



Program Studi Teknik Informatika  
Institut Teknologi Sumatera

---

---

MODUL PRAKTIKUM

# **Pengenalan Antarmuka Sistem Operasi dan Virtual Machine**

Modul ke-1

---

---

Mata Kuliah : Sistem Operasi (IF25-12007)  
Institusi : Institut Teknologi Sumatera  
Dosen : Nama Dosen Pengampu, S.T., M.T.  
Asisten : Zakhi Al-Ghifari  
Elsa Elisa Yohana Sianturi

Semester : Genap 2025/2026

---

---

## Contents

<b>1</b>	<b>Tujuan dan <i>Output</i> Praktikum</b>	<b>3</b>
1.1	Tujuan Praktikum . . . . .	3
1.2	<i>Output</i> Praktikum . . . . .	3
<b>2</b>	<b>Sistem Operasi</b>	<b>4</b>
2.1	Definisi Sistem Operasi . . . . .	4
2.2	Fungsi Utama Sistem Operasi . . . . .	5
2.3	Jenis Jenis Sistem Operasi . . . . .	5
2.4	Antarmuka Sistem Operasi . . . . .	7
<b>3</b>	<b><i>Virtual Machine</i></b>	<b>8</b>
3.1	Pengertian <i>Virtual Machine</i> . . . . .	8
3.2	Jenis Jenis <i>Virtual Machine</i> . . . . .	8

## 1 Tujuan dan *Output* Praktikum

### 1.1 Tujuan Praktikum

Setelah menyelesaikan praktikum ini, mahasiswa diharapkan mampu:

1. Menjelaskan apa itu sistem operasi dalam bahasa sederhana.
2. Mengidentifikasi peran sistem operasi dalam kehidupan sehari-hari.
3. Membedakan antarmuka *GUI* (*Graphical User Interface*) dan *CLI* (*Command Line Interface*) berdasarkan pengalaman penggunaan.
4. Memahami konsep dasar *virtual machine* secara konseptual.
5. Menjelaskan mengapa *virtual machine* digunakan dalam praktikum sistem operasi.

### 1.2 *Output* Praktikum

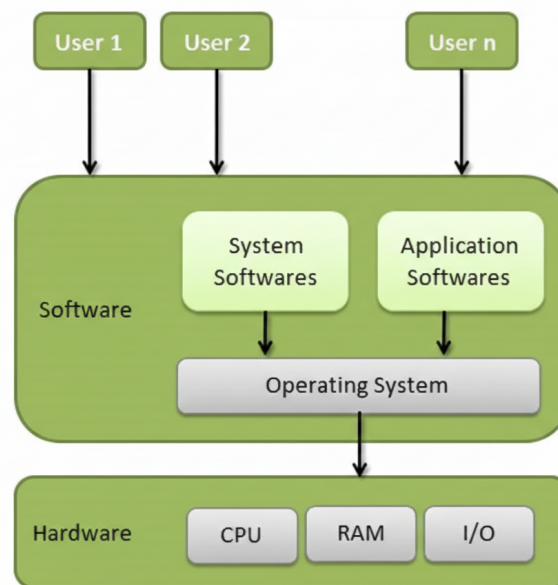
Pada akhir praktikum ini, mahasiswa diharapkan menghasilkan:

1. Pemahaman tentang definisi dan fungsi sistem operasi sebagai perangkat lunak yang mengelola sumber daya komputer.
2. Pengetahuan tentang perbedaan antara *GUI* dan *CLI*, termasuk kelebihan dan kekurangan masing-masing.
3. Pemahaman konsep *virtual machine* sebagai simulasi perangkat keras yang memungkinkan menjalankan beberapa sistem operasi pada satu mesin fisik.

## 2 Sistem Operasi

### 2.1 Definisi Sistem Operasi

Sistem Operasi (*Operating System/OS*) merupakan perangkat lunak sistem yang berfungsi sebagai penghubung antara pengguna (*user*), perangkat lunak aplikasi, dan perangkat keras komputer. Hubungan tersebut dapat dipahami melalui arsitektur berlapis sebagaimana ditunjukkan pada Gambar 1.



Gambar 1: Arsitektur Sistem Operasi

Pada lapisan paling atas terdapat *User* (*User 1, User 2, ..., User n*) yang merepresentasikan individu atau entitas yang menggunakan sistem komputer. Pengguna tidak berinteraksi langsung dengan perangkat keras, melainkan melalui perangkat lunak.

