

**ANALISIS EFISIENSI PENGGUNAAN *AUTOMASI* DALAM
INFRASTRUKTUR PENGEMBANGAN APLIKASI PADA
PRIVATE CLOUD MENGGUNAKAN METODE *T-TEST***

SKRIPSI

Diajukan untuk Melengkapi Salah Satu Syarat

Memperoleh Gelar Sarjana Komputer



OLEH :

RIZKI PUTRA RAMADHAN

09011182126024

PROGRAM STUDI SISTEM KOMPUTER

JURUSAN SISTEM KOMPUTER

FAKULTAS ILMU KOMPUTER

UNIVERSITAS SRIWIJAYA

2025

**ANALISIS EFISIENSI PENGGUNAAN *AUTOMASI* DALAM
INFRASTRUKTUR PENGEMBANGAN APLIKASI PADA
PRIVATE CLOUD MENGGUNAKAN METODE *T-TEST***

SKRIPSI

Disajikan untuk Melengkapi Salah Satu Syarat

Memperoleh Gelar Sarjana Komputer



OLEH :

RIZKI PUTRA RAMADHAN

0901118212624

JURUSAN SISTEM KOMPUTER

FAKULTAS ILMU KOMPUTER

UNIVERSITAS SRIWIJAYA

2025

HALAMAN PENGESAHAN

SKRIPSI

ANALISIS EFISIENSI PENGGUNAAN AUTOMASI DALAM INFRASTRUKTUR PENGEMBANGAN APLIKASI PADA PRIVATE CLOUD MENGGUNAKAN METODE T-TEST

Sebagai salah satu syarat untuk penyelesaian studi di

Program Studi SI Sistem Komputer

Oleh:

RIZKI PUTRA RAMADHAN

09011182126024

Pembimbing 1 : Kemahyanto Exaudi, S.Kom., M.T
NIP. 198405252023211018

Pembimbing 2 : Adi Hermansyah, S.Kom., M.T
NIP. 198904302024211001

Mengetahui

Ketua Jurusan Sistem Komputer



Dr. Ir. Sukemi, M.T
196612032006041001

AUTHENTICATION PAGE

SKRIPSI

ANALYSIS OF AUTOMATION EFFICIENCY IN APPLICATION DEVELOPMENT INFRASTRUCTURE ON PRIVATE CLOUD USING T-TEST METHOD

Submitted as one of the requirements for completing the
Bachelor's Degree Program in Computer Systems

By:

RIZKI PUTRA RAMADHAN

09011182126024

Advisor 1 : Kemahyanto Exaudi, S.Kom., M.T
NIP. 198405252023211018

Advisor 2 : Adi Hermansyah, S.Kom., M.T
NIP. 198904302024211001

Approved by,

Head of Computer System Departement



Dr. Ir. Sukemi, M.T
196612032006041001

HALAMAN PERSETUJUAN

Telah diuji dan lulus pada:

Hari : Jumat
Tanggal : 25 Juli 2025

Tim Penguji

Ketua : Prof. Ir. Deris Stlawan, M.T., Ph.D.

Penguji : Dr. Ir. Ahmad Heryanto, M.T.

Pembimbing 1 : Kemahyanto Exaudi, S.Kom., M.T

Pembimbing 2 : Adi Hermansyah, S.Kom., M.T

Mengetahui, 26/8/25
Ketua Jurusan Sistem Komputer



LEMBAR PERNYATAAN

Yang bertanda tangan di bawah ini :

Nama : Rizki Putra Ramadhan

Nim : 09011182126024

Judul : Analisis Efisiensi Penggunaan Automasi dalam Infrastruktur
Pengembangan Aplikasi pada Private Cloud menggunakan Metode T-
Test

Hasil pemeriksaan *Software* Turnitin: 1%

Menyatakan bahwa laporan skripsi saya merupakan hasil karya sendiri dan bukan hasil penjiplakan atau plagiat. Apabila ditemukan unsur penjiplakan atau plagiat dalam laporan tugas akhir ini, maka saya bersedia menerima sanksi akademik dari Universitas Sriwijaya.

Demikian, pernyataan ini saya buat dalam keadaan sadar dan tanpa paksaan dari siapapun.



Indralaya, 27 Agustus 2025



Rizki Putra Ramadhan
NIM. 09011182126024

HALAMAN PERSEMBAHAN

1. Untuk Ayah dan Mama, dengan rasa hormat dan cinta yang tak terhingga, saya persembahkan karya ini dengan sepenuh hati. Terima kasih atas setiap doa yang menguatkan, setiap dukungan yang setia mengiringi, dan setiap pengorbanan yang tak pernah kalian ungkapkan. Tanpa kalian, diriku ini tidak akan bisa menempuh pendidikan sampai setinggi ini. Semoga pencapaian ini menjadi wujud kecil dari rasa terima kasih saya atas segalanya yang telah kalian berikan, dan menjadi kebahagiaan yang turut kalian rasakan.
2. Untuk Adikku, terima kasih atas kebersamaan, dukungan, dan semangat yang selalu kau berikan. Perjalanan ini lebih ringan karena ada adikku yang selalu menyemangati kakakmu ini, dalam suka maupun duka. Kehadiranmu menjadi penguat di setiap langkah, dan sumber keceriaan di tengah lelah. Skripsi ini aku persembahkan juga untukmu, bagian penting dalam cerita ini.

