

# 1 Manajemen Perangkat Keras & Perintah Dasar Sistem Operasi

Bab ini membahas cara Linux mendeteksi dan mengelola perangkat keras, modul kernel, driver perangkat, serta perintah-perintah dasar terminal Linux yang esensial untuk praktikum Ubuntu Server 22.04. Fokus utama bab ini adalah keterampilan praktik: mengidentifikasi perangkat, mengecek driver/modul kernel, memahami perangkat di `/dev`, melakukan manipulasi teks dengan `grep/sed/awk`, mengelola proses, dan memantau kondisi sistem.

## 1.1 Deteksi Perangkat Keras di Linux

### Bagaimana Linux Dapat Mendeteksi Hardware?

Hardware merupakan bagian yang penting dalam penggunaan sistem operasi. Memastikan hardware dapat terbaca pada sistem operasi merupakan hal esensial. Dalam prosesnya, kita juga sering menambahkan hardware baru pada komputer kita, baik pada personal computer (PC) ataupun pada server. Oleh karena itu penting untuk mengetahui apakah hardware yang kita pasang sudah terdeteksi oleh sistem operasi. Selain untuk memastikan hardware terbaca dengan benar, kita juga dapat mengetahui pada bagian I/O mana hardware itu terpasang. Kita juga perlu memastikan driver yang terpasang telah sesuai.

Pada sistem operasi Linux, informasi perangkat keras dapat terlihat melalui beberapa tahapan proses, yaitu **kernel** melakukan enumerasi/scan perangkat (contoh: PCI/USB/storage), menampilkan informasi melalui **sysfs** (`/sys`), kemudian **udev** membuat node perangkat di `/dev`. Setelah menyelesaikan bab ini, diharapkan,

- Anda dapat **melihat dafatar perangkat** (misal: NIC, disk, USB) dengan menggunakan perintah seperti `lspci/lsusb/lsblk`.
- Anda dapat **melihat driver/modul** yang meng-handle perangkat (misal: nama modul kernel) dari output `lspci -k` atau `lsmod`.

## 1.1 Deteksi Perangkat Keras di Linux

- Anda dapat **melihat jejak deteksi** pada log kernel dengan `dmesg` atau `journalctl -k`.

## Fungsi Perintah yang Digunakan

Berikut perintah-perintah yang umum digunakan untuk deteksi hardware:

- `lscpu`: menampilkan arsitektur CPU, jumlah core/thread, fitur virtualisasi, dsb.
- `free -h`: ringkasan penggunaan RAM dan swap dalam format mudah dibaca.
- `lspci`: daftar perangkat PCI (misal GPU, NIC, controller).
- `lsusb`: daftar perangkat USB.
- `lsblk`: daftar block device (disk/partisi) beserta mount point dan filesystem.
- `blkid`: menampilkan UUID/TYPE filesystem dari sebuah device.
- `dmesg`: menampilkan pesan kernel (termasuk deteksi perangkat saat boot).
- `journalctl -k`: menampilkan log kernel berbasis systemd journal.

### Tips

Beberapa perintah mungkin perlu paket tambahan. Pada Ubuntu Server 22.04, Anda dapat memasang utilitas penting ini:

```
1 sudo apt update  
2 sudo apt install -y pciutils usbutils lshw dmidecode
```

## Praktikum 2.1 — Identifikasi CPU dan Memori

**Tujuan:** memahami spesifikasi CPU dan kondisi memori pada server/VM.

### Langkah-langkah:

1. Tampilkan informasi CPU:

```
1 lscpu
```

Kode 1.1: Melihat informasi CPU

2. Tampilkan ringkasan memori:

```
1 free -h
```

Kode 1.2: Melihat penggunaan memori

3. (Opsional) cek informasi hardware dari DMI/BIOS (butuh sudo):

```
1 sudo dmidecode -t system
```

Kode 1.3: Informasi DMI (opsional)

### Latihan 2.1

Catat: (1) jumlah CPU(s), core/thread, (2) total RAM, (3) total swap. Jelaskan perbedaan RAM vs swap dalam 2–3 kalimat.

## Praktikum 2.2 — Identifikasi Perangkat PCI/USB dan Driver

**Tujuan:** mengenali perangkat PCI/USB dan melihat driver/modul yang dipakai.

### Catatan

Jika Anda menggunakan VirtualBox, Anda perlu melakukan proses **mounting** device USB dari komputer host agar dapat dibaca oleh sistem operasi pada VirtualBox.

