

**LAPORAN MINGGU KE-12**  
**INSTALASI LINUX DAN ENVIROMENT CODING(PYTHON / C)**



**Dosen Pengampu:**

Ferdi Chahyadi, S.Kom., M.Cs

Disusun Oleh:

Siti Umayah	2401020018
Anika	2401020029
Aldi Syaputra	2401020024
Fauziah Sal Sabillah	2401020034

**JURUSAN TEKNIK INFORMATIKA**  
**FAKULTAS TEKNIK DAN TEKNOLOGI KEMARITIMAN**  
**UNIVERSITAS MARITIM RAJA ALI HAJI**  
**TANJUNGPINANG**

**2025**

## **ABSTRAK**

Laporan ini membahas progress pelaksanaan proyek Monitoring Resource System (Linux Task Manager Advanced) pada minggu ke-12 yang fokus pada instalasi Linux dan persiapan environment coding menggunakan Python/C. Proyek ini bertujuan untuk membuat aplikasi monitoring real-time yang dapat memantau penggunaan CPU, RAM, disk, proses sistem, serta mampu menghentikan proses menggunakan signal handling. Pada minggu ke-12 ini, telah berhasil dilakukan instalasi sistem operasi Linux, konfigurasi environment development, instalasi Python beserta library pendukung seperti psutil untuk monitoring resource system, serta testing dasar untuk memastikan environment siap digunakan. Hasil testing menunjukkan bahwa semua komponen yang dibutuhkan telah terinstall dengan baik dan berfungsi sesuai dengan kebutuhan proyek. Dengan selesainya tahap persiapan ini, proyek siap dilanjutkan ke tahap implementasi monitoring CPU/RAM pada minggu berikutnya.

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Sistem operasi Linux merupakan platform yang powerful untuk pengembangan aplikasi sistem, khususnya aplikasi yang berinteraksi langsung dengan kernel dan resource sistem. Dalam era komputasi modern, monitoring resource sistem menjadi sangat penting untuk memastikan performa optimal dan mengidentifikasi bottleneck yang dapat mengganggu kinerja sistem.

Task Manager atau System Monitor adalah salah satu aplikasi yang sangat krusial dalam manajemen sistem operasi. Aplikasi ini memungkinkan pengguna dan administrator sistem untuk memantau penggunaan resource secara real-time, mengidentifikasi proses yang mengonsumsi resource berlebihan, dan melakukan tindakan korektif seperti menghentikan proses yang bermasalah.

Proyek ini mengembangkan aplikasi monitoring resource system yang advanced dengan fitur-fitur seperti monitoring real-time CPU, RAM, disk usage, manajemen proses, visualisasi grafik penggunaan resource, dan kemampuan signal handling untuk mengelola proses sistem. Pengembangan aplikasi ini menggunakan bahasa pemrograman Python dengan library psutil yang menyediakan interface untuk mengakses informasi sistem dan proses.

#### 1.2 Tahapan Proyek

Proyek ini dilaksanakan dalam lima minggu dengan tahapan sebagai berikut:

1. Minggu 11: Membuat Proposal Proyek
2. Minggu 12: Instalasi Linux + Environment Coding (Python/C)
3. Minggu 13: Implementasi CPU/RAM Monitoring
4. Minggu 14: Implementasi Kill Process dan Pengujian Stress
5. Minggu 15: Laporan & Demo Program

Laporan ini membahas pelaksanaan pada minggu ke-12, yaitu tahap instalasi Linux dan persiapan environment coding.

### **1.3 Scope Minggu ke-12**

Pada minggu ke-12 ini, fokus utama adalah mempersiapkan infrastruktur development yang diperlukan untuk mengembangkan aplikasi monitoring resource system. Aktivitas meliputi instalasi sistem operasi Linux, setup environment development, instalasi Python dan library pendukung, serta melakukan testing dasar untuk memastikan semua komponen berfungsi dengan baik.

### **1.4 RUMUSAN MASALAH**

Berdasarkan tujuan proyek dan tahapan yang harus dilalui pada minggu ke-12, berikut adalah rumusan masalah yang perlu diselesaikan:

1. Bagaimana melakukan instalasi sistem operasi Linux dengan konfigurasi yang optimal untuk development aplikasi sistem?
2. Apa saja tools dan library yang dibutuhkan untuk mengembangkan aplikasi monitoring resource system?
3. Bagaimana cara menginstall dan mengkonfigurasi Python environment beserta library pendukung seperti psutil untuk monitoring sistem?
4. Bagaimana memastikan bahwa environment development yang telah disetup dapat berfungsi dengan baik untuk mengakses informasi resource sistem?
5. Apa saja kendala yang mungkin dihadapi dalam proses instalasi dan setup environment, serta bagaimana solusinya?

