MODUL JOBSHEET PRAKTIKUM MANDIRI

ADMINISTRASI SISTEM JARINGAN FASE F (KELAS XI)

Project-Based Learning (PjBL): Manajemen Layanan Linux Server Menggunakan Systemd/Systemctl, Kontrol Lifecycle Daemon, Analisis Log (Journalctl), Resolusi Konflik Port, dan Automasi Layanan pada Virtual Machine

A. IDENTITAS PRAKTIKUM

Parameter Identitas	Keterangan & Spesifikasi
Satuan Pendidikan	SMK Negeri 22 Jakarta
Bidang Keahlian	Teknologi Informasi
Program Keahlian	Teknik Komputer dan Jaringan (TKJ)
Mata Pelajaran	Administrasi Sistem Jaringan (ASJ)
Fase / Kelas	F / Kelas XI TKJ
Semester / Tahun Pelajaran	Ganjil (Semester 1) / 2026/2027
Model Pembelajaran	Project-Based Learning (PjBL)
Metode Pembelajaran	Praktikum Mandiri Laboratorium, Eksplorasi Teknis, Problem Solving, Diskusi
Alokasi Waktu	36 Jam Pelajaran (6 Pertemuan @ 6 x 45 Menit / 36 Jam Praktik)
Kode Tujuan Pembelajaran	TP 5.2.1 (Manajemen Layanan, Kontrol Daemon, dan Optimasi Service Dasar Server)
Nama Penyusun	Fernandez Eko NS

B. CAPAIAN PEMBELAJARAN (CP) & ALUR TUJUAN PEMBELAJARAN (ATP)

1. Capaian Pembelajaran (CP) Fase F (Elemen ASJ):
Pada akhir Fase F, peserta didik mampu menginstalasi sistem operasi jaringan; menjelaskan konsep; menginstalasi services; serta mengonfigurasi dan menguji remote server, DHCP server, DNS server, FTP server, file server, web server, mail server, database server, Control Panel Hosting, Shared Hosting Server, Dedicated Hosting Server, Virtual Private Server, VPN server, serta sistem kontrol dan monitoring.

2. Alur Tujuan Pembelajaran (ATP) - Posisi Ke-2 Kelas XI:
• TP 5.2.1: Menerapkan manajemen layanan sistem operasi jaringan (service management), mengontrol lifecycle daemon server, mengonfigurasi automasi saat booting, serta melakukan audit log layanan sesuai SOP kerja.
• Alur Proyek PjBL:
1. Merefleksikan & Merencanakan (TP 1): Analisis arsitektur init system Linux (systemd vs SysVinit), pemetaan Process ID 1 (PID 1), dan perencanaan skema kontrol layanan jaringan dasar (SSH, Web Server, DNS).
2. Mengaplikasikan Kontrol Service (TP 2): Eksekusi manajemen siklus hidup layanan (lifecycle) menggunakan utilitas systemctl (start, stop, restart, reload, status) serta pengujian waktu respons daemon.
3. Mengaplikasikan Automasi & Proteksi (TP 3): Konfigurasi automasi layanan saat booting (enable/disable), pemanfaatan fitur isolasi layanan (mask/unmask), dan analisis parameter Unit Files.
4. Manajemen Port & Resolusi Konflik (TP 4): Pemantauan socket dan port jaringan aktif (ss, netstat, lsof), identifikasi bentrokan port (port conflict), dan modifikasi binding address layanan.
5. Audit Log & Pembuatan Service Mandiri (TP 5): Analisis log sistem secara real-time menggunakan journalctl dan rsyslog, serta pembuatan berkas unit systemd custom (Custom Systemd Service) untuk automasi skrip pemeliharaan server.
6. Evaluasi, Hardening, & Dokumentasi (TP 6): Uji ketahanan layanan (reboot & recovery test), pembuatan titik cadangan (VM Snapshot), analisis pemecahan masalah (troubleshooting), dan penyusunan laporan kerja teknis.

C. TUJUAN PEMBELAJARAN (TP) OPERASIONAL

1. TP 5.2.1.1: Peserta didik mampu menjelaskan arsitektur systemd sebagai init system utama pada Linux Debian, serta membedakan fungsi perintah kontrol dasar systemctl.

2. TP 5.2.1.2: Peserta didik mampu mengoperasikan perintah systemctl untuk mengelola lifecycle layanan jaringan (memulai, menghentikan, me-restart, dan memuat ulang konfigurasi tanpa memutus koneksi klien).

3. TP 5.2.1.3: Peserta didik mampu mengonfigurasi automasi startup daemon (enable/disable), menerapkan proteksi layanan kritis menggunakan teknik masking, dan memetakan dependensi antar-layanan (list-dependencies).

4. TP 5.2.1.4: Peserta didik mampu mendiagnosis konflik port jaringan menggunakan perintah utilitas ss -tulpn dan lsof, serta memodifikasi parameter konfigurasi layanan untuk menyelesaikan bentrokan port (misalnya antara Apache dan Nginx).

5. TP 5.2.1.5: Peserta didik mampu melakukan investigasi kesalahan sistem menggunakan journalctl, membuat berkas Custom Systemd Unit Service mandiri, serta mendokumentasikan prosedur manajemen layanan ke dalam laporan teknis.

D. KOMPETENSI INDUSTRI

Modul praktikum ini dirancang untuk membekali peserta didik dengan standar kompetensi teknis Junior Network & System Administrator di dunia kerja industri IT:

1. Hard Skills (Kompetensi Teknis):
• Penguasaan mendalam terhadap System and Service Manager (systemd) dan utilitas antarmuka baris perintah (systemctl, systemd-analyze).
• Administrasi siklus hidup daemon server jaringan (OpenSSH, Nginx, Apache2, BIND9, dan Cron).
• Audit dan inspeksi port jaringan aktif, protokol transport (TCP/UDP), dan Process ID (PID) menggunakan socket statistics (ss, netstat) dan lsof.
• Analisis forensic logging dan pemantauan aktivitas kernel/service secara real-time menggunakan journalctl dan fasilitas syslog (/var/log/).
• Pengembangan dan penulisan skrip automasi server melalui pembentukan Custom Systemd Unit Files (.service & .timer) pada direktori /etc/systemd/system/.
• Manajemen recovery layanan: memproses perintah systemctl daemon-reload saat terjadi perubahan struktur berkas unit.

2. Soft Skills (Kompetensi Kerja Industri):
• Kepatuhan Prosedur (SOP): Selalu memeriksa status layanan dan log sebelum dan sesudah melakukan perubahan konfigurasi pada server produksi.
• Problem Solving & Root Cause Analysis (RCA): Kemampuan mandiri dalam menelusuri akar masalah saat layanan gagal berjalan (Job failed / Unit entered failed state) melalui pembacaan pesan log kesalahan.
• Manajemen Dokumentasi Teknis: Mencatat setiap parameter port binding, nama layanan, dan riwayat modifikasi service ke dalam logbook administrasi sistem.