### Langkah-langkah:

1. Lihat daftar perangkat PCI:

```
1 lspci
```

Kode 1.4: Daftar perangkat PCI (ringkas)

2. Lihat perangkat PCI beserta driver kernel yang digunakan:

```
1 lspci -nnk
```

Kode 1.5: Melihat driver perangkat PCI

3. Fokus pada NIC (Ethernet) untuk mencari modul driver:

```
1 lspci -nnk | grep -A3 -i ethernet
```

Kode 1.6: Mencari info NIC dan drivernya

4. Lihat perangkat USB:

```
1 lsusb
```

Kode 1.7: Daftar perangkat USB

## 1.1 Deteksi Perangkat Keras di Linux

5. Lihat topologi USB (tree):

```
1 lsusb -t
```

Kode 1.8: Topologi perangkat USB

### Latihan 2.2

Temukan 1 perangkat PCI (misal NIC) dan tuliskan: *Vendor:Device ID* (angka heksadesimal), nama driver/modul kernel, dan deskripsi singkat fungsinya.

## Praktikum 2.3 — Identifikasi Storage dan Filesystem

**Tujuan:** memahami disk/partisi dan filesystem yang terpasang.

**Langkah-langkah:**

1. Lihat daftar disk/partisi:

```
1 lsblk -f
```

Kode 1.9: Melihat block device dan filesystem

2. Tampilkan UUID dan tipe filesystem:

```
1 sudo blkid
```

Kode 1.10: Melihat UUID filesystem

3. Lihat mount point untuk root filesystem:

```
1 findmnt /
```

Kode 1.11: Melihat device untuk root filesystem

### Contoh: Interpretasi Cepat

Jika `findmnt /` menunjukkan root berada di `/dev/sda2`, maka pada `lsblk -f` Anda bisa lihat tipe filesystem (misal ext4) dan UUID partisi tersebut. UUID penting untuk konfigurasi mounting di `/etc/fstab`.

## 1.2 Modul Kernel dan Driver Perangkat

### Konsep Inti: Modul Kernel

**Modul kernel** adalah komponen kernel yang dapat dimuat (*load*) atau dilepas (*unload*) saat sistem berjalan. Banyak driver perangkat (misal NIC, filesystem tertentu) tersedia sebagai modul agar kernel fleksibel dan ukuran kernel inti tetap efisien.

Di Ubuntu, modul biasanya tersimpan pada,

```
1 /lib/modules/$(uname -r)/
```

Kode 1.12: Lokasi modul kernel

### Fungsi Perintah yang Digunakan

- `uname -r`: melihat versi kernel yang sedang berjalan.
- `lsmod`: daftar modul kernel yang sedang aktif.
- `modinfo <modul>`: detail modul (deskripsi, parameter, file .ko, dependensi).
- `modprobe <modul>`: memuat modul beserta dependensinya.
- `modprobe -r <modul>`: melepas modul (jika aman dan tidak sedang digunakan).

#### Peringatan

Jangan mencoba melepas modul yang terkait storage, filesystem, atau network yang sedang dipakai (contoh: modul driver disk, `ext4`, modul NIC), karena bisa membuat sistem hang atau koneksi SSH terputus.

### Praktikum 2.4 — Melihat Modul Aktif dan Informasinya

**Tujuan:** mengenal modul aktif dan keterkaitannya dengan perangkat.

**Langkah-langkah:**

1. Cek versi kernel:

```
1 uname -r
```

Kode 1.13: Cek versi kernel

2. Tampilkan daftar modul aktif:

```
1 lsmod | head
```

Kode 1.14: Daftar modul aktif

3. Pilih salah satu modul (contoh aman: `loop`) dan lihat detailnya:

```
1 modinfo loop
```

Kode 1.15: Detail modul dengan modinfo

4. Muat modul (jika belum aktif), lalu verifikasi:

```
1 sudo modprobe loop  
2 lsmod | grep -i loop
```

Kode 1.16: Load modul dan verifikasi

5. (Opsional) lihat pesan kernel terbaru:

```
1 dmesg -T | tail -n 20
```

Kode 1.17: Cek log kernel terbaru

## Praktikum 2.5 — Konfigurasi Auto-load dan Blacklist

**Tujuan:** memahami cara membuat modul otomatis dimuat atau diblokir.

**Fungsi konfigurasi:**

- `/etc/modules-load.d/*.conf`: daftar modul yang di-load saat boot.
- `/etc/modprobe.d/*.conf`: aturan modprobe, termasuk blacklist modul.

