

**OPTIMASI *DEPLOYMENT* APLIKASI BERBASIS SERVER KE
CLOUD ENVIRONMENT DENGAN METODE *CONTINUOUS
DEPLOYMENT***

TUGAS AKHIR

**Diajukan untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer**



OLEH:

QORY AMANAH PUTRA

09011281924152

JURUSAN SISTEM KOMPUTER

FAKULTAS ILMU KOMPUTER

UNIVERSITAS SRIWIJAYA

2024

HALAMAN PENGESAHAN

HALAMAN PENGESAHAN

Optimasi *Deployment* Aplikasi Berbasis Server ke *Cloud Environment* dengan Metode *Continuous Deployment*

TUGAS AKHIR

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

Oleh

Qory Amanah Putra

09011281924152

Palembang, 18 Juni 2024

Mengetahui,

Ketua Jurusan Sistem Komputer



Dr. Ir. H. Sukemi, M.T.
NIP.196612032006041001

Pembimbing Tugas Akhir

Ahmad Heryanto, S. Kom., M.T.
NIP. 198701222015041002

HALAMAN PERSETUJUAN

HALAMAN PERSETUJUAN

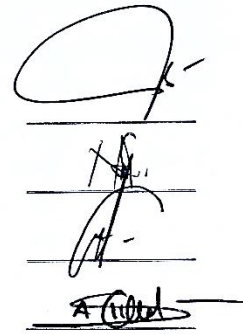
Telah diuji dan lulus pada:

Hari : Senin

Tanggal : 10 Juni 2024

Tim Penguji:

1. Ketua : Kemahyanto Exaudi, M.T.
2. Sekretaris : Nurul Affah, M.Kom.
3. Penguji : Dr. Ahmad Zarkasi, M.T.
4. Pembimbing : Ahmad Heryanto, S.Kom., M.T.



Four handwritten signatures are shown, each on a horizontal line. The signatures correspond to the names listed in the 'Tim Penguji' section: Kemahyanto Exaudi, Nurul Affah, Dr. Ahmad Zarkasi, and Ahmad Heryanto.

Mengetahui, 10 Juni 2024



Dr. Ir. H. Sukemi, M.T.

NIP.196612032006041001

HALAMAN PERNYATAAN

HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Qory Amanah Putra

NIM : 09011281924152

Judul : Optimasi *Deployment* Aplikasi Berbasis Server ke *Cloud Environment* dengan Metode *Continuous Deployment*

Hasil pengecekan software Turnitin : 19%.

Menyatakan bahwa laporan tugas akhir 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 siapa pun.

Palembang, 25 Juni 2024



Qory Amanah Putra

NIM. 09011281924152

HALAMAN PERSEMBAHAN

“Pray for the best, prepare for the worst”

Penulis:

“Qory Amanah Putra”

Skripsi ini saya persembahkan untuk:

Ibunda dan Almarhum Ayah tercinta yang telah menjadi pendukung,
penyemangat, tempat pulang, dan tempat bersandar pertama bagi penulis.

KATA PENGANTAR

Puji dan syukur kepada Tuhan Yang Maha Esa, karena berkat Rahmat dan Karunia-Nya penulis dapat menyelesaikan penyusunan Tugas Akhir ini dengan judul “**Optimasi *Deployment* Aplikasi Berbasis Server ke *Cloud Environment* dengan Metode *Continuous Deployment*”**”.

Laporan ini merupakan salah satu syarat untuk memenuhi sebagian kurikulum dan syarat untuk kelulusan Mata Kuliah Skripsi pada Jurusan Sistem Komputer di Universitas Sriwijaya. Adapun pembahasan dalam laporan ini adalah menjelaskan mengenai pembuatan arsitektur dan algoritma untuk melakukan *deployment* aplikasi berbasis server dengan metode *Continuous Deployment* untuk diterapkan ke lingkungan Awan yang diimplementasikan menjadi sebuah sistem.

Pada kesempatan ini, penulis ingin mengucapkan terima kasih kepada pihak yang telah memberikan bantuan, dorongan, motivasi, semangat, dan bimbingan dalam penyusunan proposal skripsi ini, yakni :

1. Kepada Tuhan Yang Maha Esa, yang telah melimpahkan Berkat dan Rahmat-Nya.
2. Orang tua serta kakak dan adik yang menjadi penyemangat utama dalam penulisan tugas akhir.
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 Universitas Sriwijaya.
5. Bapak Iman Saladin B. Azhar, S.Kom., M.MSi. selaku Dosen Pembimbing Akademik.
6. Bapak Ahmad Heryanto, S. Kom, M.T. selaku Dosen Pembimbing Tugas Akhir.
7. Mbak Sari, Kak Yopi, dan Kak Angga selaku admin Jurusan Sistem Komputer.
8. Muhammad Sultan Alif dan Sa’ad Abdillah Waqas yang selalu menemani dan menyemangati di rumah kontrakan.
9. Teman-teman Sistem Komputer 2019 Indralaya

10. Kakak tingkat Sistem Komputer Universitas Sriwijaya yang lainnya.

Dalam penyusunan Tugas Akhir ini saya menyadari sepenuhnya bahwa laporan ini masih jauh dari kata sempurna, oleh karena itu saya mengharapkan saran dan kritik dari semua pihak yang berkenan demi laporan yang lebih baik lagi.

Akhir kata, saya harap semoga Tugas Akhir ini dapat bermanfaat serta dapat memberikan pengetahuan dan wawasan bagi semua pihak yang membutuhkannya.

Palembang, Juni 2024

Penulis,

Qory Amanah Putra

NIM. 09011281924152

Abstrak

OPTIMASI *DEPLOYMENT* APLIKASI BERBASIS SERVER KE *CLOUD ENVIRONMENT* DENGAN METODE *CONTINUOUS DEPLOYMENT*

Qory Amanah Putra (09011281924152)

Fakultas Ilmu Komputer, Universitas Sriwijaya

Email: qorvamanahputra0708@gmail.com

Abstrak

Continuous Deployment (CD) merupakan metode penyebaran aplikasi ke server produksi secara otomatis dan berkelanjutan. Penelitian ini bertujuan untuk mengeksplorasi penerapan metode *Continuous Deployment* dalam penyebaran aplikasi berbasis server ke lingkungan *Cloud*. Penelitian ini melibatkan pengujian terhadap 26 aplikasi berbasis server yang bersifat *open source*. Hasil pengujian kemudian dibandingkan dengan penelitian sebelumnya. Temuan menunjukkan peningkatan kinerja pada 4 dari 5 parameter yang diuji, dengan peningkatan sebesar 30,5% pada time-cost, 6,13% pada penggunaan RAM, 8,95% pada penggunaan CPU, dan 86% pada pembacaan memori.