MOTTO

“Lakukan apa yang kau mau sekarang. Saat hatimu bergerak, jangan kau larang.
Hidup ini tak ada artinya, Maka kau bebas mengarang maknanya seorang.”

(Hindia)

“I want to be a stronger person.”

(Thorfinn)

“Melangkahlah, melangkahlah, melangkahlah walau hanya selangkah. Kita tidak bisa mengubah masa lalu, namun kita selalu bisa memilih untuk menjadi pribadi yang lebih baik hari ini.”

KATA PENGANTAR

Puji dan syukur atas kehadiran Allah SWT yang telah melimpahkan rahmat, kasih sayang dan karunia-Nya, sehingga penulis diberikan kekuatan serta kemudahan agar dapat menyelesaikan Skripsi dengan judul “Analisis Efisiensi Penggunaan Automasi Dalam Infrastruktur Pengembangan Aplikasi Pada Private Cloud Menggunakan Metode T-Test”.

Skripsi ini disusun untuk menganalisis dan membandingkan efisiensi antara metode manual dan automasi dalam proses provisioning infrastruktur pengembangan aplikasi berbasis Proxmox Virtual Environment (PVE). Automasi dalam penelitian ini diterapkan menggunakan Terraform dan Ansible, dengan evaluasi performa didasarkan pada parameter teknis seperti penggunaan CPU, RAM, ruang penyimpanan, aktivitas I/O, serta durasi provisioning. Penelitian ini menggunakan pendekatan kuantitatif dan metode uji statistik t-test untuk mengukur signifikansi perbedaan antara kedua pendekatan tersebut.

Selama penulisan Skripsi ini, penulis banyak mendapatkan ide, bantuan, serta saran dari semua pihak, baik secara langsung maupun tak langsung. Oleh karena itu, pada kesempatan ini penulis menyampaikan ucapan terima kasih kepada:

1. Allah SWT yang telah melimpahkan berkah serta nikmat kesehatan dan kesempatan kepada penulis dalam menyelesaikan proposal tugas akhir.
2. Kedua orang tua dan adik saya. Ayah, Mama, Adek Dhita dan Keluarga yang selalu memberikan semua dukungan kepada saya dalam setiap usaha saya termasuk pembuatan tugas akhir ini.
3. Bapak Prof. Dr. Erwin, S.Si., M.Si, selaku Dekan Fakultas Ilmu Komputer Universitas Sriwijaya.
4. Bapak Dr. Ir. Sukemi, M.T., selaku Ketua Jurusan Sistem Komputer Fakultas Ilmu Komputer Universitas Sriwijaya.
5. Bapak Kemahyanto Exaudi, S.Kom., M.T selaku Dosen Pembimbing 1 Tugas Akhir yang telah berkenan meluangkan waktunya guna membimbing,

memberikan saran dan motivasi serta bimbingan terbaik untuk penulis dalam menyelesaikan Tugas Akhir ini.

6. Bapak Adi Hermansyah, S.Kom., M.T selaku Dosen Pembimbing 2 Tugas Akhir yang telah berkenan meluangkan waktunya guna membimbing, memberikan saran dan motivasi serta bimbingan terbaik untuk penulis dalam menyelesaikan Tugas Akhir ini.
7. Bapak Prof. Dr. Erwin, S.Si., M.Si, selaku Dosen Pembimbing Akademik penulis pada Program Studi Sistem Komputer.
8. Bapak Dr. Ir. Ahmad Heryanto, M.T., yang telah memberikan Ide Proyek Akhir ini kepada Penulis dan sudah membantu memberikan pemecahan masalah.
9. Kak Angga selaku admin Jurusan Sistem Komputer Reguler yang telah membantu mengurus seluruh berkas.
10. Sahabat-sahabat penulis, INGPO, yang selalu kebersamai penulis dalam empat tahun ini serta membagi tawa dan duka bersama-sama.
11. Anggota Tim Cloud yaitu: Bona, Quddus, Ikhlash dan Tiar yang sudah berjuang bersama dengan penulis dari awal sampai akhir.
12. Dan semua pihak yang telah membantu dalam penulisan tugas akhir.

Penulis menyadari bahwa laporan ini belum mencapai kata sempurna. Untuk itu kritik dan saran yang membangun serta kemakluman sangat diperlukan sehingga penulis dapat berkembang untuk kedepannya. Akhir kata penulis berharap, semoga proposal tugas akhir ini bermanfaat dan berguna bagi banyak pihak baik dari yang terlibat, pembaca, serta penulis sendiri.

Indralaya, 27 Agustus 2025



Rizki Putra Ramadhan
NIM. 09011182126024

Efficiency Analysis of Automation in Application Development Infrastructure on Private Cloud Using T-Test Method

Rizki Putra Ramadhan (09011182126024)

Dept. of Computer Systems, Faculty of Computer Science, Universitas Sriwijaya

Email: pramadhan087@gmail.com

ABSTRACT

This study analyzes the efficiency differences between manual and automated provisioning methods in application infrastructure within a private cloud using Proxmox Virtual Environment (PVE). Automation was implemented using Terraform for provisioning and Ansible for configuration, while manual provisioning used the Proxmox interface. Performance was evaluated by monitoring CPU, RAM, disk usage, I/O activity, and provisioning duration, with data collected via Prometheus and visualized using Grafana. A quantitative experimental approach was used, with statistical analysis through an independent two-sample t-test. Results indicate that automation consistently offers better efficiency in provisioning speed and resource utilization, while also minimizing configuration errors. These findings support the adoption of automation tools to improve the reliability and availability of private cloud services.

