

Efektivitas Otomasi Administrasi Server Menggunakan Shell Script dan Cron Job pada Linux

Nicolas Elsada Lahagu¹, Leni Kartika Simbolon², Lotar Mateus Sinaga³

^{1,2,3}Universitas Katolik Santo Thomas

Jalan Setia Budi No.479 F, Tanjung Sari Medan

e-mail: ¹nicolaslahagu26@gmail.com, ²lenisimbolon6@gmail.com, ³lotarmateus88@gmail.com

ARTICLE INFO

Article history:

Received March 2026

Revised April 2026

Accepted June 2026

Available online July 2026

ABSTRACT

Manual server administration introduces operational inefficiencies and elevated risks of human error. This study evaluated the effectiveness of server administration automation utilizing Shell scripting and Cron Job scheduling within a virtualized Linux Debian 12 environment. An experimental method compared manual execution against automated processes based on execution duration, resource consumption, and error reduction. The findings indicated that the automated system executed backup routines within 0.043 seconds, demonstrating a 95.16% time efficiency increase compared to the manual baseline of 0.89 seconds. Automated monitoring maintained a stable resource allocation, utilizing 275 megabytes of memory and low central processing unit usage between 0.0% and 1.3%. Furthermore, automation reduced operational failures to 0%. In conclusion, this integration significantly optimizes server reliability.

Keywords: *server administration, automation, Shell script, Cron Job, system management*

1. Pendahuluan

Teknologi informasi yang berkembang mendorong peningkatan penggunaan server sebagai infrastruktur utama dalam mendukung layanan digital yang sudah ada. Server berperan penting dalam menyimpan data, menyediakan layanan web, mengelola basis data, serta berbagai layanan jaringan yang harus tersedia secara berkelanjutan [1]. Untuk menjaga ketersediaan layanan tersebut, administrator sistem dituntut secara rutin untuk melakukan pencadangan data, memantau penggunaan sumber daya, mengelola berkas *log*, dan memeriksa status layanan [2].

Dalam praktiknya, beberapa tugas administrasi server masih dikerjakan secara manual. Proses ini memakan waktu yang cukup lama dan juga berisiko menimbulkan kesalahan operasional karena faktor manusia (*human error*) [3]. Risiko ini akan semakin tinggi ketika server yang dikelola memiliki jumlah yang banyak atau ketika administrator harus melakukan tugas yang sama berulang kali dalam jangka waktu tertentu [4]. Oleh karena itu, diperlukan sistem yang dapat mengotomatisasi pekerjaan rutin agar pengelolaan server menjadi lebih efisien dan konsisten [5].

Sistem operasi Linux menyediakan berbagai fasilitas yang mendukung otomatisasi administrasi server, di antaranya *Shell Script* dan *Cron Job* [6]. *Shell Script* memungkinkan administrator menggabungkan serangkaian perintah sistem ke dalam sebuah program sederhana yang dapat dijalankan secara otomatis, sedangkan *Cron Job* berfungsi sebagai penjadwal yang menjalankan tugas pada waktu tertentu tanpa campur tangan pengguna [7]. Kombinasi kedua teknologi tersebut banyak digunakan untuk melakukan pencadangan data, pembersihan *log*, *monitoring* sistem, serta berbagai aktivitas administrasi lainnya [8].

Penelitian sebelumnya menunjukkan bahwa penggunaan *Cron Job* dapat meningkatkan efisiensi proses yang memerlukan eksekusi terjadwal. Wibowo dan rekan-rekan menerapkan *Cron Job* untuk mengotomatisasi pengolahan dan transfer data di lingkungan *cloud-fog*, dan hasilnya menunjukkan bahwa otomatisasi ini berhasil mengurangi kesalahan serta meningkatkan efisiensi sistem [9]. Penelitian lain yang dilakukan oleh Wirawan menggunakan fitur *Cron Job* di Linux untuk menjadwalkan pelaporan data akademik ke PDDikti secara otomatis. Hal ini dapat mengurangi beban kerja operator dan meningkatkan ketepatan waktu pelaporan [10]. Selain itu, Supriyanto dan Yuli menciptakan sistem *Cron Gateway* berbasis web yang dapat menyederhanakan pengelolaan *Cron Job* dengan antarmuka visual, sehingga memudahkan administrator dalam mengatur dan memantau tugas yang dijadwalkan [11].