**Langkah demo (gunakan modul aman, contoh `loop`):**

1. Buat file auto-load:

```
1 echo "loop" | sudo tee /etc/modules-load.d/loop.conf
```

Kode 1.18: Menambahkan modul untuk auto-load (demo)

2. Simulasikan verifikasi (tanpa reboot) dengan memastikan modul sudah aktif:

```
1 lsmod | grep -i loop
```

Kode 1.19: Verifikasi modul aktif

3. (Opsional, konsep) blacklist modul:

```
1 # echo "blacklist loop" | sudo tee /etc/modprobe.d/
      blacklist-loop.conf
```

Kode 1.20: Contoh blacklist modul (jangan diterapkan sembarangan)

### Catatan

Auto-load dan blacklist sering dipakai untuk: menghindari konflik driver, menonaktifkan modul yang tidak diperlukan, atau menerapkan kebijakan keamanan.

## 1.3 Sistem File dan /dev di Linux

### Konsep: /dev sebagai Device Node

Folder /dev berisi **device node** (file khusus) yang merepresentasikan perangkat. Ada dua kategori utama,

- **Block device**: akses berbasis blok (contoh: disk /dev/sda, partisi /dev/sda1).
- **Character device**: akses berbasis stream karakter (contoh: terminal /dev/tty, device serial).

### Fungsi Perintah yang Digunakan

- `ls -l /dev/...`: melihat tipe file dan major/minor number.
- `lsblk`: melihat mapping disk/partisi dan mount point.
- `udevadm info`: melihat atribut udev dari sebuah device node.
- `udevadm monitor`: memonitor event saat perangkat muncul/hilang.

### Praktikum 2.6 — Mengenali Block vs Character Device

**Tujuan:** membedakan perangkat disk vs terminal.

**Langkah-langkah:**

1. Lihat detail salah satu disk (sesuaikan dengan perangkat Anda, misal sda):

## 1.4 Perintah Dasar Terminal Linux

```
1 ls -l /dev/sda
```

Kode 1.21: Melihat detail device node disk

2. Lihat detail device terminal:

```
1 ls -l /dev/tty
```

Kode 1.22: Melihat detail device node terminal

3. Lihat disk dan partisi untuk mengaitkan dengan /dev:

```
1 lsblk
```

Kode 1.23: Mapping disk/partisi

### Latihan 2.3

Dari output `ls -l`, jelaskan perbedaan penanda file untuk block device dan character device. (Hint: karakter pertama pada permission string)

## Praktikum 2.7 — Melihat Informasi udev

**Tujuan:** melihat metadata yang dipakai udev untuk membuat device node.

**Langkah-langkah:**

1. Cek atribut udev untuk disk:

```
1 udevadm info --query=all --name=/dev/sda | head -n 30
```

Kode 1.24: Melihat atribut udev untuk disk

2. (Opsional) monitor event udev (jalankan, lalu colok/lepas USB pada mesin fisik):

```
1 sudo udevadm monitor
```

Kode 1.25: Monitor event udev (opsional)

## 1.4 Perintah Dasar Terminal Linux

### Konsep: Terminal sebagai Antarmuka Administrasi

Pada sistem operasi Linux, khususnya yang digunakan pada server, termasuk Ubuntu Server, terminal adalah antarmuka utama administrasi. Penguasaan terhadap peng-

gunaan terminal sangat krusial dikarenakan terminal merupakan antarmuka utama untuk dapat mengakses sistem operasi. Kemampuan dasar terminal yang **wajib** dikuasai adalah, navigasi direktori, operasi file, membaca file/log, memahami permission, dan menggunakan bantuan perintah.

## Fungsi Perintah yang Digunakan

- Navigasi: `pwd`, `ls`, `cd`
- Operasi file/direktori: `mkdir`, `touch`, `cp`, `mv`, `rm`
- Membaca file: `cat`, `less`, `head`, `tail`
- Bantuan: `man`, `-help`, `apropos`
- Redirection & pipe: `>`, `>>`, `2>`, `|`, `tee`

### Peringatan

Perintah `rm -rf` dapat menghapus direktori beserta isinya tanpa konfirmasi. Untuk praktikum, biasakan memakai `rm -ri` atau pastikan path yang Anda hapus benar.