Keywords: Infrastructure Automation, Private Cloud, Proxmox, Terraform, Ansible, T-Test, System Efficiency.

Analisis Efisiensi Penggunaan Automasi Dalam Infrastruktur Pengembangan Aplikasi Pada *Private Cloud* Menggunakan Metode *T-Test*

Rizki Putra Ramadhan (09011182126024)

Jurusan Sistem Komputer, Fakultas Ilmu Komputer, Universitas Sriwijaya

Email : pramadhan087@gmail.com

ABSTRAK

Dalam era digital yang terus berkembang, efisiensi dalam penyediaan dan pengelolaan infrastruktur aplikasi menjadi aspek krusial, terutama pada lingkungan *private cloud*. Penelitian ini bertujuan untuk menganalisis perbedaan efisiensi antara metode manual dan automasi dalam proses *provisioning* infrastruktur pengembangan aplikasi berbasis *Proxmox Virtual Environment (PVE)*. Automasi dilakukan menggunakan alat *Terraform* untuk *provisioning* dan *Ansible* untuk konfigurasi layanan, sedangkan proses manual dilakukan secara konvensional melalui antarmuka *Proxmox*. Evaluasi performa dilakukan dengan memantau parameter seperti penggunaan *CPU*, *RAM*, *disk space*, aktivitas *I/O*, serta durasi *provisioning*, yang dikumpulkan menggunakan *Prometheus* dan divisualisasikan melalui *Grafana*. Penelitian ini menerapkan pendekatan kuantitatif dengan metode eksperimen dan analisis statistik menggunakan uji *t-test* dua sampel independen guna menentukan signifikansi perbedaan antara kedua metode. Hasil penelitian menunjukkan bahwa metode automasi secara konsisten memberikan efisiensi yang lebih tinggi dalam hal kecepatan *provisioning* dan pemanfaatan sumber daya, serta mengurangi kesalahan konfigurasi. Temuan ini menegaskan bahwa integrasi alat automasi pada *private cloud* dapat meningkatkan keandalan dan ketersediaan layanan secara signifikan.

Kata Kunci : Automasi Infrastruktur, *Private Cloud*, *Proxmox*, *Terraform*, *Ansible*, *t-test*, Efisiensi Sistem.

DAFTAR ISI

HALAMAN PENGESAHAN.....	i
AUTHENTICATION PAGE	ii
HALAMAN PERSETUJUAN	iii
LEMBAR PERNYATAAN	iv
HALAMAN PERSEMBAHAN	v
KATA PENGANTAR.....	vi
ABSTRACT	viii
ABSTRAK	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	4
1.3 Batasan Masalah	4
1.4 Tujuan	5
1.5 Manfaat	5
1.6 Sistematika Penulisan	6
BAB II TINJAUAN PUSTAKA	8
2.1 Penelitian Terdahulu	8
2.2 <i>Cloud Computing</i>	44
2.2.1 <i>Private Cloud</i>	44
2.2.2 <i>Public Cloud</i>	45
2.2.3 <i>Proxmox Virtual Server</i>	45
2.3 <i>Infrastructure as a Code</i>	46
2.3.1 <i>Terraform</i>	47

2.3.2 <i>Ansible</i>	49
2.4 Parameter Evaluasi.....	51
2.4 Metode <i>t-test</i>	52
BAB III METODOLOGI PENELITIAN	56
3.1 Kerangka Kerja Penelitian	56
3.2 Perancangan <i>Topologi</i> Jaringan	60
3.3 Perancangan Skenario Pengujian	62
3.4 Implementasi <i>Topologi</i> dan Skenario Pengujian	65
3.4.1 Lingkungan Implementasi	65
3.4.2 Konfigurasi <i>Automation</i> pada tiap Skenario	68
3.5 Metode Pengumpulan Data.....	72
3.5.2 Persiapan	72
3.5.3 Pengumpulan Data Pembuatan VM.....	75
3.5.4 Pengumpulan Data Layanan <i>Web Server</i>	77
3.5.5 Pengumpulan Data Layanan <i>Monitoring</i>	79
3.5.6 Pengumpulan Data Layanan <i>Docking</i>	81
3.5.7 Pengumpulan Data Layanan <i>Cloud Storage</i>	83
3.6 Analisis Data	85
3.6.1 Persiapan Data	87
3.6.2 Uji <i>Normalitas</i>	94
3.6.3 Uji Homogenitas	98
3.6.4 Uji <i>t-test</i> Dua Sampel Independen	101
BAB IV HASIL DAN ANALISIS.....	120
4.1 Pendahuluan.....	120
4.2 Analisis Hasil Uji <i>Normalitas</i>	120
4.2.1 Penggunaan CPU dan RAM	121
4.2.2 Penggunaan <i>Disk Space</i>	121
4.2.3 Aktivitas <i>Read/Write Disk</i> dan I/O	122
4.2.4 Durasi <i>Provisioning</i>	122

