

1. Pengenalan Sistem Operasi & Instalasi

Bab ini memperkenalkan konsep fundamental sistem operasi, arsitektur kernel, perbandingan sistem operasi modern, serta proses instalasi dan konfigurasi sistem operasi dengan fokus praktis menggunakan Linux.

1.1. Definisi dan Fungsi Sistem Operasi

Sistem operasi adalah perangkat lunak sistem yang mengelola sumber daya perangkat keras komputer dan menyediakan layanan umum untuk program aplikasi. Sistem operasi bertindak sebagai perantara antara pengguna dengan perangkat keras komputer.

1.1.1. Peran Ganda Sistem Operasi

Sistem operasi memiliki dua peran utama:

1. **Resource Manager** — Mengelola dan mengalokasikan sumber daya sistem (CPU, memory, disk, I/O devices) secara efisien dan adil antar berbagai program yang berjalan.
2. **Extended Machine** — Menyediakan abstraksi dari hardware yang kompleks, memberikan interface yang lebih mudah digunakan untuk programmer dan pengguna.

Catatan

Tanpa sistem operasi, setiap program harus memiliki driver sendiri untuk setiap perangkat keras, yang sangat tidak efisien dan sulit dipelihara.

1.1.2. Fungsi Utama Sistem Operasi

Sistem operasi modern memiliki lima fungsi utama:

Process Management

Sistem operasi bertanggung jawab untuk:

- **Process scheduling** — Menentukan proses mana yang mendapat CPU time
- **Process creation dan termination** — Membuat dan mengakhiri proses
- **Process synchronization** — Mengkoordinasikan multiple processes
- **Inter-process communication (IPC)** — Memfasilitasi komunikasi antar proses

Contoh implementasi:

- **Windows:** Task Manager menampilkan proses yang berjalan dan penggunaan sumber daya
- **Linux:** Perintah `ps`, `top`, dan `htop` untuk memantau proses
- **macOS:** Activity Monitor untuk mengelola proses

Memory Management

Fungsi memory management meliputi:

- **Memory allocation** — Memberikan memori ke proses yang membutuhkan
- **Virtual memory** — Menggunakan disk sebagai extension dari RAM
- **Memory protection** — Mencegah satu proses mengakses memori proses lain
- **Paging dan swapping** — Teknik untuk optimasi penggunaan memori

Contoh: Virtual Memory

Ketika RAM penuh, sistem operasi memindahkan sebagian data ke disk (swap space di Linux, pagefile di Windows) sehingga aplikasi dapat terus berjalan, meskipun dengan kinerja yang lebih lambat.

File Management

Sistem operasi menyediakan:

- **Organisasi file system** — Struktur hirarkis (direktori, subdirektori)

- **Operasi file** — Membuat, membaca, menulis, menghapus file
- **Kontrol akses** — Izin (permissions) untuk keamanan
- **Mounting file system** — Mengintegrasikan perangkat penyimpanan ke sistem

Contoh file systems:

- Windows: NTFS (New Technology File System)
- macOS: APFS (Apple File System)
- Linux: ext4, XFS, Btrfs

I/O Management

Manajemen Input/Output mencakup:

- **Device drivers** — Perangkat lunak untuk berkomunikasi dengan perangkat keras
- **Buffering** — Penyimpanan sementara untuk kelancaran operasi I/O
- **Interrupt handling** — Merespon sinyal dari perangkat keras
- **Spooling** — Antrean untuk perangkat seperti printer

Security dan Protection

Aspek keamanan meliputi:

- **Authentication** — Verifikasi identitas pengguna (password, biometrik)
- **Authorization** — Kontrol akses ke sumber daya berdasarkan izin (permissions)
- **Encryption** — Proteksi data (BitLocker di Windows, FileVault di macOS)
- **Auditing** — Pencatatan aktivitas sistem untuk pemantauan keamanan

1.2. Jenis dan Kategori Sistem Operasi

Sistem operasi dapat dikategorikan berdasarkan berbagai kriteria:

1.2.1. Berdasarkan Platform

Desktop OS Dirancang untuk komputer pribadi dengan antarmuka pengguna yang ramah. Contoh: Windows 10/11, macOS, Ubuntu Desktop, Linux Mint.

Server OS Dioptimasi untuk menjalankan layanan dan melayani banyak klien. Biasanya tanpa GUI atau dengan GUI minimal. Contoh: Windows Server, Ubuntu Server, Red Hat Enterprise Linux, FreeBSD.

Mobile OS Dirancang untuk smartphone dan tablet dengan antarmuka sentuh dan efisiensi daya. Contoh: Android, iOS, HarmonyOS.

Real-Time OS (RTOS) Untuk sistem tertanam (embedded systems) dengan batasan waktu yang ketat. Digunakan di otomotif, kontrol industri, perangkat medis. Contoh: FreeRTOS, VxWorks, QNX.

Embedded OS Untuk perangkat dengan sumber daya terbatas (IoT, router, perangkat pintar). Contoh: Embedded Linux, Windows IoT.

