BAB I

LANDASAN TEORI

1.1 Sistem Operasi

Sistem operasi adalah kumpulan program komputer yang mengelola sumber daya perangkat keras dan perangkat lunak komputer OPE[**?**]. Sistem operasi merespon proses yang diberikan dengan cara mengelola proses dan sumber daya sistem yang dibutuhkan oleh pemakai dan proses pada program. Sebagai dasar umum dari perangkat lunak, sistem operasi melakukan tugas dasar seperti mengkontrol dan mengalokasikan, mengatur prioritas permintaan proses, mengontrol perangkat input dan output, memfasilitasi sistem jaringan dan memanajemen filesistem. Kebanyakan sistem operasi terdapat aplikasi yang menyediakan antarmuka kepada sumber daya yang dikelola oleh sistem operasi. Aplikasi ini memiliki interpreter perintah sebagai dasar antarmuka dengan user, tetapi baru-baru ini hampir semua sistem operasi menggunakan grafik sebagai antarmuka dengan user dengan alasan untuk mempermudah penggunaan sistem operasi. Sistem operasi berada pada lapisan atas dalam komputer, pada posisi sebelum aplikasi atau sama dengan aplikasi dan sesudah kernel. Berikut adalah gambaran umum dari lapisan arsitektur komputer yang menunjukan posisi sistem operasi dan kernel :



Gambar 1.1: Lapisan Arsitektur Komputer (sumber KER[**?**])

Selain sistem operasi yang telah dijelaskan dan kernel yang akan dibahas pada subbab tersendiri, lapisan lainnya akan penulis bahas berikut ini :

1. **Aplikasi** adalah program yang menggunakan sistem komputer atas tugas yang diberikan atau ingin dilakukan oleh user terhadap sistem komputer APP[**?**]. Aplikasi biasanya tidak langsung berinteraksi dengan perangkat keras sistem komputer tetapi berinteraksi dengan sistem operasi yang memanajemen perangkat keras yang tersedia. Contoh program yang termasuk kedalam aplikasi adalah program pengolah kata MS Office Word.

2. **Asembler** adalah sebuah ”penterjemah” dari program-program yang dibuat oleh bahasa pemprograman tingkat tinggi seperti C++ kedalam bahasa yang lebih dimengerti oleh sistem komputer ASS[**?**].

3. **Firmware** adalah perangkat lunak yang terletak pada perangkat keras. Firmware biasanya disimpan dalam flash Read Only Memory (ROM) atau sebagai file image binari yang dapat di update oleh user FIR[**?**]. Digunakan sebagai pengatur perangkat keras agar mampu berjalan secara independen, contohnya firmware pada modem ADSL yang menyebabkan modem dapat berfungsi dengan baik walaupun tidak ada user yang mengaturnya.

4. **Perangkat keras** adalah bagian fisik pada sistem komputer termasuk didalamnya sirkuit elektronik COM[**?**]. Yang membedakan dengan perangkat lunak adalah perangkat lunak berjalan didalam perangkat keras. Perangkat keras jarang sekali diganti tidak seperti perangkat lunak yang mudah dibuat, dirubah, atau dihapus dari komputer.

1.1.1 Sejarah Sistem Operasi

Pada masa awal sistem komputer, komputer tidak dilengkapi oleh sistem operasi. Komputer hanya bekerja sampai program selesai atau program mengalami kerusakan. Ketika sistem komputer semakin maju, otomatisasi pengerjaan program mulai dilakukan. Sebagian besar proses program dapat dilakukan secara otomatis, seperti penentuan

nomor urut, atau pencatatan waktu. Untuk menghindari error karena banyaknya otomatisasi yang dilakukan, pembuat sistem komputer melengkapi komputernya dengan pustaka standar untuk memperkecil resiko error program.

Pustaka yang disertakan oleh pembuat sistem komputer, menjadi cikal bakal sistem operasi. Pada saat itu pustaka yang digunakan oleh program berjalan sebelum program dieksekusi bukan sebaliknya. Pustaka ini telah memudahkan dalam perancangan program karena beberapa fungsi dasar tidak perlu ada didalam program. Tetapi penggunaan istilah sistem operasi belum dikenal, pustaka ini dikenal dengan istilah monitor karena sifatnya yang memonitor otomatisasi yang dilakukan sistem komputer.