4.3 Analisis Hasil Uji Homogenitas.....	123
4.3.1 Penggunaan CPU dan RAM	123
4.3.2 Penggunaan <i>Disk Space</i>	124
4.3.3 Aktivitas <i>Read/Write Disk</i> dan I/O	124
4.3.4 Durasi <i>Provisioning</i>	125
4.4 Analisis Hasil Uji <i>t-test</i> Dua Sampel Independen.....	126
4.4.1 Penggunaan CPU dan RAM	126
4.4.2 Penggunaan <i>Disk Space</i>	127
4.4.3 Aktivitas <i>Read/Write Disk</i> dan I/O	127
4.4.4 Durasi <i>Provisioning</i>	128
4.5 Analisis Penggunaan Sumber Daya	128
4.5.1 Penggunaan CPU	128
4.5.2 Penggunaan RAM.....	132
4.5.3 Penggunaan <i>Disk Space</i>	136
4.5.4 Aktivitas <i>Read</i> dan <i>Write Disk</i>	139
4.5.5 Aktivitas <i>Read</i> dan <i>Write I.O</i>	144
4.5.6 Durasi <i>Provisioning</i>	148
BAB V KESIMPULAN DAN SARAN	153
5.1 Kesimpulan	153
5.2 Saran	154
DAFTAR PUSTAKA.....	155

DAFTAR GAMBAR

Gambar 2.1 Struktur Kode Terraform	48
Gambar 2.2 Struktur Playbook Ansible	50
Gambar 3.1 Kerangka Kerja Penelitian.....	57
Gambar 3.2 Alur Model Penelitian	59
Gambar 3.3 Topologi Jaringan Penelitian	61
Gambar 3.4 Tampilan WEB GUI Proxmox	66
Gambar 3.5 Template VM.....	66
Gambar 3.6 Tampilan WEB GUI VSphere.....	67
Gambar 3.7 Dashboard Grafana	67
Gambar 3.8 Alur Kerja Automasi dan Tapping	69
Gambar 3.9 Contoh Code Terraform.....	70
Gambar 3.10 Contoh Code Ansible	71
Gambar 3.11 Flowchart Pengumpulan Data	73
Gambar 3.12 Terraform dijalankan dari VM Utama.....	74
Gambar 3.13 VM langsung dibentuk pada Node Proxmox	74
Gambar 3.14 Ansible dijalankan pada VM Hasil Automasi	74
Gambar 3.15 Dashboard Grafana saat proses Automasi dijalankan	75
Gambar 3.16 Flowchart Analisa Data	86
Gambar 3.17 Flowchart Pengumpulan Data	88
Gambar 3.18 Kode Program Uji Normalitas	95
Gambar 3.19 Kode Program Transformasi Data	96
Gambar 3.20 Kode Program Uji Homogenitas	99
Gambar 3.21 Kode Program Uji t-test.....	102
Gambar 3.22 Kode Program Uji Mann-Whitney	103
Gambar 4.1 Contoh Visualisasi Penggunaan CPU	129
Gambar 4.2 Perbandingan Rata - rata Penggunaan CPU	131
Gambar 4.3 Contoh Visualisasi Penggunaan RAM	132
Gambar 4.4 Perbandingan Rata - rata Penggunaan RAM.....	134

Gambar 4.5 Contoh Visualisasi Disk Usage.	136
Gambar 4.6 Perbandingan Rata - rata Penggunaan Disk Space.....	138
Gambar 4.7 Contoh Visualisasi Aktivitas R/W Disk	140
Gambar 4.8 Perbandingan Rata - rata Aktivitas R/W Disk.....	142
Gambar 4.9 Contoh Visualisasi Aktivitas R/W I.O	144
Gambar 4.10 Perbandingan Rata - rata Aktivitas R/W I.O	146
Gambar 4.11 Perbandingan Rata - rata Durasi Provisioning	148

DAFTAR TABEL

Tabel 2.1 Penelitian terdahulu yang dijadikan sebagai acuan.....	8
Tabel 3.1 Skenario Pengujian Automasi dan Manual	64
Tabel 3.2 Data Pembuatan VM	76
Tabel 3.3 Data Layanan Web Server	78
Tabel 3.4 Data Layanan Monitoring	80
Tabel 3.5 Data Layanan Docking.....	82
Tabel 3.6 Data Layanan Cloud Storage	84
Tabel 3.7 Rata - rata Penggunaan Sumber Daya.....	89
Tabel 3.8 Hasil Uji Normalitas	96
Tabel 3.9 Hasil Uji Homogenitas	100
Tabel 3.10 Hasil Statistik Uji t-test	103
Tabel 3.11 Kesimpulan Uji t-test	105
Tabel 3.12 Hasil Uji Mann-Whitney pada Data yang tidak normal.....	114

DAFTAR LAMPIRAN

Lampiran 1 Repository Tugas Akhir.....	161
Lampiran 2 Hasil Cek Turnit In.....	162
Lampiran 3 Lembar Keterangan Pengecekan Similarity.....	163
Lampiran 4 Form Perbaikan Penguji.....	165
Lampiran 5 Form Perbaikan Pembimbing 1.....	166
Lampiran 6 Form Perbaikan Pembimbing 2.....	167

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada era digital yang semakin berkembang pesat, kebutuhan akan aplikasi yang cepat, efektif, dan *skalabel* meningkat. Penggunaan aplikasi *desktop*, *mobile*, dan berbasis *website* menjadi sangat umum untuk kebutuhan pribadi dan bisnis[1]. Oleh karena itu, pembuatan dan persiapan infrastruktur yang mendukung aplikasi ini harus dilakukan secara efektif untuk memungkinkan peningkatan jumlah pengguna tanpa mengganggu kinerja atau ketersediaan layanan. Menerapkan sistem otomatisasi dalam pengelolaan infrastruktur adalah salah satu cara untuk memenuhi tuntutan ini. *Automatisasi* mempercepat waktu yang dibutuhkan untuk memulai pengembangan aplikasi karena memungkinkan penyediaan dan pengaturan lingkungan pengembangan dari awal.