1.2.2. Perbandingan Sistem Operasi Populer

Tabel 1.1 menunjukkan perbandingan sistem operasi modern yang paling banyak digunakan.

Table 1.1.: Perbandingan Sistem Operasi Modern

| OS | Kernel Type | Primary Use | License | Market Share |
|---------|--------------|------------------|-------------|--------------|
| Windows | Hybrid | Desktop/Business | Proprietary | Desktop: 75% |
| macOS | Hybrid (XNU) | Creative/Dev | Proprietary | Desktop: 15% |
| Linux | Monolithic | Server/Dev | Open Source | Server: 60% |
| Android | Linux-based | Mobile | Open Source | Mobile: 70% |
| iOS | Hybrid (XNU) | Mobile | Proprietary | Mobile: 28% |

Tips

Pemilihan sistem operasi bergantung pada kebutuhan:

- **Gaming & Aplikasi Bisnis:** Windows (kompatibilitas terbaik)
- **Pekerjaan Kreatif:** macOS (standar industri untuk video/audio editing)
- **Server & Cloud:** Linux (stabilitas, keamanan, tanpa biaya lisensi)
- **Development:** macOS atau Linux (tools berbasis Unix)

1.3. Arsitektur Kernel Sistem Operasi

Kernel adalah inti dari sistem operasi yang memiliki kontrol penuh atas seluruh sistem. Kernel menyediakan abstraksi antara hardware dan software.

1.3.1. Apa itu Kernel?

Kernel adalah program yang selalu berjalan (resident) di memori dan memiliki akses eksklusif ke perangkat keras. Kernel bertanggung jawab untuk:

- Manajemen proses tingkat rendah (low-level)
- Manajemen memori
- Device drivers
- System calls
- Penanganan interrupt

1.3.2. Kernel Space vs User Space

Sistem operasi modern memisahkan memori menjadi dua area:

Kernel Space (Ring 0) Area memori dengan privilege tertinggi. Hanya kode kernel yang berjalan di sini. Akses penuh ke semua perangkat keras dan memori.

User Space (Ring 3) Area memori untuk aplikasi pengguna. Akses terbatas ke perangkat keras. Harus melalui system calls untuk mengakses layanan kernel.

Catatan

Protection Rings: x86 architecture memiliki 4 privilege levels (Ring 0-3), tapi modern OS hanya menggunakan Ring 0 (kernel) dan Ring 3 (user). Ring 1 dan 2 jarang digunakan.

1.3.3. Jenis Arsitektur Kernel

Monolithic Kernel

Karakteristik:

- Semua layanan (file system, device drivers, manajemen memori) berjalan di kernel space

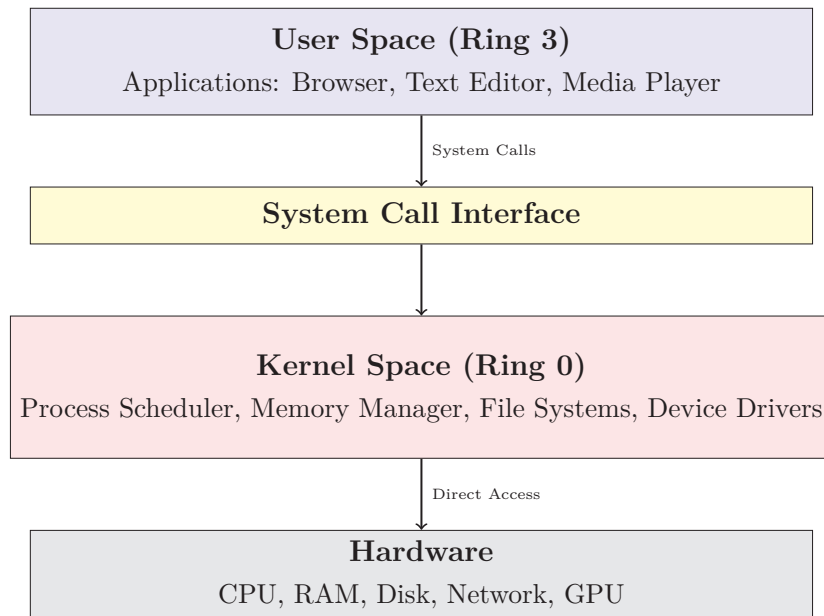


Figure 1.1.: Arsitektur Layer Sistem Operasi

- Binary tunggal yang besar
- Pemanggilan fungsi langsung antar komponen (cepat)

Keuntungan:

- **Kinerja tinggi** — Tidak ada overhead untuk context switching
- **Komunikasi efisien** — Pemanggilan fungsi lebih cepat dari IPC
- **Desain lebih sederhana** — Semua dalam satu address space

Kerugian:

- **Keandalan** — Bug di satu driver dapat menyebabkan crash seluruh sistem
- **Keamanan** — Permukaan serangan lebih luas
- **Pemeliharaan** — Sulit untuk mengisolasi dan debug

Contoh: Linux, traditional Unix, BSD

Microkernel

Karakteristik:

- Kernel minimal dengan hanya layanan esensial (IPC, penjadwalan dasar, memori tingkat rendah)
- Sebagian besar layanan (file systems, drivers, network stack) di user space
- Message passing untuk komunikasi antar komponen

Keuntungan:

- **Keandalan** — Crash layanan tidak mempengaruhi kernel
- **Keamanan** — Permukaan serangan lebih kecil, isolasi lebih baik
- **Modular** — Mudah menambah/menghapus layanan
- **Cocok untuk real-time** — Perilaku yang dapat diprediksi

Kerugian:

- **Overhead kinerja** — Context switches dan IPC lebih lambat
- **Kompleksitas** — Message passing menambah kompleksitas

Contoh: MINIX 3, QNX, seL4, L4 family

Hybrid Kernel

Karakteristik:

- Mengkombinasikan ide dari monolithic dan microkernel
- Beberapa layanan di kernel space untuk kinerja
- Beberapa layanan di user space untuk modularitas
- Keseimbangan antara kecepatan dan keandalan

Contoh:

- **Windows NT kernel** — Digunakan di Windows, Xbox
- **macOS XNU (X is Not Unix)** — Mach microkernel + BSD components

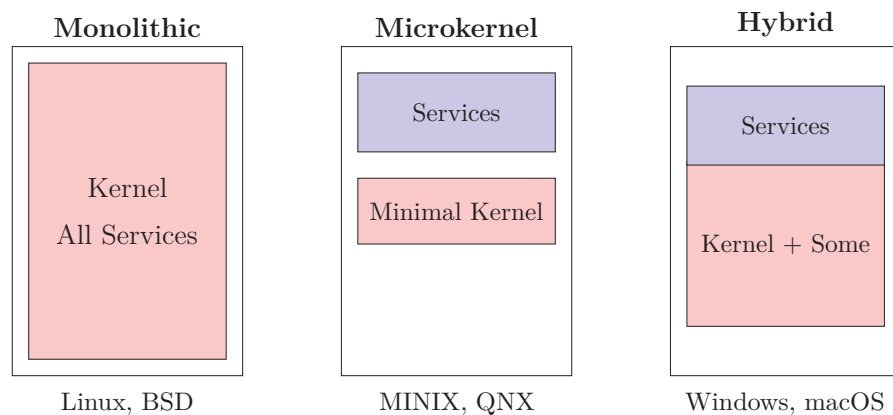


Figure 1.2.: Perbandingan Arsitektur Kernel

1.3.4. Perbandingan Arsitektur Kernel

Table 1.2.: Comparison of Kernel Architectures

| Aspek | Monolithic | Microkernel | Hybrid |
|-----------------|------------|-------------|----------------|
| Performance | Excellent | Good | Very Good |
| Reliability | Moderate | Excellent | Good |
| Security | Moderate | Excellent | Good |
| Complexity | Moderate | High | High |
| Maintainability | Moderate | Good | Moderate |
| Examples | Linux, BSD | QNX, MINIX | Windows, macOS |

Contoh: Tanenbaum-Torvalds Debate (1992)

Perdebatan famous antara Andrew Tanenbaum (creator of MINIX, microkernel) dan Linus Torvalds (creator of Linux, monolithic):

Tanenbaum: "Linux is obsolete" - argued monolithic kernels are outdated

Torvalds: Defended pragmatic monolithic design - performance dan simplicity

Outcome: Both approaches have merits. Linux dominates servers dengan monolithic-modular design, sementara QNX microkernel dominates automotive systems.

1.3.5. System Calls

System call adalah interface antara user space program dan kernel services. Ini adalah controlled entry point ke kernel.

Kategori System Calls:

1. **Process Control:** `fork()`, `exec()`, `exit()`, `wait()`
2. **File Management:** `open()`, `read()`, `write()`, `close()`
3. **Device Management:** `ioctl()`, `read()`, `write()`
4. **Information Maintenance:** `getpid()`, `time()`, `uname()`
5. **Communication:** `pipe()`, `socket()`, `send()`, `receive()`

Contoh: System Call Flow

Ketika program membuka file dengan `fopen()` di C:

1. Application calls library function `fopen()`
2. Library function calls system call `open()`
3. CPU switches to kernel mode (Ring 0)
4. Kernel validates parameters dan permissions
5. Kernel interacts dengan file system dan disk driver
6. Kernel returns file descriptor
7. CPU switches back to user mode (Ring 3)
8. Control returns to application

Proses ini sama conceptually di semua OS, tapi implementation details berbeda.

1.4. Evolusi Sistem Operasi

1.4.1. Timeline Singkat

Perkembangan sistem operasi telah melalui beberapa era penting:

1950s–1960s: Era Awal Batch processing systems, program cards, no interactive computing. Contoh: IBM 701, UNIVAC.

1960s: Time-Sharing CTSS (Compatible Time-Sharing System) di MIT - multiple users sharing computer time. Multics project (ambitious tapi gagal, inspires Unix).

