

# STRUKTUR SISTEM OPERASI

## Konsep Dasar Sistem Operasi

- Komponen Sistem Operasi
- Layanan Sistem Operasi
- System Calls
- Pemrograman Sistem
- Struktur sistem
- Mesin Virtual
- System Generation
- Rancangan Sistem

## 1. Komponen 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
- Manajemen Memori Utama
- Manajemen Berkas
- Manajemen I/O
- Manajemen Penyimpanan Sekunder
- Jaringan
- Sistem Proteksi
- Command-Interpreter System

### 1.1. Manajemen Proses

- Proses adalah sebuah program yang sedang dijalankan (eksekusi).
- Suatu proses memerlukan sumber daya pada saat eksekusi: CPU time, memori, berkas dan peranti I/O
- Sistem operasi bertanggung jawab terhadap aktifitas yang berhubungan dengan manajemen proses:
  - Pembuatan dan penghapusan proses
  - Penundaan dan pelanjutan proses
  - Penyedia mekanisme untuk:
    - Sinkronisasi antar proses
    - Komunikasi antar proses
    - Penanganan Deadlock

### 1.2. Manajemen Memori Utama

- Memori sebagai tempat penyimpanan instruksi/data dari program.
- Penyimpanan yang cepat sehingga dapat mengimbangi kecepatan eksekusi instruksi CPU
- Terdiri dari "array words/bytes" yang besar
- Alamat digunakan untuk mengakses data (shared oleh CPU dan I/O devices)
- Umumnya main memory bersifat "volatile" – tidak permanen. Isinya akan hilang jika komputer dimatikan.
- Sistem operasi bertanggung jawab untuk aktivitas berikut yang berhubungan dengan manajemen memori:
  - melacak pemakaian memori (siapa dan berapa besar?).
  - memilih program mana yang akan di-load ke memori ketika bisa digunakan.
  - alokasi dan dealokasi memori sesuai yang dibutuhkan

### 1.3. Manajemen File

- Berkas adalah kumpulan informasi yang berhubungan (sesuai dengan tujuan pembuat berkas tersebut). Biasanya berkas merepresentasikan program dan data.
- Sistem operasi bertanggung jawab untuk aktivitas berikut yang berhubungan dengan manajemen berkas:
  - pembuatan dan penghapusan berkas
  - pembuatan dan penghapusan direktori
  - Mendukung primitif untuk manipulasi berkas dan direktori
  - memetakan berkas pada sistem sekunder
  - Backup berkas pada media penyimpanan yang stabil (nonvolatile)

### 1.4. Manajemen sistem I/O

- Sistem I/O terdiri dari:
  - Sistem buffer: menampung sementara data dari/ke peranti I/O
  - Spooling: melakukan penjadwalan pemakaian I/O sistem supaya lebih efisien (antrian dsb)
  - Antarmuka devices-driver yang umum: menyediakan device driver yang umum sehingga sistem operasi dapat seragam (buka, baca, tulis, tutup)
  - Drivers untuk spesifik perangkat keras: menyediakan driver untuk melakukan operasi rinci/detail untuk perangkat keras tertentu.

### 1.5. Manajemen Penyimpanan Sekunder

- Penyimpanan sekunder: Penyimpanan Permanen  
Karena memori utama bersifat sementara dan kapasitasnya terlalu kecil, maka untuk menyimpan semua data dan program secara permanen, sistem komputer harus menyediakan penyimpanan sekunder untuk dijadikan back-up memori utama.
- Sistem Operasi bertanggung jawab dalam aktivitas yang berhubungan dengan manajemen penyimpanan sekunder:
  - manajemen ruang kosong
  - alokasi penyimpanan
  - penjadwalan disk

### 1.6. Jaringan (Sistem Terdistribusi)

- Sistem Terdistribusi adalah kumpulan prosesor yang tidak berbagi memori atau clock. Setiap memiliki memori lokal masing-masing.
- Prosesor-prosesor dalam sistem terhubung dalam jaringan komunikasi.
- Sistem terdistribusi menyediakan akses pengguna ke bermacam-macam sumber daya. Akses tersebut menyebabkan:
  - Peningkatan kecepatan komputasi
  - peningkatan penyediaan data
  - peningkatan keandalan