E. SKENARIO PROJECT-BASED LEARNING (PjBL)

Konteks Industri/Proyek:
SMK Negeri 22 Jakarta sedang melanjutkan tahap penggelaran infrastruktur jaringan pada unit Teaching Factory. Setelah Server Fondasi (Base Server) berhasil dibangun dan

dikonfigurasi dengan IP Statis (192.168.10.10/24) pada proyek sebelumnya, tim teknisi bersiap memasang berbagai layanan utama seperti Web Server (Nginx & Apache2), DNS Server (BIND9), dan Remote Admin (SSH Server). Namun, sebelum layanan tersebut diaktifkan untuk pengguna sekolah, Kepala Unit IT mensyaratkan bahwa setiap teknisi wajib menguasai manajemen layanan Linux Server.

INSTRUKSI KERJA PROYEK (WORK ORDER):
Anda ditugaskan sebagai Tim Teknisi Sistem & Jaringan. Tugas Anda adalah melakukan manajemen komprehensif terhadap seluruh layanan (service daemon) di dalam Virtual Machine Debian Server Anda. Anda diminta untuk:
1. Menguji kontrol hidup-mati layanan SSH dan memonitor status PID-nya.
2. Mengonfigurasi agar layanan jaringan otomatis berjalan saat server dinyalakan (boot auto-start), dan melakukan penguncian (masking) pada layanan yang berpotensi menimbulkan kerentanan.
3. Mensimulasikan dan menyelesaikan konflik port 80 yang terjadi akibat pemasangan dua layanan web server sekaligus (Apache2 dan Nginx).
4. Menggunakan journalctl untuk menelusuri log kesalahan penulisan sintaks konfigurasi.
5. Membangun sebuah Layanan Mandiri (Custom Systemd Service) bernama tkj-monitor.service yang bertugas mencatat status sistem sekolah setiap kali server di-restart.
Proyek diakhiri dengan uji keandalan sistem (reboot test) dan pembuatan cadangan Snapshot VM baru.

F. PERTANYAAN PEMANTIK

1. Mengapa sistem operasi Linux modern (seperti Debian 12/13 dan Ubuntu) beralih dari sistem inisiasi klasik (SysVinit) ke systemd? Apa keunggulan systemd dalam hal kecepatan booting server dan manajemen dependensi antar-layanan?

2. Apa perbedaan fundamental antara perintah systemctl stop <service>, systemctl disable <service>, dan systemctl mask <service>? Kapan seorang System Administrator harus menggunakan teknik masking?

3. Mengapa dalam sebuah sistem operasi server, dua layanan yang berbeda (misalnya Apache2 dan Nginx) tidak dapat dijalankan secara bersamaan jika keduanya dikonfigurasi untuk mendengarkan (listen) pada nomor Port TCP yang sama (Port 80)? Bagaimana sistem operasi merespons bentrokan tersebut?

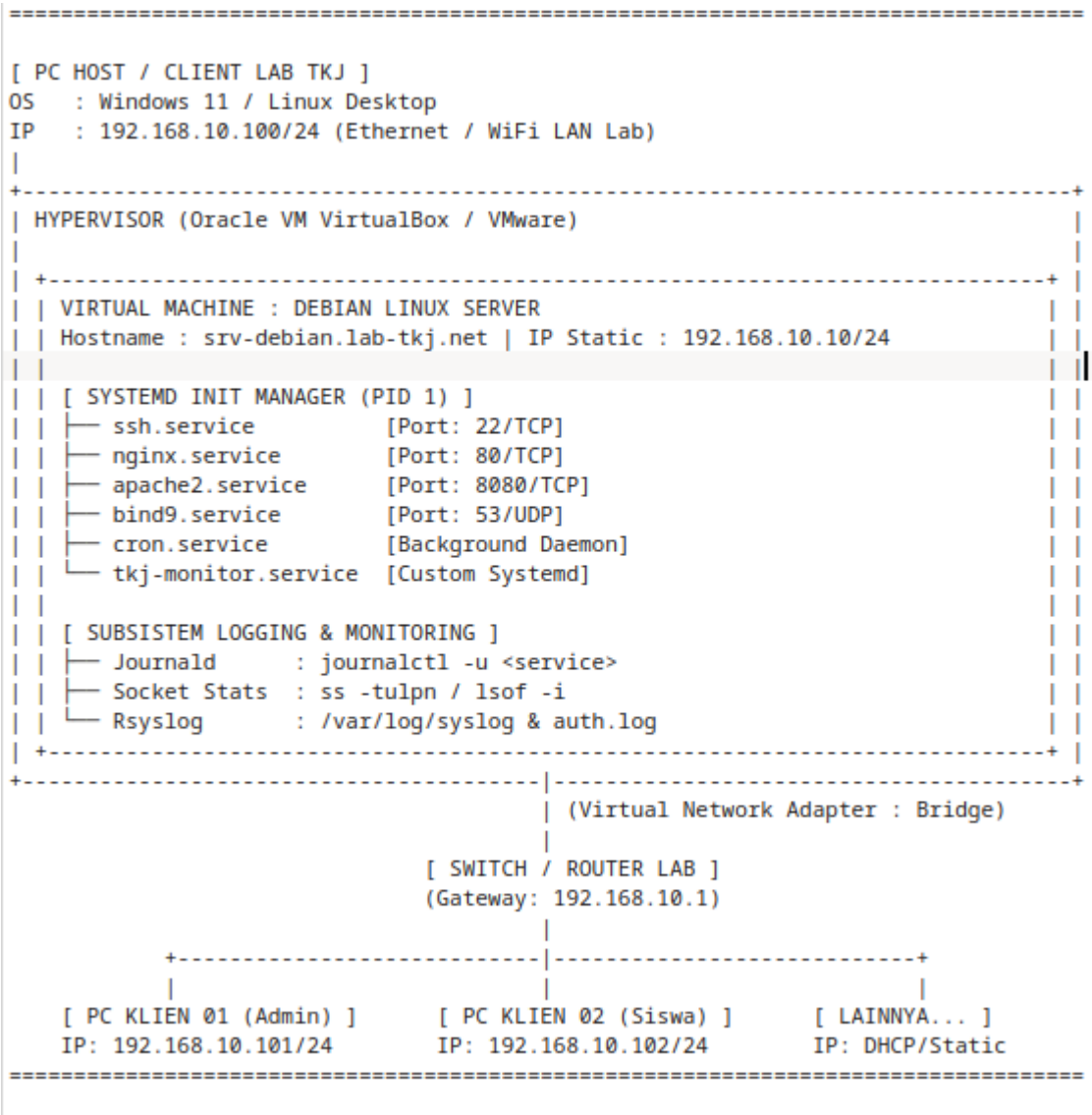
4. Ketika sebuah layanan gagal dijalankan (failed to start), pesan kesalahan di terminal umumnya sangat singkat. Mengapa utilitas journalctl menjadi senjata utama teknisi untuk menemukan baris kode yang salah atau penyebab crash pada layanan?