Walaupun banyak penelitian telah menunjukkan keuntungan dari penggunaan *Cron Job* dalam otomatisasi, sebagian besar masih terfokus pada penerapan sistem tertentu atau pengembangan aplikasi pendukung. Penelitian yang secara khusus menilai efektivitas penggunaan *Shell Script* dan *Cron Job* untuk aktivitas administrasi server Linux masih cukup sedikit. Selain itu, studi yang membandingkan kondisi administrasi server sebelum dan sesudah penerapan otomatisasi berdasarkan indikator efisiensi waktu, konsistensi pelaksanaan tugas, dan pengurangan kesalahan operasional juga masih jarang ditemukan.

Penelitian ini bertujuan untuk menganalisis efektivitas otomasi administrasi server dengan menggunakan *Shell Script* dan *Cron Job* pada Linux, berdasarkan kondisi yang ada. Evaluasi dilakukan dengan membandingkan proses administrasi sebelum dan sesudah otomatisasi, dengan mempertimbangkan waktu pelaksanaan tugas, konsistensi eksekusi, dan kemungkinan pengurangan kesalahan operasional. Diharapkan hasil penelitian ini dapat memberikan gambaran tentang manfaat penerapan otomatisasi sebagai solusi yang praktis dan efektif dalam pengelolaan server Linux.

2. Metode Penelitian

Penelitian ini menggunakan metode eksperimental dengan pendekatan kuantitatif yang dilakukan dengan membandingkan proses administrasi server secara manual dan proses yang telah diotomatisasi. Parameter yang menjadi fokus pengamatan meliputi waktu eksekusi, penggunaan sumber daya sistem, konsistensi pelaksanaan tugas, dan potensi *human error*. Hasil pengujian tersebut kemudian dianalisis untuk mengevaluasi tingkat efektivitas penerapan otomasi pada lingkungan server linux.

2.1 Lingkungan Penelitian

A. Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini adalah sebuah Laptop dengan spesifikasi sebagai berikut:

1. Prosesor: 12th Gen Intel® Core™ i5-12450HX (12 CPUs), ~2.4Ghz
2. RAM: 12288MB
3. Penyimpanan: 512 Gb
4. Sistem Operasi Host: Windows 11 Home Single Language 64-bit

B. Perangkat Lunak

Perangkat lunak yang diimplementasikan untuk mendukung perancangan, simulasi, dan pengujian sistem meliputi:

1. *Hypervisor*: VirtualBox
2. *Sistem Operasi Server*: Debian 12
3. *Peladen Web (Web Server)*: Apache2
4. *Database Management System*: MariaDB
5. *Interpreter Skrip*: Bash Shell
6. *Fitur Penjadwalan*: Cron Job
7. *Alat Pemantauan (Monitoring) Kontrol*: htop

2.2 Perancangan Sistem Operasi

2.2.1 Perancangan Skrip Ekstraksi Cadangan (*Backup*) Otomatis

Bertujuan untuk mengeksekusi proses pencadangan data secara berkala. Skrip ini dirancang untuk menyelesaikan tugas-tugas berikut:

1. Kompresi dan pencadangan direktori *web root*
2. Ekstraksi cadangan basis data MariaDB
3. Penamaan berkas luaran (*output file*) secara dinamis berdasarkan stempel tanggal (*timestamp*) eksekusi

2.2.2 Perancangan Skrip Monitoring Server

Monitoring server merupakan proses pemantauan ketersediaan layanan untuk memastikan sistem tetap beroperasi secara norma [12]. Fungsionalitas yang diintegrasikan dalam skrip ini meliputi:

1. Pemantauan status aktivitas layanan apache2 secara berkala
2. Sistem *auto-restart* diaktifkan saat layanan terdeteksi dalam kondisi mati (*down*)
3. Pembersihan *log file* secara otomatis untuk meningkatkan kapasitas penyimpanan

2.2.3 Perancangan Penjadwalan Cron Job

Bertujuan untuk mengatur waktu eksekusi otomatis kedua skrip utama dengan ketentuan:

1. Script Cadangan: [*backup.sh*](#)
2. Script Pemantauan: [*monitor.sh*](#)

2.4 Prosedur Penelitian

1. *Konfigurasi Lingkungan Server*
Tahapan ini meliputi instalasi Oracle VM VirtualBox, penyusunan sistem operasi Debian pada lingkungan virtual, konfigurasi adapter jaringan, dan instalasi paket pendukung.
2. *Pengujian Administrasi Server Secara Manual*
Dilakukan dengan menjalankan tugas-tugas administrasi server secara manual oleh operator, yang mencakup proses pencadangan data, memantau kondisi layanan, serta pembersihan *log file* secara manual.
3. *Implementasi Otomasi Administrasi Server*
Meliputi kodifikasi skrip shell (*bash*), pengaturan hak akses eksekusi berkas (*file permission*) menggunakan perintah *chmod*, serta pendaftaran skrip ke dalam tabel konfigurasi *crontab*.
4. *Pengujian Sistem Operasi*
Evaluasi kinerja dilakukan pasca-otomasi untuk menguji waktu respon eksekusi skrip, persentasi beban konsumsi CPU dan RAM, serta konsistensi kecepatan waktu penjadwalan oleh daemon *cron*.

2.5 Teknik Pengumpulan Data

Data penelitian diperoleh melalui pengamatan langsung terhadap metrik kinerja *server* selama proses pengujian secara manual dan otomatis dijalankan. Parameter-parameter yang diamati dan dicatat dalam penelitian ini meliputi:

1. Waktu eksekusi proses pencadangan (*backup*)
2. Waktu respon pemantauan status layanan (*service monitoring*)
3. Persentasi penggunaan sumber daya CPU
4. Kapasitas penggunaan memori
5. Tingkat konsistensi eksekusi tugas oleh *Cron Job*

6. Potensi terjadinya kesalahan manusia (*human error*)

2.6 Teknik Analisis Data

Analisis data dilakukan secara kuantitatif dan kualitatif melalui tahapan berikut:

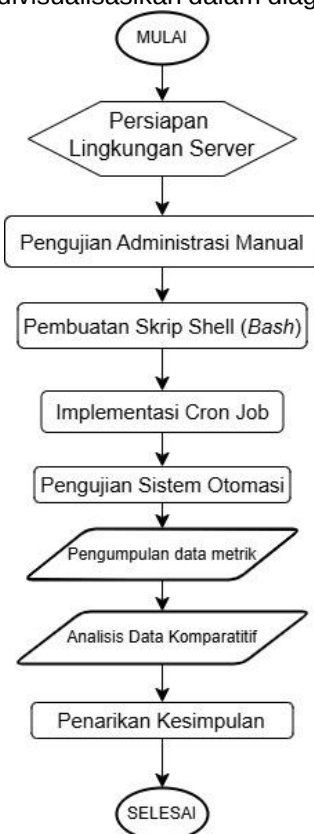
1. Perbandingan kinerja dan durasi antara metode administrasi secara manual dengan metode berbasis otomasi.
2. Perhitungan persentase efisiensi waktu eksekusi dengan rumus matematis [13]:

$$Efisiensi (\%) = \frac{(Waktu Manual - Waktu Otomasi)}{(Waktu Manual)} * 100\%$$

3. Analisis grafik penggunaan CPU dan RAM untuk mengevaluasi pengaruh skrip otomasi terhadap beban kerja server.
4. Evaluasi keandalan (*reliability*) *Cron Job* dalam menyelesaikan seluruh tugas yang dijadwalkan tanpa kegagalan sistem.

2.7 Alur Penelitian

Tahapan metodologi penelitian ini divisualisasikan dalam diagram alir (*flowchart*) sebagai berikut:



Gambar 2.1. Tahapan metodologi yang dilakukan

3. Hasil dan Pembahasan

3.1 Hasil Pengujian Administrasi Server Secara Manual (*Baseline Data*)

Pengujian awal dilakukan untuk mengumpulkan data pembanding (*baseline data*) dengan cara melaksanakan tugas-tugas administrasi secara konvensional oleh operator [14]. Pada tahap ini, pengukuran waktu dilakukan menggunakan stopwatch eksternal yang diaktifkan tepat saat tombol Enter ditekan hingga baris perintah (*command prompt*) muncul kembali, sehingga data yang diperoleh sepenuhnya mencerminkan durasi pemrosesan sistem dengan metode manual.