## Praktikum 2.8 — Membuat Workspace Praktikum

**Tujuan:** membuat area kerja aman untuk semua latihan bab ini.

### Langkah-langkah:

1. Buat direktori praktikum dan masuk ke dalamnya:

```

1 mkdir -p ~/praktikum-os/week02
2 cd ~/praktikum-os/week02
3 pwd

```

Kode 1.26: Membuat workspace praktikum

2. Buat beberapa file contoh:

```

1 touch notes.txt data.log config.txt
2 ls -lah

```

Kode 1.27: Membuat file contoh

3. Isi file log contoh (simulasi):

```

1 echo "INFO: service started" >> data.log

```

## 1.5 Manipulasi Teks

```
2 echo "WARN: disk usage high" >> data.log
3 echo "ERROR: failed to connect" >> data.log
4 cat data.log
```

Kode 1.28: Mengisi file log contoh

4. Baca file dengan less:

```
1 less data.log
```

Kode 1.29: Membaca file dengan less

# 1.5 Manipulasi Teks

## Konsep: Data Administrasi = Teks

Banyak aktivitas administrasi Linux menghasilkan teks, seperti, log, konfigurasi, output perintah. Tool seperti `grep`, `sed`, dan `awk` adalah tools untuk menyaring, mengubah, dan meringkas teks secara cepat.

### 1.5.1 Perintah grep

#### Fungsi grep

`grep` digunakan untuk mencari baris yang cocok dengan pola (pattern) di file atau stream. Sangat berguna untuk mencari error pada log atau mencari konfigurasi tertentu.

#### Praktikum 2.9 — Pencarian Pola dengan grep

##### Langkah-langkah:

1. Cari baris yang mengandung `ERROR` pada `data.log`:

```
1 grep "ERROR" data.log
```

Kode 1.30: grep sederhana

2. Cari tanpa memperhatikan huruf besar/kecil:

```
1 grep -i "error" data.log
```

Kode 1.31: grep case-insensitive

3. Tampilkan nomor baris:

```
1 grep -n "WARN" data.log
```

Kode 1.32: grep dengan nomor baris

4. Tampilkan baris yang **tidak** cocok (invert match):

```
1 grep -v "INFO" data.log
```

Kode 1.33: grep invert match

#### Latihan 2.4

Gunakan grep untuk menampilkan hanya baris yang mengandung INFO atau WARN dari data.log. (Hint: gunakan grep -E dengan pola alternatif)

## 1.5.2 Perintah sed

### Fungsi sed

**sed** adalah stream editor untuk melakukan transformasi teks, terutama substitusi (cari-ganti). Umum dipakai untuk memperbarui konfigurasi atau melakukan normalisasi output.

#### Praktikum 2.10 — Substitusi dengan sed (Aman di File Latihan)

##### Langkah-langkah:

1. Siapkan file konfigurasi latihan:

```
1 cat > config.txt << 'EOF'
2 PORT=8080
3 MODE=dev
4 SERVICE_NAME=myserver
5 EOF
6 cat config.txt
```

Kode 1.34: Membuat file config latihan

2. Ganti **dev** menjadi **prod** (tanpa mengubah file asli):

```
1 sed 's/MODE=dev/MODE=prod/' config.txt
```

Kode 1.35: sed substitusi tanpa in-place

## 1.5 Manipulasi Teks

3. Terapkan perubahan langsung ke file (-i):

```
1 sed -i 's/MODE=dev/MODE=prod/' config.txt  
2 cat config.txt
```

Kode 1.36: sed in-place

4. Ganti semua kemunculan kata (g untuk global), contoh ubah `myserver` menjadi `node`:

```
1 sed -i 's/myserver/node/g' config.txt  
2 cat config.txt
```

Kode 1.37: sed global replacement

### Peringatan

Gunakan `sed -i` dengan hati-hati jika mengedit file sistem. Untuk administrasi nyata, praktik aman adalah membuat backup: `sed -i.bak 's/old/new/g' file.conf`

## 1.5.3 Perintah awk

### Fungsi awk

`awk` digunakan untuk memproses teks berbasis kolom/field. Sangat efektif untuk meringkas output perintah seperti `df`, `ps`, atau file berformat kolom.