### 1.7. Sistem Proteksi

- Proteksi berkenaan dengan mekanisme untuk mengontrol akses yang dilakukan oleh program, prosesor, pengguna sistem maupun pengguna sumber daya.
- Mekanisme Proteksi harus:
  - membedakan antara penggunaan yang sah dan yang tidak sah.
  - spesifikasi kontrol untuk diterima
  - menyediakan alat untuk pemberlakuan sistem.

### 1.8. Command-Interpreter System

- Sistem Operasi menunggu instruksi dari pengguna (command driven).

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

## 2. Pelayanan Sistem Operasi

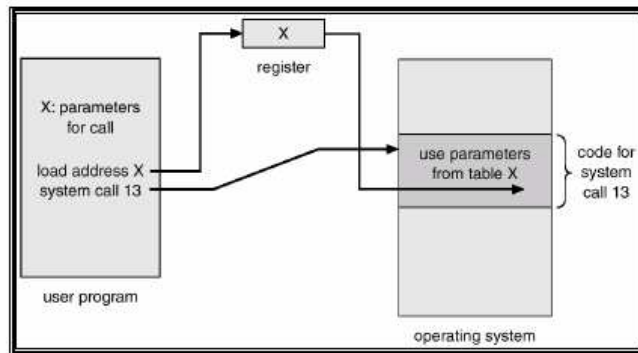
- Eksekusi program: memload program ke memory dan menjalankannya (run)
- Operasi I/O: pengguna tidak bisa mengontrol I/O secara langsung ( untuk efisiensi dan keamanan), sistem harus bisa menyediakan mekanisme utk melakukan operasi I/O
- Manipulasi sistem berkas: membaca, menulis, membuat, dan menghapus file
- Komunikasi: pertukaran informasi, dapat dilaksanakan melalui shared memory atau message passing
- Deteksi error: mempertahankan kestabilan dengan mendeteksi error (pada CPU, perangkat keras memori, I/O, program pengguna) dan jika bisa, memperbaikinya

### **Pelayanan Tambahan**

- Lebih diarahkan kepada upaya untuk menjaga efisiensi sistem, bukan untuk membantu pengguna
- Alokasi sumber daya: mengalokasikan sumber daya kepada beberapa pengguna atau tugas yang dijalankan pada saat yang bersamaan
- Accounting: menentukan berapa banyak dan berapa lama users menggunakan sumber daya sistem
- Proteksi: menjaga semua akses ke sumber daya sistem terkontrol

## 3. System calls

- Suatu set '*extended instructions*' yang merupakan interface antara program-program dengan sistem operasi.
- System calls menyediakan antarmuka antara proses (program yang sedang dijalankan) dan sistem operasi.
- Biasanya tersedia sebagai instruksi bahasa rakitan
- Beberapa sistem mengizinkan system calls dibuat langsung dari bahasa pemrograman tingkat tinggi
- Beberapa bahasa pemrograman tingkat tinggi (contoh: C, C++) telah didefinisikan untuk menggantikan bahasa rakitan untuk sistem pemrograman
- Tiga metode umum yang digunakan dalam memberikan parameter kepada sistem operasi
  - Melalui register
  - Menyimpan parameter dalam blok atau tabel pada memori dan alamat blok tersebut diberikan sebagai parameter dalam register
  - Menyimpan parameter (push) ke dalam stack (oleh program), dan melakukan pop off pada stack (oleh sistem operasi)
  - Memberikan Parameter dalam Tabel



Sumber: Silberschatz, et.al, Operating System Concepts, 6th ed, .2003, New York: John Wiley & Son.Inc , page 65