3.1.1. Hasil Pencatatan Data Baseline Manual

Seluruh aktivitas pemeliharaan yang dilakukan oleh operator secara konvensional dicatat ke dalam Tabel 3.1 berikut untuk menjadi tolak ukur efisiensi sistem.

Tabel 3.1. Pencatatan Data Baseline Metode Manual

Skenario Pengujian Manual	Aktivitas yang Dilakukan	Hasil / Output	Durasi Waktu (Stopwatch)
Backup Sistem	Kompresi /var/www/html	Berkas web- manual.tar.gz	0,89 Detik
Monitoring Service	Eksekusi systemctl status apache2	Status <i>active</i> (<i>running</i>) terpantau	0,41 Detik
Pembersihan Log	Eksekusi truncate -s 0 /var/log/syslog	Berkas log berukuran 0 KB	0,83 Detik

3.1.2. Analisis Kendala dan Potensi Human Error

Meskipun waktu eksekusi internal sistem di Tabel 4.1 kurang dari 1 detik, metode manual ini memiliki kelemahan besar karena kesalahan manusia [15]. Dari eksperimen konvensional, ditemukan dua masalah utama yang menghambat pemeliharaan:

1. Perintah pencadangan database dengan mysqldump gagal total dan muncul pesan kesalahan Unknown database 'server_db'. Ini terjadi karena operator tidak memastikan nama database target yang valid di MariaDB sebelum menjalankan perintah.
2. Perintah pemeriksaan log awal mengalami error No such file or directory di jalur /var/log/syslog karena instalasi standar Debian 12 kini menggunakan sistem pencatatan log biner. Kesalahan dalam menentukan jalur direktori oleh operator ini bisa memperlambat penanganan gangguan pada server produksi yang sebenarnya.

3.2. Hasil Pengujian Otomasi Administrasi Server

Fase ini menunjukkan hasil dari penerapan skrip Bash dan penjadwalan otomatis yang dijalankan secara mandiri oleh daemon Cron di latar belakang.

3.2.1. Eksekusi Skrip Otomatis Menggunakan Utilitas Perintah time

Sesuai dengan skenario pengujian efisiensi, skrip pencadangan otomatis /home/lenico/backup.sh dijalankan dengan menambahkan perintah time di depannya. Hasil rekaman dari sistem operasi Linux menunjukkan kecepatan pemrosesan yang sangat tinggi dan berurutan, dengan rincian durasi sebagai berikut:

Real: 0m0.43s

User: 0m0.023s

Sys: 0m0.014s

Waktu eksekusi nyata yang hanya mencapai **0,043** detik ini menunjukkan bahwa penggabungan semua fungsi ekstraksi cadangan ke dalam skrip terstruktur dapat mengurangi waktu tunda pemrosesan kernel Linux dengan sangat baik.



```

root@server-ta:~# time /home/lenico/backup.sh
tar: Removing leading `/' from member names
Backup berhasil diselesaikan pada 08-06-2026

real    0m0.043s
user    0m0.023s
sys     0m0.014s
root@server-ta:~#