G. ALAT DAN BAHAN

Kategori Perangkat	Spesifikasi / Nama Perangkat	Keterangan & Fungsi Teknis
Perangkat Keras / PC Lab	PC / Laptop Lab TKJ (Intel Core i3 / AMD Ryzen 3 atau lebih tinggi)	CPU Min. 4 Cores/Threads, RAM Min. 8 GB, Storage Min. 100 GB Free Space. Fitur Virtualisasi (VT-x / AMD-V) aktif di BIOS.
Perangkat Lunak Virtualisasi	Oracle VM VirtualBox 7.x / VMware Workstation Pro / Proxmox VE	Hypervisor untuk mengelola mesin virtual Linux Debian Server dari proyek praktikum sebelumnya.
Mesin Virtual Target (VM)	VM SMKN22-Debian-Server-XI (Hasil Snapshot Bab Sebelumnya)	Mesin virtual Debian 12/13 Server dengan IP Statis (192.168.10.10/24), partisi manual (Root, Swap, Home), dan akun Sudo aktif.
Sistem Operasi Klien (Host)	Windows 10/11 atau Linux Desktop	Berfungsi sebagai terminal klien untuk melakukan akses remote SSH dan pengujian konektivitas ke layanan server.
Konektivitas Jaringan	Jaringan LAN Laboratorium dengan Akses Internet / Router Lokal	Digunakan untuk mengunduh paket simulasi layanan (apache2, nginx, bind9) melalui repositori APT lokal/internet.
Paket Layanan & Utility Lab	systemd, apache2, nginx, bind9, cron, htop, lsof, net-tools, iproute2, curl	Paket aplikasi layanan jaringan yang akan dikelola, dipantau port-nya, dan di-troubleshooting selama praktikum.

H. TOPOLOGI DAN ARSITEKTUR LAYANAN

Praktikum ini menggunakan arsitektur Single-Host Virtualization dengan Multi-Daemon Management. Mesin virtual server terhubung ke jaringan LAN laboratorium melalui mode Bridged Adapter. Di dalam sistem operasi tamu (Guest OS), systemd bertindak sebagai induk dari seluruh proses (PID 1) yang mengontrol siklus hidup daemon layanan jaringan.



I. RANCANGAN KONFIGURASI KLASTER / LAYANAN

Parameter Konfigurasi	Nilai / Spesifikasi Implementasi
Nama Mesin Virtual (VM Name)	SMKN22-Debian-Server-XI
Alokasi Sumber Daya VM	Memory: 2048 MB (2 GB) | Processor: 2 vCPU Cores | Storage: 20 GB
Alamat IP Statis & Gateway	IP: 192.168.10.10/24 | Gateway: 192.168.10.1 | Hostname: srv-debian
Layanan Utama Target Audit	1. OpenSSH Server (ssh.service) - Status: Active / Enabled 2. Cron Daemon (cron.service) - Status: Active / Enabled 3. Nginx Web Server (nginx.service) - Status: Active / Enabled (Port 80) 4. Apache2 Web Server (apache2.service) - Status: Modified (Port 8080) 5. BIND9 DNS Server (named.service) - Status: Masked / Protected
Alokasi Pemetaan Port (TCP/UDP)	• Port 22 (TCP): Layanan OpenSSH Server • Port 80 (TCP): Layanan Nginx Web Server • Port 8080 (TCP): Layanan Apache2 Web Server (Hasil resolusi konflik port) • Port 53 (TCP/UDP): Layanan BIND9 DNS Server (Kondisi terisolasi/masked)
Identitas Custom Unit Service	Nama File: /etc/systemd/system/tkj-monitor.service Script Eksekusi: /usr/local/bin/tkj-check.sh Deskripsi: SMKN 22 Jakarta - automated System & Service Health Monitor
Target Audit Log Systemd	Utiliti journalctl (parameter: -u, -f, -xe, --since today, -b)
Akun Operator & Sudo	Akun Admin: sysadmin (Grup Sudo aktif) | Kata Sandi: AdminTKJ#2026!

Nama Snapshot Cadangan Baru	Snapshot-02-Service-Management-Mastered

J. SINTAKS PROJECT-BASED LEARNING (PjBL)

Pelaksanaan praktikum modul ini dijadwalkan selama 36 Jam Pelajaran (6 Pertemuan @ 6 JP) sesuai Program Semester Ganjil (No. 2), yang mencakup bulan Agustus, September, dan Oktober dengan alur sintaks:

1. Penentuan Pertanyaan Mendasar (Start with the Essential Question - Pertemuan 1 / 6 JP): Guru memberikan studi kasus tentang insiden server crash di industri akibat kebocoran memori layanan dan konflik port antar-aplikasi. Peserta didik menganalisis mengapa penguasaan systemctl dan journalctl menjadi fondasi utama teknisi sebelum mengelola layanan jaringan.

2. Mendesain Perencanaan Proyek (Design a Plan for the Project - Pertemuan 1 / 6 JP): Peserta didik memeriksa kondisi cadangan Snapshot VM dasar, menyusun skema pemetaan port laboratorium (192.168.10.10:80 dan 8080), merancang arsitektur skrip monitoring layanan mandiri, dan mendaftar paket layanan yang akan diinstal (nginx, apache2, bind9).

3. Menyusun Jadwal & Alur Kerja (Create a Schedule - Pertemuan 2 & 3 / 12 JP): Peserta didik dan guru menyepakati jadwal implementasi proyek selama minggu efektif pembelajaran:
• Pertemuan 2 (6 JP): Instalasi paket simulasi layanan, eksekusi perintah dasar lifecycle (start, stop, restart, reload), dan eksperimen fitur proteksi auto-start serta masking.
• Pertemuan 3 (6 JP): Diagnosis pemantauan port aktif (ss, lsof), simulasi bentrokan port 80 antara Apache2 dan Nginx, serta penyelesaian konflik melalui rekonfigurasi port.

4. Pelaksanaan & Monitor Proyek (Monitor the Students and the Progress - Pertemuan 4 & 5 / 12 JP): • Pertemuan 4 (6 JP): Audit log kesalahan sistem menggunakan variasi parameter journalctl dan rsyslog, investigasi proses PID, dan manajemen sumber daya layanan dengan htop.
• Pertemuan 5 (6 JP): Pembuatan skrip pemrograman bash dan merakit berkas Custom Systemd Unit Service (tkj-monitor.service) di direktori sistem, lalu mengaktifkannya secara permanen.

5. Pengujian & Validasi Hasil (Assess the Outcome - Pertemuan 6 / 3 JP): Peserta didik melakukan Acceptance Test: uji ketahanan reboot server untuk memverifikasi automasi service, validasi pemisahan port 80 dan 8080, pengujian skrip custom unit, dan verifikasi status masked service.

6. Evaluasi & Refleksi (Evaluate the Experience - Pertemuan 6 / 3 JP): Peserta didik membuat titik cadangan sistem baru (Snapshot-02), menganalisis kendala teknis yang

ditemukan, melengkapi lembar evaluasi kerja, dan menyerahkan dokumentasi proyek kepada guru pengampu.