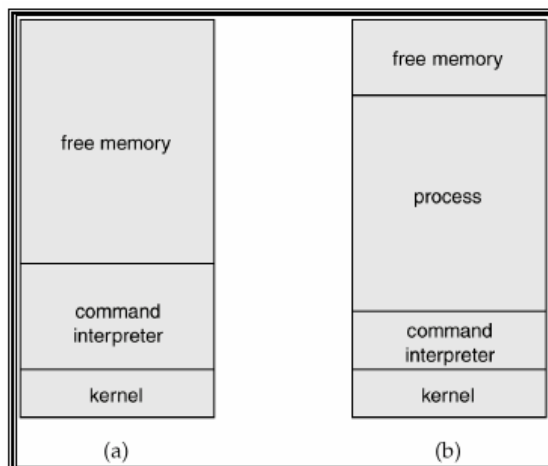
### 3.1. Jenis System Calls

- Pengendalian proses
- Manajemen berkas
- Manajemen Peranti
- Mempertahankan informasi
- Komunikasi

#### 3.1.1. Process Control

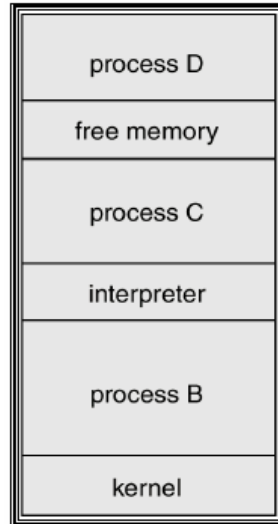
- selesai, abort
- Load, eksekusi
- Membuat dan mengakhiri proses
- Mengambil dan mengeset atribut proses
- Menunggu waktu
- Wait event, signal event
- Alokasi dan pengosongan memori

Eksekusi MS-DOS



Sumber: Silberschatz, et.al, Operating System Concepts, 6th ed, .2003, New York: John Wiley & Son.Inc , page 68

UNIX Menjalankan Multiple Program



Sumber: Silberschatz, et al, Operating System Concepts, 6th ed, .2003, New York: John Wiley & Son. Inc, page 69

### 3.1.2. Manajemen Berkas

- Membuat dan menghapus berkas
- Membuka dan menutup berkas
- Read, write, reposition
- Mengambil dan mengeset atribut berkas

### 3.1.3. Manajemen Peranti

- Meminta peranti, melepaskan peranti
- Read, write, reposition
- Mengambil dan mengeset atribut peranti

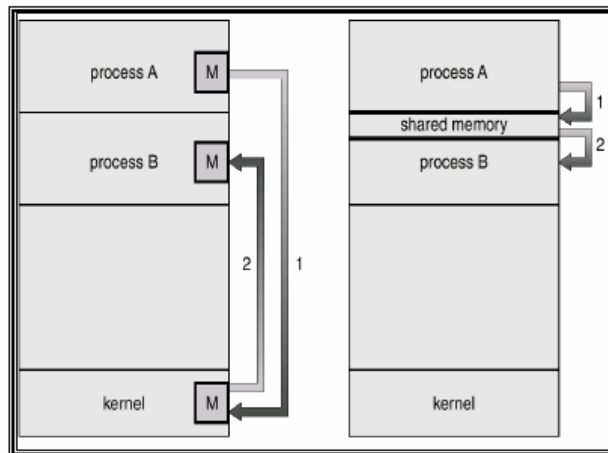
### 3.1.4. Information Maintenance

- Mengambil dan mengeset waktu dan tanggal
- Mengambil dan mengeset system data
- Mengambil proses, berkas atau atribut peranti
- Mengeset proses, berkas atau atribut peranti

### 3.1.5. Komunikasi

- Menciptakan, menghapus hubungan komunikasi
  - Mengirim dan menerima pesan
  - Mentransfer status informasi
  - Attach atau deattach remote device
- Komunikasi dapat dilakukan melalui message passing atau shared memory
- a. Message-passing model. Informasi saling ditukarkan melalui fasilitas yang telah ditentukan oleh sistem operasi
  - b. Shared-memory model. Proses-proses menggunakan map memory untuk mengakses daerah-daerah dimemori dengan proses-proses yang lain.