Menerapkan sistem otomatisasi dalam pengelolaan infrastruktur menjadi salah satu solusi yang efektif untuk memenuhi tuntutan ini. *Automatisasi* dapat mempercepat waktu yang dibutuhkan untuk memulai pengembangan aplikasi, karena memungkinkan penyediaan dan pengaturan lingkungan pengembangan dari awal secara cepat dan konsisten[2][3]. Dengan alat-alat otomatisasi yang tepat, organisasi dapat mengurangi kesalahan manual, meningkatkan kecepatan, dan mengoptimalkan penggunaan sumber daya.

Proxmox Virtual Enviroment atau *PVE*, adalah platform *virtualisasi* yang berfungsi untuk pembuatan dan pemeliharaan *Virtual Machine* dan kontainer pada *Private Cloud*. *PVE* memungkinkan organisasi membuat dan mengelola infrastruktur *virtualisasi* di dalam lingkungan mereka sendiri yang dapat memberikan kontrol, keamanan, dan *kustomisasi* yang lebih tinggi sehingga menjadi poin penting untuk elastisitas. *Elastilitas* ini memberikan organisasi kemampuan untuk mengelola sumber daya dengan lebih efisien dan responsif terhadap perubahan kebutuhan. Dengan fitur-fitur seperti *skalabilitas* vertikal dan horizontal, pengelolaan *cluster*, dan dukungan

untuk kontainer, *PVE* memungkinkan penerapan dan pengelolaan aplikasi yang lebih dinamis dalam lingkungan *virtualisasi*[4].

Penggunaan alat *automasi* seperti *Terraform*, *Ansible*, *Kubernetes*, *Docker*, serta *CI/CD* (*Continuous Integration and Continuous Deployment*) berperan penting untuk pengelolaan aplikasi[5]. *Terraform* adalah alat *Infrastructure as a Code (IaaS)* yang digunakan untuk mengotomatiskan pengembangan dan manajemen infrastruktur *cloud*. Dengan menggunakan *Terraform*, akan memudahkan pengguna untuk membuat dan mengonfigurasi *VM Nodes* dan sumber daya lainnya sesuai dengan kebutuhan aplikasi[6].

Sementara itu, *Ansible* digunakan untuk mengotomasi proses instalasi dan konfigurasi perangkat lunak di *Virtual Machine* atau Kontainer yang telah disiapkan sebelumnya[7]. Dengan *Ansible*, semua alat yang diperlukan untuk menjalankan aplikasi dapat terinstal dan terkonfigurasi dengan konsisten, mengurangi potensi kesalahan manual yang sering terjadi[8]. Selain itu, penggunaan *Docker* untuk mengemas aplikasi ke dalam kontainer juga memberikan banyak keuntungan, seperti efisiensi sumber daya dan *portabilitas* yang tinggi[9]. Aplikasi dapat berjalan dengan cara yang sama di berbagai lingkungan tanpa masalah kompatibilitas.

Kubernetes berfungsi sebagai platform orkestrasi kontainer yang memungkinkan manajemen aplikasi berbasis kontainer dalam skala besar[9][10]. Dengan *Kubernetes*, aplikasi dapat dijalankan secara otomatis, disebarkan, dan diskalakan dengan lebih mudah, memastikan bahwa aplikasi dapat berjalan dengan optimal meskipun terjadi lonjakan beban[7]. *Kubernetes* juga membantu dalam mengelola kegagalan dan melakukan pemulihan secara otomatis, sehingga meningkatkan ketahanan sistem.

Dalam konteks pengelolaan infrastruktur, otomatisasi menawarkan efisiensi yang signifikan dibandingkan dengan proses manual. Proses manual sering kali rentan terhadap kesalahan manusia dan memerlukan waktu yang lebih lama untuk menyelesaikan tugas-tugas rutin, seperti penyediaan *server* dan konfigurasi lingkungan[8][11]. Sebaliknya, otomatisasi memungkinkan penyediaan dan

pengaturan infrastruktur dilakukan dalam hitungan menit, bukan jam atau hari. Hal ini tidak hanya menghemat waktu tetapi juga memastikan konsistensi dalam konfigurasi, mengurangi variabilitas yang dapat menyebabkan masalah di kemudian hari.

Penelitian sebelumnya sering kali hanya berfokus pada salah satu alat *automasi*, seperti *Terraform* untuk *provisioning* atau *Ansible* untuk konfigurasi[1], [5], [8]. Penelitian ini menawarkan pendekatan yang lebih komprehensif dengan mengintegrasikan semua alat tersebut untuk menciptakan *pipeline automasi end-to-end*. Terlebih, penelitian sebelumnya banyak berfokus pada *automasi* infrastruktur di lingkungan *public cloud* seperti *AWS*, *Google Cloud Platform*, atau *Azure*[5][12][13]. Penelitian ini memfokuskan pada penerapan di *private cloud* berbasis *PVE*, yang menawarkan tingkat kontrol, keamanan, dan *kustomisasi* yang lebih tinggi. Meskipun memberikan kontribusi pada deskripsi implementasi, penelitian-penelitian yang telah disebutkan sebelumnya memiliki kelemahan utama pada tahap analisis data karena tidak memiliki validasi melalui metode yang terstruktur seperti pengujian asumsi statistik atau perhitungan signifikan yang dapat mendukung temuan.