1969: Lahirnya Unix Ken Thompson dan Dennis Ritchie di Bell Labs menciptakan Unix. Simple, portable (ditulis dalam C), filosofi "do one thing well".

1980s: Era PC MS-DOS (1981) untuk IBM PC. Apple Macintosh (1984) dengan GUI. Windows 1.0 (1985) sebagai shell di atas DOS.

1991: Linux Linus Torvalds, mahasiswa dari Finland, merilis Linux kernel 0.01. Free, open source, Unix-compatible.

1993: Windows NT Microsoft merilis Windows NT dengan hybrid kernel architecture untuk enterprise market.

2001: Mac OS X Apple merilis macOS X berbasis Darwin (BSD Unix + Mach microkernel). UNIX certified.

2000s: Modern Era Windows XP unifies consumer dan NT lines. Linux dominates server market. Open source momentum.

2007–2008: Mobile Revolution iOS (2007) dan Android (2008) mendefinisikan ulang computing dengan mobile-first approach.

2010s: Cloud Era Linux dominates cloud computing (AWS, Google Cloud, Azure). Containers (Docker) revolutionize deployment.

2020s: Current Hybrid computing, ARM processors (Apple Silicon), WSL brings Linux to Windows, security-focused development.

1.5. Ekosistem Sistem Operasi Modern

1.5.1. Windows Ecosystem

Platform:

- **Windows 10/11** - Consumer dan Professional editions
- **Windows Server** - Enterprise server OS
- **Windows IoT** - Embedded dan industrial devices

Karakteristik:

- Desktop market dominance (75% market share)
- Excellent backward compatibility
- Large software ecosystem
- Enterprise integration (Active Directory, Group Policy)
- Gaming support terbaik

Teknologi:

- Filesystem: NTFS
- Package management: Microsoft Store, winget, Chocolatey
- Development: Visual Studio, .NET, PowerShell
- Virtualization: Hyper-V

1.5.2. macOS Ecosystem

Karakteristik:

- Premium hardware-software integration (Apple Silicon)
- UNIX certified (POSIX compliant)
- Excellent for creative professionals
- iOS/iPadOS ecosystem integration
- Developer-friendly (Unix tools + graphical environment)

Teknologi:

- Filesystem: APFS (Apple File System)
- Package management: App Store, Homebrew
- Development: Xcode, Swift, Objective-C
- Security: Gatekeeper, FileVault, Secure Enclave

1.5.3. Linux Ecosystem

Distribusi Linux:

Linux bukan single OS, tapi collection of distributions yang share Linux kernel tapi berbeda dalam package management, default software, dan target audience.

Family Utama:

- **Debian Family** - Ubuntu, Debian, Linux Mint
 - Package management: APT (dpkg)
 - Stability-focused, large repository
 - Popular untuk desktop dan server
- **Red Hat Family** - Fedora, RHEL, CentOS Stream, Rocky Linux, AlmaLinux
 - Package management: DNF/YUM (RPM)
 - Enterprise-oriented
 - Cutting-edge technologies
- **Independent** - Arch Linux, Gentoo, Slackware
 - Rolling release atau highly customizable
 - Advanced users

- DIY philosophy

Mengapa Linux untuk Praktik?

1. **Open Source** - Gratis, source code available untuk learning
2. **Educational** - Transparent, excellent documentation
3. **Industry Relevance** - Dominates server, cloud, supercomputers
 - 100% of TOP500 supercomputers run Linux
 - 90% of public cloud workloads
 - Most web servers
4. **Skills Transferable** - Unix concepts apply to macOS, BSDs
5. **Career Value** - High demand for Linux skills in DevOps, SRE, cloud engineering
6. **Flexibility** - Complete control, customization
7. **Legal** - Can install unlimited copies untuk learning

Catatan

Concepts yang dipelajari di Linux applicable ke semua Unix-like systems karena POSIX compliance. Skills seperti shell scripting, file permissions, process management work di Linux, macOS, dan BSDs.

Table 1.3.: Perbandingan Distribusi Linux Populer

| Distribusi | Base | Package Manager | Target User |
|---------------|-------------|-----------------|-------------------------|
| Ubuntu | Debian | APT | Beginners, General |
| Debian | Independent | APT | Stability, Servers |
| Fedora | Red Hat | DNF | Developers, Latest Tech |
| CentOS Stream | Red Hat | DNF | Enterprise, Testing |
| Arch Linux | Independent | Pacman | Advanced, Customization |

1.6. Instalasi Sistem Operasi

1.6.1. Konsep Instalasi Universal

Instalasi sistem operasi meliputi tahapan umum berikut, berlaku untuk semua OS:

1. **Persiapan** - Unduh ISO, buat media bootable, backup data

2. **Boot** - Konfigurasi BIOS/UEFI, boot dari media instalasi
3. **Partisi** - Alokasi ruang disk, buat partisi
4. **Instalasi** - Salin file sistem, instal bootloader
5. **Konfigurasi** - Buat pengguna, konfigurasi jaringan, pengaturan awal

1.6.2. BIOS vs UEFI

Table 1.4.: Comparison: BIOS vs UEFI

| Aspect | BIOS (Legacy) | UEFI (Modern) |
|-----------------|---------------|---------------|
| Introduced | 1970s | 2000s |
| Boot Mode | 16-bit | 32-bit/64-bit |
| Partition Table | MBR | GPT |
| Disk Limit | 2 TB | 9.4 ZB |
| Boot Time | Slower | Faster |
| Secure Boot | No | Yes |

Peringatan

Modern systems (post-2010) umumnya menggunakan UEFI. Secure Boot harus disabled atau configured untuk install beberapa Linux distros.