Setelah penggunaan sistem komputer semakin meluas, sistem komputer mulai dilengkapi dengan program yang bertugas menyediakan operasi dasar komputer, seperti manajemen perangkat keras atau penjadwalan proses perangkat lunak. Pada era komputer desktop, program dasar tersebut kembali dilengkapi dengan beberapa aplikasi seperti manajemen file. Pada masa ini sistem operasi mulai dikenal oleh banyak orang, terutama dengan lahirnya UNIX dan MS-DOS. Tetapi arti dari sistem operasi mulai berubah dari istilah aslinya. Sistem operasi yang berarti program yang menyediakan operasi dasar komputer, berubah menjadi kumpulan aplikasi yang menyediakan fasilitas yang lebih luas dibandingkan sebelumnya. Sedangkan program yang menyediakan operasi dasar sistem komputer kini disebut dengan *kernel*.

1.1.2 Kernel

Kernel adalah komponen utama dari kebanyakan sistem operasi, tanggung jawab dari kernel adalah memanajemen sumber daya sistem dan komunikasi antara perangkat keras dengan perangkat lunak. Dalam konsep sistem operasi yang menerapkan prinsip abtraksi, kernel adalah abtraksi yang paling rendah dalam sistem operasi. Kernel hanya menyediakan abstraksi kepada aplikasi, aplikasi yang akan mengontrol sumber daya tersebut.

Ada berbagai metode yang digunakan kernel untuk mengelola fungsi-fungsinya. *Monolithic* kernel menjalankan semua fungsinya dalam satu alamat memori yang sama

untuk meningkatkan performa sistem. Sedangkan pada *microkernel* sebagian fungsi sistem operasi dijalankan diluar memori kernel. Tujuannya untuk meningkatkan pemeliharaan dan mempertahankan modulitasnya. Terdapat banyak kemungkinan untuk meningkatkan kedua metode sistem operasi diatas.

1.1.3 Perbedaan Sistem Operasi Dengan Kernel

Jika melihat pada gambar **??** halaman **??** maka dapat terlihat perbedaan antara posisi kernel dengan sistem operasi. Sistem operasi dan aplikasi membutuhkan kernel untuk mengabstraksi lapisan perangkat keras, sedangkan kernel dapat berjalan tanpa adanya sistem operasi. Karena setiap lapisan memfasilitisasi lapisan yang berada diatasnya.

Kernel tidak dapat langsung digunakan oleh user karena kernel hanya berinteraksi dengan sistem operasi atau aplikasi. Setiap lapisan arsitektur komputer harus dilewati secara berurutan, sehingga untuk mengakses kernel user harus melewati lapisan sistem operasi. Sedangkan Sistem operasi dan aplikasi dapat digunakan secara langsung oleh user. Sistem operasi dan aplikasi yang akan berinteraksi dengan user kemudian meneruskan permintaan user kepada kernel.

1.1.4 Fungsi Sistem Operasi

Fungsi sistem operasi atas sistem komputer hampir sama dengan fungsi yang terdapat pada kernel OPE[**?**], yaitu :

1. Memanajemen proses yang berlangsung.
2. Manajemen memori.
3. Manajemen disk.
4. Manajemen jaringan.
5. Manajemen keamanan data.

Terdapat pula fungsi sistem operasi yang berbeda dengan kernel. Namun tidak semua sistem operasi menyediakan fungsi ini :

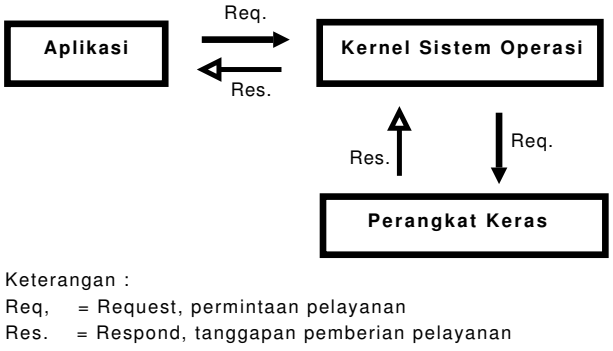
1. Manajemen perangkat keras input/output.