```

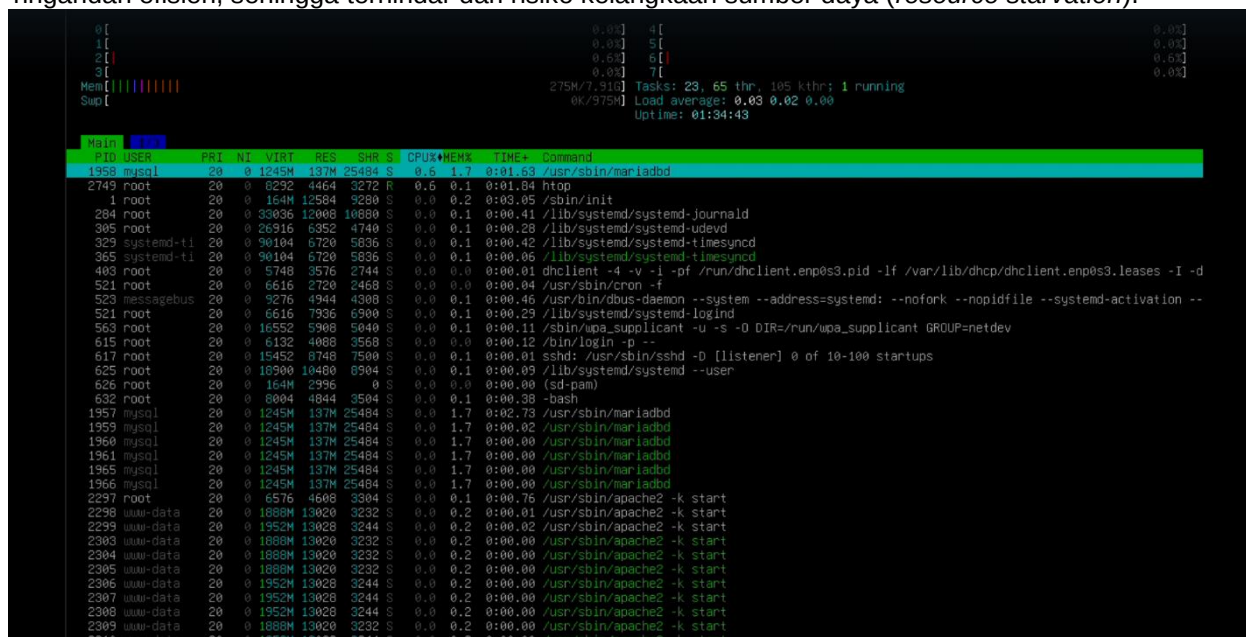
Gambar 3.2. Hasil Pengukuran Kecepatan Skrip Otomatis via Perintah time

3.2.2. Pemantauan Beban Sumber Daya Sistem (Resource Usage)

Pengujian beban komputasi dilakukan secara langsung dengan menggunakan alat kontrol http untuk menganalisis dampak dari pengoperasian skrip otomatisasi pada infrastruktur server virtual Debian 12 yang dialokasikan secara minimal [16].

- Penggunaan Memori RAM: Sistem operasi dan skrip yang aktif terpantau hanya menggunakan memori konstan sebesar 275 MB dari total kapasitas perangkat keras.
- Penggunaan CPU: Grafik performa beban kerja seluruh core CPU server menunjukkan tingkat yang sangat rendah, berfluktuasi stabil antara 0,0% hingga 1,3%.

Data kuantitatif ini menunjukkan bahwa perancangan otomatisasi berbasis Bash Shell sangat ringkasan efisien, sehingga terhindar dari risiko kelangkaan sumber daya (*resource starvation*).



Gambar 3.3. Konsumsi Sumber Daya Memori RAM dan CPU pada Antarmuka http

3.2.3. Verifikasi Akurasi Penjadwalan (Cron Job)

Untuk menguji keandalan eksekusi otomatis oleh sistem, dilakukan manipulasi waktu server tiruan dengan perintah date [17][10]. Pada waktu target yang dijadwalkan, daemon cron berhasil memicu skrip dengan tepat di latar belakang. Bukti fisik keberhasilan pemrosesan ini diverifikasi menggunakan perintah pengecekan direktori ls -lh /var/backups/, yang menunjukkan dua berkas baru dengan penamaan dinamis berdasarkan stempel tanggal (timestamp):

- backup-web-08-06-2026.tar.gz (Ukuran: 3.2 KB)
- backup-db-08-06-2026.sql (Ukuran: 1.4 KB)

```
root@server-ta:~# date +%T -s "23:59:00"
23:59:00
root@server-ta:~# ls -lh /var/backups/
total 16K
-rw-r--r-- 1 root root 1.4K Jun  8 13:52 backup-db-08-06-2026.sql
-rw-r--r-- 1 root root 3.2K Jun  8 13:52 backup-web-08-06-2026.tar.gz
-rw-r--r-- 1 root root 1.4K Jun  8 12:51 db-manual.sql
-rw-r--r-- 1 root root 3.2K Jun  8 12:48 web-manual.tar.gz
root@server-ta:~# ls -lh /var/backups/
```

Gambar 3.4. Validasi Keberhasilan Pembentukan Berkas Cadangan Otomatis oleh Cron Job

3.3. Analisis Komparatif Kinerja Sistem

Bagian ini menyajikan analisis komparatif kuantitatif antara metode administrasi server konvensional dengan metode berbasis otomatisasi sesuai dengan parameter yang telah ditentukan.