1.6.3. Metode Deployment

Virtual Machine (Recommended untuk Learning)

Keuntungan:

- Safe experimentation - tidak affect host system
- Snapshots untuk easy recovery
- Multiple OS secara parallel
- No hardware risk

Software:

- VirtualBox (free, cross-platform)
- VMware Workstation Player (free)
- QEMU/KVM (Linux native, free)
- Hyper-V (Windows built-in)

Dual Boot

Install multiple OS pada satu machine. User memilih OS saat boot.

Use case: Native performance, hardware access, production use

Risk: Partition errors, bootloader conflicts, data loss jika tidak careful

Bare Metal

Single OS installation, full hardware control.

Use case: Production servers, dedicated workstations

1.6.4. Praktik: Instalasi Ubuntu Server 22.04 LTS

Prerequisites

- VirtualBox installed di host machine
- Ubuntu Server 22.04 LTS ISO downloaded dari <https://ubuntu.com/download/server>
- Minimum 2GB RAM, 25GB disk space untuk VM

Tips

Ubuntu Server dipilih karena:

- LTS (Long Term Support) - 5 years updates
- Industry standard untuk servers
- Excellent documentation
- Large community support

Step-by-Step Installation

1. Create Virtual Machine

1. Buka VirtualBox, klik
2. Name: **Ubuntu-Server-Lab**
3. Type: Linux, Version: Ubuntu (64-bit)
4. Memory: 2048 MB (2 GB RAM)

5. Hard disk: Create virtual hard disk

- Type: VDI (VirtualBox Disk Image)
- Storage: Dynamically allocated
- Size: 25 GB

2. Configure VM Settings

- Settings → Storage → Controller: IDE → Add optical drive
- Select Ubuntu Server ISO
- Network: NAT atau Bridged (untuk network access)

3. Boot and Install

1. Start VM, akan boot dari ISO
2. Select language: English
3. Keyboard configuration: sesuai keyboard anda
4. Network configuration:
 - DHCP (automatic) - recommended untuk beginning
 - Atau configure static IP (advanced)
5. Proxy: Leave blank (unless needed)
6. Mirror: Default Ubuntu archive mirror

4. Partitioning

Pilihan partitioning:

Option 1: Guided - use entire disk (Recommended untuk beginners)

- Automatic partitioning
- Sets up /, /boot/efi, dan swap
- Simple, fast

Option 2: Custom storage layout (Advanced)

- Manual partition creation
- Dijelaskan detail di section [1.7](#)

5. Profile Setup

```
1 Your name: John Doe
2 Your server's name: ubuntu-server
3 Pick a username: john
```

```
4 Choose a password: [strong password]
5 Confirm your password: [repeat password]
```

Kode 1.1: User account information

6. SSH Setup

- Install OpenSSH server: **YES** (highly recommended)
- Import SSH identity: No (skip untuk now)

Catatan

OpenSSH server memungkinkan remote access ke server via terminal. Essential untuk production servers.

7. Featured Server Snaps

Optional packages:

- Docker - container runtime
- Microk8s - Kubernetes
- (Skip untuk basic installation)

8. Installation Progress

- Installation akan download dan install packages (15-20 minutes)
- Wait hingga "Installation complete!" muncul
- Select "Reboot Now"
- Remove installation media (VirtualBox auto-ejects)

9. First Login

```
1 # Login dengan username yang dibuat
2 ubuntu-server login: john
3 Password: [enter password]
4
5 # Verify system information
6 $ uname -r
7 5.15.0-56-generic
8
9 $ cat /etc/os-release
10 NAME="Ubuntu"
11 VERSION="22.04.1 LTS (Jammy Jellyfish)"
12
13 $ ip a
```

```
14 # Should show network interfaces dengan IP address
```

Kode 1.2: Login dan verify installation

Contoh: Snapshot VM

Setelah instalasi successful, create snapshot di VirtualBox:

1. Power off VM
2. VirtualBox → VM → Snapshots → Take
3. Name: "Fresh Install"

Ini memungkinkan restore ke state bersih kapanpun dibutuhkan.

1.7. Partisi Disk dan Sistem File

1.7.1. Konsep Partisi

Partisi adalah logical division dari physical disk. Setiap partisi diperlakukan sebagai independent storage unit oleh OS.

Mengapa perlu partisi?