2. Antarmuka yang berupa grafik.

1.1.5 Arsitektur Sistem Operasi

Terdapat banyak sistem operasi yang dibuat atas dasar penelitian, hobi, atau bisnis, tetapi dari sekian banyak sistem operasi hanya terdapat beberapa desain arsitektur kernel sistem operasi. Seperti telah dijelaskan sebelumnya, kernel adalah program yang menyediakan operasi dasar dari sistem komputer. Dua jenis arsitektur kernel yaitu *monolithic* kernel dan *microkernel* adalah jenis arsitektur yang umum digunakan.

Arsitektur *monolithic* kernel memiliki ciri dimana semua kode kernel dijalankan pada suatu proses dan pada lokasi memori yang sama MON[**?**]. Dengan kata lain terdapat satu program kernel yang akan mengatur semua proses pada sistem. Keuntungan dari desain ini adalah komunikasi yang cepat antara fungsi-fungsi di dalam kernel. Beberapa orang juga berpendapat desain ini lebih mudah untuk di implementasikan dibandingkan arsitektur lainnya. Akan tetapi dikarenakan semua fungsi kernel berjalan didalam satu proses yang sama berakibat jika salah satu fungsi gagal dalam melakukan fungsinya, akan menyebabkan seruruh sistem operasi mengalami kegagalan proses. Jika fungsi-fungsi kernel terus ditambah maka kode yang harus dipelihara akan menjadi lebih besar. Kode kernel yang cenderung saling berkaitan mengakibatkan jika terjadi perubahan dalam kode kernel perubahan itu harus selalu dimonitor, dan jika kode kernel menjadi terlalu besar maka monitoring kode kernel akan sulit dilakukan. Contoh sistem operasi yang menggunakan arsitektur ini adalah Linux dan Windows9x. Berikut ini adalah gambar struktur *monolithic* kernel secara umum didalam sistem komputer:



Gambar 1.2: Struktur *Monolithic* Kernel (sumber MON[**?**])

Desain arsitektur kernel yang lain adalah *microkernel*. Kontras terhadap desain *monolithic* kernel, *microkernel* membagi beberapa prosesnya kedalam beberapa pemproses lain yang disebut ”server” MIC[**?**]. Keunggulan utama dari desain ini adalah kode yang lebih mudah dipelihara serta apabila terjadi kegagalan proses suatu server tidak akan mempengaruhi server lainnya. Kekurangannya pemprosessan yang lebih lambat dibandingkan *monolithic* kernel. Ini disebabkan *microkernel* mengandalkan komunikasi antara server independen untuk menjalankan sistemnya. Selanjutnya mengenai *microkernel* akan dibahas lebih lanjut pada BAB III.

Selain arsitektur kernel yang dijelaskan diatas, terdapat arsitektur kernel lain yang umumnya berupa perubahan dari desain diatas. Sebagai contoh *nanokernel* yang merupakan perubahan dari *microkernel*. *Nanokernel* mengurangi fungsi yang terdapat pada kernel dengan tujuan agar kernel benar-benar independen, dan menggunakan prinsip ”server” dalam sistem operasinya. Arsitektur rubahan yang lain adalah *hybrid* kernel yang menggabungkan fungsi-fungsi penting kernel kedalam sebuah program dengan alasan kecepatan dan fungsi-sungsi lainnya yang tidak vital dipisahkan kedalam ”server”, desain ini merupakan gabungan dari kecepatan pada desain *monolithic* kernel dan kestabilan sistem operasi yang dimiliki *microkernel*. Selain itu terdapat sistem operasi yang mengurangi fungsi yang dimiliki kernel sehingga tanggung jawab kernel terhadap aplikasi semakin kecil yaitu arsitektur *exokernel* EXO[**?**] DAW[**?**] DAW[**?**], yang akan dibahas lebih lanjut pada BAB IV.

1.2 Perkakas

1.2.1 Netwide Assembler (NASM)

NASM adalah sebuah kompiler untuk bahasa assembler. NASM dapat digunakan untuk menulis program 16-bit dan 32-bit, NASM hanya digunakan untuk jenis prosesor x86 NAS[**?**]. Program didapat dari *http://nasm.sourceforge.net/*.