Kata Kunci: *Cloud Computing, Cloud Environment, Deployment Automation, Continuous Deployment.*

Palembang, 24 Juni 2024

Mengetahui,

Ketua Jurusan Sistem Komputer

Pembimbing Tugas Akhir



Dr. Ir. H. Sukemi, M.T.
NIP.196612032006041001

Ahmad Heryanto, S. Kom., M.T.
NIP. 198701222015041002

Abstract

OPTIMIZATION OF DEPLOYMENT OF SERVER-BASED APPLICATIONS TO CLOUD ENVIRONMENT USING CONTINUOUS DEPLOYMENT

Qory Amanah Putra (09011281924152)

Faculty of Computer Science, Sriwijaya University

Email: qoryamanahputra0708@gmail.com

Abstract

Continuous Deployment (CD) is a method of deploying applications to production servers automatically and continuously. This research aims to explore the application of the Continuous Deployment method in deploying server-based applications to a Cloud environment. This research involved testing 26 open-source server-based applications. The test results were then compared with previous research. Findings show performance improvements in 4 of the 5 parameters tested, with improvements of 30.5% in time-cost, 6.13% in RAM usage, 8.95% in CPU usage, and 86% in disk reads.

Keywords: *Cloud Computing, Cloud Environment, Deployment Automation, Continuous Deployment.*

Palembang, June 2024

Acknowledge,

Head of Computer System
Departments



Dr. Ir. H. Sukemi, M.T.
NIP.196612032006041001

Supervisor

Ahmad Heryanto, S. Kom., M.T.
NIP. 198701222015041002

DAFTAR ISI

HALAMAN PENGESAHAN	i
HALAMAN PERSETUJUAN	ii
HALAMAN PERNYATAAN	iii
HALAMAN PERSEMBAHAN	iv
KATA PENGANTAR	v
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xvi
DAFTAR LAMPIRAN	xviii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah	4
1.3. Batasan Masalah.....	5
1.4. Tujuan Penelitian.....	5
1.5. Manfaat Penelitian	5
1.6. Metodologi Penelitian	5
1.7. Sistematika Penulisan	7
BAB II TINJAUAN PUSTAKA	8
2.1. Pendahuluan	8
2.2. Deployment Automation	14
2.2.1. Pengertian <i>Deployment</i>	14
2.2.2. Pengertian <i>Automation</i>	14
2.2.3. Pengertian Deployment Automation.....	15
2.3. CI/CD	15

2.3.1.	Konsep CI/CD	15
2.3.2.	Continuous Integration	17
2.3.3.	Continuous Deployment	17
2.3.4.	Alat CI/CD	17
2.4.	Cloud Computing	18
2.4.1.	Pengertian Cloud Computing	18
2.4.2.	Pengertian Cloud Environment.....	19
2.4.3.	Model Layanan <i>Cloud Computing</i>	19
2.4.4.	Model Deployment Cloud Computing	20
2.4.5.	Karakteristik <i>Cloud Computing</i>	21
2.5.	Version Control System (VCS)	22
2.5.1.	Centralized VCS	23
2.5.2.	Distributed VCS.....	23
2.6.	SSH	23
2.7.	Software Repository	24
2.8.	Kompleksitas Waktu	25
2.8.1.	Big-O Notation	26
2.8.2.	Big Omega Notation	28
2.8.3.	Big Theta Notation.....	29
2.8.4.	Istilah dalam Kompleksitas Waktu	29
2.8.5.	Contoh Algoritma dan Kompleksitas Waktunya.....	30
2.9.	Teknik Analisis Data	31
2.9.1.	Uji Normalitas	31
2.9.2.	Uji Signifikansi.....	33
2.9.3.	Uji Post-Hoc	35
2.10.	Paramiko	36
2.11.	Git Hooks	36
2.11.1.	Pre-Push.....	37
2.11.2.	Post-Update.....	37
2.12.	Python	38
BAB III METODOLOGI PENELITIAN		39
3.1.	Pendahuluan	39

3.2.	Desain Penelitian.....	39
3.3.	Kebutuhan Perangkat Keras dan Perangkat Lunak	41
3.3.1.	Perangkat Keras	41
3.3.2.	Perangkat Lunak	41
3.4.	Persiapan Aplikasi untuk <i>Deployment</i>	42
3.5.	Persiapan Arsitektur <i>Cloud Computing</i>	44
3.5.1.	Topologi Jaringan	46
3.5.2.	Konfigurasi Lingkungan VM Server Produksi	48
3.5.3.	Konfigurasi Lingkungan VM Git Server	48
3.6.	Skenario Sistem <i>Deployment</i>	49
3.6.1.	Skenario 1	50
3.6.1.1.	Arsitektur	50
3.6.1.2.	Algoritma Deployment	52
3.6.2.	Skenario 2	54
3.6.2.1.	Arsitektur	54
3.6.2.2.	Algoritma Deployment	55
3.6.3.	Skenario 3	57
3.6.3.1.	Arsitektur	57
3.6.3.2.	Algoritma Deployment	59
3.6.4.	Skenario Sistem Pemandangan	61
3.7.	Skenario Pengujian.....	62
3.8.	Pengolahan Data.....	64
BAB IV HASIL DAN PEMBAHASAN.....		67
4.1.	Pembuatan Arsitektur Cloud Computing	67
4.2.	Sistem Deployment	68
4.2.1.	Skenario 1	69
4.2.2.	Skenario 2	70
4.2.3.	Skenario 3	71
4.3.	Hasil Pengujian	72
4.3.1.	Hasil Pengujian Skenario 1	72
4.3.2.	Hasil Pengujian Skenario 2.....	75
4.3.3.	Hasil Pengujian Skenario 3.....	77