Lapisan berikutnya adalah *Software*, yang terdiri dari dua kategori utama:

1. **Application Software** merupakan perangkat lunak yang digunakan secara langsung oleh pengguna untuk menyelesaikan tugas tertentu, seperti pengolah kata, peramban web, atau perangkat lunak pemrograman.
2. **System Software** merupakan perangkat lunak pendukung yang membantu pengoperasian sistem secara keseluruhan, termasuk *compiler*, *interpreter*, dan utilitas sistem.

Di bawah kedua jenis perangkat lunak tersebut terdapat *Operating System*. Pada posisi inilah sistem operasi berperan sebagai pengelola dan pengendali utama sistem komputer. Sistem operasi menjadi perantara antara perangkat lunak dengan perangkat keras. Lapisan paling bawah adalah *Hardware*, yang terdiri dari:

1. CPU (*Central Processing Unit*) sebagai pemroses instruksi,
2. RAM (*Random Access Memory*) sebagai penyimpanan sementara,
3. I/O (*Input/Output Devices*) seperti *keyboard*, *mouse*, dan perangkat penyimpanan.

Struktur ini menunjukkan bahwa:

1. Pengguna → berinteraksi dengan aplikasi,

2. Aplikasi → meminta layanan dari sistem operasi,
3. Sistem operasi → mengatur dan mengalokasikan sumber daya perangkat keras,
4. Perangkat keras → mengeksekusi instruksi.

Dengan demikian, sistem operasi tidak sekadar “penghubung”, tetapi merupakan pengelola sumber daya (*resource manager*) yang mengontrol akses terhadap CPU, memori, dan perangkat I/O agar dapat digunakan secara efisien dan terorganisasi oleh berbagai aplikasi serta pengguna secara bersamaan. Tanpa sistem operasi, perangkat lunak tidak memiliki mekanisme terstandarisasi untuk mengakses perangkat keras. Akibatnya, setiap program harus berkomunikasi langsung dengan perangkat keras, yang dapat menyebabkan konflik penggunaan sumber daya dan ketidakteraturan sistem [1].

## 2.2 Fungsi Utama Sistem Operasi

Secara umum, sistem operasi memiliki beberapa fungsi utama sebagai pengelola sumber daya dalam sistem komputer, yaitu:

### 1. Manajemen Proses

Sistem operasi mengatur eksekusi program yang sedang berjalan (proses) serta melakukan pembagian waktu penggunaan prosesor (CPU). Dalam lingkungan multiprogramming, sistem operasi menentukan proses mana yang dijalankan, kapan dijalankan, dan berapa lama waktu eksekusinya.

### 2. Manajemen Memori

Sistem operasi bertanggung jawab dalam mengalokasikan dan mengelola penggunaan memori utama (RAM). Sistem operasi melacak bagian memori yang sedang digunakan, menentukan proses mana yang memperoleh alokasi memori, serta membebaskan memori ketika proses telah selesai dijalankan.

### 3. Manajemen Sistem Berkas (*File System*)

Sistem operasi mengatur penyimpanan dan pengorganisasian data pada media penyimpanan. Hal ini mencakup pengelolaan direktori, file, hak akses, serta alokasi dan dealokasi ruang penyimpanan.

### 4. Manajemen Perangkat I/O

Sistem operasi mengontrol interaksi antara perangkat lunak dan perangkat keras *input/output* seperti *keyboard*, *mouse*, *printer*, dan perangkat penyimpanan. Pengelolaan ini dilakukan melalui *driver* perangkat agar komunikasi berjalan secara terstandarisasi dan efisien.

### 5. Keamanan dan Proteksi

Sistem operasi mengatur hak akses pengguna terhadap sumber daya sistem serta melindungi data dan program dari akses yang tidak sah. Mekanisme ini mencakup autentikasi pengguna, kontrol akses, dan isolasi antar proses.

## 2.3 Jenis Jenis Sistem Operasi

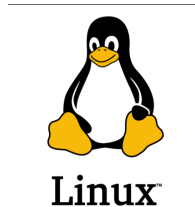
### 1. *Microsoft Windows*



Gambar 2: *Microsoft Windows*

*Microsoft Windows* merupakan sistem operasi yang dikembangkan oleh *Microsoft* dan banyak digunakan pada komputer personal di seluruh dunia. *Windows* dikenal dengan antarmuka grafis yang intuitif serta dukungan perangkat lunak yang sangat luas. Sistem operasi ini banyak digunakan dalam lingkungan perkantoran, pendidikan, dan industri karena kompatibilitasnya dengan berbagai perangkat keras dan aplikasi komersial.