- **Organization** - Separate OS, user data, applications
- **Multi-OS** - Install multiple OS pada satu disk
- **Backup/Recovery** - Easier to backup specific partitions
- **Security** - Isolate data dengan different permissions
- **Performance** - Optimize different workload types

1.7.2. Partition Tables

MBR (Master Boot Record)

Legacy scheme - digunakan dengan BIOS:

- Maximum 4 primary partitions
- Extended partition dapat contain multiple logical partitions
- Maximum disk size: 2 TB

- Partition table size: 64 bytes

GPT (GUID Partition Table)

Modern scheme - digunakan dengan UEFI:

- Praktis unlimited partitions (128 di Windows/Linux)
- Maximum disk size: 9.4 ZB (zettabytes)
- Redundant partition tables untuk reliability
- Required untuk UEFI Secure Boot
- Backward compatible dengan BIOS (Protective MBR)

1.7.3. Skema Partisi Linux

Minimum Partitioning

| | | | |
|--------|--------|------|-----------------|
| 1 / | 20 GB | ext4 | Root filesystem |
| 2 swap | 2-4 GB | swap | Virtual memory |

Kode 1.3: Minimal partition scheme

Recommended Partitioning

| | | | |
|-------------|--------|-------|------------------------------|
| 1 /boot/efi | 512 MB | FAT32 | EFI System Partition (UEFI) |
| 2 /boot | 1 GB | ext4 | Kernel dan bootloader |
| 3 / | 20 GB | ext4 | Root filesystem |
| 4 /home | 50+ GB | ext4 | User data |
| 5 swap | 4 GB | swap | Virtual memory |

Kode 1.4: Recommended partition scheme

Penjelasan Mount Points:

/ (root) Sistem files, installed programs, configurations

/boot Kernel, initramfs, GRUB bootloader

/boot/efi EFI bootloader (UEFI systems only)

/home User directories dan personal data

swap Virtual memory, hibernation support

Tips**Why separate /home?**

Benefits:

- Reinstall OS tanpa lose user data
- Easier backup (backup only /home)
- Upgrade distro dengan preserve settings
- Different filesystem atau encryption untuk data

1.7.4. Swap Space

Swap adalah disk space used sebagai extension dari RAM (virtual memory).

When is swap used?

- Saat RAM penuh, inactive pages moved to swap
- Hibernation (suspend-to-disk) - save RAM contents to swap

Swap Sizing Guidelines:

Table 1.5.: Rekomendasi Ukuran Swap

| RAM Size | Swap Size |
|--------------------|-----------------|
| ≤ 2 GB | 2x RAM |
| 2–8 GB | Same as RAM |
| > 8 GB | 4–8 GB fixed |
| Hibernation needed | RAM size + 2 GB |

Peringatan

Swap bukanlah pengganti RAM yang tidak mencukupi. Menambahkan RAM jauh lebih efektif daripada membuat swap yang besar.

1.7.5. Filesystem Types

Cross-Platform Comparison**Linux Filesystems**

ext4 (Fourth Extended Filesystem)

Table 1.6.: Perbandingan Filesystem

| Filesystem | Platform | Max File | Journaling | Use Case |
|------------|-----------|----------|------------|-------------------|
| NTFS | Windows | 16 TB | Yes | Windows system |
| APFS | macOS | 8 EB | CoW | Modern macOS |
| ext4 | Linux | 16 TB | Yes | Linux default |
| XFS | Linux | 8 EB | Yes | Large files |
| Btrfs | Linux | 16 EB | CoW | Advanced features |
| exFAT | Universal | 16 EB | No | USB drives |

- Default untuk most Linux distributions
- Mature, stable, well-tested
- Journaling untuk crash recovery
- Good performance untuk general use

XFS

- High-performance filesystem
- Excellent untuk large files dan parallel I/O
- Default di Red Hat Enterprise Linux
- Cannot shrink (only grow)

Btrfs (B-tree filesystem)

- Modern copy-on-write filesystem
- Built-in snapshots dan compression
- RAID support
- Still maturing (production-ready di openSUSE)

1.7.6. Praktik: Manual Partitioning

Untuk advanced users, manual partitioning allows full control.

Durante Ubuntu installation, select "Custom storage layout":

| | Device | Size | Type | Mount Point | Filesystem |
|---|-----------|--------|------|-------------|------------|
| 1 | /dev/sda1 | 512 MB | EFI | /boot/efi | FAT32 |
| 3 | /dev/sda2 | 1 GB | ext4 | /boot | ext4 |
| 4 | /dev/sda3 | 20 GB | ext4 | / | ext4 |
| 5 | /dev/sda4 | 50 GB | ext4 | /home | ext4 |
| 6 | /dev/sda5 | 4 GB | swap | [swap] | swap |

