

LAPORAN RANCANG BANGUN SYSTEM DAEMON UNTUK MONITORING LOG SECARA OTOMATIS PADA LINUX

Dosen Pengampu :

Ir. Eko Prayetno, S.T., M.Eng



Disusun oleh :

1. Fachrezi Bachry (2401020010)
2. Willy Hadipermana (2401020019)
3. Haikal Fachry Akbar (2401020027)
4. Muhammad Faiz (241020040)

**FAKULTAS TEKNIK DAN TEKNOLOGI KEMARITIMAN
UNIVERTAS MARITIM RAJA ALI HAJI**

KATA PENGANTAR

Puji syukur kami haturkan kehadiran Allah Swt. yang telah melimpahkan rahmat dan hidayah-Nya sehingga kami bisa menyelesaikan karya ilmiah tentang “Rancang Bangun System Daemon Untuk Monitoring Log Secara Otomatis Pada Linux”.

Tidak lupa juga kami mengucapkan terima kasih kepada semua pihak yang telah turut memberikan kontribusi dalam penyusunan karya ilmiah ini. Tentunya, tidak akan bisa maksimal jika tidak mendapat dukungan dari berbagai pihak.

Sebagai penyusun, kami menyadari bahwa masih terdapat kekurangan, baik dari penyusunan maupun tata bahasa penyampaian dalam karya ilmiah ini. Oleh karena itu, kami dengan rendah hati menerima saran dan kritik dari pembaca agar kami dapat memperbaiki karya ilmiah ini.

Kami berharap semoga karya ilmiah yang kami susun ini memberikan manfaat dan juga inspirasi untuk pembaca.

DAFTAR ISI

DAFTAR ISI	ii
A. Abstrak.....	1
B. Pendahuluan	1
C. Rumusan Masalah & Tujuan Project.....	2
D. Perancangan Arsitektur.....	2
E. Implementasi	3
F. Pengeujian & Analisis	4
G. Kesimpulan	5
H. Daftar Pustaka	5
I. Lampiran(konfigurasi, kode, screenshot)	5
J. Bukti Kerja	6

A. Abstrak

Proyek ini bertujuan untuk merancang dan mengimplementasikan sebuah system daemon berbasis Linux yang berfungsi untuk melakukan pemantauan (monitoring) sistem secara real-time. Daemon ini secara berkala mengumpulkan informasi sistem seperti penggunaan CPU, RAM, disk, status pembaruan paket, riwayat backup, dan log sistem. Data tersebut kemudian diproses dan disajikan dalam bentuk dashboard web interaktif yang dapat diakses melalui browser. Dashboard ini menampilkan informasi dalam format visual yang mudah dipahami, seperti progress bar, tabel, dan notifikasi status. Implementasi menggunakan kombinasi bash scripting, Python, dan JavaScript dengan arsitektur client-server yang ringan. Hasil pengujian menunjukkan bahwa sistem mampu berjalan stabil, memperbarui data setiap 5 detik, dan memberikan informasi yang akurat tentang kondisi sistem.

B. Pendahuluan

Dalam lingkungan sistem operasi Linux, pemantauan sistem adalah aktivitas penting untuk memastikan stabilitas, keamanan, dan kinerja sistem. Administrasi sistem secara manual seringkali memakan waktu dan rentan terhadap human error. Oleh karena itu, diperlukan solusi otomatis yang dapat memantau berbagai aspek sistem secara real-time dan memberikan notifikasi atau laporan visual.

Proyek ini mengembangkan sebuah daemon yang berjalan di latar belakang (background service) untuk mengumpulkan data sistem secara periodik. Data tersebut kemudian diolah dan disajikan melalui antarmuka web yang responsif. Dengan demikian, administrator sistem dapat memantau kesehatan sistem dengan mudah tanpa perlu mengakses server melalui command line.

C. Rumusan Masalah & Tujuan Project

Rumusan Masalah:

1. Bagaimana cara mengotomatisasi pengumpulan data sistem (CPU, RAM, disk, update, backup, log) di Linux?
2. Bagaimana menyajikan data tersebut dalam bentuk visual yang mudah dipahami?
3. Bagaimana memastikan sistem dapat berjalan terus-menerus (daemon) dan diperbarui secara real-time?

Tujuan Project:

1. Membuat daemon yang berjalan di latar belakang untuk monitoring sistem.
2. Mengembangkan dashboard web yang menampilkan data sistem secara real-time.
3. Mengimplementasikan mekanisme pembaruan data otomatis dengan interval tertentu.
4. Menyediakan antarmuka yang intuitif dan informatif bagi administrator sistem.