K. LANGKAH KERJA PRAKTIKUM MANDIRI

> **PERINGATAN K3LH & SOP:**
> *Bekerja di tingkat root atau menggunakan perintah sudo systemctl memiliki wewenang penuh untuk mematikan layanan kritis server. Jangan pernah melakukan stop atau mask pada layanan jaringan utama (seperti networking.service atau ssh.service) jika Anda sedang melakukan konfigurasi secara remote melalui SSH, karena hal tersebut akan memutuskan koneksi terminal Anda secara permanen! Selalu pastikan Anda memiliki akses langsung ke jendela konsol VirtualBox.*

Langkah 1: Persiapan VM & Verifikasi Arsitektur Systemd (PID 1)

1. Nyalakan PC Host laboratorium, buka aplikasi Oracle VM VirtualBox Manager, dan pilih mesin virtual SMKN22-Debian-Server-XI.
2. (Opsi SOP) Pastikan VM berada pada kondisi bersih dengan melakukan restore ke Snapshot-01-Base-OS-Installed-Clean. Klik tombol Start.
3. Login ke dalam terminal server menggunakan akun administratif:
   • Username: sysadmin | Kata Sandi: AdminTKJ#2026!
4. Verifikasi bahwa sistem operasi Debian Anda dikelola oleh systemd sebagai Process ID 1 (induk dari seluruh proses sistem):

```
ps -p 1 -o pid,comm,user
```

Output harus menunjukkan PID 1, command systemd, dan user root.
5. Periksa waktu yang dibutuhkan oleh server untuk menyelesaikan proses booting dan memuat seluruh layanan sistem:

```
systemd-analyze
systemd-analyze blame | head -n 10
```

Perintah blame akan mengurutkan 10 layanan yang memakan waktu paling lama saat server dinyalakan.

Langkah 2: Manajemen Siklus Hidup Layanan Dasar (Lifecycle Management)

1. Kita akan menggunakan layanan OpenSSH Server (ssh.service) dan Cron (cron.service) sebagai objek eksperimen siklus hidup layanan. Periksa status layanan SSH:

```
sudo systemctl status ssh
```

Perhatikan indikator warna hijau Active: active (running), nomor Main PID, dan alokasi memori yang digunakan.
2. Lakukan uji coba menghentikan (stop) layanan internal cron daemon:

```
sudo systemctl stop cron
sudo systemctl status cron
```

Status kini akan berubah menjadi Active: inactive (dead).
3. Nyalakan kembali (start) layanan cron dan verifikasi perubahan nomor Main PID-nya:

```
sudo systemctl start cron
sudo systemctl status cron
```

Perhatikan bahwa nomor Main PID telah berubah karena sistem menghasilkan proses baru.
4. Perbedaan restart dan reload:
• restart: Mematikan proses sepenuhnya lalu menyalakannya kembali (PID berubah, koneksi klien yang sedang berjalan akan terputus seketika).
• reload: Meminta daemon membaca ulang berkas konfigurasi tanpa mematikan proses utama (PID tetap sama, koneksi klien tetap aman).
• Uji coba me-reload layanan SSH tanpa memutus sesi Anda:

```
sudo systemctl reload ssh
sudo systemctl status ssh
```

Langkah 3: Kontrol Automasi Booting (Enable/Disable) & Proteksi Layanan (Masking)

Layanan yang tidak digunakan namun terinstal di server berpotensi menjadi celah keamanan. Kita dapat menonaktifkan automasinya (disable), atau menguncinya secara total (mask) agar tidak bisa dinyalakan oleh program atau aktor jahat sekalipun.

1. Untuk pengujian proteksi, kita akan menginstal paket layanan BIND9 DNS Server terlebih dahulu:

```
sudo apt update && sudo apt install bind9 bind9utils -y
```

2. Periksa status automasi booting pada layanan BIND9 (named.service):

```
systemctl is-enabled named
```

Output default umumnya adalah enabled (akan otomatis aktif saat VM di-restart).
3. Nonaktifkan automasi booting layanan BIND9 (disable):

```
sudo systemctl disable named
systemctl is-enabled named
```

Output berubah menjadi disabled. Layanan tidak akan menyala otomatis saat booting, TETAPI teknisi masih bisa menyalakannya secara manual dengan perintah systemctl start named.
4. Menerapkan Proteksi Total (Masking):
Kunci layanan BIND9 secara permanen agar sistem memetakan unit file-nya ke /dev/null:

```
sudo systemctl stop named
sudo systemctl mask named
sudo systemctl status named
```

Status akan menunjukkan Loaded: masked (/dev/null; bad).
5. Uji coba paksa menyalakan layanan yang telah di-masking:

```
sudo systemctl start named
```

Sistem akan menolak dengan pesan error: Failed to start named.service: Unit named.service is masked. (Proteksi berhasil!)
6. Untuk mengembalikan layanan agar dapat digunakan kembali di masa depan (unmask):

```
sudo systemctl unmask named
sudo systemctl enable named
```

Langkah 4: Investigasi Port & Resolusi Konflik Port 80 (Apache2 vs Nginx)

Di lapangan, teknisi sering mengalami error server saat memasang aplikasi baru karena port jaringan yang dibutuhkan telah diduduki oleh layanan lain. Kita akan mensimulasikan bentrokan Port 80 TCP.

1. Pasang paket layanan Nginx Web Server dan pastikan berjalan lancar di Port 80:

```
sudo apt install nginx -y
sudo systemctl status nginx
```

2. Periksa port jaringan yang sedang mendengarkan (listening socket) pada sistem menggunakan utilitas ss (socket statistics):

```
sudo ss -tulpn | grep -E '(80|22)'
```

Anda akan melihat proses nginx menduduki port 80 dan sshd menduduki port 22.
3. Simulasi Konflik Port: Sekarang, pasang paket Apache2 Web Server ke dalam server yang sama:

```
sudo apt install apache2 -y
```

Perhatikan pesan error di layar terminal saat instalasi selesai! Apache2 akan gagal menyala dengan pesan: Job for apache2.service failed because the control process exited with error code.
4. Diagnosis penyebab kegagalan Apache2 menggunakan status dan lsof:

```
sudo systemctl status apache2
sudo lsof -i :80
```

Anda akan melihat pesan kesalahan: Address already in use: AH00072: make_sock: could not bind to address [::]:80. Hal ini membuktikan Nginx telah menduduki Port 80 terlebih dahulu!
5. Resolusi Konflik (Solusi Teknis):
Kita harus memisahkan port kedua layanan tersebut. Kita akan mengubah port Apache2 dari port 80 menjadi Port 8080, sementara Nginx tetap di Port 80.