Mekanisme Komunikasi



Sumber: Silberschatz, et al, Operating System Concepts, 6th ed., 2003, New York: John Wiley & Son, Inc., page 72

#### 4. Pemrograman sistem (Sistem Program)

- Pemrograman sistem menyediakan lingkungan yang memungkinkan pengembangan program dan eksekusi berjalan dengan baik
- Dapat dikategorikan :
  - Manajemen / manipulasi berkas: membuat, menghapus, copy, rename, print, memanipulasi berkas dan direktori
  - Informasi Status: tanggal, jam, jumlah memori dan disk yang tersedia, jumlah pengguna, dan informasi tentang status lainnya
  - Modifikasi Berkas: modifikasi isi berkas
  - Mendukung bahasa pemrograman: Kompilator, perakitan, interpreter
  - Loading dan eksekusi program: absolute loaders, relocatable loaders, linkage editors, overlay loaders
  - Komunikasi: menyediakan mekanisme komunikasi antara proses, pengguna, dan sistem komputer yang berbeda
  - Program-program aplikasi. Sistem operasi harus menyokong program-program yang berguna untuk menyelesaikan permasalahan secara umum atau membentuk operasi-operasi secara umum, seperti kompilasi, pemformatan teks, paket plot, sistem basis data, spreadsheet, paket analisis statistik dan games.
  - Sistem program yang paling penting adalah command interpreter (mengambil dan menerjemahkan user-specified command selanjutnya)

#### 5. Struktur Sistem

- Struktur Sederhana
- Monolithic sistem
- Metode pendekatan Terlapis
- Sistem dengan Client-Server
- Sistem Berorientasi Objek
- Mikrokernell

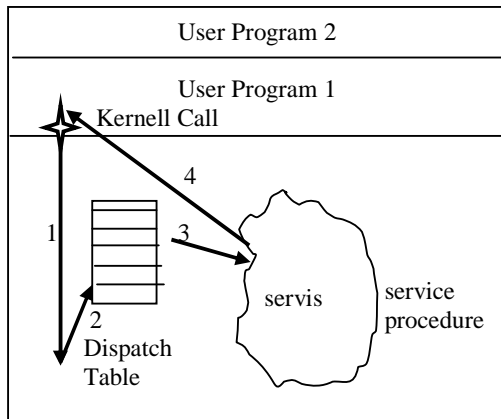
##### 5.1. Struktur Sederhana

- Dimulai dengan sistem yang kecil, sederhana dan terbatas kemudian berkembang dengan cakupan original
- Struktur sistem MS-DOS: disusun untuk mendukung fungsi yang banyak pada ruang yang kecil
- Struktur Sistem UNIX  
Terdiri dari 2 bagian:

- Kernel :
  - antarmuka
  - device drivers
- Program Sistem

**5.2. Sistem Monolitik**

- Konsep : “Sistem operasi sebagai kumpulan prosedur dimana prosedur dapat saling dipanggil oleh prosedur lain di sistem bila diperlukan”.
- *Kernel* berisi semua layanan yang disediakan sistem operasi untuk pemakai.
- Contoh : Sistem Operasi Unix menggunakan konsep *kernel loadable modules*, yaitu :
  - Bagian-bagian *kernel* terpenting berada dimemori utama secara tetap.
  - Bagian-bagian esensi lain berupa modul yang dapat ditambahkan ke kernel saat diperlukan dan dicabut begitu tidak digunakan lagi diwaktu jalan (*run-time*).
- The ‘big mess’
- Tidak berstruktur
- Kernel call / supervisor call
- User mode
- Kernel mode / supervisor mode



User Program  
Runs in  
User Mode

Operating System  
runs in  
Kernel Mode

Gambar Struktur Sistem Monolithic

**5.3. Pendekatan Terlapis**

- Konsep : “Sistem operasi dibentuk secara hirarki berdasarkan lapisan-lapisan, dimana lapisan-lapisan memberi layanan lapisan lebih atas”.
- Contoh :
  - ✓ Sistem Operasi THE (Technique Hogesschool Endiche), Djikstra, 1968.
  - ✓ Sistem Operasi MULTICS (menggunakan *concentric rings*).

