*, berisi informasi tentang *port serial*.
- c. *Network*, berisi informasi yang berhubungan dengan *network* untuk yang sedang aktif atau *logon*.
- d. *Software*, berisi informasi dan *setting software*.

#### 🚩 *HKEY\_PENGGUNAS*

Berisi informasi tentang desktop dan pengguna *setting* untuk tiap pengguna yang berhak login ke komputer tersebut. Tiap pengguna mempunyai sebuah *subkey*. Jika hanya terdapat satu pengguna maka nama *subkey* tersebut adalah ".*default*"

#### 🚩 *HKEY\_CURRENT\_CONFIG*

Berisi informasi tentang konfigurasi *hardware*, berhubungan dengan *HKEY\_LOCAL\_MACHINE*.

#### 🚩 *HKEY\_DYN\_DATA*

Berisi informasi tentang *plug and play*. [Klik-kanan.com,2006]

### 2.8. *File Sistem*

Pada *Windows 2003*, *File Sistem* yang disarankan adalah NTFS. Jenis *file sistem* ini lebih cepat, aman, dan mendukung ukuran hardisk yang lebih besar dibandingkan dengan FAT dan FAT32. Jika pada *Windows 95* bisa melakukan booting dari disket kemudian mengakses drive c secara langsung, maka hal yang sama tidak dapat dilakukan pada sistem NTFS. NTFS yang digunakan pada *Windows 2003* adalah NTFS yang lebih baru dibandingkan dengan NT dan 2000. NTFS versi 3.1 ini telah mendukung kuota, enkripsi *file* dan lain-lain. NTFS adalah satu-satunya *file sistem* yang harus digunakan ketika ingin menggunakan *Active Directory* karena sistem FAT tidak didukung. [S'to,2004]