### Praktikum 2.11 — Ekstraksi Kolom dengan awk

#### Langkah-langkah:

1. Lihat output `df -h`:

```
1 df -h
```

Kode 1.38: Output `df -h`

2. Ambil kolom filesystem dan persentase pemakaian:

```
1 df -h | awk 'NR==1 {print $1, $5, $6} NR>1 {print $1,  
$5, $6}'
```

Kode 1.39: awk print kolom tertentu

3. Filter hanya yang pemakaian disk di atas 80%:

```
1 df -h | awk 'NR==1 || ($5+0) > 80 {print $1, $5, $6}'
```

Kode 1.40: awk filter berdasarkan kondisi

## 1.6 Manajemen Proses

### Konsep: Proses, PID, dan Sinyal

Setiap program yang berjalan direpresentasikan sebagai **proses** dengan **PID** (Process ID). Linux menggunakan **sinyal** untuk mengontrol proses, misalnya:

- **SIGTERM** (15): meminta proses berhenti secara baik (graceful).
- **SIGKILL** (9): memaksa proses berhenti segera (force).

### 1.6.1 Perintah ps

#### Fungsi ps

**ps** menampilkan snapshot proses pada saat perintah dijalankan. Berguna untuk mencari PID, melihat pemakaian CPU/MEM, dan memfilter proses tertentu.

#### Praktikum 2.12 — Melihat Proses dengan ps

##### Langkah-langkah:

1. Tampilkan semua proses (format BSD):

```
1 ps aux | head
```

Kode 1.41: ps aux

2. Cari proses tertentu (misal **sshd**):

```
1 ps aux | grep -i sshd
```

Kode 1.42: Filter proses dengan grep

## 1.6.2 Perintah top

### Fungsi top

`top` menampilkan pemakaian resource secara real-time: CPU, memori, serta proses paling berat. Ini alat cepat untuk diagnosis “server terasa lambat”.

#### Praktikum 2.13 — Monitoring Real-time dengan top

Langkah-langkah:

1. Jalankan `top`:

```
1 top
```

Kode 1.43: Menjalankan top

2. Amati nilai load average, pemakaian CPU, dan proses teratas. Tekan `q` untuk keluar.

### Tips

Altenatif lainnya, Anda dapat menggunakan, `htop` atau `btop`. Kedua tools tersebut lebih nyaman digunakan secara visual.

```
1 sudo apt install -y htop btop
```

## 1.6.3 Perintah kill

### Fungsi kill

`kill` mengirim sinyal ke proses berdasarkan PID. Sinyal yang dikirimkan merupakan sinyal untuk menghentikan proses. Sinyal default adalah `SIGTERM`.

#### Praktikum 2.14 — Menghentikan Proses dengan kill

Langkah-langkah:

1. Jalankan proses dummy di background:

```
1 sleep 300 &
```

Kode 1.44: Membuat proses dummy

2. Cari PID proses `sleep`:

```
1 ps aux | grep -E "sleep 300" | grep -v grep
```

Kode 1.45: Mencari PID sleep

3. Hentikan dengan SIGTERM:

```
1 kill <PID_ANDA>
```

Kode 1.46: Mengirim SIGTERM

4. Verifikasi proses berhenti:

```
1 ps aux | grep -E "sleep 300" | grep -v grep
```

Kode 1.47: Verifikasi proses sudah berhenti

5. (Opsional) Jika proses sulit untuk dihentikan dan Anda membutukan untuk menghentikan proses tersebut, gunakan SIGKILL:

```
1 kill -9 <PID_ANDA>
```

Kode 1.48: Mengirim SIGKILL (opsional)

## 1.7 Pemantauan Sistem

### Konsep: “Health Check” Server

Pemantauan sistem berfokus pada tiga pertanyaan praktis:

1. Apakah CPU/memori sedang terbebani?
2. Apakah disk hampir penuh?
3. Apakah service penting berjalan normal dan log tidak menunjukkan error?

### Fungsi Perintah yang Digunakan

- Disk: df -h (kapasitas), du -sh (ukuran direktori).
- Load & memori: uptime, free -h.
- Network (ringkas): ip a, ip r, ss -tulpn.
- Service: systemctl status, systemctl -failed.
- Log: journalctl -xe, journalctl -u <service>.