4.3.4.	Hasil Pengujian Sistem Pemandang	80
4.3.5.	Hasil Pengujian Penggunaan RAM dan CPU	81
4.3.6.	Hasil Pengujian <i>Throughput</i> Disk.....	82
4.4.	Perbandingan Hasil Pengujian	84
4.4.1.	Time-Cost Keseluruhan	84
4.4.2.	Time-Cost Pre-Deployment	87
4.4.3.	Time-Cost Deployment.....	90
4.4.4.	Time-Cost Post-Deployment	92
4.5.	Perbandingan Hasil Pengujian dengan Sistem Pemandang.....	94
4.6.	Validasi Hasil	101
4.6.1.	Uji Normalitas Saphiro-Wilk.....	101
4.6.2.	Uji Kruskal-Wallis	104
4.6.3.	Uji Dunn	107
4.7.	Perbandingan Hasil Pengujian berdasarkan Penelitian Terkait.....	108
BAB V KESIMPULAN DAN SARAN		110
5.1.	Kesimpulan	110
5.2.	Saran.....	110
DAFTAR PUSTAKA.....		111
LAMPIRAN.....		A

DAFTAR GAMBAR

Gambar 2.1. Model deployment automation	15
Gambar 2.2. CI/CD.....	15
Gambar 2.3. Model layanan <i>Cloud Computing</i>	19
Gambar 2.4. <i>Version Control System</i>	22
Gambar 2.5. Arsitektur SSH.....	24
Gambar 2.6. Kompleksitas waktu.....	25
Gambar 2.7. Model uji normalitas.....	32
Gambar 2.8. Pemilihan uji signifikansi.....	34
Gambar 2.9. Pemilihan uji post-hoc	35
Gambar 2.10. Git hooks.....	36
Gambar 3.1. Desain penelitian.....	40
Gambar 3.2. Arsitektur <i>Cloud</i>	45
Gambar 3.3. Arsitektur Cloud Computing di Google Cloud Platform.....	46
Gambar 3.4. Topologi Jaringan.....	47
Gambar 3.5. Topologi Jaringan di Google Cloud Platform.....	47
Gambar 3.6. Alur kerja sistem automasi deployment skenario 1.....	51
Gambar 3.7. Arsitektur sistem automasi deployment skenario 1.....	51
Gambar 3.8. Algoritma script deployment skenario 1.....	52
Gambar 3.9. Alur kerja sistem automasi deployment skenario 2.....	54

Gambar 3.10. Arsitektur sistem automasi deployment skenario 2.	55
Gambar 3.11. Algoritma script deployment skenario 2.....	55
Gambar 3.12. Alur kerja sistem automasi deployment skenario 3.	58
Gambar 3.13. Arsitektur sistem automasi deployment skenario 3.	59
Gambar 3.14. Algoritma script deployment skenario 3.....	59
Gambar 3.15. Arsitektur sistem pengujian.	62
Gambar 3.16. Skenario pengujian.	64
Gambar 3.17. Alur Pemrosesan Data.....	66
Gambar 4.1. Struktur file repo.json.	69
Gambar 4.2. Script deployment skenario 1. (a) c1. (b) c2 . (c) c3. (d) c4.....	70
Gambar 4.3. Script deployment skenario 2. (a) c1. (b) c3. (c) c4.....	71
Gambar 4.4. Script deployment skenario 3 simbol c3.	72
Gambar 4.5. Waktu <i>deployment</i> skenario 1.	73
Gambar 4.6. Waktu pemrosesan tahap <i>pre-action</i> pada skenario 1.....	74
Gambar 4.7. Waktu deployment skenario 2.....	76
Gambar 4.8. Waktu pemrosesan tahap <i>pre-action</i> pada skenario 2.....	77
Gambar 4.9. Waktu deployment skenario 3.....	78
Gambar 4.10. Waktu pemrosesan tahap <i>pre-action</i> pada skenario 2.....	79
Gambar 4.11. Waktu deployment skenario sistem pembanding.....	80
Gambar 4.12. Penggunaan CPU dan RAM pada server <i>Cloud</i>	82

Gambar 4.13. <i>Throughput read disk</i>	83
Gambar 4.14. <i>Throughput write disk</i>	83
Gambar 4.15. hasil pengujian skenario 1, skenario 2, dan skenario 3.....	86
Gambar 4.16. Hasil pengujian tahap <i>pre-action</i> tiga skenario utama.....	89
Gambar 4.17. Hasil pengujian tahap <i>action</i> tiga skenario utama.	91
Gambar 4.18. Hasil pengujian tahap <i>post-action</i> tiga skenario utama.	93
Gambar 4.19. Hasil pengujian seluruh skenario.....	96
Gambar 4.20. Hasil pengujian tahapan <i>pre-action</i> seluruh skenario.....	97
Gambar 4.21. Hasil pengujian tahapan <i>action</i> seluruh skenario.	98
Gambar 4.22. Hasil pengujian tahapan <i>post-action</i> seluruh skenario.	99
Gambar 4.23. Radar chart nilai rata – rata hasil pengujian.	100
Gambar 4.24. Persebaran data Skenario 1	101
Gambar 4.25. Persebaran data Skenario 2	102
Gambar 4.26. Persebaran data Skenario 3	103
Gambar 4.27. Persebaran data Skenario Pembandingan.....	103

DAFTAR TABEL

Tabel 2.1. Penelitian Rujukan Utama.....	8
Tabel 3.2. Spesifikasi perangkat lunak.....	41
Tabel 3.3. Aplikasi berbasis server untuk pengujian sistem.....	42
Tabel 3.4. Spesifikasi layanan <i>Cloud Computing</i>	44
Tabel 3.5. Kompleksitas waktu skenario 1.....	53
Tabel 3.6. Kompleksitas waktu skenario 2.....	56
Tabel 3.7. Kompleksitas waktu skenario 3.....	60
Tabel 4.1. Sampel data hasil pengujian skenario 1.	72
Tabel 4.2. Data deskriptif hasil pengujian skenario 1 (detik).	73
Tabel 4.3. Data deskriptif hasil pengujian skenario 2 (detik).	75
Tabel 4.4. Data deskriptif hasil pengujian skenario 3 (detik).	78
Tabel 4.5. Data deskriptif hasil pengujian skenario pembandingan (detik).....	80
Tabel 4.6. Penggunaan CPU dan RAM pada server <i>Cloud</i>	82
Tabel 15 <i>Throughput disk</i> pada server <i>Cloud</i>	84
Tabel 4.8. Data statistik deskriptif skenario utama (detik).....	84
Tabel 4.9. Data statistik deskriptif tahap pre-action skenario utama (detik).....	87
Tabel 4.10. Data statistik deskriptif tahap <i>action</i> skenario utama (detik).....	90
Tabel 4.11. Data tahap post-action skenario utama (detik).	92
Tabel 4.12. Hasil Uji Normalitas Saphiro-Wilk.....	104

Tabel 4.13. Nilai n , R_i , dan $n(R_i - R)^2$	105
Tabel 4.14. Nilai statistik Kruskal-Wallis	106
Tabel 4.15. Hasil Uji Dunn dari setiap skenario dengan skenario penguji	107
Tabel 4.16. Persentase keunggulan sistem yang dibuat.	108
Tabel 4.17. Perbandingan terhadap penelitian terkait	109

DAFTAR LAMPIRAN

1. Form revisiA
2. Surat keterangan *similarity*C

BAB I PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi memudahkan manusia dalam menjalani kehidupan sehari-hari, mulai dari bangun tidur hingga nantinya beristirahat kembali. Perkembangan teknologi ini tak lepas dari semakin besarnya kebutuhan manusia dalam berbagai bidang kehidupan, salah satunya adalah bidang *Cloud Computing*.