### **1.5 TUJUAN PROJECT**

#### **1. Tujuan Umum**

Mengembangkan aplikasi monitoring resource system berbasis Linux yang mampu memantau penggunaan CPU, RAM, disk, dan proses secara real-time, serta memiliki kemampuan untuk mengelola proses menggunakan signal handling.

## 2. Tujuan Khusus Minggu ke-12

Pada minggu ke-12 ini, tujuan khusus yang ingin dicapai adalah:

1. Melakukan instalasi sistem operasi Linux dengan konfigurasi yang sesuai untuk development aplikasi sistem
2. Mempersiapkan environment development yang mencakup IDE/text editor, compiler, dan tools pendukung lainnya
3. Menginstall Python beserta package manager (pip) untuk mengelola library yang dibutuhkan
4. Menginstall library psutil sebagai library utama untuk mengakses informasi resource sistem dan proses
5. Menginstall library pendukung lainnya seperti matplotlib untuk visualisasi grafik (jika diperlukan)
6. Melakukan testing dasar untuk memastikan bahwa environment yang disetup dapat mengakses informasi CPU, RAM, dan disk usage
7. Mendokumentasikan proses instalasi beserta kendala dan solusi yang ditemukan

## 3. Output yang Diharapkan

Pada akhir minggu ke-12, diharapkan:

- Sistem Linux terinstall dan berjalan dengan baik
- Python environment siap digunakan untuk development
- Library psutil dan pendukung lainnya terinstall dengan sukses
- Testing dasar menunjukkan bahwa monitoring resource dapat dilakukan
- Dokumentasi lengkap proses instalasi dan konfigurasi
- Environment siap untuk tahap implementasi pada minggu berikutnya

## **BAB II**

### **METODOLOGI**

#### **2.1 Metode Pelaksanaan**

Pelaksanaan minggu ke-12 dilakukan dengan tahapan sistematis sebagai berikut:

##### **Tahap 1: Persiapan**

- Menentukan distribusi Linux yang akan digunakan
- Mempersiapkan media instalasi (USB bootable / Virtual Machine)
- Backup data penting (jika dual boot)

##### **Tahap 2: Instalasi Linux**

- Melakukan instalasi sistem operasi Linux
- Konfigurasi partisi dan bootloader
- Setup user dan password
- Update sistem ke versi terbaru

##### **Tahap 3: Setup Environment Development**

- Instalasi Python dan pip
- Instalasi C/C++ compiler (gcc, g++)
- Instalasi IDE/text editor
- Instalasi Git dan tools pendukung

##### **Tahap 4: Instalasi Library**

- Instalasi psutil melalui pip
- Instalasi library pendukung lainnya
- Verifikasi instalasi library

##### **Tahap 5: Testing**

- Membuat script testing sederhana

- Testing akses ke informasi CPU
- Testing akses ke informasi RAM
- Testing akses ke informasi Disk
- Testing akses ke daftar proses

#### **Tahap 6: Dokumentasi**

- Screenshot hasil instalasi
- Dokumentasi kendala dan solusi
- Pembuatan laporan

## **BAB III**

### **HASIL DAN PEMBAHASAN**

#### **3.1 Instalasi Linux**

Distribusi yang Dipilih: Ubuntu 24.04 LTS

Spesifikasi Hardware:

- Processor: AMD Athlon 300U with Radeon Vega Mobile Gfx 2.40 GHz
- RAM : 8. 00 GB
- Storage: 512 GB SSD
- Alokasi untuk Linux: 50 GB

Metode Instalasi:

- Dual Boot dengan Windows
- Virtual Machine (VirtualBox/VMware)
- WSL2 (Windows Subsystem for Linux)
- Native Installation

Proses Instalasi:

Instalasi dilakukan dengan langkah-langkah berikut:

1. Pembuatan Media Instalasi
  - a) Download ISO file Ubuntu 22.04 LTS dari website resmi
  - b) Membuat bootable USB menggunakan Rufus/Etcher
  - c) Ukuran USB: minimum 4 GB
2. Konfigurasi BIOS/UEFI
  - a) Disable Secure Boot (jika diperlukan)
  - b) Set boot priority ke USB
  - c) Restart dan masuk ke installer