- Buat cadangan dan edit berkas port Apache2:

```
sudo cp /etc/apache2/ports.conf /etc/apache2/ports.conf.bak
sudo nano /etc/apache2/ports.conf
```

- Cari baris sintaks Listen 80 dan ubah menjadi Listen 8080. Simpan berkas (CTRL+O, ENTER, CTRL+X).
- Edit juga berkas virtual host default Apache2:

```
sudo nano /etc/apache2/sites-enabled/000-default.conf
```

- Ubah baris paling atas dari <VirtualHost *:80> menjadi <VirtualHost *:8080>. Simpan berkas.

6. Restart layanan Apache2 dan verifikasi bahwa kedua web server kini dapat berjalan berdampingan secara harmonis:

```
sudo systemctl restart apache2
sudo systemctl status apache2
sudo ss -tulpn | grep -E '(80|8080)'
```

Output ss akan menampilkan nginx pada port 80 dan apache2 pada port 8080. Konflik berhasil diselesaikan!
7. Uji akses kedua layanan dari terminal server menggunakan utilitas curl:

```
curl -I http://192.168.10.10:80
curl -I http://192.168.10.10:8080
```

Port 80 harus menghasilkan header Server: nginx, sedangkan Port 8080 menghasilkan header Server: Apache.

Langkah 5: Audit Log Layanan secara Real-Time Menggunakan Journalctl

Kapan pun sebuah layanan mengalami kegagalan fungsi, langkah pertama yang wajib dilakukan seorang System Administrator bukanlah menginstal ulang paket, melainkan membaca log journald menggunakan utilitas journalctl.

1. Untuk meninjau log spesifik milik suatu layanan (misalnya Apache2), gunakan parameter -u (unit):

```
sudo journalctl -u apache2 --no-pager | tail -n 20
```

Perintah ini menampilkan 20 baris catatan log terakhir yang dihasilkan oleh proses Apache2, termasuk riwayat error saat terjadi bentrokan port di Langkah 4.
2. Memantau Log secara Real-Time (Follow Mode):
Buka terminal, lalu jalankan pemantauan log SSH secara live (seperti fitur tail -f):

```
sudo journalctl -u ssh -f
```

(Biarkan terminal ini terbuka). Sekarang, buka aplikasi Command Prompt (CMD) atau PuTTY dari PC Host Windows/Linux, lalu coba lakukan login SSH ke IP server (ssh sysadmin@192.168.10.10).
Perhatikan bagaimana layar terminal VM Anda langsung menampilkan baris log baru secara instan saat ada aktivitas login klien! Tekan CTRL+C untuk keluar dari mode pemantauan.

3. Penyaringan Log Lanjutan untuk Investigasi Forensik:

• Melihat log error yang terjadi pada hari ini saja:

```
sudo journalctl --since today --priority=err
```

• Melihat log khusus dari sesi booting yang sedang berjalan saat ini (-b):

```
sudo journalctl -b -u nginx
```

• Melihat log sistem lengkap dengan detail kode kesalahan di akhir (tail exit):

```
sudo journalctl -xe
```

Langkah 6: Pembuatan Layanan Mandiri (Custom Systemd Unit Service)

Untuk membuktikan pemahaman mendalam terhadap arsitektur systemd, teknisi TKJ akan membangun sebuah service baru dari nol yang bertugas memonitor performa server sekolah secara otomatis setiap kali sistem dinyalakan.

1. Buat sebuah skrip bash pemantau kesehatan sistem di direktori /usr/local/bin/:

```
sudo nano /usr/local/bin/tkj-check.sh
```

Masukkan kode skrip automasi berikut ke dalam editor:

```
#!/bin/bash
# Skrip Automasi Monitoring Kesehatan Server - SMKN 22 Jakarta

LOG_FILE="/var/log/tkj-health-monitor.log"
TIMESTAMP=$(date '+%Y-%m-%d %H:%M:%S')
HOSTNAME=$(hostname -f)
UPTIME=$(uptime -p)
RAM_FREE=$(free -h | awk '/^Mem:/ {print $4}')

echo "============================================================" >> $LOG_FILE
echo "[*] SMKN 22 JAKARTA - SERVICE MONITORING REPORT" >> $LOG_FILE
echo " -> Waktu Check : $TIMESTAMP" >> $LOG_FILE
echo " -> Hostname    : $HOSTNAME" >> $LOG_FILE
echo " -> Uptime      : $UPTIME" >> $LOG_FILE
echo " -> Free RAM    : $RAM_FREE" >> $LOG_FILE
echo "[*] Status Layanan Kritis:" >> $LOG_FILE
```

```
for svc in ssh nginx apache2 cron; do
   STATUS=$(systemctl is-active $svc 2>/dev/null)
   echo "    - Service $svc : [ $STATUS ]" >> $LOG_FILE
done
echo "============================================================" >> $LOG_FILE
```

(Simpan berkas dengan CTRL+O, ENTER, CTRL+X).
2. Berikan hak akses eksekusi (executable permission) pada skrip tersebut:

```
sudo chmod +x /usr/local/bin/tkj-check.sh
```

3. Uji coba eksekusi skrip secara manual terlebih dahulu untuk memastikan tidak ada syntax error:

```
sudo /usr/local/bin/tkj-check.sh
cat /var/log/tkj-health-monitor.log
```

Pastikan laporan status server telah tercipta dengan rapi di dalam berkas log!
4. Membuat Berkas Systemd Unit (.service):
Sekarang, kita rakit berkas konfigurasi service di dalam direktori sistem /etc/systemd/system/:

```
sudo nano /etc/systemd/system/tkj-monitor.service
```

Tuliskan struktur parameter unit systemd berikut dengan teliti:

```
[Unit]
Description=SMKN 22 Jakarta - Automated System & Service Health Monitor
After=network.target nginx.service apache2.service

[Service]
Type=oneshot
ExecStart=/usr/local/bin/tkj-check.sh
RemainAfterExit=yes
User=root

[Install]
WantedBy=multi-user.target
```

(Simpan berkas konfigurasi unit).
Penjelasan Parameter:

• After=: Menjamin layanan ini baru berjalan SESUDAH network, nginx, dan apache2 selesai menyala.
• Type=oneshot: Ideal untuk skrip yang melakukan satu tugas sekali jalan lalu selesai.
• WantedBy=multi-user.target: Meminta agar layanan ini diaktifkan pada mode normal server CLI (Multi-User Runlevel).
5. Registrasi dan Aktivasi Custom Service:
Setiap kali ada penambahan atau modifikasi berkas di /etc/systemd/system/, kita WAJIB memberi tahu daemon utama systemd untuk memuat ulang indeks unitnya:

```
sudo systemctl daemon-reload
```

6. Aktifkan layanan custom kita agar berjalan otomatis setiap kali server booting, lalu nyalakan layanannya saat ini:

```
sudo systemctl enable tkj-monitor.service
sudo systemctl start tkj-monitor.service
sudo systemctl status tkj-monitor.service
```

Status layanan harus menunjukkan Active: active (exited) dan pesan Loaded: loaded (...; enabled; ...). Kata 'exited' adalah normal dan sukses untuk tipe layanan oneshot!