Pembahasan mengenai *Cloud Computing* tidak luput dari perkembangan sumber daya komputasi (*computational resources*) fisik yang stabil sehingga mendorong tren di bidang sumber daya virtual (*virtual resources*) [1], [2]. Sumber daya virtual ini dapat diimplementasikan sebagai mesin virtual atau VM (*Virtual Machine*), jaringan virtual, *hard drive* virtual dan lainnya. Pemanfaatan teknik virtualisasi pada sumber daya komputasi fisik adalah cara efektif untuk mengurangi *energy-cost* [3]. Dibandingkan dengan sumber daya fisiknya, terdapat penurunan kinerja pada sumber daya virtual yang masih dalam batas wajar. Namun sumber daya virtual memiliki keunggulan yang luar biasa karena sifatnya yang fleksibel. Efek dari fleksibilitas inilah yang mendorong semakin berkembangnya *Cloud Computing* [1].

Cloud Computing sudah diaplikasikan oleh berbagai perusahaan dan instansi baik pada kelas kecil hingga korporasi. Maraknya penerapan *Cloud Computing* dikarenakan memiliki banyak keunggulan dibandingkan dengan infrastruktur *on-premise* seperti biaya yang lebih murah, ketersediaan sumber daya yang instan, fleksibilitas, hemat waktu, dan keandalannya [4], [5].

Cloud Computing cocok untuk menyediakan aplikasi berbasis server yang bersifat *published* atau siap dipakai seperti Web App, Rest API, AI model, Aplikasi Penyimpanan, dan Aplikasi Kolaborasi [6]. Aplikasi yang disediakan untuk publik baik itu memerlukan autentikasi atau tidak, tentunya akan terus menerus berkembang sesuai dengan kebutuhan. Perkembangan aplikasi ini menuntut

developer (pengembang) untuk dapat mengirimkan perubahan ke *Cloud Environment* dengan cepat dan tepat agar dapat *publish* dan digunakan.

Proses publikasi perubahan ke lingkungan produksi dinamakan dengan *deployment* (penyebaran). Awalnya proses *deployment* aplikasi dilakukan secara konvensional, yaitu dengan mengirimkan *file* yang mengalami perubahan ke server produksi menggunakan protokol FTP (*File Transfer Protocol*). Cara lainnya yaitu menggunakan *Version Control System* (VCS) untuk pelacakan perubahan yang kemudian di-*push* ke server pusat VCS dan melakukan *pull* pada aplikasi di server produksi. Proses manual ini menimbulkan kesulitan, membutuhkan waktu yang lama, dan setiap kali ada perubahan kode maka harus mengulangi langkah yang sama untuk melakukan *deployment* [7]. Hal ini memicu pembuatan *Deployment Automation* dengan tujuan menghemat waktu (*time-cost*) dan meningkatkan produktivitas pengembang dengan melakukan proses *deployment* secara otomatis.

Deployment Automation atau otomatisasi proses *deployment* adalah proses yang memungkinkan penyebaran aplikasi ke lingkup produksi berjalan secara runtut, teratur, dan otomatis dengan satu pemicu. Salah satu model *Deployment Automation* adalah praktik *Continuous Deployment* yang merupakan bagian dari praktik berkelanjutan atau lebih dikenal dengan istilah CI/CD.

Praktik *Continuous Deployment* (CD) secara otomatis menyebarkan perubahan aplikasi ke lingkungan produksi setelah berhasil melalui proses sebelumnya yang sudah ditetapkan. Dalam CD, sesaat setelah pengembang melakukan perubahan, perubahan tersebut akan langsung diterapkan atau disebarkan ke lingkungan produksi melalui *deployment pipeline* yang biasanya dipicu dengan pendekatan berbasis *push* [8], namun bisa juga menggunakan pendekatan lain seperti *merge* sesuai kebutuhan aplikasi.

Dalam penerapannya, perusahaan besar seperti Facebook, Flickr, Etsy mengadopsi CD bahkan dari awal tahun 2000-an. Pada tahun 2009, Flickr melakukan *deployment* dengan *Continuous Deployment* dengan rata-rata 10 kali dalam sehari. Pada tahun 2011, Etsy yang merupakan perusahaan *e-commerce* telah memanfaatkan CD dan melakukan *deployment* lebih dari 11.000 kali. Data yang

lebih besar berasal dari Facebook, yang dari tahun 2008 hingga 2014 telah melakukan total 1.159.303 *commit*, di mana 705.631 *commit* untuk aplikasi *website*, 68.272 *commit* untuk aplikasi Ios, 146.658 *commit* untuk aplikasi Android, dan 238.742 *commit* untuk aplikasi *back-end* [9].

Terdapat beberapa contoh layanan yang menyediakan sistem untuk melakukan CD. Beberapa layanan terkenal seperti Github Actions [10], Travis [11], Jenkins [7], [12], TeamCity [7], dan Circle-CI. Layanan-layanan tersebut mengotomatiskan perubahan kode dari beberapa kontributor ke dalam repositori pusat tempat *build* otomatis, *testing* (pengujian), dan pemeriksaan kualitas kode dijalankan. Layanan – layanan di atas menyediakan berbagai kebutuhan untuk pengembang. Namun terkadang pengembang hanya membutuhkan satu atau beberapa layanan sederhana untuk menjalankan automasi proses *deployment* alih-alih menggunakan layanan terpadu, *compact* dan kompleks. Layanan sederhana akan dapat berjalan lebih cepat dan sumber daya yang digunakan lebih ringan karena bersifat *standalone* dan tidak terikat dengan layanan lain.