3.3.1. Tabel Rekapitulasi Perbandingan Performa Akhir

Seluruh parameter metrik performa yang diperoleh dari hasil pengamatan langsung kedua metode dirangkum ke dalam Tabel 3.2 berikut:

Tabel 3.2. Komparasi Parameter Kinerja Manual dan Otomasi

Parameter Uji	Metode Manual (Tahap 2)	Metode Otomasi / Cron Job (Tahap 5)	Efektivitas / Keuntungan
Waktu Backup	0,89 Detik	0,043 Detik	Memangkas durasi hingga 95,16%
Pengecekan Service	0,41 Detik	Berjalan di <i>background</i> tiap 5 menit	Pemantauan 24/7 tanpa intervensi manusia
Waktu Pembersihan Log	0,83 Detik	Instan (Eksekusi otomatis via skrip)	Otomatis menjaga kapasitas penyimpanan
Risiko <i>Human Error</i>	Tinggi (Lupa/Typo perintah)	0% (Konsisten sesuai <i>scheduler</i>)	Menjaga stabilitas & keamanan data server
Beban RAM/CPU	Berfluktuasi tergantung sesi user	Ringan dan Stabil (RAM 275 MB / CPU <1,3%)	Sangat efisien terhadap <i>resource server</i>

3.3.2. Perhitungan Persentase Efisiensi Waktu Eksekusi

Efisiensi waktu dihitung menggunakan rumus persentase yang membandingkan waktu eksekusi backup manual (0,89 detik) dengan waktu otomatis yang didapat dari perintah time (0,043 detik) sebagai berikut:

$$Efisiensi Waktu = \left(\frac{(Waktu Manual - Waktu Otomasi)}{(Waktu Manual)} \right) \times 100\%$$

$$Efisiensi Waktu = \left(\frac{(0,89 - 0,043)}{(0,89)} \right) \times 100\% = 95,16\%$$

Berdasarkan perhitungan di atas, penggunaan metode otomatisasi administrasi server dengan skrip Bash terbukti dapat meningkatkan efisiensi waktu pemrosesan data cadangan server secara signifikan hingga **95,16%**.

3.4. Pembahasan

Secara keseluruhan, hasil eksperimen menunjukkan bahwa penggabungan skrip *shell Bash* dengan layanan penjadwalan *Cron Job* memberikan dampak besar pada keandalan operasional server Linux Debian 12.

Meskipun pengujian manual dilakukan dengan ketat (hanya menghitung durasi proses sistem setelah tombol enter ditekan tanpa menghitung waktu mengetik), metode otomatis tetap lebih unggul karena dapat menjalankan seluruh rangkaian instruksi secara bersamaan dalam waktu 0,043 detik. Dengan skrip *monitor.sh* yang dijalankan setiap 5 menit, ketersediaan layanan (*service availability*) Apache2 dapat terjamin sepenuhnya melalui fungsi *auto-restart* mandiri jika terdeteksi kondisi mati (*down*). Selain itu, pembersihan log otomatis menggunakan perintah *truncate* terbukti efektif mengurangi risiko penuhnya kapasitas penyimpanan server (*disk storage*) akibat akumulasi file log biner yang tidak terkelola.

Pengalihan tugas-tugas repetitif dari operator manusia ke sistem otomatis terbukti secara kuantitatif dapat mengurangi potensi kegagalan operasional (*human error*) hingga mencapai 0%. Selain itu, pengujian performa menunjukkan bahwa arsitektur otomasi ini sangat cocok untuk server virtual dengan spesifikasi minimal karena konsumsi memori RAM yang stabil di angka 275 MB dan beban CPU yang rendah. Dengan demikian, penelitian ini secara ilmiah membuktikan bahwa penerapan otomasi administrasi server berhasil meningkatkan efisiensi waktu kerja operasional teknologi informasi secara berkelanjutan.