## 2. *GNU/Linux*



Gambar 3: *GNU/Linux*

*GNU/Linux* adalah sistem operasi berbasis kernel *Linux* yang bersifat *open-source*. Sistem operasi ini dikenal karena stabilitas, keamanan, dan fleksibilitasnya dalam berbagai kebutuhan komputasi. *Linux* banyak digunakan pada *server*, sistem jaringan, dan lingkungan pengembangan perangkat lunak. Tersedia dalam berbagai distribusi seperti *Ubuntu*, *Debian*, dan *Fedora*, yang dirancang untuk kebutuhan pengguna yang berbeda.

## 3. *UNIX*



Gambar 4: *UNIX*

*UNIX* merupakan sistem operasi yang dikembangkan pada akhir tahun 1960-an dan menjadi dasar bagi banyak sistem operasi modern. *UNIX* dirancang dengan konsep *multiuser* dan *multitasking* yang kuat, sehingga banyak digunakan dalam sistem komputasi skala besar dan lingkungan akademik. Arsitektur dan filosofi desain *UNIX* memberikan pengaruh signifikan terhadap pengembangan *Linux* dan *macOS*.

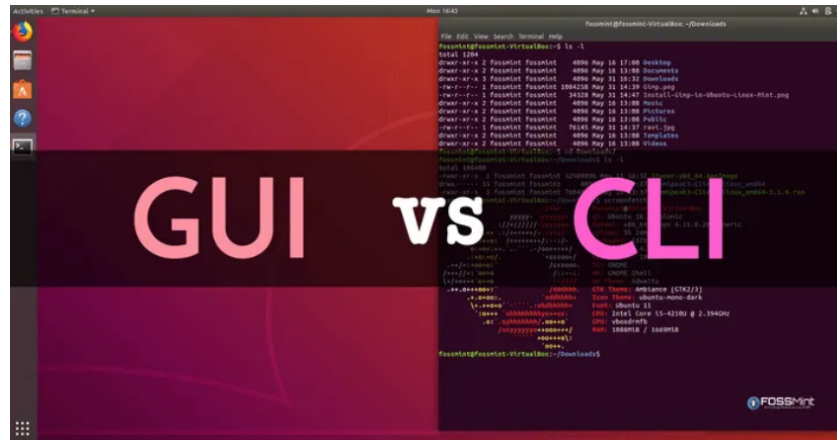
## 4. *macOS*



Gambar 5: *macOS*

*macOS* adalah sistem operasi yang dikembangkan oleh *Apple* untuk perangkat komputer *Macintosh*. Sistem operasi ini berbasis *UNIX* dan dikenal dengan desain antarmuka yang konsisten serta integrasi yang erat dengan ekosistem perangkat *Apple*. *macOS* banyak digunakan dalam bidang desain grafis, multimedia, dan pengembangan aplikasi karena stabilitas serta optimalisasi perangkat keras dan perangkat lunaknya.

## 2.4 Antarmuka Sistem Operasi



Gambar 6: *Command Line Interface (CLI)*

Antarmuka sistem operasi merupakan mekanisme yang memungkinkan pengguna berinteraksi dengan sistem komputer. Melalui antarmuka ini, pengguna dapat memberikan perintah, menjalankan aplikasi, serta mengakses sumber daya sistem. Secara umum, terdapat dua bentuk utama antarmuka, yaitu *Graphical User Interface (GUI)* dan *Command Line Interface (CLI)*. *GUI* menggunakan elemen visual seperti jendela (*window*), ikon, menu, dan tombol sehingga lebih intuitif dan mudah digunakan, terutama oleh pengguna pemula. Sementara itu, *CLI* memungkinkan interaksi melalui perintah berbasis teks dengan mengetikkan instruksi tertentu untuk menjalankan program, mengelola file, dan mengakses konfigurasi sistem. *CLI* banyak digunakan pada lingkungan *server*, administrasi sistem, dan pengembangan perangkat lunak karena lebih efisien, fleksibel, serta mendukung automasi melalui skrip. *GUI* unggul dalam kemudahan penggunaan, sedangkan *CLI* memberikan kontrol yang lebih detail terhadap sistem.

### 3 *Virtual Machine*

#### 3.1 *Pengertian Virtual Machine*

*Virtual Machine* (VM) adalah representasi komputer secara virtual yang berjalan di atas sistem operasi utama melalui teknologi virtualisasi. Dengan pendekatan ini, satu perangkat keras fisik dapat menjalankan lebih dari satu sistem operasi secara bersamaan. Secara konsep, VM dapat dipahami sebagai “komputer di dalam komputer”. Sistem operasi utama disebut *Host Operating System*, sedangkan sistem operasi di dalam VM disebut *Guest Operating System*. Proses virtualisasi tersebut dikelola oleh perangkat lunak yang disebut *hypervisor*. Dalam pembelajaran sistem operasi, VM berguna karena menyediakan lingkungan terisolasi untuk eksperimen, instalasi, dan konfigurasi tanpa mengganggu sistem utama.

