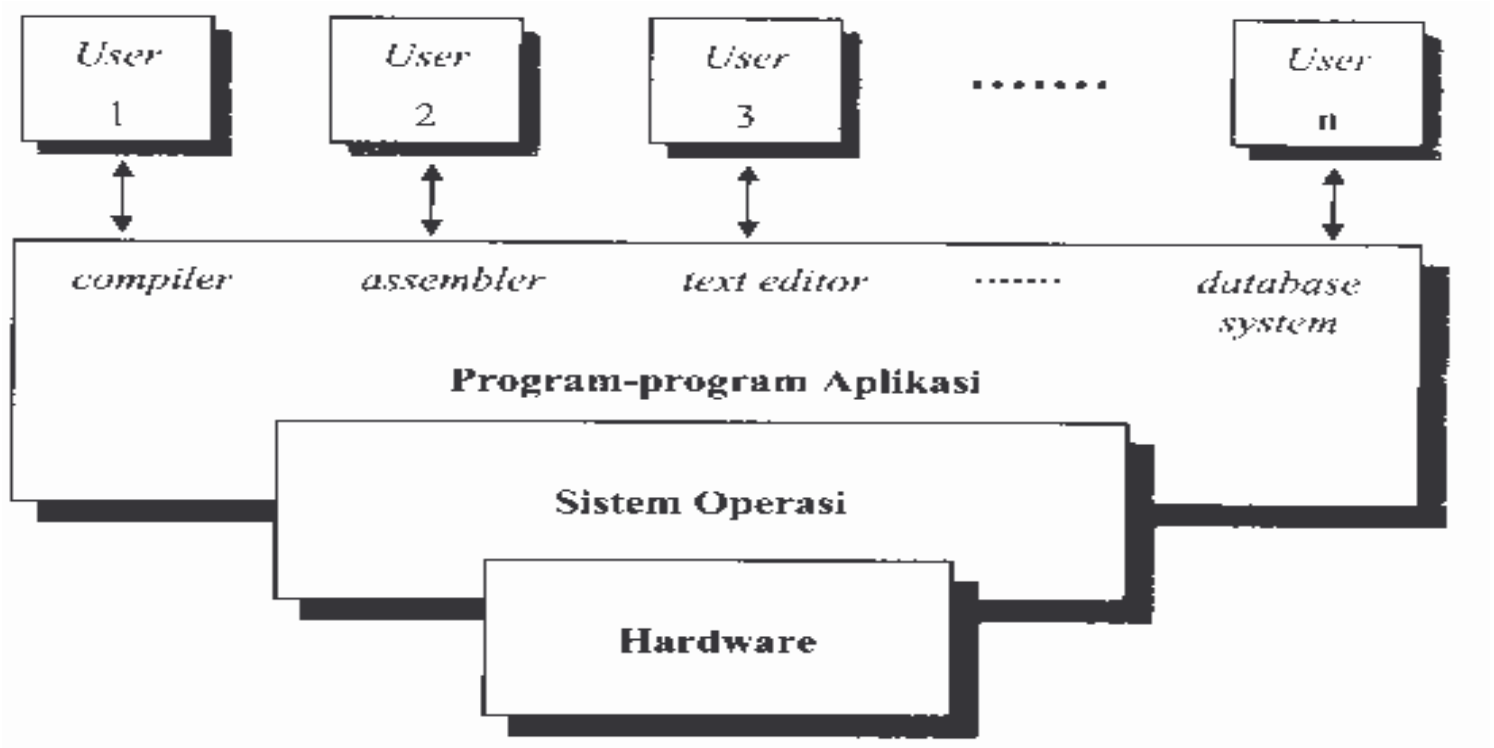
Pengenalan Sistem Operasi

Sistem Operasi

- Definisi
  - Sistem operasi adalah program yang bertindak sebagai perantara antara user dengan komputer hardware
- Maksud
  - Memberikan lingkungan dimana user dapat mengeksekusi program
- Tujuan
  - Primer : agar sistem komputer sesuai dengan kegunaan
  - Sekunder : menggunakan hardware dengan efisien

Apakah Sistem Operasi ?