Tabel 1. Lapisan-lapisan pada Sistem Operasi THE

Layer	Function
5	The Operator
4	User Programs
3	I/O Management
2	Operator-Process Communication
1	Memory & Drum Management
0	Processor Allocation & Multiprogramming

- Lapisan adalah implementasi dari objek abstrak yang merupakan enkapsulasi dari data dan operasi yang bisa memanipulasi data tersebut
- Lapisan paling bawah: perangkat keras

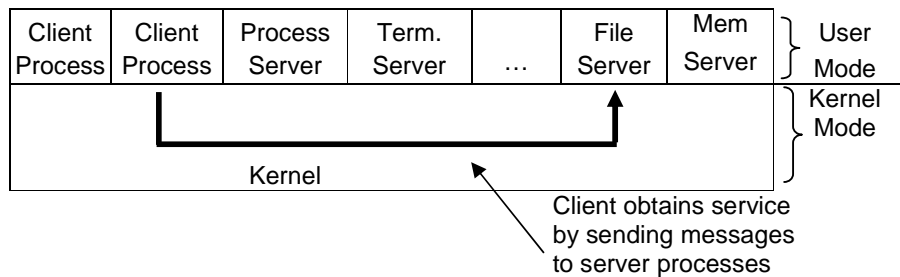
- Lapisan paling atas: antarmuka pengguna  
Tingkatan Desain Sistem Operasi

Level	nama	objek
13	shell	user programming environment
12	proses pengguna	proses pengguna
11	direktori	direktori
10	peranti	peranti eksternal
9	sistem berkas	berkas
8	komunikasi	pipa
7	memori virtual	segmen, halaman
6	penyimpanan sekunder lokal	blok data, saluran peranti
5	proses primitif	proses primitif, semafor, ready list
4	interupsi	program penanganan interupsi
3	prosedur	prosedur, call-stack, tampilan
2	set instruksi	stack, microprogram interpreter, scalar, dan array data
1	sirkuit elektronik	register, gerbang, bus, dll

- Keuntungan: modularitas
  - mempermudah debug dan verifikasi sistem
  - lapisan pertama bisa didebug tanpa mengganggu sistem yang lain
- Kesulitan:
  - hanya bisa menggunakan lapisan dibawahnya
  - tidak efisien dibandingkan tipe yang lain

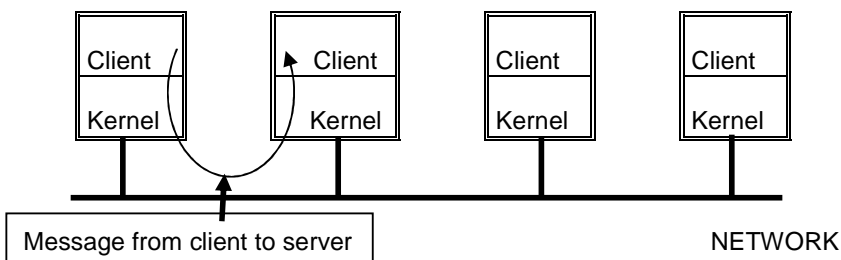
#### 5.4. Sistem dengan Client-Server

- ✓ Konsep : “*Server* adalah proses yang menyediakan layanan, dan *Client* adalah proses yang memerlukan / meminta layanan. Proses *client* yang memerlukan layanan mengirim pesan ke *server* dan menanti pesan jawaban. Proses *server* setelah melakukan tugas yang diminta, mengirim hasil dalam bentuk pesan jawaban ke proses *client*. *Server* hanya menanggapi permintaan *client* dan tidak memulai percakapan dengan *client*”.