Pada penelitian [9], [13], diterangkan bahwa Facebook menggunakan praktik *Continuous Deployment* sebagai strategi pengembangan dan *deployment* perangkat lunak untuk memfasilitasi *rapid innovation* dan *frequent update* pada platformnya. Penerapan CD dipilih karena memungkinkan eksperimen langsung menggunakan *A/B test*, di mana fitur – fitur baru segera diterapkan agar dapat dinikmati oleh pengguna sebenarnya, sehingga teknisi dapat membandingkan dampak fitur-fitur ini terhadap perilaku pengguna dengan situs saat ini.

Pada penelitian [7], [14] penulis menjelaskan bahwa latar belakang penerapan CD adalah kebutuhan bisnis yang terus berkembang sehingga membutuhkan praktik automasi, karena proses *deployment* secara tradisional memakan waktu yang lama dan setiap ada perubahan walaupun kecil harus dilakukan *deployment* manual kembali. Penggunaan CD didasarkan pada kelebihan dari praktik ini, yaitu dapat mengurangi kesalahan yang rawan, menghemat waktu, dan *developer* selalu dapat mengetahui bagaimana aplikasi dikonfigurasi jika melakukan *build* menggunakan proses CD. Selain itu, CD dapat mempercepat

waktu pemasaran, meningkatkan kualitas produk, meningkatkan kepuasan pelanggan, rilis yang andal, meningkatkan produktivitas dan efisiensi [14].

Berbagai penelitian sudah dilakukan untuk melakukan optimasi deployment aplikasi menggunakan model *Continuous Deployment* yang berhasil menghemat waktu *deployment* hingga 27% [15] dengan rata – rata waktu *pre-deployment* 2,509 hingga 11,5 detik [7], [15], [16], [17] dan rata – rata waktu *post-deployment* 0,4025 detik hingga 3 detik [18].

Pada penelitian [7] melakukan deployment aplikasi berbasis kontainer ke lingkungan Cloud menggunakan Jenkins *CI/CD Pipeline* dengan waktu rata – rata *pre-action* 7,75 detik. Pada penelitian [15] menggunakan orkestrasi kontainer untuk melakukan *deployment* aplikasi ke lingkungan awan dengan waktu keseluruhan *deployment* 8 menit 43 detik yang lebih cepat dibandingkan cara tradisional yaitu dengan waktu 11 menit 57 detik. Pada penelitian [16] menggunakan kontainer untuk membandingkan proses deployment aplikasi ansible dengan aplikasi monolitik di mana mendapatkan rata – rata waktu *pre-action* sebesar 5,2 detik. Pada penelitian [17] melakukan deployment aplikasi web berbasis dengan waktu *pre-action* selama 2,509 detik. Pada penelitian [19] melakukan deployment aplikasi server untuk *mobile* dengan rata – rata penggunaan RAM sebesar 1128 MB.

Pada penulisan Tugas Akhir ini, penulis menggunakan metode *Continuous Deployment* pada proses *deployment* aplikasi ke *Cloud Environment* untuk mengurangi *time cost* dan penggunaan sumber daya komputasi.

1.2. Perumusan Masalah

Adapun rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana cara mengaplikasikan metode *Continuous Deployment* untuk automasi proses *deployment*?
2. Bagaimana cara mengurangi *time-cost* dan penggunaan sumber daya komputasi menggunakan metode *Continuous Deployment*?
3. Bagaimana tingkat keunggulan sistem yang dibuat dibandingkan dengan sistem yang sudah ada?

1.3. Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Penelitian dilakukan dengan menguji beberapa aplikasi berbasis server dari repositori yang bersifat *open source* untuk di-*deploy* menggunakan sistem yang dibuat.
2. Parameter hasil yang dilihat adalah *time-cost* atau waktu yang dibutuhkan untuk melakukan *deployment* serta penggunaan RAM dan CPU pada server *Cloud*.
3. Perbandingan hasil penelitian menggunakan hasil pengujian *Continuous Deployment* dengan layanan Github Action.
4. Penelitian ini tidak membahas masalah keamanan jaringan baik dari sisi pengembang maupun *Cloud Environment*.

1.4. Tujuan Penelitian

Penelitian ini memiliki tujuan sebagai berikut:

1. Membuat sistem *deployment automation* menggunakan metode *Continuous Deployment*.
2. Mengetahui seberapa unggul sistem yang dibuat berdasarkan *time-cost*.

1.5. Manfaat Penelitian

Adapun manfaat dalam penelitian ini adalah sebagai berikut:

1. Memudahkan pengembang dalam proses *deployment*.
2. Menghemat *time-cost* proses *deployment* menggunakan *Deployment Automation*.

1.6. Metodologi Penelitian

Di bawah ini merupakan tahapan metodologi dalam penulisan Tugas Akhir ini, sebagai berikut:

1. Metode Studi Pustaka dan Literatur

Pada metode ini penulis mencari dan mengumpulkan referensi berupa literatur yang terdapat pada jurnal, buku dan internet mengenai automasi *deployment*, dan CI/CD dalam penerapannya ke Lingkungan *Cloud*.

2. Metode Konsultasi

Pada metode ini peneliti melakukan konsultasi kepada pihak-pihak yang memiliki pengetahuan serta wawasan yang baik dalam mengatasi permasalahan yang ditemui pada penulisan tugas akhir Optimasi *Deployment* Aplikasi Berbasis Server ke *Cloud Environment* dengan Metode *Continuous Deployment* di antaranya dosen dan praktisi.

3. Metode Pembuatan Algoritma

Pada metode ini peneliti melakukan pembuatan algoritma dan menganalisisnya hingga menemukan algoritma yang paling unggul menggunakan metode perhitungan kompleksitas waktu atau *time complexity*.

4. Metode Pengujian

Pada metode ini penulis melakukan dua buah pengujian. Pengujian pertama dilakukan dengan menggunakan sistem yang dibuat menggunakan metode *Continuous Deployment*. Pengujian kedua dilakukan untuk mendapatkan data variabel yang diuji menggunakan salah satu platform *Continuous Deployment*, yaitu Github Actions. Kedua pengujian ini menggunakan aplikasi berbasis server dari repositori bersifat *open source* yang sama.

5. Metode Analisis dan Penarikan Kesimpulan

Hasil dari pengujian pada tugas akhir ini akan dianalisis menggunakan metode analisis uji signifikansi untuk menguji apakah terdapat perbedaan signifikan hasil pengujian antara setiap skenario dari

sistem yang dibuat, dan juga antara sistem yang dibuat dengan sistem pembandingan.