D. Perancangan Arsitektur

Sistem ini dirancang dengan arsitektur modular yang terdiri dari tiga komponen utama:

1. Daemon / Service Backend
 - Berjalan sebagai systemd service di Linux.
 - Mengeksekusi script bash/Python secara berkala untuk mengumpulkan data.
 - Menyimpan data dalam format JSON (data.json).
2. Web Dashboard (Frontend)
 - Dibangun dengan HTML, CSS, dan JavaScript.
 - Mengambil data dari data.json menggunakan fetch() API.
 - Memperbarui tampilan secara real-time dengan interval 5 detik.
3. Data Flow

Daemon Service → Monitoring Scripts → Output Processing → data.json → Web Dashboard

4. Struktur File:

```
Integrated-System-Monitor-Backup-Daemon-Hybrid-Mode-main/
├── Integrated_daemon.py      # Main Python daemon script
├── README.md                 # Documentation
├── setup_complete.sh         # Installation script
├── config/
│   └── config.json           # Configuration file
├── output/
│   ├── integrated-monitor.service # Systemd service file
│   ├── backups/              # Backup files
│   │   └── backup_*.zip
│   ├── logs/                  # System logs
│   │   └── updates.log
│   └── web/                    # Frontend dashboard
│       ├── index.html
│       ├── css/
│       │   └── style.css
│       ├── js/
│       │   └── script.js
│       └── data.json           # Data output untuk dashboard
└── laporan/                   # Documentation/report
```

E. Implementasi

1. Daemon Service

- Script bash/Python dijadwalkan menggunakan cron atau systemd timer.
- Contoh tugas yang dijalankan:
 - Mengecek pembaruan paket (pacman -Qu untuk Arch Linux).
 - Memeriksa penggunaan CPU, RAM, disk.
 - Melakukan backup otomatis dan mencatat log.

2. Data Generator

Data sistem dikumpulkan dan disimpan dalam data.json dengan struktur:

```
{
  "last_check": "timestamp",
  "resources": { "cpu": 1.2, "ram": {...}, "disk": {...} },
  "updates": { "count": 19, "list": [...] },
  "backup": { "current": {...}, "history": [...] },
  "logs": { "update_logs": [...] }
}
```

3. Dashboard Web

- HTML (index.html): Struktur halaman dengan section untuk setiap jenis informasi.
- CSS (style.css): Styling untuk kartu, progress bar, tabel, dan layout responsif.
- JavaScript (script.js):
 - Fungsi fetchData() untuk mengambil data JSON.
 - Fungsi updateDashboard() untuk memperbarui UI.
 - Pembaruan otomatis dengan setInterval().

4. Konfigurasi

File config.json berisi pengaturan seperti:

```
{
  "log_files": ["/var/log/syslog", ...],
  "keywords": ["error", "warning", ...],
  "refresh_interval": 5
}
```

F. Pengeujian & Analisis

1. Pengujian Fungsional

- Daemon: Diuji dengan menjalankan service dan memverifikasi bahwa data JSON diperbarui sesuai interval.
- Dashboard: Diakses melalui browser lokal dan jaringan, responsif di berbagai ukuran layar.
- Data Update: Dipastikan UI memperbarui data setiap 5 detik tanpa perlu refresh manual.

2. Hasil Pengujian

- Sistem berhasil menampilkan:
 - Penggunaan CPU: 1.2%
 - Penggunaan RAM: 4.8 GB / 15.43 GB (31.1%)
 - Pembaruan tersedia: 19 paket
 - Status backup: sukses
 - Log update: 5 file log terbaru

3. Analisis Kinerja

- Interval 5 detik memberikan keseimbangan antara real-time update dan beban sistem.
- Dashboard ringan dan cepat karena hanya menggunakan vanilla JavaScript tanpa framework berat.
- Daemon berjalan dengan konsumsi resource minimal.

G. Kesimpulan

Proyek ini berhasil mengimplementasikan sistem daemon untuk monitoring log dan sistem di Linux, serta dashboard web untuk visualisasi data real-time. Sistem ini dapat membantu administrator dalam memantau kesehatan sistem dengan lebih efisien dan intuitif. Ke depan, sistem dapat dikembangkan dengan fitur notifikasi (email/telegram), autentikasi akses, dan grafik historis.

H. Daftar Pustaka

I. Lampiran(konfigurasi, kode, screenshot)

J. Bukti Kerja