Gambar Model Client Server

- Model Client Server pada distributed system



Gambar Model Client Server pada Distributed System



### 5.5. Sistem berorientasi objek

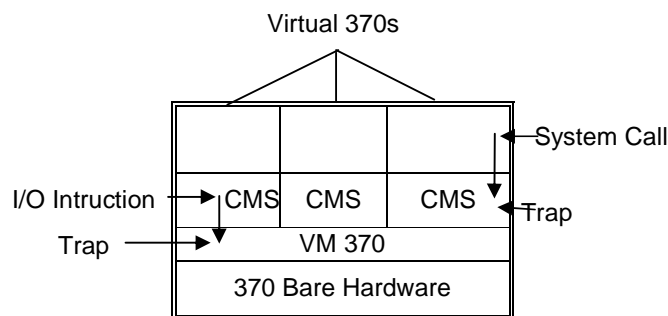
- ✓ Konsep : “Layanan diimplementasikan sebagai objek”.
- ✓ Model ini terstruktur dan memisahkan antara layanan yang disediakan dan implementasinya.
- ✓ Contoh :
  - Sistem operasi X-kernel
  - Sistem operasi MS-Windows NT telah mengadopsi beberapa teknologi berorientasi objek, tapi belum secara keseluruhan.

### 5.5. Mikrokernel

- Menyusun sistem operasi dengan menghapus semua komponen yang tidak esensial dari kernel, dan mengimplementasikannya sebagai sistem program dan level pengguna
- Fungsi utama: mendukung fasilitas komunikasi antara program klien dan bermacam-macam layanan yang juga berjalan di user-space
- Keuntungan:
  - ketika layanan baru akan ditambahkan ke user-space, kernel tidak perlu di modif
  - OS lebih mudah ditempatkan pada suatu desain perangkat keras ke desain lainnya
  - mendukung keamanan reliabilitas lebih
- Contoh sistem operasi: Tru64 UNIX, MacOSX, QNX

### 6. MesinVirtual

- ✓ Konsep : “Awalnya struktur ini membuat seolah-olah *user* mempunyai seluruh komputer dengan simulasi atas pemroses yang digunakan. Sistem operasi melakukan simulasi mesin nyata. Mesin hasil simulasi digunakan *user*, mesin maya merupakan tiruan 100% atas mesin nyata. Semua *user* diberi ilusi mempunyai satu mesin yang sama-sama canggih”.
- ✓ Contoh :
  - Sistem operasi MS-Windows NT dapat menjalankan aplikasi MS-Dos, OS/2 mode teks, dan Win 16.
  - Pengembang Linux membuat DOSEMU agar aplikasi MS-Dos dapat dijalankan di Linux, WINE agar aplikasi MS-Windows dapat dijalankan di Linux, iBCS agar aplikasi SCO-Unix dapat dijalankan di Linux.
- ✓ CP/CMS
- ✓ VM/370
- ✓ Virtual Machine Monitor
- ✓ CMS (Conversational Monitor System)



Gambar Sistem Virtual Mesin

- Mesin virtual mengambil pendekatan terlapis sebagai kesimpulan logis. Mesin virtual memperlakukan hardware dan sistem operasi seolah-olah berada pada level yang sama sebagai perangkat keras.
- Pendekatan Mesin virtual menyediakan sebuah antarmuka yang identik dengan underlying bare hardware.

- Sistem Operasi membuat ilusi dari banyak proses, masing-masing dieksekusi pada prosesor sendiri dengan virtual memorinya sendiri.
- VM dibuat dengan pembagian sumber daya oleh komputer fisik. Sumber daya dari komputer fisik dibagi untuk membuat VM
- Penjadwalan CPU bisa menciptakan penampilan seakan-akan pengguna mempunyai prosesor sendiri
- Spooling dan system data bisa menyediakan virtual card readers dan virtual line printers
- Sebuah time-sharing terminal user yang normal melayani sebagaimana operator konsulat
- VM software membutuhkan ruang di dalam disk untuk menyediakan memori virtual dan spooling, yaitu sebuah disk virtual
- Keuntungan Penggunaan Mesin virtual
  - Keamanan bukanlah masalah
  - VM mempunyai perlindungan lengkap pada berbagai sistem sumber daya
  - Tidak ada pembagian sumber daya secara langsung. Pembagian disk mini dan jaringan diimplementasikan pada perangkat lunak
  - VM sistem adalah kendaraan yang “sempurna” untuk penelitian dan pengembangan sistem operasi
  - Dengan VM perubahan suatu bagian tidak akan mempengaruhi komponen yang lain
- Kerugian Penggunaan VM
  - VM sulit diimplementasikan karena banyak syarat yang dibutuhkan untuk menyediakan duplikat yang tepat dari underlying machine
  - Harus punya virtual -user mode dan virtual-monitor mode yang keduanya berjalan di Physical mode. Akibatnya, saat instruksi yang hanya membutuhkan virtual monitor mode dijalankan, register berubah dan bisa berefek pada virtual user mode, bahkan bisa me-restart VM
  - Waktu yang dibutuhkan I/O bisa lebih cepat (karena ada spooling), tapi bisa lebih lambat (karena diinterpreted)