1.7. Sistematika Penulisan

Berikut sistematika penulisan dalam penelitian ini:

BAB I PENDAHULUAN

Pada Bab I penelitian ini, mencantumkan latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada Bab II penelitian ini, mencantumkan penjelasan dari teori-teori yang berkaitan dengan *Deployment Automation*, *Continuous Deployment*, dan teori-teori lainnya yang berkaitan dengan penelitian skripsi.

BAB III METODOLOGI

Pada Bab III penelitian ini, mencantumkan tahapan-tahapan penelitian yang dilakukan berupa proses perancangan arsitektur *Cloud*, perancangan arsitektur sistem dan skenario, serta perancangan pengujian yang digunakan pada skripsi ini.

BAB IV HASIL DAN PEMBAHASAN

Pada Bab IV penelitian ini, mencantumkan proses tahapan-tahapan penelitian dengan analisis dari hasil percobaan setiap skenario, serta analisis tingkat keunggulan sistem yang dibuat.

BAB V KESIMPULAN DAN SARAN

Pada Bab V penelitian ini, mencantumkan beberapa kesimpulan yang ditarik oleh peneliti dari hasil penjelasan dari bab-bab sebelumnya dan memberikan saran yang nantinya akan digunakan untuk penelitian selanjutnya.

DAFTAR PUSTAKA

- [1] A. Ciuffoletti, 'Automated Deployment of a Microservice-based Monitoring Infrastructure', in *Procedia Computer Science*, Elsevier B.V., 2015, pp. 163–172. Doi: 10.1016/j.procs.2015.09.232.
- [2] S. Vila, F. Guirado, and J. L. L rida, 'Cloud computing virtual machine consolidation based on stock trading forecast techniques', *Future Generation Computer Systems*, vol. 145, pp. 321–336, Aug. 2023, doi: 10.1016/j.future.2023.03.018.
- [3] E.  ner and A. H.  zer, 'An energy-aware combinatorial auction-based virtual machine scheduling model and heuristics for green cloud computing', *Sustainable Computing: Informatics and Systems*, vol. 39, Sep. 2023, doi: 10.1016/j.suscom.2023.100889.
- [4] M. G. Avram, 'Advantages and Challenges of Adopting Cloud Computing from an Enterprise Perspective', *Procedia Technology*, vol. 12, pp. 529–534, 2014, doi: 10.1016/j.protcy.2013.12.525.
- [5] D. Truong, 'How cloud computing enhances competitive advantages: A research model for small businesses', 2010. [Online]. Available: <https://www.researchgate.net/publication/273447113>
- [6] R. Buyya, J. Broberg, and A. Goscinski, 'CLOUD COMPUTING Principles and Paradigms Edited by', 2011. [Online]. Available: www.wowebook.com
- [7] S. Mysari and V. Bejgam, 'Continuous Integration and Continuous Deployment Pipeline Automation Using Jenkins Ansible', in *International Conference on Emerging Trends in Information Technology and Engineering, ic-ETITE 2020*, Institute of Electrical and Electronics Engineers Inc., Feb. 2020. Doi: 10.1109/ic-ETITE47903.2020.239.
- [8] M. Shahin, M. Ali Babar, and L. Zhu, 'Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices', *IEEE Access*, vol. 5. Institute of Electrical and Electronics

- Engineers Inc., pp. 3909–3943, 2017. Doi: 10.1109/ACCESS.2017.2685629.
- [9] T. Savor, M. Douglas, M. Gentili, L. Williams, K. Beck, and M. Stumm, ‘Continuous deployment at Facebook and OANDA’, in *Proceedings – International Conference on Software Engineering*, IEEE Computer Society, May 2016, pp. 21–30. Doi: 10.1145/2889160.2889223.
- [10] B. Pachev, G. K. Stuart, and C. Dawson, *Continuous Integration for HPC with Github Actions and Tapis*. 2022.
- [11] F. Zampetti, S. Geremia, G. Bavota, and M. Di Penta, ‘CI/CD Pipelines Evolution and Restructuring: A Qualitative and Quantitative Study’, 2021. [Online]. Available: <https://madnight.github.io/githut/>
- [12] F. Fathurahman *et al.*, ‘AUTOMATE DEPLOYMENT APLIKASI WEB MENGGUNAKAN METODE GITOPS PADA KUBERNETES CLUSTER (Studi kasus: Web Riset Informatika Universitas Muhammadiyah Malang)’, *Jurnal: ElektriKA Borneo (JEB)*, vol. 9, no. 1, pp. 8–14, 2023.
- [13] D. G. Feitelson, E. Frachtenberg, and K. L. Beck, ‘Development and deployment at facebook’, *IEEE Internet Comput*, vol. 17, no. 4, pp. 8–17, 2013, doi: 10.1109/MIC.2013.25.
- [14] S. A. I. B. S. Arachchi and I. Perera, ‘Continuous integration and continuous delivery pipeline automation for agile software project management’, in *MERCon 2018 – 4th International Multidisciplinary Moratuwa Engineering Research Conference*, Institute of Electrical and Electronics Engineers Inc., Jul. 2018, pp. 156–161. Doi: 10.1109/MERCon.2018.8421965.
- [15] N. S. Aji and A. L. Dwi, ‘IMPLEMENTASI CONTINUOUS INTEGRATION/CONTINUOUS DELIVERY (CI/CD) PADA PERFORMANCE TESTING DEVOPS’, 2022.