Langkah 7: Pengujian Terpadu & Uji Ketahanan (Acceptance & Reboot Test)

Untuk membuktikan bahwa seluruh konfigurasi layanan Anda stabil dan tangguh terhadap reboot sistem:
1. Bersihkan isi berkas log monitoring kita agar kita bisa melihat hasil pengujian baru:

```
sudo truncate -s 0 /var/log/tkj-health-monitor.log
```

2. Lakukan reboot pada server virtual Anda:

```
sudo reboot
```

3. Setelah server menyala kembali dan Anda login sebagai sysadmin, periksa apakah layanan mandiri kita bekerja secara otomatis saat booting:

```
cat /var/log/tkj-health-monitor.log
```

Jika laporan baru terisi secara otomatis dengan status ssh [active], nginx [active], dan apache2 [active], maka Proyek PjBL Manajemen Layanan Anda dinyatakan Berhasil Sempurna!

Langkah 8: Pembuatan Snapshot Mesin Virtual (Backup Point)

Anda telah berhasil menguasai manajemen layanan, memisahkan konflik port jaringan, dan menciptakan systemd service mandiri. Amankan kondisi server master ini sebelum melangkah ke modul materi berikutnya!

1. Matikan server Linux Debian dengan perintah:

```
sudo shutdown -h now
```

2. Pada aplikasi VirtualBox Manager di PC Host, klik pada VM SMKN22-Debian-Server-XI.
3. Klik menu Machine Tools -> pilih Snapshots.
4. Klik tombol Take (Ambil) ikon kamera.
5. Beri nama snapshot baru Anda: Snapshot-02-Service-Management-Mastered
   • Description: Modul Manajemen Layanan selesai. Nginx Port 80 & Apache2 Port 8080 aktif berdampingan. Custom service tkj-monitor.service terdaftar dan enabled. BIND9 terinstal dan standby.
6. Klik OK.

L. PERINTAH KONFIGURASI & SPESIFIKASI KODE YAML/SCRIPT

Berikut adalah ringkasan skrip bash (Cheat Sheet / Automated Verifier) yang dapat dijalankan di terminal server untuk melakukan audit cepat terhadap seluruh status manajemen layanan dan port di laboratorium:

```
#!/bin/bash
#
================================================================================
# SCRIPT AUTOMASI VERIFIKASI MANAJEMEN LAYANAN LINUX (LAB ASJ SMKN 22 JAKARTA)
#
================================================================================

echo
"================================================================================"
echo "[1] MEMERIKSA ARSITEKTUR INIT SYSTEM & BOOT TIME..."
echo " -> Init Process (PID 1) : $(ps -p 1 -o comm=) (User: $(ps -p 1 -o user=))"
echo " -> Waktu Booting Server : $(systemd-analyze | awk '{print $0}')"
echo
"================================================================================"
```

```
echo "[2] MEMERIKSA STATUS & AUTOMASI LAYANAN KRITIS..."
printf "%-15s %-15s %-15s %-10s\n" "NAMA LAYANAN" "STATUS KETERANGAN" "AUTOMASI BOOT"
"MAIN PID"
echo "--------------------------------------------------------------------------------"
services=("ssh" "nginx" "apache2" "cron" "named" "tkj-monitor")
for svc in "${services[@]}"; do
   state=$(systemctl is-active $svc 2>/dev/null)
   enabled=$(systemctl is-enabled $svc 2>/dev/null)
   pid=$(systemctl show --property MainPID --value $svc 2>/dev/null)
   if [ "$pid" == "0" ]; then pid="-"; fi
   printf "%-15s %-15s %-15s %-10s\n" "$svc" "[$state]" "[$enabled]" "$pid"
done
echo
"================================================================================"

echo "[3] MEMERIKSA ALOKASI PORT JARINGAN AKTIF (SOCKET AUDIT)..."
echo "--------------------------------------------------------------------------------"
sudo ss -tulpn | grep -E '(nginx|apache2|sshd|named)' | awk '{print " -> Protokol: " $1 "\tListen: " $5
"\tProcess: " $7}'
echo
"================================================================================"

echo "[4] MEMERIKSA HASIL EKSEKUSI CUSTOM SYSTEMD SERVICE..."
if [ -f "/var/log/tkj-health-monitor.log" ]; then
   echo " -> Berkas Log Terdeteksi. 4 Baris Laporan Terakhir:"
   tail -n 4 /var/log/tkj-health-monitor.log | sed 's/^/    /'
else
   echo " [!] PERINGATAN: Berkas /var/log/tkj-health-monitor.log tidak ditemukan!"
fi
echo
"================================================================================"
```

M. LEMBAR HASIL PRAKTIKUM

Lengkapi lembar observasi laboratorium berikut berdasarkan hasil pengujian nyata pada mesin virtual Debian Server Anda:

Tahapan & Parameter Uji	Target Hasil yang Diharapkan	Status & Catatan Hasil Pengujian
1. Verifikasi Init System (PID 1)	Perintah ps -p 1 mengembalikan nama proses systemd dengan	[ ] Ya [ ] Tidak Nama Command PID 1: .....

	kepemilikan user root.	
2. Uji Lifecycle Layanan Cron	Layanan cron.service berhasil di-stop (inactive) dan di-start kembali (active running).	[ ] Ya [ ] Tidak Perubahan Nomor PID: .....
3. Uji Reload OpenSSH Server	Perintah systemctl reload ssh berhasil dieksekusi tanpa memutus sesi SSH klien aktif.	[ ] Ya [ ] Tidak Status SSH Klien: .....
4. Proteksi Layanan BIND9 (Mask)	Layanan named.service berstatus masked (/dev/null) dan menolak saat di-start.	[ ] Ya [ ] Tidak Pesan Error Masking: .....
5. Resolusi Konflik Port 80	Nginx aktif di Port 80 dan Apache2 aktif di Port 8080 tanpa ada bentrokan (AH00072).	[ ] Ya [ ] Tidak Port Apache2 Baru: .....
6. Uji Akses Web via Curl	curl http://192.168.10.10:80 merespons Nginx; port :8080 merespons Apache.	[ ] Ya [ ] Tidak Header Server 8080: .....
7. Real-time Logging Journalctl	Perintah journalctl -u ssh -f menampilkan log secara langsung saat klien login.	[ ] Ya [ ] Tidak Respon Log Real-time: .....
8. Eksekusi Skrip Health Check	Berkas /usr/local/bin/tkj-check.sh dapat dieksekusi dan membuat log di /var/log/.	[ ] Ya [ ] Tidak Hak Akses Skrip: .....
9. Custom Systemd Unit File	Berkas /etc/systemd/system/tkj-monitor.service terdaftar dan berstatus enabled.	[ ] Ya [ ] Tidak Status Unit Service: .....
10. Uji Ketahanan Reboot	Setelah reboot, log	[ ] Ya [ ] Tidak

VM	/var/log/tkj-health-monitor.log otomatis terisi laporan baru.	Waktu Check di Log: .....
11. Cadangan Snapshot VM	Snapshot baru bernama Snapshot-02-Service-Management-Mastered berhasil dibuat.	[ ] Ya [ ] Tidak Waktu Snapshot: .....