- Sistem komputer dibagi menjadi 4 komponen : hardware, sistem operasi, program aplikasi dan user



Peranan Sistem Operasi

- Bertindak sebagai "pemerintah"
  - Mempengaruhi penggunaan komponen sistem komputer yang tepat : h/w, s/w dan data
  - Memberi lingkungan sehingga program dapat berguna
- Dipandang sebagai "resource allocator"
  - Manajer dari resource : CPU time, memory space, file storage I/O device dll
  - Memberi resource bagi program tertentu dan user sesuai kebutuhan
  - Menentukan permintaan yang diberikan resource sehingga sistem komputer berjalan efisien dan fair
- Dipandang sebagai "control program"
  - Mengontrol perangkat I/O dan program user yang berbeda
  - Mengontrol eksekusi program user untuk mencegah error dan penggunaan komputer yang tidak tepat

Sistem Pendahulu

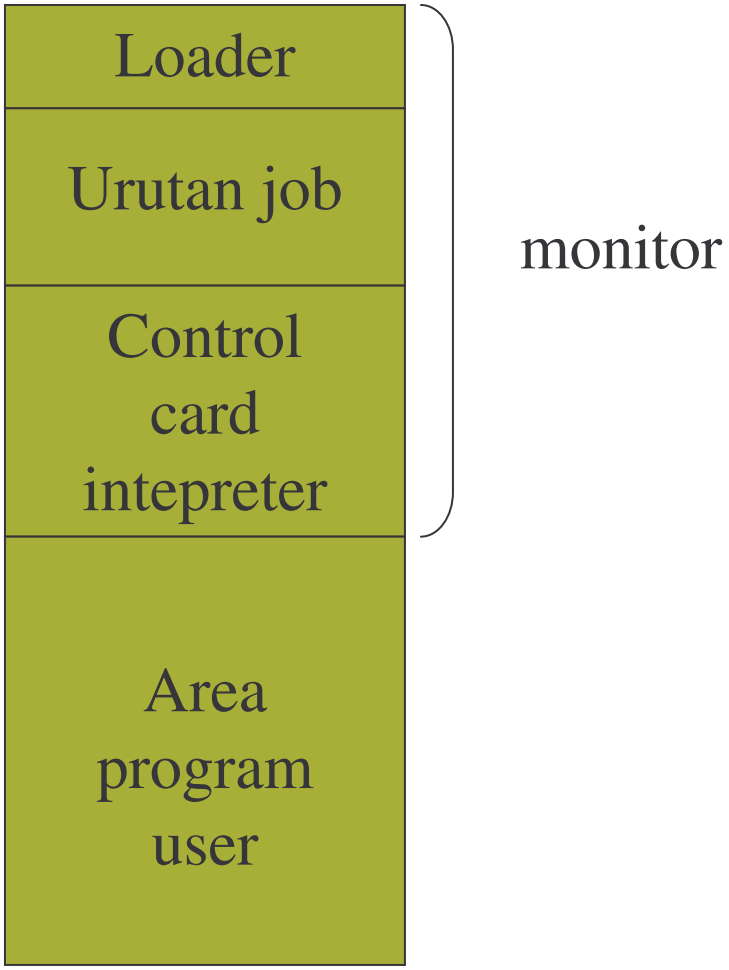
- Karakteristik :
  - Mesin sangat besar
  - Programmer (operator) menulis program dan mennjalankan program langsung dari console
  - Setiap satu step dikerjakan manual
- Tersedia hardware dan software tambahan
  - Hardware : card reader, line printer, magnetic tape
  - Software : assembler, loader, linkers
- “device driver” : subroutine untuk device khusus seperti buffer, flag, register, control bit dan status bit
- Compiler FORTRAN, COBOL

Sistem Batch Sederhana

- IDE : Pengumpulan job-job yang sejenis sebagai satu kelompok
- Contoh : Operator menerima satu job FORTRAN, kemudian satu job COBOL dan satu job FORTRAN lain
  - UNBATCH : job FORTRAN $\rightarrow$ job COBOL $\rightarrow$ job FORTRAN
  - BATCH : job FORTRAN $\rightarrow$ job COBOL
- Masalah :
  - Bila job berhenti, operator harus memeriksa console unuk menentukan mengapa program berhenti
  - Bila operator mengerjakan event sampai komputer restart, CPU menjadi idle

Bentuk Sistem Batch (1)

- Resident monitor :
  - Job dikerjakan sesuai urutan secara otomatis, tersedia program kecil resident di memori yang berisi urutan job yang akan berpindah yang disebut resident monitor
  - Control card interpreter untuk menentukan program yang dieksekusi
  - Loader menyimpan program sistem dan program aplikasi ke memory



Bentuk Sistem Batch (2)

- Overlap operasi I/O dengan CPU
  - Alasan : dengan resident monitor, CPU kemungkinan masih idle karena kecepatan mekanik perangkat I/O lebih lambat daripada perangkat elektronik
  - Off-line processing
    - komputer tidak lagi terhambat kecepatan card reader dan line printer tetapi terbatas magnetic tape yang lebih cepat
  - Spooling (Simultaneously Peripheral Operation On-Line)
    - Bentuk pemrosesan dimana sistem disk langsung berhubungan dengan card reader (perangkat input) dan printer (perangkat output) secara simultan
    - Menggunakan buffer sangat besar, untuk membaca dan mempersiapkan file output