Penelitian ini menggunakan metode analisis *t-test* untuk memvalidasi perbedaan efisiensi antara metode manual dan otomatisasi. *t-test* dipilih karena cocok untuk membandingkan Rata - rata dua kelompok, seperti waktu *provisioning* antara metode manual dan otomatisasi[14]. Selain itu, *t-test* relevan untuk ukuran sampel kecil hingga sedang, yang sering digunakan dalam pengujian infrastruktur. Metode ini membantu menentukan apakah perbedaan antara dua kelompok signifikan secara statistik atau hanya kebetulan[15]. *t-test* juga sederhana dan efisien, sehingga mudah diterapkan dan menghasilkan hasil yang jelas untuk interpretasi. Dalam konteks efisiensi waktu, metode ini efektif untuk mengevaluasi perbedaan dalam penghematan waktu *provisioning* melalui otomatisasi.

Penelitian ini bertujuan untuk merancang dan mengimplementasikan solusi *automasi* aplikasi menggunakan alat-alat seperti *Terraform*, *Ansible*, *Kubernetes*, *CI/CD*, dan *Docker* dalam lingkungan *private cloud* berbasis *PVE*. Fokus utama dari

penelitian ini adalah untuk mengkaji bagaimana penerapan *automasi* dapat meningkatkan efisiensi pengelolaan aplikasi dan infrastruktur, serta memastikan ketersediaan layanan yang lebih baik melalui pendekatan yang *skalabel* dan terkelola dengan baik.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, perumusan masalah dalam penelitian ini adalah:

1. Bagaimana merancang dan mengimplementasikan solusi otomatisasi aplikasi menggunakan alat-alat seperti *Terraform*, *Ansible*, *Kubernetes*, *CI/CD*, dan *Docker* dalam lingkungan *private cloud* berbasis PVE?
2. Apa saja tantangan yang dihadapi dalam penerapan otomatisasi aplikasi dan infrastruktur, serta bagaimana solusi yang dapat diterapkan untuk mengatasi tantangan tersebut?
3. Sejauh mana penerapan alat otomatisasi dapat meningkatkan efisiensi pengelolaan aplikasi dan infrastruktur, serta ketersediaan layanan di lingkungan *private cloud*?

1.3 Batasan Masalah

Berikut batasan masalah dari penelitian ini, yaitu

1. Penelitian ini hanya akan membahas penggunaan alat otomatisasi tertentu, yaitu *Terraform*, *Ansible*, *Kubernetes*, *CI/CD*, dan *Docker* dalam konteks *private cloud* berbasis PVE.
2. Analisis yang dilakukan akan terbatas pada aspek efisiensi pengelolaan aplikasi dan infrastruktur, serta ketersediaan layanan, tanpa memperluas ke aspek lain seperti biaya atau dampak sosial.

1.4 Tujuan

Adapun tujuan dari penulisan Tugas Akhir ini, yaitu untuk :

1. Mengembangkan dan menerapkan solusi otomatisasi aplikasi menggunakan alat seperti *Terraform*, *Ansible*, *Kubernetes*, *CI/CD*, dan *Docker* dalam lingkungan *private cloud* berbasis *Proxmox VE* (PVE) dan *vSphere*.
2. Menganalisis efisiensi pengelolaan aplikasi dan infrastruktur melalui penerapan otomatisasi, serta membandingkan performa metode manual dan otomatisasi dari berbagai aspek sumber daya.
3. Mengidentifikasi tantangan dalam implementasi otomatisasi dan merumuskan solusi untuk meningkatkan ketersediaan layanan, mengurangi *downtime*, dan meningkatkan keandalan sistem.
4. Melakukan evaluasi berbasis data, termasuk penerapan uji statistik (uji t) untuk mengukur signifikansi perbedaan performa antara metode manual dan otomatisasi, khususnya dalam hal waktu *provisioning*, penggunaan CPU, RAM, *disk*, dan I/O.

1.5 Manfaat

Manfaat dari penelitian ini adalah sebagai berikut :

1. Memberikan kontribusi terhadap pengembangan solusi otomatisasi yang lebih efisien dalam pengelolaan aplikasi dan infrastruktur *cloud*, yang berdampak pada peningkatan performa dan keandalan sistem.
2. Menjadi referensi bagi peneliti dan praktisi dalam memahami penerapan otomatisasi *provisioning*, serta menyajikan analisis berbasis statistik (uji t) sebagai dasar validasi perbandingan antar metode.
3. Membantu pengembang dan profesional TI dalam mengidentifikasi dan mengatasi tantangan umum dalam otomatisasi, guna meminimalkan *downtime* dan meningkatkan kontinuitas layanan.
4. Menyediakan hasil penelitian yang dapat langsung diterapkan dalam sistem nyata untuk manajemen aplikasi dan infrastruktur secara otomatis, khususnya

dalam konteks beban tinggi atau lingkungan yang memerlukan *provisioning* cepat dan efisien.

1.6 Sistematika Penulisan

Di sini guna mempermudah penyusunan Tugas Akhir dan juga membuat isi dari setiap bab yang ada pada penelitian Tugas Akhir ini lebih jelas, maka dibuat sistematika penulisan, yaitu sebagai berikut :

BAB I PENDAHULUAN

Bab pertama berisi tentang penjabaran secara sistematis yang berupa bahasan atau topik dari penelitian, bab pertama ini meliputi latar belakang masalah, tujuan, manfaat, perumusan masalah, batasan masalah, serta metodologi yang digunakan untuk penelitian.