#### **3.2 Setup Environment Development**

Instalasi Python:

```
gita@DESKTOP-RUU77IK:/mnt/c/Users/lenovo$ sudo apt install python3 python3-pip -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.12.3-0ubuntu2).
python3 set to manually installed.
```

```
gita@DESKTOP-RUU77IK:/mnt/c/Users/lenovo$ pip3 --version
Command 'pip3' not found, but can be installed with:
sudo apt install python3-pip
```

```
gita@DESKTOP-RUU77IK:/mnt/c/Users/lenovo$ sudo apt install python3-pip -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
libdrm-nouveau2 libdrm-radeon1 libgl1-amber-dri
libglapi-mesa liblvm17t64 libxcb-dri2-0
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
build-essential bzip2 cpp cpp-13 cpp-13-x86-64-linux-gnu
cpp-x86-64-linux-gnu dpkg-dev fakeroot g++ g++-13
g++-13-x86-64-linux-gnu g++-x86-64-linux-gnu gcc gcc-13
gcc-13-base gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu
javascript-common libalgorithm-diff-perl
libalgorithm-diff-xs-perl libalgorithm-merge-perl libaom3
libasan8 libatomic1 libc-dev-bin libc-devtools libc6-dev
libcc1-0 libcrypt-dev libde265-0 libdpkg-perl libexpat1-dev
libfakeroot libfile-fcntllock-perl libgcc-13-dev libgd3
libgomp1 libheif-plugin-aomdec libheif-plugin-aomenc
libheif-plugin-libde265 libheif1 libhwasan0 libisl23 libitm1
libjs-jquery libjs-sphinxdoc libjs-underscore liblsan0
libmpc3 libpython3-dev libpython3.12-dev libquadmath0
libstdc++-13-dev libtsan2 libubsan1 libxpm4 linux-libc-dev
lto-disabled-list make manpages-dev python3-dev
python3-wheel python3.12-dev rpcsvc-proto zlib1g-dev
```

### 3.3 Testing Environment

Setelah semua komponen terinstall, dilakukan testing untuk memastikan environment siap digunakan.

#### Script Testing 1: Basic System Info

```
# test_basic.py

import psutil

import platform
```

```
print("=="*50)

print("SYSTEM INFORMATION")

print("=="*50)

# OS Information

print(f"System: {platform.system() }")

print(f"Node Name: {platform.node() }")

print(f"Release: {platform.release() }")

print(f"Version: {platform.version() }")

print(f"Machine: {platform.machine() }")

print(f"Processor: {platform.processor() }")

print("\n" + "=="*50)

print("RESOURCE INFORMATION")

print("=="*50)

# CPU Information

print(f"Physical cores: {psutil.cpu_count(logical=False) }")

print(f"Total cores: {psutil.cpu_count(logical=True) }")

print(f"CPU Usage: {psutil.cpu_percent(interval=1) }%")

# Memory Information

memory = psutil.virtual_memory()

print(f"\nTotal Memory: {memory.total / (1024**3):.2f} GB")

print(f"Available Memory: {memory.available / (1024**3):.2f} GB")

print(f"Used Memory: {memory.used / (1024**3):.2f} GB")
```

```

print(f"Memory Usage: {memory.percent}%")

# Disk Information

disk = psutil.disk_usage('/')

print(f"\nTotal Disk: {disk.total / (1024**3):.2f} GB")

print(f"Used Disk: {disk.used / (1024**3):.2f} GB")

print(f"Free Disk: {disk.free / (1024**3):.2f} GB")

print(f"Disk Usage: {disk.percent}%")

```

### Hasil Testing 1:

```

(venv) gita@DESKTOP-RUU77IK:~/MRS/script_basic_sistem_info/process_monitoring$ python script_basic_sistem_info.py
=====
SYSTEM INFORMATION
=====
System: Linux
Node Name: DESKTOP-RUU77IK
Release: 6.6.87.2-microsoft-standard-WSL2
Version: #1 SMP PREEMPT_DYNAMIC Thu Jun  5 18:30:46 UTC 2025
Machine: x86_64
Processor: x86_64

=====
RESOURCE INFORMATION
=====
Physical cores: 2
Total cores: 4
CPU Usage: 0.0%

Total Memory: 2.78 GB
Available Memory: 2.35 GB
Used Memory: 0.43 GB
Memory Usage: 15.4%

Total Disk: 1006.85 GB
Used Disk: 2.29 GB
Free Disk: 953.34 GB
Disk Usage: 0.2%
(venv) gita@DESKTOP-RUU77IK:~/MRS/script_basic_sistem_info/process_monitoring$ |

```

### Script Testing 2: process monitoring

```

# test_process.py

import psutil

print("=*80)

print("RUNNING PROCESSES")

print("=*80)