Multiprogramming System

- Beberapa job yang siap dieksekusi dikumpulkan dalam suatu pool
- Beberapa job yang siap dieksekusi diletakkan di memori utama. Memori utama dibagi menjadi beberapa partisi, foreground partition untuk program dengan prioritas lebih tinggi, background partition untuk program dengan prioritas lebih rendah
- Jika job yang dieksekusi menunggu beberapa task (contoh : input dari keyboard, maka diganti job berikutnya
- Tugas sistem operasi menangani perpindahan/switch dari proses tsb
- Contoh : MS-Windows 3.0, Windows NT, OS/2, Macintosh System 7

Time Sharing System

- Disebut juga multitasking
- Sama dengan multiprogramming system tapi waktu prosesnya dibatasi
- Waktu maksimum yang digunakan disebut quantum time
- Keuntungan : tingkat kebersamaannya tinggi
- Kerugian : switching time besar sehingga utilitas rendah
- CPU burst > quantum-time maka proses dihentikan sementara dan mengantri di posisi ekor dari ready queue & CPU menjalankan proses berikutnya

Multiprocessing System

- Sistem memiliki lebih dari satu prosessor untuk menjalankan satu atau lebih program, menggunakan bus, clock, memori dan peralatan lain secara bersama-sama
- Disebut juga *tightly coupled system*
- Dibagi menjadi :
  - Symmetric Multiprocessing, tiap prosessor mempunyai sistem operasi yang sama
  - Asymmetric Multiprocessing , satu prosessor berfungsi sebagai master prosessor (mengatur penjadwalan dan mengalokasikan kerja tiap-tiap prosessor) dan prosessor-prosessor lain berfungsi sebagai slave
- Contoh : MS Windows NT, UNIX, Linux

Distributed System

- Disebut juga *loosely coupled system*
- Kumpulan prosessor yang tidak menggunakan memory atau clock secara bersama-sama
- Keuntungan :
  - Pemakaian resource secara bersama-sama → resource yang ada pada suatu tempat dapat digunakan oleh tempat yang lainnya
  - Kecepatan komputasi → dibagi menjadi sub-komputasi yang masing-masing dikerjakan tiap-tiap prosessor yang ada
  - Reliabilitas → jika proses dikerjakan beberapa prosessor, maka jika salah satu prosessor gagal masih ada prosessor lain yang dapat mengerjakan
  - Komunikasi →dimungkinkan transfer data dari satu program ke program lainnya
- Contoh : AMOEBA, MACH

Real Time System

- Jika suatu operasi memerlukan ketepatan waktu dari prosessor atau aliran data
- Bentuk real time system :
  - Hard Real Time → critical task dapat diselesaikan tepat waktu
  - Soft Real Time → memberikan prioritas pada critical task dibandingkan dengan task yang lainnya sehingga critical task tsb selesai dikerjakan

Sistem Operasi dalam berbagai sudut pandang

Pemakai & administrator sistem

- ✓ Antarmuka yang disediakan aplikasi dalam menyelesaikan masalah yang dihadapi
- ✓ tidak berurusan dengan arsitektur komputer, sebatas menggunakan *command-language* untuk meminta layanan sistem operasi
- ✓ *Command-language* terdapat di shell
  - text-based shell, contoh : MS-DOS, UNIX[TD]
  - GUI based shell, contoh : MS-Windows 95/98, X-Windows

Pemrogram

- ✓ Membuat aplikasi untuk pemakai dengan menggunakan bahasa pemrograman
- ✓ Bertanggung jawab mengelola dan mengendalikan seluruh perangkat komputer
- ✓ Level :
  - Program utilitas → membantu penciptaan program, manajemen berkas, mengendalikan perangkat masukan/keluaran, tugas dasar lain
  - Service-interface → pustaka rutin untuk melakukan layanan
  - System Call (API – Application programming interface) → tata cara pemanggilan di program aplikasi untuk memperoleh layanan SO, contoh : command-language (type, copy, del) dikonversi dan dieksekusi sebagai rangkaian system call yang memberi fasilitas pengendalian atas operasi sistem lebih baik dan pengaksesan ke fasilitas hardware lebih langsung

Perancang Sistem Operasi

- ✓ Membuat sistem operasi yang dapat mempermudah dan menyamankan terutama untuk pemrogram dalam membuat aplikasi-aplikasi
- ✓ Menghindari rincian operasi perangkat keras

Sistem Operasi dalam berbagai sudut pandang