Kode 1.5: Example manual partition layout

Post-Installation Filesystem Commands

```
1 # Check disk usage
2 df -h
3
4 # Check disk partitions
5 lsblk
6 sudo fdisk -l
7
8 # Check filesystem type
9 lsblk -f
10 blkid
11
12 # Mount additional filesystem manually
13 sudo mkdir /mnt/data
14 sudo mount /dev/sdb1 /mnt/data
15
16 # Permanent mount - edit /etc/fstab
17 cat /etc/fstab
18 # Format: <device> <mount point> <type> <options> <dump> <
    pass>
19
20 # UUID-based mount (preferred)
21 UUID=xxx-xxx-xxx /data ext4 defaults 0 2
```

Kode 1.6: Filesystem verification dan management

Catatan**UUID vs Device Names**

Use UUID instead of `/dev/sdX` di `fstab` karena:

- Device names can change (`sda` → `sdb` jika add disk)
- UUID adalah persistent identifier
- More reliable untuk production systems

Get UUID: `blkid` atau `lsblk -f`

1.8. Konfigurasi Post-Installation

Setelah instalasi completed, several konfigurasi penting harus dilakukan.

1.8.1. System Updates

Mengapa penting?

- Security patches untuk vulnerabilities
- Bug fixes dan stability improvements
- New features dan driver updates

```
1 # Update package lists from repositories
2 sudo apt update
3
4 # Upgrade installed packages to latest versions
5 sudo apt upgrade
6
7 # Full system upgrade (handles dependencies)
8 sudo apt full-upgrade
9
10 # Remove packages yang tidak dibutuhkan
11 sudo apt autoremove
12
13 # Clean package cache
14 sudo apt clean
```

Kode 1.7: Update sistem Ubuntu/Debian

1.8.2. Software Installation Essentials

```
1 # Development tools
2 sudo apt install -y build-essential
3
4 # Version control
5 sudo apt install -y git
6
7 # Network utilities
8 sudo apt install -y curl wget net-tools
9
```

```
10 # Text editors
11 sudo apt install -y vim nano
12
13 # System monitoring
14 sudo apt install -y htop iotop
15
16 # Complete essentials in one command
17 sudo apt install -y \
18     build-essential \
19     git \
20     curl \
21     wget \
22     vim \
23     htop \
24     tree \
25     net-tools \
26     openssh-server
```

Kode 1.8: Install essential development tools

1.8.3. Verifikasi Koneksi Jaringan

Pastikan sistem dapat terhubung ke internet.

```
1 # Check IP address
2 ip addr show
3
4 # Test internet connectivity
5 ping -c 4 google.com
6
7 # Check DNS
8 cat /etc/resolv.conf
```

Kode 1.9: Check network status

Tips

Jika menggunakan VirtualBox, pastikan network adapter di-set ke "NAT" atau "Bridged Adapter" untuk koneksi internet.

1.8.4. Akses SSH (Opsional)

Jika ingin remote access ke server melalui SSH (jika sudah terinstall saat instalasi):

```
1 # Check if SSH installed dan running
2 sudo systemctl status ssh
3
4 # Get IP address untuk remote access
5 ip addr show
```

Kode 1.10: Check SSH status

Contoh: Remote Login via SSH

Dari komputer lain pada jaringan yang sama:

```
1 ssh username@192.168.1.100
```

Ganti alamat IP dengan alamat IP server.

1.8.5. Menampilkan Informasi Sistem

Install Neofetch

Neofetch adalah tool populer untuk menampilkan informasi sistem dengan visual yang menarik.

```
1 # Install neofetch
2 sudo apt install -y neofetch
3
4 # Run neofetch
5 neofetch
```

Kode 1.11: Install neofetch

Neofetch akan menampilkan:

- Logo distro Linux
- Informasi OS (distro, versi, kernel)
- CPU dan memory usage
- Disk space
- Uptime sistem
- Dan informasi lainnya dalam format yang mudah dibaca

Tips

Add `neofetch` ke file `~/.bashrc` agar otomatis tampil setiap login:

```
1 echo "neofetch" >> ~/.bashrc
```

Perintah Informasi Sistem Dasar

```
1 # Informasi OS
2 cat /etc/os-release
3 hostnamectl
4
5 # Versi kernel
6 uname -r
7
8 # Informasi CPU
9 lscpu
10
11 # Informasi memory
12 free -h
13
14 # Informasi disk
15 df -h
16 lsblk
17
18 # System uptime
19 uptime
20
21 # Who is logged in
22 who
```

Kode 1.12: Perintah informasi sistem dasar

Contoh: Praktik: Explore System Info

Coba jalankan perintah berikut dan perhatikan output:

1. `neofetch` – Tampilan lengkap dengan visual
2. `free -h` – Check RAM usage
3. `df -h` – Check disk space
4. `lscpu` – Detail CPU
5. `uptime` – Berapa lama sistem running

1.9. Rangkuman

Dalam bab ini, kita telah mempelajari konsep fundamental sistem operasi dan praktik instalasi:

1.9.1. Konsep Sistem Operasi

- **Definisi:** OS adalah resource manager dan extended machine
- **Fungsi Utama:** Process, Memory, File, I/O Management, dan Security
- **Jenis OS:** Desktop, Server, Mobile, Real-Time, Embedded
- **Kernel Architectures:**
 - Monolithic (Linux) - Performance
 - Microkernel (QNX) - Reliability
 - Hybrid (Windows, macOS) - Balance
- **System Calls:** Interface between user space dan kernel services