### 6.1. Java Virtual Machine

- Program Java yang telah dicompile adalah platform-neutral byte codes yang dieksekusi oleh Java Virtual Machine(JVM)
- JVM terdiri dari:
  - pengeload kelas
  - pemverifikasi kelas
  - runtime interpreter
  - Just In-Time (JIT) kompilator meningkatkan kinerja
  - Java Development Environment (JDE) terdiri dari sebuah compile time environment yang mengubah java sourcescode menjadi bytecode, dan sebuah run time environment yang menyediakan Java platform system untuk tuan rumah

### 7. Perancangan Sistem

- Masalah: menentukan tujuan dan spesifikasi sistem.
- Perancangan sistem dipengaruhi oleh perangkat keras dan jenis sistem sehingga kebutuhannya akan lebih sulit untuk dispesifikasikan.
- Kebutuhan terdiri dari tujuan pengguna dan tujuan sistem.
- Pengguna ingin sistem yang enak digunakan, mudah dipelajari, terpercaya, aman, dan cepat. Tapi itu semua sebenarnya tidak dibutuhkan oleh sebuah sistem.
- Sistem ingin mudah dirancang dan diimplmentasikan, fleksibel, terpercaya, error yang minimal, dan efisien.

### Mekanisme dan Kebijakan

- Mekanisme menjelaskan bagaimana melakukan sesuatu, kebijakan menentukan apa yang akan dilakukan

- Pemisahan kebijakan dari mekanisme adalah hal yang sangat penting, ini, mengijinkan fleksibilitas yang tinggi jika kebijakan akan diubah suatu saat.
- Kebijakan penting untuk semua alokasi sumber daya dan menjadwalkan masalah, menentukan perlu atau tidaknya mengalokasikan sumber daya.
- Mekanisme yang menentukan apa dan bagaimana

### **Implementasi Sistem**

- Secara tradisional, sistem operasi ditulis dalam bahasa rakitan, tapi sekarang sering dibuat dalam bahasa tingkat tinggi.
- Keuntungan ditulis dalam bahasa tingkat tinggi adalah
  - kodenya bisa ditulis dengan lebih cepat
  - lebih padat
  - mudah dimengerti dan didebug
- Sistem operasi yang ditulis dengan bahasa tingkat tinggi akan mudah dipindahkan ke perangkat keras lain, tapi bisa mengurangi kecepatan dan membutuhkan penyimpanan yang lebih banyak.

### **8. System Generation**

- Sistem operasi dirancang untuk dapat dijalankan pada berbagai jenis mesin, sistemnya harus dikonfigurasi untuk setiap komputer.
- Program Sysgen mendapatkan informasi mengenai konfigurasi khusus tentang sistem perangkat keras dari sebuah data, antara lain sebagai berikut:
  - CPU apa yang digunakan, pilihan yang diinstal
  - Berapa banyak memori yang tersedia
  - Peralatan yang tersedia
  - Sistem operasi pilihan apa yang diinginkan atau parameter apa yang digunakan
  - Satu kali info diperoleh, bisa digunakan dengan berbagai cara
- Booting: memulai komputer dengan me-load kernel.
- Bootstrap program: kode yang disimpan di code ROM yang dapat menempatkan kernel, memasukkannya ke dalam memori, dan memulai eksekusinya.