N. TROUBLESHOOTING & PEMECAHAN MASALAH (TP 5.2.1.5)

Dalam mengelola daemon dan unit file, teknisi kerap dihadapkan pada kendala konfigurasi. Tabel berikut merangkum diagnosis dan solusi untuk pesan kesalahan umum di laboratorium:

Gejala / Pesan Error yang Muncul	Kemungkinan Penyebab Utama	Solusi Diagnostik & Langkah Perbaikan
Error: Failed to start service: Unit is masked.	Layanan yang ingin dinyalakan telah dikunci (masked) secara simbolik ke /dev/null oleh administrator sebelumnya untuk alasan keamanan.	1. Periksa status penguncian layanan: systemctl status <name> 2. Lepaskan kunci proteksi menggunakan perintah: sudo systemctl unmask <name> 3. Nyalakan kembali layanan: sudo systemctl start <name>
Error: Job for apache2.service failed... AH00072: make_sock: could not bind to address [::]:80	Terjadi bentrokan port jaringan (Port Conflict). Ada layanan lain (dalam kasus ini Nginx) yang telah menduduki Port 80 TCP terlebih dahulu.	1. Periksa proses penguasa port 80: sudo ss -tulpn | grep :80 2. Pilih salah satu solusi: matikan layanan penguasa port, atau ubah port layanan kedua. 3. Ubah port Apache di /etc/apache2/ports.conf menjadi Listen 8080, lalu restart.

Error: Warning: The unit file, source configuration file or drop-ins of <service> changed on disk. Run 'systemctl daemon-reload' to reload units.	Anda telah mengedit atau menambahkan berkas konfigurasi di /etc/systemd/system/, namun memori indeks internal daemon systemd belum diperbarui.	1. Jangan langsung melakukan start/restart pada layanan tersebut. 2. Eksekusi pembaruan indeks systemd: sudo systemctl daemon-reload 3. Setelah indeks dimuat ulang, jalankan perintah kontrol layanan yang Anda inginkan.
Layanan langsung mati dengan status Active: active (exited) saat di-start	Hal ini bukanlah error jika parameter pada unit file menggunakan Type=oneshot. Namun jika ini terjadi pada layanan daemon terus-menerus (seperti web server), berarti ada kesalahan script binding.	1. Periksa tipe layanan di dalam berkas .service. Jika bernilai Type=oneshot dan RemainAfterExit=yes, status exited adalah normal (tugas sukses dijalankan). 2. Jika terjadi pada layanan reguler, periksa log detailnya: sudo journalctl -u <service> -e
Error: Failed to enable unit: Unit file /etc/systemd/system/tkj-monitor.service has no [Install] section.	Berkas Custom Systemd Unit yang Anda buat tidak memiliki blok konfigurasi [Install] atau parameter WantedBy=, sehingga systemd tidak tahu pada runlevel apa layanan ini harus dipasang.	1. Buka kembali berkas unit Anda: sudo nano /etc/systemd/system/tkj-monitor.service 2. Tambahkan blok sintaks berikut pada bagian paling bawah berkas: [Install] WantedBy=multi-user.target 3. Simpan, lakukan daemon-reload, lalu ulangi perintah enable.
Kapasitas disk partisi root (/) penuh akibat membengkaknya berkas log sistem (/var/log/journal/)	Layanan server menghasilkan pesan error/log secara berlebihan (log spamming) dalam jangka waktu lama, dan	1. Periksa ukuran penggunaan disk oleh log systemd: journalctl --disk-usage 2. Bersihkan log lama

	batas penyimpanan log journald belum dibatasi.	dengan membatasi ukuran maksimalnya (misal sisakan 100 MB): sudo journalctl --vacuum-size=100M 3. Batasi permanen di /etc/systemd/journald.conf dengan mengatur SystemMaxUse=100M.

O. ANALISIS & STUDI KASUS TEKNIS

1. Analisis Evolusi Init System (systemd vs SysVinit): Pada era Linux lama, proses booting dilakukan secara sekuensial (satu per satu) menggunakan skrip SysVinit. Mengapa pendekatan sekuensial tersebut membuat waktu nyala server menjadi sangat lambat? Jelaskan bagaimana konsep Parallel Start-up dan Socket Activation pada systemd mampu mempercepat proses booting secara signifikan!

2. Komparasi Keamanan Layanan (disable vs mask): Dalam sebuah audit keamanan pusat data, seorang auditor menemukan bahwa layanan Telnet yang tidak aman hanya di-disable oleh teknisi, bukan di-mask. Mengapa auditor tersebut menyatakan sistem masih rentan? Berikan contoh skenario nyata bagaimana sebuah layanan yang berstatus disabled tetap bisa aktif tanpa disadari oleh administrator!

3. Analisis Resolusi Konflik Port Jaringan: Pada Langkah 4, kita memecahkan konflik port dengan memisahkan Apache2 ke port 8080 dan Nginx di port 80. Jika dalam arsitektur industri sebenarnya kita ingin agar kedua web server tersebut tetap dapat diakses oleh pengguna luar melalui satu domain dan tetap menggunakan Port 80, teknologi arsitektur jaringan apa yang harus kita terapkan? (Petunjuk: Kaitkan dengan konsep Reverse Proxy).

4. Analisis Keandalan Forensik Log (journald vs Plain Text Log): Utiliti journalctl membaca data log dari basis data biner (binary journal), sedangkan log tradisional disimpan dalam teks biasa di /var/log/syslog. Apa keunggulan log biner dalam menjaga integritas bukti forensik saat server mengalami serangan peretasan? Mengapa peretas lebih sulit memanipulasi berkas log biner milik systemd?

5. Konsep Watchdog & Self-Healing Service: Pada parameter berkas unit systemd, terdapat atribut Restart=always dan RestartSec=3s. Jelaskan bagaimana kombinasi kedua parameter tersebut dapat menciptakan mekanisme Self-Healing Server (pemulihan otomatis)! Apa yang akan terjadi pada layanan web server sekolah jika terjadi crash tiba-tiba saat pengumuman kelulusan siswa, bila parameter tersebut diaktifkan?

P. REFLEKSI PESERTA DIDIK

Aspek Refleksi Pembelajaran	Catatan Evaluasi Diri Peserta Didik
1. Kompetensi Teknis Baru yang Diperoleh	Tuliskan minimal 3 keahlian teknis baru yang Anda kuasai setelah menyelesaikan proyek manajemen layanan Linux ini: 1. .... 2. .... 3. ....
2. Kendala & Hambatan Selama Praktikum	Jelaskan kesulitan teknis, pesan error, atau kegagalan konfigurasi layanan yang Anda alami selama praktikum berlangsung: ....
3. Solusi & Langkah Perbaikan yang Diterapkan	Bagaimana cara Anda memecahkan masalah atau error teknis tersebut? (Sebutkan perintah diagnostik atau referensi log yang digunakan): ....