BAB II TINJAUAN PUSTAKA

Bab kedua ini berisi tentang dasar dan kerangka teori, serta kerangka pikir penelitian, di mana pada bab ini membahas mengenai *proxmox server*, konsep *cloud computing*, metode pengembangan aplikasi, *devops*, analisis *t-test*, dan lain sebagainya yang berkaitan langsung terhadap penelitian ini.

BAB III METODOLOGI PENELITIAN

Bab ketiga membahas secara sistematis mengenai proses secara bertahap dan terperinci tentang langkah-langkah yang dilakukan dalam penelitian, guna mencari, mengumpulkan dan menganalisa data yang dihasilkan selama tahap perbandingan konfigurasi *automasi* dan manual layanan.

BAB IV HASIL DAN ANALISA

Bab keempat ini membahas tentang hasil pengujian (eksperimen) dari proses yang telah dilakukan. Serta pada bab ini analisis akan dilakukan sesuai dengan data yang telah dikumpulkan dari hasil pengujian.

BAB V KESIMPULAN DAN SARAN

Bab kelima atau yang terakhir ini berisi tentang kesimpulan dan saran berdasarkan hasil dan analisa dari penelitian yang telah dilakukan.

DAFTAR PUSTAKA

- [1] C. Lekkala, “Automating *Infrastructure* Management with *Terraform*: Strategies and Impact on Business Efficiency,” no. November 2022, 2022, doi: 10.5281/zenodo.12737493.
- [2] M. R. Apriansyah, “Pengembangan Media Pembelajaran Video Berbasis Animasi Mata Kuliah Ilmu Bahan Bangunan Di Program Studi Pendidikan Teknik Bangunan Fakultas Teknik Universitas Negeri Jakarta,” *J. PenSil*, vol. 9, no. 1, pp. 9–18, 2020, doi: 10.21009/jpensil.v9i1.12905.
- [3] M. H. R. Istifarulah and R. Tiaharyadini, “*DevOps, Continuous Integration* and *Continuous Deployment* Methods for *Software Deployment Automation*,” *JISA(Jurnal Inform. dan Sains)*, vol. 6, no. 2, pp. 116–123, 2023, doi: 10.31326/jisa.v6i2.1751.
- [4] Y. Al-Dhuraibi, F. Paraiso, N. Djarallah, and P. Merle, “Elasticity in *Cloud Computing*: State of the Art and Research Challenges,” *IEEE Trans. Serv. Comput.*, vol. 11, no. 2, pp. 430–447, 2018, doi: 10.1109/TSC.2017.2711009.
- [5] V. Marella, “Optimizing *DevOps Pipelines* with *Automation : Ansible* and *Terraform* in *AWS Environments*,” pp. 285–294, 2024.
- [6] I. Operations, “East African Journal of Information Technology The Implementation of *Infrastructure as Code Template* for *Low-cost Cloud*,” vol. 7, no. 1, pp. 462–474, 2024, doi: 10.37284/eajit.7.1.2638.IEEE.
- [7] I. P. Hariyadi and K. Marzuki, “Implementation Of Configuration Management *Virtual Private Server* Using *Ansible*,” *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 19, no. 2, pp. 347–357, 2020, doi: 10.30812/matrik.v19i2.724.
- [8] A. Khumaidi, “Implementation of *Devops* Method for *Automation* of *Server*

- Management Using *Ansible*,” *J. Transform.*, vol. 18, no. 2, p. 199, 2021, doi: 10.26623/transformatika.v18i2.2447.
- [9] B. R. Cherukuri, “*Containerization in cloud computing : comparing Docker and Kubernetes for scalable web applications*,” vol. 13, no. 01, pp. 3302–3315, 2024.
- [10] M. Lucas, “Leveraging Micro services and *Containerization for Scalable Software Solutions*,” no. November, 2024.
- [11] G. Vanam, “*INFRASTRUCTURE AUTOMATION IN CLOUD COMPUTING : A SYSTEMATIC REVIEW OF TECHNOLOGIES , IMPLEMENTATION PATTERNS , AND ORGANIZATIONAL IMPACT*,” no. January, 2025, doi: 10.34218/IJCET.
- [12] V. Marella, I. Researcher, and R. Applications, “*Journal of Sustainable Solutions*,” vol. 1, no. 4, pp. 144–152, 2024.
- [13] R. Molleti and E. Engineers, “*End-To-End Cloud Infrastructure Automation End-To-End Cloud Infrastructure Automation*,” no. January 2019, 2024.
- [14] I. Journal, E. Issn, and I. No, “*Test Case Automation : Transforming Software Testing in the Digital Era Test Case Automation : Transforming Software Testing in the Digital Era*,” vol. 6, no. 5, pp. 52–58, 2024.
- [15] P. Mishra, U. Singh, and C. M. Pandey, “*Application of Student ’ s t - test , Analysis of Variance , and Covariance*,” pp. 407–411, 2019, doi: 10.4103/aca.ACA.
- [16] V. Jayaram, J. Chase, and S. R. Sankiti, “*Accelerated Cloud Infrastructure Development Using Terraform*,” no. September, 2024, doi: 10.5281/zenodo.13935111.
- [17] P. Ganesan, “*DevOps Automation for Cloud Native Distributed Applications* Available online www.jsaer.com *DevOps Automation for Cloud Native*

