

## Komponen dan struktur sistem operasi

Nama : Rudi Setiawan  
NIM : 055318532

---

## Komponen Sistem Operasi

---

### Manajemen Proses

Mengatur proses (program yang berjalan), termasuk penjadwalan, sinkronisasi, dan komunikasi antar-proses.

Tugas utamanya adalah memastikan efisiensi dan keadilan dalam penggunaan CPU, terutama dalam lingkungan multiprogramming. OS menggunakan Process Control Block (PCB) untuk melacak status setiap proses.

### Manajemen Memori Utama

Mengalokasikan dan mendealokasikan memori utama (RAM) ke proses dan melacak penggunaannya.

Fungsinya melibatkan proteksi memori (memastikan satu proses tidak mengganggu memori proses lain) dan mekanisme paging atau segmentation untuk memuat dan mengakses instruksi secara efisien.

### Manajemen File

Mengelola penyimpanan data pada media sekunder, termasuk pembuatan, penghapusan, dan pengorganisasian file serta direktori.

Menyediakan abstraksi logis (konsep file dan direktori) kepada pengguna, menyembunyikan detail fisik penyimpanan data, dan menangani hak akses (permissions).

### Manajemen I/O

Mengelola perangkat Input/Output (I/O) dan driver perangkat yang menghubungkannya dengan sistem.

Bertanggung jawab atas sistem buffering, caching, dan spooling untuk mengoptimalkan kinerja I/O, serta menyediakan antarmuka umum (device-independent interface) untuk driver.

### Manajemen Penyimpanan Sekunder

Atau sering disebut Manajemen Disk.

Mengelola disk penyimpanan jangka panjang (HDD, SSD), termasuk ruang kosong (free-space management) dan penjadwalan disk.

Ini mengatur block mana yang bebas dan yang terpakai pada disk, dan menggunakan algoritma penjadwalan (seperti SCAN atau C-SCAN) untuk meminimalkan waktu pencarian (seek time).

## **Sistem Jaringan (Sistem Terdistribusi)**

Menyediakan mekanisme untuk komunikasi dan akses sumber daya di jaringan.

OS modern menyediakan fitur untuk protokol jaringan (TCP/IP), keamanan, dan implementasi sistem file terdistribusi (distributed file systems).

## **Sistem Proteksi**

Mengontrol akses ke sumber daya sistem untuk mencegah kerusakan atau penggunaan yang tidak sah.

Mekanisme ini menggunakan domain proteksi dan daftar kontrol akses (ACL) untuk memverifikasi bahwa proses atau pengguna memiliki otorisasi yang benar sebelum mengakses objek sistem (seperti file atau memori).

## **Sistem Command Interpreter**

Antarmuka (shell terminal atau GUI) yang menerima dan menjalankan perintah dari pengguna.

Ini adalah bagian Sistem Operasi yang paling sering dilihat user. Meskipun secara teknis bukan inti kernel, ia adalah komponen penting yang menjembatani interaksi manusia-komputer. Contohnya adalah CMD/PowerShell di Windows atau Bash di Linux.

---

## Struktur Sistem Informasi

---

### **Struktur Sederhana**

Tidak memiliki pemisahan yang jelas antara modul atau lapisan; seluruh fungsionalitas dijalankan dalam ruang kernel, yang rentan terhadap kegagalan.

Contohnya: MS-DOS

### **Struktur Monolitik**

Seluruh komponen SO bekerja dalam ruang kernel yang besar. Efisien, tetapi sulit dikelola dan di-debug.

Contoh: UNIX System

### **Pendekatan Terlapis (Layered Approach)**

Sistem Operasi dibagi menjadi lapisan-lapisan, di mana setiap lapisan hanya dapat berinteraksi dengan lapisan di bawah dan di atasnya. Memudahkan modularitas dan debugging.

Contoh:

Sistem Operasi THE yang dikembangkan oleh Edsger W. Dijkstra

### **Microkernel**

Memindahkan sebagian besar fungsionalitas Sistem Operasi (seperti manajemen I/O dan driver) dari kernel ke user-space. Kernel hanya berisi fungsi dasar seperti komunikasi antar-proses dan manajemen memori. Meningkatkan keandalan dan keamanan.

Contoh:

Mach oleh Richard Rashid di Carnegie Mellon University (CMU)

### Kernel Hibrida (Hybrid Kernel)

Menggabungkan aspek dari kernel monolitik dan microkernel untuk mendapatkan performa yang baik dengan modularitas tertentu.

Contoh:

Windows, MacOS