- [16] F. Anggawie, D. Sulisty Kusumo, and D. Richasdy, 'PERANCANGAN SISTEM OTOMATISASI CONTINUOUS DEPLOYMENT BERBASIS MONOLITHIC REPOSITORY', *TELKATIKA*, vol. 2, no. 1, 2022.
- [17] N. Singh, 'CI/CD Pipeline for Web Applications', *Int J Res Appl Sci Eng Technol*, vol. 11, no. 5, pp. 5218–5226, May 2023, doi: 10.22214/ijraset.2023.52867.
- [18] M. R. Pratama and D. Sulistiyo Kusumo, 'Implementation of Continuous Integration and Continuous Delivery (CI/CD) on Automatic Performance Testing', in *2021 9th International Conference on Information and Communication Technology (ICoICT)*, IEEE, Aug. 2021, pp. 230–235. Doi: 10.1109/ICoICT52021.2021.9527496.
- [19] S. Deng *et al.*, 'Optimal Application Deployment in Resource Constrained Distributed Edges', *IEEE Trans Mob Comput*, vol. 20, no. 5, pp. 1907–1923, May 2021, doi: 10.1109/TMC.2020.2970698.
- [20] A. W. W. NUGRAHA, I. ROSYADI, and F. KHOERULLATIF, 'Penerapan DevOps pada Sistem Tertanam dengan ESP8266 menggunakan Mekanisme Over the Air', *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 9, no. 3, p. 678, Jul. 2021, doi: 10.26760/elkomika.v9i3.678.
- [21] A. Ahmed and G. Pierre, 'Docker Container Deployment in Fog Computing Infrastructures', pp. 1–8, 2018, doi: 10.1109/EDGE.2018.00008i.
- [22] A. El Amine, O. Brun, S. Abdellatif, and P. Berthou, 'Shortening the Deployment Time of SFCs by Adaptively Querying Resource Providers', 2021, doi: 10.1109/GLOBECOM46510.2021.9685356i.
- [23] M. H. Rifa'i Istifarulah and R. Tiaharyadini, 'DevOps, Continuous Integration and Continuous Deployment Methods for Software Deployment Automation', 2023.

- [24] I. C. Donca, O. P. Stan, M. Misaros, D. Gota, and L. Miclea, ‘Method for Continuous Integration and Deployment Using a Pipeline Generator for Agile Software Projects’, *Sensors*, vol. 22, no. 12, Jun. 2022, doi: 10.3390/s22124637.
- [25] M. Lestari, A. Iriani, and H. Hendry, ‘Information Technology Governance Design in DevOps-Based E-Marketplace Companies Using COBIT 2019 Framework’, *INTENSIF: Jurnal Ilmiah Penelitian dan Penerapan Teknologi Sistem Informasi*, vol. 6, no. 2, pp. 233–252, Aug. 2022, doi: 10.29407/intensif.v6i2.18104.
- [26] A. R. Sabau, S. Hacks, and A. Steffens, ‘Implementation of a continuous delivery pipeline for enterprise architecture model evolution’, *Softw Syst Model*, vol. 20, no. 1, pp. 117–145, Feb. 2021, doi: 10.1007/s10270-020-00828-z.
- [27] L. N. Vijouyeh, M. Sabaei, J. Santos, T. Wauters, B. Volckaert, and F. De Turck, *Efficient Application Deployment in Fog-enabled Infrastructures*. 2020.
- [28] J. Liu and Q. Huang, ‘An Application Deployment Approach for City IoT Applications in Resource-Constrained Edge Computing Environments’, 2020.
- [29] V. Arulkumar and R. Lathamanju, ‘Start to Finish Automation Achieve on Cloud with Build Channel: By DevOps Method’, in *Procedia Computer Science*, Elsevier B.V., 2019, pp. 399–405. Doi: 10.1016/j.procs.2020.01.032.
- [30] K. C. Wu, W. Y. Liu, and S. Y. Wu, ‘Dynamic Deployment and Cost-Sensitive Provisioning for Elastic Mobile Cloud Services’, *IEEE Trans Mob Comput*, vol. 17, no. 6, pp. 1326–1338, Jun. 2018, doi: 10.1109/TMC.2017.2771427.

- [31] Z. Sampedro, A. Holt, and T. Hauser, ‘Continuous integration and delivery for HPC: Using Singularity and Jenkins’, in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Jul. 2018. Doi: 10.1145/3219104.3219147.
- [32] M. Shahin, M. Ali Babar, and L. Zhu, ‘Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices’, *IEEE Access*, vol. 5. Institute of Electrical and Electronics Engineers Inc., pp. 3909–3943, 2017. Doi: 10.1109/ACCESS.2017.2685629.
- [33] M. Shahin, M. Zahedi, M. Ali Babar, and L. Zhu, ‘Adopting continuous delivery and deployment: Impacts on team structures, collaboration and responsibilities’, in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Jun. 2017, pp. 384–393. Doi: 10.1145/3084226.3084263.
- [34] M. Shahin, M. Ali Babar, M. Zahedi, and L. Zhu, ‘Beyond Continuous Delivery: An Empirical Investigation of Continuous Deployment Challenges’, 2017. [Online]. Available: <http://www.qsrinternational.com/>
- [35] P. Perera, R. Silva, and I. Perera, ‘Improve software quality through practicing DevOps’, in *17th International Conference on Advances in ICT for Emerging Regions, ICTer 2017 – Proceedings*, Institute of Electrical and Electronics Engineers Inc., Jul. 2017, pp. 13–18. Doi: 10.1109/ICTER.2017.8257807.
- [36] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano, ‘DevOps’, *IEEE Softw*, vol. 33, no. 3, pp. 94–100, May 2016, doi: 10.1109/MS.2016.68.
- [37] M. Shahin, M. A. Babar, and L. Zhu, ‘The Intersection of Continuous Deployment and Architecting Process: Practitioners’ Perspectives’, in *International Symposium on Empirical Software Engineering and Measurement*, IEEE Computer Society, Sep. 2016. Doi: 10.1145/2961111.2962587.

- [38] A. A. U. Rahman, E. Helms, L. Williams, and C. Parnin, ‘Synthesizing Continuous Deployment Practices Used in Software Development’, in *2015 Agile Conference*, New York, NY, USA: IEEE, Aug. 2015, pp. 1–10. Doi: 10.1109/Agile.2015.12.
- [39] A. Dearle, ‘Software Deployment, Past, Present and Future’, 2007.
- [40] Muhammad Touseef, Saira Siddiqui, and Nabeela Farah, ‘Understanding the Role of Automation in Society and its Impact on Labour Market’, *International Journal of Asian Business and Management*, vol. 2, no. 4, pp. 543–558, Aug. 2023, doi: 10.55927/ijabm.v2i4.3042.
- [41] K. Gallaba, ‘Improving the Robustness and Efficiency of Continuous Integration and Deployment’, 2019. [Online]. Available: https://github.com/software-rebels/hansel_and_gretel
- [42] P. M. Mell and T. Grance, ‘The NIST definition of cloud computing’, Gaithersburg, MD, 2011. Doi: 10.6028/NIST.SP.800-145.
- [43] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, ‘Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility’, *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009, doi: <https://doi.org/10.1016/j.future.2008.12.001>.
- [44] M. Armbrust *et al.*, ‘A View of Cloud Computing’, *Commun. ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010, doi: 10.1145/1721654.1721672.
- [45] R. S. Couto *et al.*, ‘Building an IaaS cloud with droplets: a collaborative experience with OpenStack’, *Journal of Network and Computer Applications*, vol. 117, pp. 59–71, Sep. 2018, doi: 10.1016/j.jnca.2018.05.016.
- [46] S. Dash and S. K. Pani, ‘E-Governance Paradigm Using Cloud Infrastructure: Benefits and Challenges’, in *Procedia Computer Science*, Elsevier B.V., 2016, pp. 843–855. Doi: 10.1016/j.procs.2016.05.274.