- Distributed Applications,” no. February 2020, 2024, doi: 10.5281/zenodo.13753321.
- [18] S. Pattanayak, P. Murthy, and A. Mehra, “Integrating AI into *DevOps pipelines* : *Continuous integration* , *continuous delivery* , and *automation* in infrastructural management : Projections for future Integrating AI into *DevOps pipelines* : *Continuous integration* , *continuous delivery* , and automatio,” no. September, 2024, doi: 10.30574/ijjsra.2024.13.1.1838.
- [19] S. Amgothu and G. Kankanala, “AI / ML – *DevOps Automation*,” no. November, 2024.
- [20] M. Moez *et al.*, “Comprehensive Analysis of *DevOps*: Integration, *Automation*, Collaboration, and *Continuous Delivery*,” *Bull. Bus. Econ.*, vol. 13, no. 1, 2024, doi: 10.61506/01.00253.
- [21] J. Kovacs, B. Ligetfalvi, and R. Lovas, “Automated *debugging* mechanisms for orchestrated *cloud infrastructures* with active *control* and global evaluation,” *IEEE Access*, no. August, pp. 143193–143214, 2024, doi: 10.1109/ACCESS.2024.3467228.
- [22] M. Kondo, H. Langi, Y. Putung, and V. Lengkong, “Performance Analysis of *Cloud Computing* Based *E-Commerce Server* Using *PROXMOX Virtual Environment*,” no. January, pp. 741–745, 2023, doi: 10.5220/0011876000003575.
- [23] Y. Ariyanto, “Single *Server-Side* and Multiple *Virtual Server-Side* Architectures: Performance Analysis on *Proxmox Ve* for *E-Learning* Systems,” *J. Eng. Technol. Ind. Appl.*, vol. 9, no. 44, pp. 25–34, 2023, doi: 10.5935/jetia.v9i44.903.
- [24] K. Marzuki, M. I. Kholid, I. P. Hariyadi, and L. Z. A. Mardedi, “Automation of *Open VSwitch*-Based *Virtual Network* Configuration Using *Ansible* on *Proxmox Virtual Environment*,” *Int. J. Electron. Commun. Syst.*, vol. 3, no. 1, p. 11, 2023,

doi: 10.24042/ijecs.v3i1.16524.

- [25] V. Oleksiuk and O. Oleksiuk, “The practice of developing the academic *cloud* using the *Proxmox* VE platform,” *Educ. Technol. Q.*, vol. 2021, no. 4, pp. 605–616, 2021, doi: 10.55056/etq.36.
- [26] V. Arulkumar and R. Lathamaju, “Start to Finish *Automation* Achieve on *Cloud* with *Build Channel*: By *DevOps* Method,” *Procedia Comput. Sci.*, vol. 165, no. 2019, pp. 399–405, 2019, doi: 10.1016/j.procs.2020.01.032.
- [27] R. Nirek, “Available online www.jsaer.com Comparative Study of Hypervisor Technologies for Running Enterprise Applications on Linux VMs,” vol. 8, no. 3, pp. 278–284, 2021.
- [28] M. Zboril and V. Svatá, “Performance comparison of *cloud virtual machines*,” *J. Syst. Inf. Technol.*, vol. 27, no. 2, pp. 197–213, 2025, doi: 10.1108/JSIT-02-2022-0040.
- [29] K. Chengeta, “Comparing the performance between *Virtual Machines* and *Containers* using deep learning credit models,” pp. 1–8, 2021, doi: 10.1145/3487923.3487934.
- [30] G. I. Radchenko, A. B. A. Alaasam, and A. N. Tchernykh, “Comparative analysis of *virtualization* methods in Big Data processing,” *Supercomput. Front. Innov.*, vol. 6, no. 1, pp. 48–79, 2019, doi: 10.14529/jsfi190107.
- [31] R. M. Karamagi and S. Ally, “iMedPub Journals Analysis of Wasted *Computing Resources* in Robert Method Karamagi 1 * Data Centers in terms of CPU , RAM and HDD and Said Ally Abstract *Virtualization*,” pp. 1–11, 2020, doi: 10.36648/computer-science-engineering-survey.08.02.08.American.
- [32] A. H. Hussein, H. S. Fouda, M. E. S. Nasr, and H. H. Abdullah, “A Novel *T-Test* Based Spectrum Sensing Technique for SIMO and MIMO Cognitive Radio Networks,” *IEEE Access*, vol. 8, pp. 140279–140292, 2020, doi: 10.1109/ACCESS.2020.3012375.

- [33] M. R. Salmanpour *et al.*, “Machine Learning Evaluation Metric Discrepancies across Programming Languages and Their Components in Medical Imaging Domains: Need for Standardization,” *IEEE Access*, vol. 13, no. March, 2025, doi: 10.1109/ACCESS.2025.3549702.
- [34] A. Potochnik, M. Colombo, and C. Wright, “Statistics and Probability,” *Recipes Sci.*, no. Table 2, pp. 167–206, 2018, doi: 10.4324/9781315686875-6.
- [35] K. Shafi *et al.*, “Measuring performance through enterprise *resource* planning system implementation,” *IEEE Access*, vol. 7, pp. 6691–6702, 2019, doi: 10.1109/ACCESS.2018.2884900.
- [36] M. F. Saiyed and I. Al-Anbagi, “A Genetic Algorithm- and *t-Test*-Based System for DDoS Attack Detection in IoT Networks,” *IEEE Access*, vol. 12, no. January, pp. 25623–25641, 2024, doi: 10.1109/ACCESS.2024.3367357.