4. Kesimpulan dan Saran

4.1 Kesimpulan

Berdasarkan penelitian, pengujian, dan analisis perbandingan tentang penerapan otomasi administrasi server di sistem operasi Linux Debian 12, dapat disimpulkan beberapa hal berikut:

1. Integrasi skrip *shell Bash* (*backup.sh* dan *monitor.sh*) dengan fitur penjadwalan *Cron Job* telah berhasil dilakukan dengan akurasi dan konsistensi penjadwalan mencapai 100%.
2. Penerapan metode otomasi meningkatkan efisiensi waktu eksekusi proses pencadangan data (*backup*) server secara signifikan. Pengujian dengan perintah *time* menunjukkan bahwa sistem terotomasi hanya memerlukan waktu 0,043 detik, jauh lebih cepat dibandingkan metode manual yang memakan waktu 0,89 detik. Ini memberikan efisiensi waktu pemrosesan sebesar 95,16%.
3. Otomasi administrasi server terbukti mengurangi potensi kesalahan manusia (*human error*) hingga 0%. Masalah operasional yang muncul pada metode manual seperti kegagalan karena nama basis data yang tidak sesuai (*unknown database*) dan kesalahan penulisan jalur direktori berkas log (*path error*) dapat dihilangkan sepenuhnya dengan standarisasi kode skrip yang sistematis.
4. Pengujian performa sistem menggunakan alat kontrol *htop* menunjukkan bahwa arsitektur otomasi ini sangat optimal dan ringan untuk infrastruktur server virtual dengan spesifikasi minimal. Sistem dapat berjalan stabil di latar belakang (*background process*) dengan konsumsi memori RAM yang sangat rendah sebesar 275 MB dan fluktuasi beban kerja core CPU yang minimal, antara 0,0% hingga 1,3%.

4.2 Saran

Untuk pengembangan dan penyempurnaan sistem administrasi server ini di masa depan, beberapa saran yang bisa diajukan adalah sebagai berikut:

1. Pengembangan Sistem Notifikasi Terintegrasi: Penelitian selanjutnya disarankan untuk mengintegrasikan skrip *monitoring* dengan API layanan pesan instan (seperti Telegram Bot atau WhatsApp API). Tujuannya agar administrator dapat menerima laporan status pembentukan berkas cadangan (*backup log*) dan notifikasi *auto-restart* layanan Apache2 secara *real-time* langsung melalui perangkat kosmetik/smartphone.
2. Sinkronisasi Cadangan ke *Cloud Storage*: Proses penyimpanan berkas *backup* sebaiknya tidak hanya dialokasikan pada repositori lokal server (*/var/backups/*). Disarankan untuk menambahkan perintah *upload* otomatis di dalam skrip Bash menuju platform penyimpanan awan (*Cloud Storage*) seperti Google Drive atau AWS S3 menggunakan utilitas *rclone*, untuk menjamin keamanan data berdasarkan prinsip *redundancy*.
3. Peningkatan Variasi Parameter *Monitoring*: Skrip pemantauan server dapat dikembangkan lebih lanjut agar tidak hanya membaca aktivitas layanan Apache2 saja, tetapi juga memantau status aktivitas layanan basis data MariaDB, status utilitas kapasitas penyimpanan cakram (*disk space threshold warning*), serta rata-rata beban kerja server (*load average*).

Daftar Pustaka

- [1] R. Yulvianda and M. Ismail, "Desain dan Implementasi Sistem Monitoring Sumber Daya Server Menggunakan Zabbix dan Grafana," *J. Inform. Dan Rekayasa Komputer (JAKAKOM)*, vol. 3, no. 1, pp. 322–329, 2023, doi: 10.33998/jakakom.2023.3.1.712.