1.9.2. Perbandingan Sistem Operasi

- **Windows:** Desktop dominance, gaming, business software
- **macOS:** Creative professionals, hardware integration
- **Linux:** Server dominance, open source, flexibility
- **Android/iOS:** Mobile platforms

1.9.3. Practical Skills

- Installation process (VM setup, Ubuntu Server)
- Partitioning concepts (MBR vs GPT, partition schemes)
- Filesystem understanding (ext4, swap, mount points)
- Post-installation: system updates dan basic configuration
- System information gathering dengan neofetch dan basic commands

1.9.4. Key Takeaways

1. OS concepts are universal - applicable across platforms
2. Linux excellent untuk learning: open, free, industry-relevant
3. Virtual machines provide safe learning environment
4. Proper partitioning important untuk flexibility dan recovery
5. System updates should be performed regularly
6. Basic system monitoring tools membantu understanding system state

1.10. Latihan

1.10.1. Latihan Konseptual

Latihan 1.1

Jelaskan 5 fungsi utama sistem operasi dengan contoh konkret dari minimal 2 OS berbeda (Windows, macOS, atau Linux).

Latihan 1.2

Kapan sebaiknya menggunakan Windows vs Linux vs macOS? Analisis berdasarkan use case: gaming, development, server, creative work, dan enterprise.

1.10.2. Latihan Praktikal

Latihan 1.3

Install Ubuntu Server 22.04 LTS di VirtualBox dengan langkah berikut:

1. Download Ubuntu Server ISO dari website resmi
2. Create VM baru di VirtualBox (RAM: 2GB, Disk: 25GB)
3. Install dengan automatic partitioning (guided)
4. Buat user account dengan password yang kuat
5. Reboot dan login ke sistem
6. Dokumentasikan proses instalasi dengan screenshot key steps

Latihan 1.4

Setelah instalasi Ubuntu Server, lakukan tasks berikut:

1. Update package list: `sudo apt update`
2. Upgrade packages: `sudo apt upgrade`
3. Install neofetch: `sudo apt install neofetch`
4. Jalankan neofetch dan screenshot hasilnya
5. Check disk usage dengan `df -h`
6. Check memory dengan `free -h`
7. Dokumentasikan output dari setiap command

Latihan 1.5

Eksplorasi sistem yang baru diinstall:

1. Tampilkan informasi OS: `cat /etc/os-release`
2. Tampilkan versi kernel: `uname -r`
3. List partisi: `lsblk`
4. Check network connectivity: `ping -c 4 google.com`
5. Install dan jalankan `htop` untuk melihat resource usage
6. Buat laporan singkat tentang konfigurasi sistem Anda

1.10.3. Latihan Refleksi**Latihan 1.6**

Ceritakan pengalaman Anda dengan sistem operasi:

1. Sistem operasi apa yang Anda gunakan sehari-hari? (Windows, macOS, Linux, atau lainnya)
2. Berapa lama Anda menggunakan sistem operasi tersebut?
3. Apa yang Anda sukai dari sistem operasi tersebut?
4. Apa tantangan atau masalah yang pernah Anda hadapi?
5. Apakah Anda pernah menggunakan sistem operasi lain? Bandingkan pengalaman Anda.
6. Setelah mempelajari bab ini, apakah ada sistem operasi lain yang ingin Anda coba? Mengapa?

Tulis refleksi Anda dalam 300-500 kata disertai dengan dokumentasi.

1.11. Referensi

1.11.1. Buku Rekomendasi

- **Operating System Concepts (10th Ed)** - Silberschatz, Galvin, Gagne
Classic textbook untuk OS theory
- **Modern Operating Systems (4th Ed)** - Andrew Tanenbaum
Comprehensive OS concepts dengan implementation details
- **How Linux Works (3rd Ed)** - Brian Ward
Practical guide untuk understanding Linux internals
- **The Linux Command Line (2nd Ed)** - William Shotts
Excellent resource untuk command-line mastery
- **UNIX and Linux System Administration Handbook (5th Ed)** - Nemeth et al.
Industry standard untuk system administration

1.11.2. Online Resources

Documentation:

- Linux Foundation: <https://www.linuxfoundation.org>
- Ubuntu Documentation: <https://ubuntu.com/server/docs>
- Arch Wiki: <https://wiki.archlinux.org> (excellent cross-distro reference)
- Red Hat Documentation: <https://access.redhat.com/documentation>

Interactive Learning:

- Linux Journey: <https://linuxjourney.com>
- OverTheWire Bandit: <https://overthewire.org/wargames/bandit/>
- edX: Introduction to Linux (LFS101x)

Official Sites:

- The Linux Kernel: <https://www.kernel.org>
- Microsoft Docs: <https://docs.microsoft.com>
- Apple Developer: <https://developer.apple.com/documentation>