## Praktikum 2.15 — Cek Disk, Load, dan Service

Langkah-langkah:

1. Cek penggunaan disk:

```
1 df -h
```

Kode 1.49: Cek kapasitas disk

2. Cari direktori yang besar (contoh pada /var):

```
1 sudo du -sh /var/* 2>/dev/null | sort -h | tail -n 10
```

Kode 1.50: Cek ukuran direktori (contoh /var)

3. Cek load dan uptime:

```
1 uptime
```

Kode 1.51: Cek load average

4. Cek service yang gagal:

```
1 systemctl --failed
```

Kode 1.52: Service yang gagal

5. Ambil log error terbaru (jika ada indikasi masalah):

```
1 journalctl -xe | tail -n 50
```

Kode 1.53: Log error terbaru

## Praktikum 2.16 — Monitoring Port dan Koneksi (Network Basics)

**Tujuan:** melihat interface, routing, dan port yang sedang listen (berguna untuk troubleshooting service).

Langkah-langkah:

1. Lihat interface dan IP:

```
1 ip a
```

Kode 1.54: Cek IP address

2. Lihat routing table:

```
1 ip r
```

Kode 1.55: Cek routing

3. Lihat port yang sedang listening:

```
1 sudo ss -tulpn
```

Kode 1.56: Cek port listening

### Latihan 2.5

Pilih satu port yang listening dari output `ss -tulpn`(misal port 22), lalu tuliskan service/proses yang membukanya. Jelaskan kegunaan port tersebut secara singkat.

## 1.8 Rangkuman

- Linux mendeteksi perangkat melalui kernel, sysfs (`/sys`), udev, dan membuat node perangkat di `/dev`.
- Utilitas seperti `lspci/lsusb/lblk` membantu identifikasi perangkat; `lspci -k` membantu melihat driver/modul.
- Modul kernel dapat dilihat dengan `lsmod`, dianalisis dengan `modinfo`, dan dikelola dengan `modprobe`.
- `/dev` berisi device node (block vs character device) untuk interaksi perangkat.
- Perintah dasar terminal dan konsep pipe/redirection adalah fondasi administrasi server.
- `grep/sed/awk` memudahkan pencarian, pengubahan, dan peringkasan teks/log.
- `ps/top/kill` digunakan untuk observasi proses dan pengiriman sinyal penghentian.
- Monitoring sistem mencakup disk (`df/du`), load (`uptime`), service (`systemctl`), dan log (`journalctl`).

## 1.9 Latihan

### Latihan 2.A

Jalankan `lspci -nnk`. Pilih 1 perangkat PCI dan tuliskan: nama perangkat, ID vendor:device, dan kernel driver in use.

### Latihan 2.B

Tentukan device root filesystem dengan `findmnt /`. Lalu cocokkan dengan `lsblk -f` dan tuliskan tipe filesystem serta UUID-nya.

### Latihan 2.C

Buat file `server.log` berisi minimal 10 baris dengan variasi kata: INFO, WARN, ERROR. Gunakan `grep` untuk menampilkan hanya baris ERROR.

### Latihan 2.D

Gunakan `sed` untuk mengganti semua kata `server` menjadi `node` pada file latihan. Tunjukkan sebelum dan sesudah.

### Latihan 2.E

Gunakan `df -h` lalu `awk` untuk menampilkan filesystem yang penggunaan disk di atas 70%.

### Latihan 2.F

Jalankan `sleep 600 &`. Temukan PID-nya dengan `ps`. Hentikan dengan SIGTERM. Jelaskan beda SIGTERM vs SIGKILL.

### Latihan 2.G

Gunakan `systemctl -failed`. Jika tidak ada yang gagal, pilih satu service aktif (misal `ssh`) dan tampilkan status serta 30 baris log terakhirnya.

## 1.10 Referensi Tambahan

- Manual pages: `man lspci`, `man lsusb`, `man lsblk`, `man modprobe`, `man journalctl`, `man systemctl`
- William E. Shotts, *The Linux Command Line* (dasar command line dan text processing)

### *1.10 Referensi Tambahan*

- Dokumentasi Ubuntu Server (administrasi service systemd dan troubleshooting log)