1.2.2 GNU project C and C++ compiler (GCC)

GCC merupakan kumpulan dari beberapa kompiler yang sifatnya sumber terbuka. GCC pertamakali dimulai oleh Richard Stallman pada tahun 1985. GCC mampu meng-kompile beberapa bahasa pemprograman diataranya C dan C++ GNU[**?**]. Program didapat dari *http://gcc.gnu.org/*.

1.2.3 Linux

Linux adalah sebuah sistem operasi sumber terbuka berbasiskan UNIX yang dicetuskan oleh Linus Trovalds. Sistem operasi Linux menggunakan arsitektur *monolithic* kernel untuk desain kernelnya LIN[**?**]. Program didapat dari *http://www.kernel.org/* tapi hanya berupa linux kernel saja, untuk dapat menggunakan sistem operasi linux dibutuhkan program-program pendukung lainnya. Paket program sistem operasi Linux dikenal dengan sebutan distro linux salah satu distro linuxadalah slackware linux. Slackware linux didapat di *http://www.slackware.com/*.

1.2.4 GRand Unified Bootloader (GRUB)

GRUB adalah sebuah *boot loader* yang berjalan ketika komputer dihidupkan (setelah BIOS). Tugas dari GRUB adalah me-load dan menyerahkan kontrol komputer kepada kernel sistem operasi GNU[**?**]. Program didapat dari *http://www.gnu.org/software/grub/*.

1.2.5 LD

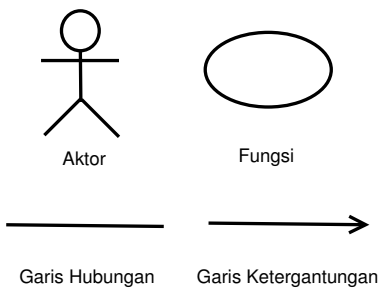
LD adalah sebuah aplikasi yang mengkombinasikan beberapa kode objek dan pustaka yang ada untuk menjadi sebuah file yang dapat di eksekusi. LD diambil dari kata

'load', dan merupakan perkakas standar bagi sistem operasi berbasis UNIX GNU[**?**]. Program termasuk kedalam gcc.

1.3 Analisa Sistem

1.3.1 Use Case Diagram

Use Case diagram adalah diagram yang mengambarkan hubungan antara aktor yang berada diluar sistem dengan fungsi-fungsi didalam sistem. Tetapi *Use Case* diagram hanya menggambarkan hubungan aktor dengan fungsi pada sistem, bukan seluruh fungsi yang terdapat pada sistem AGU[**?**]. Simbol-simbol yang digunakan untuk menggambarkan *use case* diagram yaitu :



Gambar 1.3: Daftar Simbol Use Case Diagram

Deskripsi simbol-simbol tersebut adalah :

1. **Aktor** mempresentasikan seseorang atau sesuatu (seperti perangkat atau sistem lain) yang berinteraksi dengan sistem. Aktor mungkin hanya memberi informasi atau menerima informasi dari sistem. Aktor berinteraksi dengan fungsi pada use case tapi tidak memegang kendali atas fungsi pada sistem.

2. **Fungsi Use Case** menggambarkan fungsionalitas yang terdapat dari suatu sistem, sehingga user dapat mengerti tentang kegunaan dari sistem yang dibangun. Fungsi use case digambarkan dari sudut pandang pengguna sistem, sehingga use case lebih menitik beratkan pada fungsi yang berinteraksi dengan pengguna sistem tidak semua fungsi dalam sistem akan digambarkan.

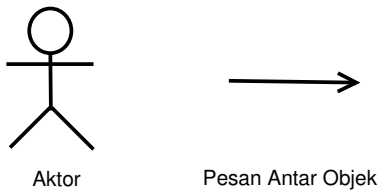
3. **Garis** menggambarkan hubungan yang terjadi antara objek-objek yang ada dalam diagram use case. Garis tanpa tanda panah menggambarkan hubungan dua arah antara objek yang dihubungkan. Sedangkan garis dengan tanda panah menggambarkan hubungan satu arah informasi dari objek asal garis kepada objek akhir garis.

1.3.2 Skenario Use Case

Skenario Use Case digunakan untuk memperjelas kejadian pada sistem dan aktor diluar sistem yang telah digambarkan pada *use case* diagram AGU[**?**]. Skenario use case digambarkan kedalam tabel, tabel menggambarkan kejadian yang terjadi antara aktor dengan sistem.