*Virtual Machine* umum dimanfaatkan untuk:

1. Pengujian perangkat lunak.
2. Simulasi instalasi sistem operasi.
3. Keamanan dan *sandboxing*.
4. Pengembangan dan penelitian sistem.

#### 3.2 *Jenis Jenis Virtual Machine*

Terdapat berbagai perangkat lunak virtualisasi yang digunakan untuk menjalankan mesin virtual. Beberapa di antaranya adalah sebagai berikut:

1. ***Oracle VM VirtualBox***



Gambar 7: *Oracle VM VirtualBox*

*Oracle VM VirtualBox* merupakan perangkat lunak virtualisasi yang memungkinkan pengguna menjalankan sistem operasi tambahan di dalam sistem operasi utama. Sebagai contoh, pengguna dengan sistem operasi *Windows* dapat menjalankan *Linux* di dalamnya tanpa mengubah konfigurasi sistem utama. *VirtualBox* banyak digunakan dalam lingkungan pendidikan karena bersifat gratis dan relatif mudah digunakan.

2. ***Parallels Desktop***

*Parallels Desktop* adalah perangkat lunak virtualisasi yang dirancang khusus untuk komputer *Macintosh* berbasis prosesor *Intel* maupun *Apple Silicon*. Perangkat lunak ini memungkinkan pengguna *macOS* menjalankan sistem operasi lain seperti *Windows* secara bersamaan dalam satu perangkat.

3. ***VMware***

*VMware Workstation* merupakan perangkat lunak virtualisasi untuk arsitektur x86 dan x86-64. *VMware* memungkinkan pembuatan beberapa mesin virtual yang dapat dijalankan secara simultan. Produk *VMware* banyak digunakan dalam lingkungan profesional dan industri karena stabilitas serta fitur manajemen yang lengkap.



#### 4. ***QEMU***

*QEMU (Quick Emulator)* adalah perangkat lunak virtualisasi dan emulasi yang bersifat *open-source*. *QEMU* mampu melakukan emulasi berbagai arsitektur prosesor serta menjalankan sistem operasi tamu pada lingkungan yang berbeda. *QEMU* sering digunakan dalam pengembangan sistem operasi dan penelitian karena fleksibilitasnya yang tinggi.

#### 5. ***Microsoft Virtual PC***

*Microsoft Virtual PC* merupakan perangkat lunak virtualisasi yang dikembangkan oleh *Microsoft* untuk sistem operasi *Windows*. Perangkat lunak ini memungkinkan pengguna menjalankan sistem operasi lain di dalam lingkungan *Windows*. Meskipun kini telah digantikan oleh teknologi lain seperti *Hyper-V*, *Microsoft Virtual PC* menjadi salah satu pelopor virtualisasi pada platform *Windows*.

#### 6. ***Xen***

*Xen* adalah *Virtual Machine Monitor (VMM)* yang dikembangkan sebagai proyek penelitian di *University of Cambridge*. *Xen* banyak digunakan dalam lingkungan *server* dan komputasi awan karena mendukung virtualisasi dengan performa tinggi.

#### 7. ***KVM (Kernel-based Virtual Machine)***

*KVM* adalah teknologi virtualisasi yang terintegrasi langsung ke dalam kernel *Linux*. *KVM* menyediakan virtualisasi penuh untuk sistem berbasis x86 dan banyak digunakan pada *server Linux* serta layanan komputasi awan.

#### 8. ***OpenStack***

*OpenStack* merupakan arsitektur layanan *Infrastructure as a Service (IaaS)* yang mendukung berbagai *hypervisor* seperti *KVM*, *VMware*, *Xen*, dan *QEMU*. *OpenStack* digunakan dalam pengelolaan infrastruktur komputasi awan berskala besar [2].

## References

- [1] Tutorials Point. (2026) Operating system overview. TutorialsPoint. Diakses tanggal 24 Februari 2026. [Online]. Available: [https://www.tutorialspoint.com/operating\\_system/os\\_overview.htm](https://www.tutorialspoint.com/operating_system/os_overview.htm)
- [2] S. M. Prasetyo, M. B. Agusti, D. A. Mahesa, F. Maulana, and A. Rafly, “Mesin virtual (virtual machine): Sekilas tentang tujuan, fungsi, keuntungan, dan pengelolaan dari mesin virtual,” *BIIKMA: Buletin Ilmiah Ilmu Komputer dan Multimedia*, vol. 1, no. 6, pp. 743–749, April 2024. [Online]. Available: <https://jurnalmahasiswa.com/index.php/biikma>