



KONSEP DAN STRUKTUR SISTEM OPERASI

**Core Jurusan Teknik Informatika
Kode MK/SKS : TIF 19417/ 4**

Struktur Sistem Operasi

2. Sistem Monolithic

Struktur SO di sistem ini tidak terstruktur SO sbg kumpulan prosedur yg masing-masing dpt saling dipanggil jika dibutuhkan.

Setiap prosedur dalam sistem ini memiliki:

- **interface yg sudah didefinisikan dgn baik berupa parameter dan hasilnya.**
- **masing -masing prosedur bebas utk saling memanggil jika dibutuhkan**

Setiap system call memiliki satu prosedur yg akan mengelolanya.

Utilitas prosedur mengerjakan suatu hal yg diinginkan oleh beberapa service procedure seperti pengambilan data dari program pemakai.

Struktur Sistem Operasi

3. Sistem Lapisan (layered System)

Sistem pertama yg dibangun dgn sistem lapisan adalah THE (dibuat di *Technische Hogeschool Eindhoven* di Belanda 1968 oleh E.W. Dijkstra dan para mahasiswanya)

Struktur THE

5	Program pemakai
4	Buffering untuk perangkat I/O
3	Operator – console device driver
2	Manajemen memori
1	Penjadwalan CPU
0	Hardware

Masalah Implementasi Sistem Lapisan

Kurang efisien dibandingkan sistem yang lain.

Contoh:

Suatu program pemakai akan mengeksekusi suatu operasi I/O.

Masalah Implementasi Sistem Lapisan

Di setiap lapisan, parameter dan data yg digunakan memiliki kemungkinan utk dimodifikasi .



KONSEP DAN STRUKTUR SISTEM OPERASI

**Core Jurusan Teknik Informatika
Kode MK/SKS : TIF 19417/ 4**

Struktur Sistem Operasi

4. Mesin Virtual

Secara konsep, Sistem komputer dibuat berdasarkan lapisan.

Hardware merupakan tingkatan terbawah dari keseluruhan sistem.

Kernel yg berjln di tingkat berikutnya menggunakan instruksi-instruksi hardware utk membuat kumpulan system call yg digunakan oleh lapisan di luarnya.

Program sistem di atas kernel dpt menggunakan system call atau instruksi-instruksi hardware.

Program sistem memperlakukan system call dan hardware di lapisan yg sama.

Beberapa sistem mengadopsi teknik ini supaya program sistem dpt dipanggil dgn mudah oleh program aplikasi.

Walaupun program sistem ada pd tingkat yg lebih tinggi dari rutin-rutin program aplikasi, program aplikasi akan melihat semua yg berada di bawahnya (dalam hirarki)

Dgn menggunakan penjadwalan CPU dan teknik virtual memori, sebuah SO dpt membuat suatu bayangan proses dlm jumlah banyak., yg masing-masing dieksekusi oleh prosesornya sendiri dengan memori (virtual) sendiri.

Secara normal, proses memiliki ciri-ciri tambahan seperti: system call dan sistem file , yg tdk disediakan oleh hardware.

Pd mesin virtual, tdk tersedia fungsi-fungsi tambahan tetapi ada interface yg identik dgn hardware yg mendasarinya

Setiap proses dilengkapi dgn salinan dari komputer yang mendasarinya.



KONSEP DAN STRUKTUR SISTEM OPERASI

**Core Jurusan Teknik Informatika
Kode MK/SKS : TIF 19417/ 4**

Sumber daya komputer secara fisik dibagi pakai utk membuat suatu mesin virtual.

Penjadwalan CPU dpt digunakan utk membagi pakai CPU dan membuat tampilan seolah-olah pemakai memiliki prosesor sendiri.

Mesin Virtual kembali dikembangkan dan muncul lagi ketika ada masalah kompatibilitas sistem.

Contoh:

Ketika komputer yg tdk berbasis prosesor Intel akan memakai program MS DOS yg berjalan di atas mesin Intel.

Bagaimana komputer dgn prosesor lebih cepat dapat menjalankan program MS-DOS?

Solusinya:

Membuat mesin Intel virtual di atas prosesor aslinya.

Sebuah program MS-DOS dpt berjalan di lingkungan tsb dan instruksi Intel diterjemahkan ke kumpulan instruksi aslinya.

Sistem Operasi MS-DOS sendiri juga dpt berjalan dalam mesin virtual ini, sehingga program-programnya dpt membuat system call masing-masing.

Jika prosesor berkecepatan lebih tinggi, maka program MS-DOS jg akan berjalan dgn cepat walaupun setiap instruksinya harus diterjemahkan ke dalam instruksi asli agar dpt dieksekusi.

Dengan kata lain, sebuah program yg seharusnya berjalan di mesin berbasis prosesor Intel dpt berjln dgn baik di mesin berbasis prosesor yg berbeda sama sekali.