1.3.3 Sequence Diagram

Sequence diagram menggambarkan hubungan antara sejumlah objek, hubungan itu ditunjukan oleh rangkaian pesan yang dikirim dan diterima oleh objek AGU[**?**]. Simbol-simbol yang digunakan untuk menggambarkan *sequence* diagram yaitu :



Gambar 1.4: Daftar Simbol Sequence Diagram

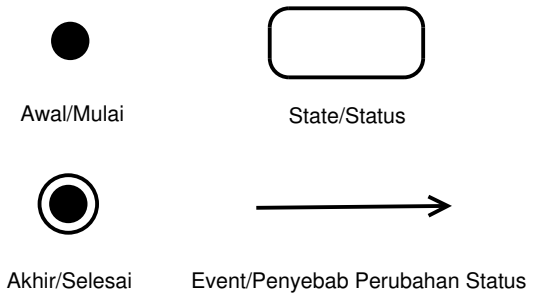
Deskripsi simbol-simbol tersebut adalah :

1. **Aktor** mempresentasikan seseorang atau sesuatu (seperti perangkat atau sistem lain) yang berinteraksi dengan sistem. Aktor mungkin hanya memberi informasi atau menerima informasi dari sistem.

2. **Garis** menggambarkan hubungan yang terjadi antara objek-objek yang ada dalam diagram sequence. Garis dengan tanda panah menggambarkan hubungan satu arah informasi dari objek asal garis kepada objek akhir garis.

1.3.4 State Diagram

State diagram menggambarkan kondisi yang dimiliki oleh suatu objek, perubahan kondisi ini dapat disebabkan oleh pesan atau karena perubahan yang bersifat berurutan AGU[**?**]. Simbol-simbol yang digunakan untuk menggambarkan *state* diagram yaitu:



Gambar 1.5: Daftar Simbol State Diagram

Deskripsi simbol-simbol tersebut adalah :

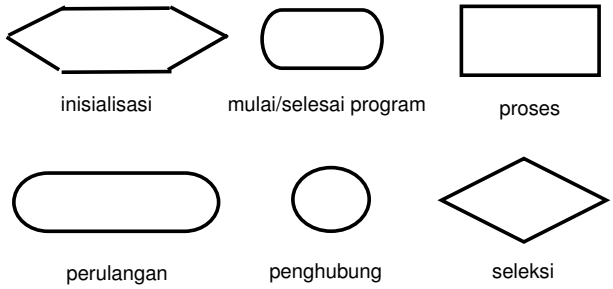
1. **State** adalah kondisi sebuah objek ketika memenuhi beberapa kondisi, melakukan beberapa action, atau menunggu event.

2. **Event penyebab perubahan status** menginpresentasikan perubahan dari state awal ke sebuah state berikutnya dapat disertai dengan action jika perubahan terjadi tidak secara otomatis.

3. **Awal atau akhir** menggambarkan mulai atau berakhirnya suatu state diagram pada sebuah sistem.

1.3.5 Flowchart

Flowchart menggambarkan alur logika pada program atau fungsi yang berjalan FLO[**?**]. Simbol-simbol yang digunakan untuk menggambarkan *flowchart* yaitu :



Gambar 1.6: Daftar Simbol Flowchart

Deskripsi simbol-simbol tersebut adalah :

1. **Inisialisasi**, simbol ini menggambarkan nilai dari variabel yang di inisialisasi oleh program.

2. **Mulai/selesai program**, menggambarkan mulai atau berakhirnya suatu program pada flowchart.

3. **Proses**, menggambarkan terjadinya proses pada suatu program. Proses yang dilakukan dapat berupa membandingkan, melakukan penghitungan, atau merubah nilai suatu variabel.

4. **Perulangan**, mengambarkan perulangan yang dilakukan program terhadap suatu atau beberapa perintah didalam program.

5. **Penghubung**, menggambarkan penggabungan dari garis-garis perintah dalam program sehingga garis yang keluar dari penghubung dapat dikurangi jumlahnya tanpa merubah perintah pada program.

6. **Seleksi**, menggambarkan pemilihan kondisi yang terjadi dalam program.