4. Rencana Pengembangan & Eksplorasi Lanjutan	Tuliskan rencana Anda untuk memperdalam keahlian automasi (misal: mencoba membuat Systemd Timer sebagai pengganti Cron, atau eksplorasi Reverse Proxy Nginx): ....

Q. RUBRIK PENILAIAN PROYEK (KKTP)

Penilaian hasil praktikum mandiri menggunakan rubrik penilaian berbasis proyek (Project-Based Learning Rubric) dengan skala 1–4, yang diturunkan dari Kriteria Ketercapaian Tujuan Pembelajaran (KKTP) Elemen ASJ Fase F (khususnya TP 5.2.1). Peserta didik dinyatakan Tercapai apabila indikator esensial terpenuhi dengan predikat minimal Baik (Skala 3).

Komponen Penilaian	Bobot	Perlu Bimbingan (1)	Cukup (2)	Baik / Tercapai (3)	Sangat Baik (4)
1. Analisis & Perencanaan Proyek	15%	Belum memahami arsitektur systemd; skema kontrol port dan layanan tidak disiapkan dengan baik.	Memahami konsep dasar PID 1, namun masih membutuhkan bimbingan dalam merencanakan pemetaan port.	Menjelaskan arsitektur systemd, merancang pemetaan port 80/8080 dengan tepat, dan menyiapkan skrip automasi.	Menganalisis keunggulan systemd dengan kritis, merancang topologi audit layanan yang sempurna, dan menyusun skema mitigasi error.
2. Kontrol Lifecycle & Proteksi	35%	Gagal mengoperasikan	Mampu menjalankan start/stop,	Menguasai seluruh siklus hidup	Sangat terampil mengontrol

		systemctl; salah membedakan restart/reload serta enable/mask.	namun gagal menerapkan teknik masking pada layanan BIND9.	layanan, sukses mempraktikkan reload tanpa putus koneksi, dan mengunci layanan (mask).	daemon, mampu memvalidasi perubahan PID, serta menerapkan proteksi berlapis pada server.
3. Resolusi Konflik & Custom Unit	25%	Tidak mampu mengatasi bentrokan port 80; gagal membuat Custom Systemd Unit File.	Mengubah port Apache setelah dibimbing intensif; skrip custom unit berhasil dibuat tetapi error saat eksekusi.	Berhasil menyelesaikan konflik Apache/Nginx (Port 80/8080); custom unit tkj-monitor.service berjalan aktif.	Menyelesaikan konflik port dengan cepat menggunakan audit socket ss, serta merakit custom unit yang rapi dan mematuh standar FHS.
4. Audit Log & Troubleshooting	15%	Tidak mampu membaca pesan log; bingung saat menghadapi error unit masked atau port crash.	Mampu membuka journalctl, tetapi lambat dalam menemukan baris kode penyebab error pada berkas konfigurasi.	Menggunakan journalctl (-u, -f, -xe) secara efektif untuk menemukan dan memperbaiki akar masalah mandiri.	Sangat mahir melakukan forensik log real-time, mengidentifikasi error dalam hitungan detik, dan mengelola kapasitas penyimpanan log.

5. Dokumentasi, SOP, & K3LH	10%	Tidak menyusun laporan kerja; melanggar SOP (mematikan layanan network via SSH jarak jauh).	Laporan kerja seadanya; masih perlu diingatkan untuk melakukan perintah systemctl daemon-reload.	Laporan teknis lengkap, mencatat logbook, mematuhi SOP keselamatan, dan sukses membuat cadangan Snapshot VM.	Laporan berstandar industri, dokumentasi sangat sistematis dan replikatif, menjaga kerapian lab, serta menjadi tutor sebaya.

R. FORMAT DAN SISTEMATIKA LAPORAN PRAKTIKUM

Peserta didik wajib menyusun laporan akhir praktikum mandiri dalam format cetak atau PDF dengan sistematika penulisan standar industri sebagai berikut:

1. HALAMAN SAMPUL / COVER: Judul Proyek (Manajemen Layanan dan Automasi Systemd Linux Server), Identitas Peserta Didik, NISN, Kelas, Nama Sekolah, dan Tahun Pelajaran.

2. BAB I: PENDAHULUAN: Latar Belakang Proyek (Urgensi manajemen layanan dan stabilitas daemon di pusat data sekolah), Tujuan Pembelajaran Operasional (TP 5.2.1), Batasan Masalah dan Spesifikasi Implementasi Layanan.

3. BAB II: LANDASAN TEORI: Arsitektur dan Evolusi Init System Linux (systemd dan Konsep PID 1), Perbedaan Siklus Hidup Layanan (start, stop, restart, reload, enable, disable, mask), Konsep Socket & Port Binding (Resolusi Bentrokan Port TCP/UDP), Sub-sistem Logging Linux (Journald, journalctl, dan Rsyslog).

4. BAB III: METODOLOGI & PERANCANGAN SISTEM: Alat dan Bahan yang Digunakan (Spesifikasi PC Host & Mesin Virtual Debian Server), Gambar Topologi Arsitektur Layanan dan Pemetaan Port Jaringan (Port 22, 80, 8080, 53), Diagram Alur (Flowchart) Prosedur Audit, Resolusi Konflik Port, dan Pembuatan Custom Service.

5. BAB IV: HASIL DAN PEMBAHASAN: Dokumentasi Langkah Kerja dan Konfigurasi (Disertai bukti tangkapan layar terminal Linux yang relevan), Bukti Resolusi Konflik Port 80 (Tangkapan layar hasil perintah ss -tulpn dan uji curl Nginx vs Apache2), Bukti Audit Log Real-time (journalctl -f) dan Hasil Eksekusi Skrip tkj-monitor.service, Pembahasan Analisis

Troubleshooting dan Temuan Kendala Teknis di Laboratorium, Jawaban Lengkap dari Pertanyaan Analisis & Studi Kasus Teknis (Bagian O).

6. BAB V: PENUTUP: Kesimpulan Hasil Implementasi Proyek Manajemen Layanan Linux Server, Saran dan Rekomendasi Pengembangan (misalnya penerapan Systemd Timer atau Reverse Proxy).

7. LAMPIRAN (APPENDIX): Salinan Teks Berkas Konfigurasi Lengkap (/etc/apache2/ports.conf, /etc/systemd/system/tkj-monitor.service, skrip /usr/local/bin/tkj-check.sh), Salinan Bukti Tampilan Antarmuka VirtualBox Manager yang menampilkan Snapshot-02, Lembar Hasil Praktikum (Bagian M) yang telah diverifikasi dan ditandatangani oleh Guru Mata Pelajaran.

Mengetahui/Menyetujui,
Kepala SMK Negeri 22 Jakarta

Jakarta, 17 Juli 2026
Guru Administrasi Sistem Jaringan

Lia Adawiyah, S.Pd.
NIP. 198004052008012031

Fernandez Eko NS
NIP. 198612162022211008