KONSEP DAN STRUKTUR SISTEM OPERASI

**Core Jurusan Teknik Informatika
Kode MK/SKS : TIF 19417/ 4**

Struktur Sistem Operasi

4. Model Client-Server

Kecenderungan SO modern adalah memindah kode-kode SO ke lapisan-lapisan di atasnya dan membuang sebanyak mungkin kode dari SO yang akan membuat suatu kernel yang minimal.

Dengan mengimplementasikan sebanyak mungkin fungsi-fungsi SO di proses-proses pemakai.

Utk meminta layanan seperti membaca sebuah blok dari suatu file, proses pemakai mengirimkan permintaan tsb ke proses server yg kemudian bekerja dan mengirimkan kembali suatu jawaban

Semua kernel bekerja menangani komunikasi antara server dan klien.

Dgn membagi SO menjadi banyak bagian, yg masing-masing hanya menangani bagian tertentu dari sistem, seperti layanan file, layanan proses, layanan terminal, layanan memori, maka setiap bagian menjadi lebih kecil dan lebih mudah diatur.

Krn semua server berjalan dalam proses user mode dan tidak dlm kernel mode, maka bagian-bagian tsb tdk memiliki akses langsung ke hardware.

Keuntungan Model Client-Server

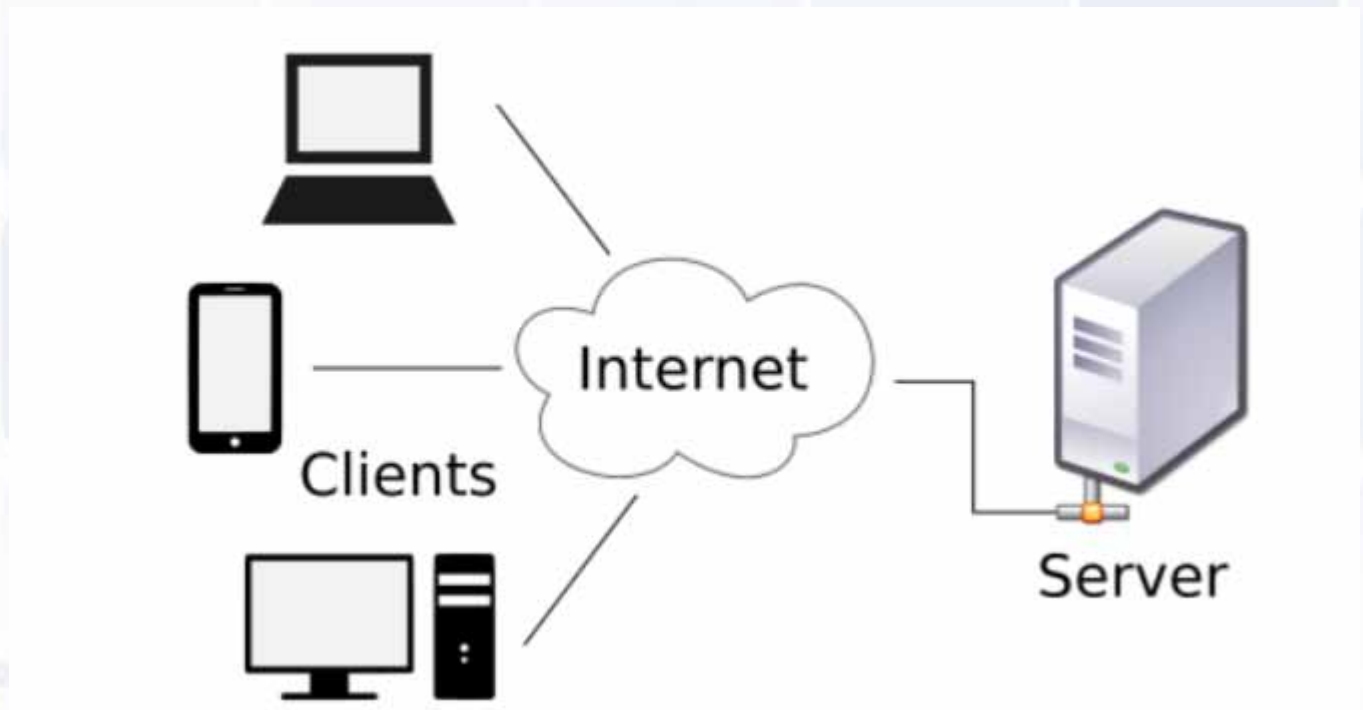
Bila ada pemicu kesalahan di file server maka layanan file mungkin akan *crash*, tetapi tdk mempengaruhi keseluruhan sistem yg akhirnya dpt mengakibatkan keseluruhan sistem tdk berfungsi.

Keuntungan Model Client-Server

Dapat diadaptasi untuk digunakan dalam sistem terdistribusi.

Jika client berkomunikasi dgn server yaitu dengan mengirimkan pesan, client tdk perlu tahu pesan tsb ditangani secara lokal dalam mesinnya sendiri atau pesan tsb dikirimkan melalui jaringan server di mesin yg lain (jarak jauh)

Gambar Model Client Server



Gambar Model Client Server Terdistribusi