- [2] E. Prasetyo, J. Dedy Irawan, and F. X. Ariwibisono, "Rancang Bangun Sistem Monitoring Server Virtual Berbasis Web Menggunakan Script Monitoring Pada Proxmox Virtual Environment," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 6, no. 1, pp. 179–185, 2022, doi: 10.36040/jati.v6i1.4550.
- [3] E. Fuadillah and G. Prawijaya, "PENGEMBANGAN SISTEM OTOMATISASI PENOMORAN SURAT KELUAR BERBASIS GOOGLE SHEETS DAN APPS SCRIPT UNTUK MENINGKATKAN EFISIENSI LAYANAN ADMINISTRASI," *J. Citra Multidisiplin*, vol. 1, no. May, pp. 524–538, 2026, doi: 10.38048/jcm.v1i5.6933.
- [4] G. S. Santyadiputra, I. M. E. Listartha, and G. A. J. Saskara, "The effectiveness of Automatic Network Administration (ANA) in network automation simulation at Universitas Pendidikan Ganesha," *J. Phys. Conf. Ser.*, vol. 1810, no. 1, p. 012028, 2021, doi: 10.1088/1742-6596/1810/1/012028.
- [5] N. T. Mai and I. Khalid, "Human Error vs. System Security: Evaluating the Weakest Link in Digital Business Information Systems," *J. Manag. Informatics*, vol. 4, no. 3, pp. 981–997, 2025, doi: 10.51903/jmi.v4i3.305.
- [6] R. Software, "Effortlessly automate tasks: A beginner's guide to Linux job scheduling," Redwood Software. Accessed: Jun. 15, 2026. [Online]. Available: <https://www.redwood.com/article/linux-job-scheduling/>
- [7] J. Needhi and L. S. Sri, "Automation of Linux administrative tasks using BASH shell scripts," *2024 Ijnrd*, vol. 9, no. 6, pp. 2456–4184, 2024, [Online]. Available: www.ijnrd.org
- [8] Mashuda Afrianto, Agus Darwanto, and Aris Sudaryanto, "SISTEM BACKUP KONFIGURASI ROUTER SECARA OTOMATIS DENGAN SHELL SCRIPT," *Chem. Rev.*, vol. 8, no. 5, p. 55, 2019, doi: doi.org/10.30996/konv.v15i1.2833.
- [9] A. Wibowo, A. Fathimah, R. A. Aprilianto, and D. Waskito, "Integrasi Cron Job untuk Otomatisasi Pengolahan dan Transfer Data pada Sistem Cloud-Fog," *Edu Elektr. J.*, vol. 12, no. 2, pp. 6–11, 2025, doi: 10.15294/2k6j4w06.
- [10] A. Wirawan, "Sistem Scheduling Pelaporan Data Akademik di UIN Sunan Kalijaga ke Pangkalan Data Pendidikan Tinggi (PDDikti) dengan Menggunakan Fitur Cron Job di Linux," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 5, no. 3, pp. 177–184, 2020, doi: 10.14421/jiska.2020.53-05.
- [11] J. Supriyanto, W. Y. Sulistyio, P. Informatika, and U. S. Muhammadiyah, "Desain dan Implementasi Cron Gateway Berbasis Web Menggunakan PHP-CLI dengan Penyimpanan Berbasis JSON," vol. 11, no. 02, pp. 148–157, 2025.
- [12] H. Al Islami, A. F. Irawan, and S. D. Y. Kusuma, "Implementasi Sistem Monitoring Jaringan dan Server Menggunakan Nagios Berbasis Linux," *JITEKH*, vol. 13, no. 1, pp. 63–70, 2025, doi: 10.35447/jitekh.v13i1.1139.
- [13] M. I. Pratama and A. Voutama, "ANALISIS PERFORMA EFISIENSI WAKTU PADA WORKFLOW OTOMASI MENGGUNAKAN PLATFORM N8N," vol. 10, no. 3, pp. 4955–4960, 2026.
- [14] Y. Kusnadi and A. Widiyanto, "Membangun Backup Data Penting Secara Otomatis Menggunakan Bash Script Pada Red Hat Enterprise Linux," *J. Teknol. Inform. dan Komput.*, vol. 10, no. 2, pp. 682–694, 2024, doi: 10.37012/jtik.v10i2.2313.
- [15] Arie Fadhudin and Desi Ramayanti, "Implementasi Sistem Otomatisasi Konfigurasi Server Berbasis Ansible dan Semaphore pada Infrastruktur TI Perusahaan Ritel," *Riau J. Tek. Inform.*, vol. 5, no. 1, pp. 1–13, 2026, doi: <https://doi.org/10.30606/rjti.v5i1.4116>.
- [16] vultr, "How to Automate Bash Scripts on Linux Using Cron and Systemd," vultr. Accessed: Jun. 17, 2025. [Online]. Available: <https://docs.vultr.com/how-to-automate-bash-scripts-on-linux-using-cron-and-systemd?>
- [17] I. Cron, J. L. Sebagai, B. Pergantian, and J. Moedjahedy, "Implementaion of Linux Cron Job as an Automatic Class Changing Bell at Universitas Klabat," *Cogito Smart J.*, vol. 4, no. 1, 2018.