- [47] S. B. Dash, H. Saini, T. C. Panda, and A. Mishra, 'A Theoretical Aspect of Cloud Computing Service Models and Its Security Issues: A Paradigm', 2014. [Online]. Available: www.ijera.com
- [48] L. B. A. Rabai, M. Jouini, A. Ben Aissa, and A. Mili, 'A cybersecurity model in cloud computing environments', *Journal of King Saud University – Computer and Information Sciences*, vol. 25, no. 1, pp. 63–75, Jan. 2013, doi: 10.1016/j.jksuci.2012.06.002.
- [49] S. Subashini and V. Kavitha, 'A survey on security issues in service delivery models of cloud computing', *Journal of Network and Computer Applications*, vol. 34, no. 1. Pp. 1–11, Jan. 2011. Doi: 10.1016/j.jnca.2010.07.006.
- [50] D. Jones *et al.*, 'Towards integrated version control of virtual and physical artefacts in new product development: Inspirations from software engineering and the digital twin paradigm', in *Procedia CIRP*, Elsevier B.V., 2021, pp. 283–288. Doi: 10.1016/j.procir.2021.05.121.
- [51] L. Bulteau, P. Y. David, and F. Horn, 'The Problem of Discovery in Version Control Systems', in *Procedia Computer Science*, Elsevier B.V., 2023, pp. 209–216. Doi: 10.1016/j.procs.2023.08.231.
- [52] M. J. Rochkind, 'The source code control system', *IEEE Trans. Software Eng*, vol. 1, no. 4, pp. 364–3370, 1975.
- [53] W. F. Tichy, 'Design, Implementation and Evaluation of a Revision Control Design, Implementation and Evaluation of a Revision Control System', 1982. [Online]. Available: <https://docs.lib.purdue.edu/cstech/323>
- [54] D. Grune, 'Concurrent Versions System, A Method for Independent Cooperation', 1986.
- [55] A. Sudhir Vatare and P. Adkar, 'Review Paper on Centralized and Distributed Version Control System', 2019.

- [56] D. J. Barrett and R. E. Silverman, *SSH, the secure shell : the definitive guide*. O'Reilly, 2001.
- [57] A. E. Hassan, 'Mining Software Repositories to Assist Developers and Support Managers', 2006.
- [58] S. Gayathri Devi, K. Selvam, and S. P. Rajagopalan, 'An abstract to calculate big o factors of time and space complexity of machine code', in *IET Conference Publications*, 2011, pp. 844–847. Doi: 10.1049/cp.2011.0483.
- [59] A. Tarek, 'A Generalized Set Theoretic Approach for Time and Space Complexity Analysis of Algorithms and Functions', 2006.
- [60] S. W. Looney and J. L. Hagan, 'Statistical Methods for Assessing Biomarkers and Analyzing Biomarker Data', in *Essential Statistical Methods for Medical Statistics*, Elsevier, 2011, pp. 27–65. Doi: 10.1016/B978-0-444-53737-9.50005-0.
- [61] A. P. King and R. J. Eckersley, 'Inferential Statistics IV: Choosing a Hypothesis Test', in *Statistics for Biomedical Engineers and Scientists*, Elsevier, 2019, pp. 147–171. Doi: 10.1016/B978-0-08-102939-8.00016-5.
- [62] R. H. RIFFENBURGH, 'Chapter Summaries', in *Statistics in Medicine*, Elsevier, 2006, pp. 533–580. Doi: 10.1016/B978-012088770-5/50067-8.
- [63] K. Fang and M. Yang, 'A Practical Approach to Model Validation', in *Model Engineering for Simulation*, Elsevier, 2019, pp. 123–161. Doi: 10.1016/B978-0-12-813543-3.00007-X.
- [64] S. F. Sawyer, 'Analysis of Variance: The Fundamental Concepts', *Journal of Manual & Manipulative Therapy*, vol. 17, no. 2, pp. 27E-38E, Apr. 2009, doi: 10.1179/jmt.2009.17.2.27e.
- [65] H. Abdi and L. J. Williams, 'Encyclopedia of Research Design', 2010. [Online]. Available: <http://www.utd.edu/>

- [66] A. Dinno, 'Nonparametric pairwise multiple comparisons in independent groups using Dunn's test', 2015.
- [67] M. L. McHugh, 'Multiple comparison analysis testing in ANOVA', 2011.
- [68] J. Loeliger and M. McCullough, *Version Control with Git: Powerful tools and techniques for collaborative software development*, 3rd ed. O'Reilly Media, Inc., 2022.
- [69] A. Rawat, 'A Review on Python Programming', *International Journal of Research in Engineering, Science and Management*, vol. 3, no. 12, pp. 8–11, 2020.
- [70] E. Bisong, 'An Overview of Google Cloud Platform Services', in *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, Berkeley, CA: Apress, 2019, pp. 7–10. Doi: 10.1007/978-1-4842-4470-8_2.
- [71] J. Moura and D. Hutchison, 'Review and analysis of networking challenges in cloud computing', *Journal of Network and Computer Applications*, vol. 60, pp. 113–129, Jan. 2016, doi: 10.1016/j.jnca.2015.11.015.
- [72] R. W. Hornung and L. D. Reed, 'Estimation of Average Concentration in the Presence of Nondetectable Values', *Appl Occup Environ Hyg*, vol. 5, no. 1, pp. 46–51, Jan. 1990, doi: 10.1080/1047322X.1990.10389587.
- [73] D. B. Richardson and A. Ciampi, 'Effects of exposure measurement error when an exposure variable is constrained by a lower limit', *Am J Epidemiol*, vol. 157, no. 4, pp. 355–363, Feb. 2003, doi: 10.1093/aje/kwf217.
- [74] S. S. Shapiro and ; M B Wilk, 'An Analysis of Variance Test for Normality (Complete Samples)', 1965.