```

```

print(f"{'PID':<10} {'NAME':<25} {'CPU %':<10} {'MEMORY %':<10}")
print("=*80)

# Get all running processes

for proc in psutil.process_iter(['pid', 'name', 'cpu_percent',
'memory_percent']):
    try:

        info = proc.info

        print(f"{info['pid']:<10} {info['name']:<25}
{info['cpu_percent']:<10.2f} {info['memory_percent']:<10.2f}")

    except (psutil.NoSuchProcess, psutil.AccessDenied,
psutil.ZombieProcess):

        pass

print("=*80)

print(f"Total processes: {len(psutil.pids())}")

```

## Hasil Testing 2:

```

(venv) gita@DESKTOP-RUU77IK:~/MRS/script_basic_sistem_info/process_monitoring$ python process_monitoring.py
=====
RUNNING PROCESSES
=====
PID      NAME          CPU %     MEMORY %
=====
1        systemd       0.00      0.44
2        init-systemd  0.00      0.07
7        init          0.00      0.06
171      cron          0.00      0.09
172      dbus-daemon   0.00      0.18
188      systemd-logind 0.00      0.29
193      atd           0.00      0.08
198      rsyslogd      0.00      0.19
210      getty         0.00      0.07
214      unattended-upgr 0.00      0.77
219      getty         0.00      0.06
323      SessionLeader 0.00      0.03
324      Relay(325)    0.00      0.04
325      bash          0.00      0.17
326      login         0.00      0.15
3741     systemd       0.00      0.37
375      (sd-pam)     0.00      0.12
395      bash          0.00      0.17
940      polkitd      0.00      0.26
2956     systemd-journald 0.00      0.47
3008     systemd-timesyncd 0.00      0.26
3059     systemd-udevd   0.00      0.21
3135     systemd-resolved 0.00      0.44
4748     wsl-pro-service 0.00      0.46
4940     python        0.00      0.44
=====

(venv) gita@DESKTOP-RUU77IK:~/MRS/script_basic_sistem_info/process_monitoring$ python script_basic_sistem_info.py

```

### **3.4 Analisis Hasil**

Berdasarkan testing yang telah dilakukan, dapat disimpulkan bahwa:

1. Library psutil berfungsi dengan baik dalam mengakses informasi sistem seperti CPU, RAM, dan disk usage
2. Akses ke informasi proses berhasil dilakukan, termasuk PID, nama proses, CPU usage, dan memory usage
3. Environment development siap digunakan untuk tahap implementasi monitoring yang lebih advanced
4. Performa sistem Linux stabil dengan resource usage yang normal

## **BAB IV**

### **KESIMPULAN**

Berdasarkan pelaksanaan kegiatan pada minggu ke-12, dapat disimpulkan bahwa:

1. Instalasi sistem operasi Linux telah berhasil dilakukan dengan konfigurasi yang optimal untuk development aplikasi sistem. Sistem dapat berjalan dengan stabil dan semua hardware terdeteksi dengan baik.
2. Environment development telah disetup dengan lengkap, mencakup Python 3.x, pip, library psutil, library pendukung seperti matplotlib, serta development tools seperti Git dan IDE.
3. Library psutil berhasil diinstall dan berfungsi dengan baik dalam mengakses informasi resource sistem. Testing menunjukkan bahwa library ini dapat mengambil data CPU usage, RAM usage, disk usage, dan informasi proses dengan akurat.
4. Testing dasar telah dilakukan dengan hasil yang memuaskan. Script testing dapat menampilkan informasi sistem secara real-time dan mengakses daftar proses yang berjalan beserta resource yang dikonsumsi.
5. Kendala yang dihadapi dapat diatasi dengan baik melalui dokumentasi dan troubleshooting yang tepat. Sebagian besar kendala berkaitan dengan permission dan konfigurasi library.
6. Environment development siap digunakan untuk tahap implementasi pada minggu ke-13, yaitu implementasi monitoring CPU/RAM secara real-time dengan fitur yang lebih advanced.

Dengan selesainya tahap persiapan ini, proyek dapat dilanjutkan sesuai timeline yang telah ditentukan. Minggu depan akan fokus pada implementasi core functionality monitoring CPU dan RAM dengan update otomatis dan visualisasi yang lebih baik.

## **BAB V**

### **REFERENSI**

psutil Documentation. (2023). *"psutil documentation"*.

Python Documentation. (2023). *"The Python Standard Library"*.

Ubuntu Documentation. (2023). *"Ubuntu Desktop Guide"*.

Linux man pages. (2023). *"Linux Programmer's Manual"*.