

PERANCANGAN DAN IMPLEMENTASI CI/CD
(*Continuous Integration/Continuous Deployment*)
MENGGUNAKAN DOCKER UNTUK OPTIMALISASI
PENGEMBANGAN APLIKASI

SKRIPSI

Oleh :
Lutfy Farisal Aryandi
201810225192



PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BHAYANGKARA JAKARTA RAYA
2025

LEMBAR PERSETUJUAN PEMBIMBING

Judul Skripsi : Implementasi Metode CI/CD
(Continuous Integration/Continuous Deployment)
Dengan Docker Container

Nama Mahasiswa : Lutfy Farisal Aryandi

Nomor Pokok Mahasiswa : 201810225192

Program Studi/Fakultas : Informatika / Ilmu Komputer

Tanggal Lulus Ujian Skripsi : 20 Januari 2025

Jakarta, 20 Januari 2025

MENYETUJUI,

Pembimbing I

Pembimbing II


Rakhmat Purnomo, S.Pd., S.Kom., M.Kom.
NIDN. 0322108201


Asep Ramdhani Mahbub, S.Kom., M.Kom.
NIDN. 0329087703

LEMBAR PENGESAHAN

Judul Skripsi : Implementasi Metode CI/CD
(Continuous Integration/Continuous Deployment)
Dengan Docker Container

Nama Mahasiswa : Lutfy Farisal Aryandi

Nomor Pokok Mahasiswa : 201810225192

Program Studi/Fakultas : Informatika / Ilmu Komputer

Tanggal Lulus Ujian Skripsi : 20 Januari 2025

Jakarta, 27 Januari 2025

MENGESAHKAN,

Ketua Tim Penguji : Adi Muhajirin, S.Kom., M.Kom., M.M.

NIDN : 0318038501

Penguji I : Allan Desi Alexander, S.T., M.Kom.

NIDN : 0305127404

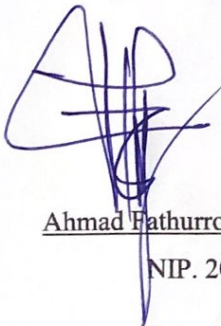
Penguji II : Rakhmat Purnomo, S.Pd., S.Kom., M.Kom.

NIDN : 0322108201

MENGETAHUI,

Ketua

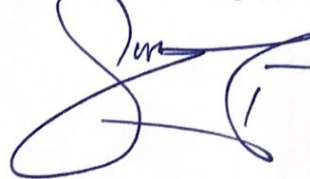
Program Studi Informatika



Ahmad Fathurrozi, SE., M.M.S.I
NIP. 2012486

Dekan

Fakultas Ilmu Komputer

Dr. Robertus Suraji, S.S., M.A

NIP. 1904406



LEMBAR PERNYATAAN BUKAN PLAGIASI

Yang bertanda tangan dibawah ini :

Nama : Lutfy Farisal Aryandi
NPM : 201810225192
Program Studi : Informatika
Fakultas : Ilmu Komputer
Judul Tugas Akhir : Implementasi Metode CI/CD (Continuous Integration /
Continuous Deployment) Dengan Docker Container

Dengan ini menyatakan bahwa hasil penulisan skripsi yang telah saya buat ini merupakan **hasil karya saya sendiri dan benar keasliannya**. Apabila dikemudian hari penulisan skripsi ini merupakan plagiat atau penjiplakan terhadap karya orang lain, maka saya bersedia mempertanggungjawabkan sekaligus bersedia menerima sanksi berdasarkan tata tertib di Universitas Bhayangkara Jakarta Raya.

Demikian pernyataan ini saya buat dalam keadaan sadar dan tidak dipaksakan dari pihak manapun.

Jakarta, 20 Januari 2025

Penulis



Lutfy Farisal Aryandi

ABSTRAK

Lutfy Farisal Aryandi. 201810225192. Implementasi Metode CI/CD (Continuous Integration/Continuous Deployment) dengan docker container. Dalam pengembangan sebuah aplikasi yang mencakup proses analisis kebutuhan, *planning*, *development*, *testing* dan *deployment*. Masalah umum yang sering terjadi sering kali berpusat dalam fase *development* seperti perubahan *requirement* atau *bug fixing*. Masalah lain selain dalam fase *development* adalah pada fase *deployment*. Fase ini merupakan fase krusial pada pengembangan sebuah aplikasi yang dimana fase terakhir sebelum kembali ke fase *development*. Keterlamabatan perilisan aplikasi sering terjadi karena ketergantungan perilisan *human input* ataupun kesulitan yang timbul dalam implementasi versi baru aplikasi tersebut. Melihat hal tersebut sebagai suatu permasalahan yang dapat menghambatnya proses perilisan aplikasi maka dilakukan proses implementasi CI/CD (*Continuous Integration Continuous Deployment*). CI/CD (*Continuous Integration Continuous Deployment*) adalah suatu praktek dimana sistem akan otomatis diintegrasikan dengan suatu layanan pihak ketiga dengan tujuan untuk code testing atau untuk menggabungkan beberapa layanan ke dalam aplikasi dalam pipeline CI/CD yang dijalankan secara otomatis jika ada *triggered event* yang memicunya seperti *event push* ke repository. Pipeline CI/CD bisa diklasifikasikan ke beberapa stage untuk memisahkan antara bagian testing, integrating, approval dan deployment. Dengan CI/CD, otomatisasi berlangsung dari sejak penulisan kode (*coding*), pengujian (*testing*), hingga penerapan (*deployment*). Contoh kasusnya ketika sebuah aplikasi rekening nasabah hendak merilis versi terbarunya. perubahan maupun penambahan kode yang dibuat para *developer* akan diintegrasikan lalu diuji, pengujian fungsional akan dilakukan otomatis. Jika sudah lolos uji, proses *deployment* akan berlangsung dengan hasil akhir aplikasi yang nantinya akan sampai ke pengguna.

Kata kunci : CICD, *Continuous Integration Continuous Deployment*, DevOps, Docker

ABSTRACT

Lutfy Farisal Aryandi. 201810225192. *Implementation of CI/CD (Continuous Integration/Continuous Deployment) Method with docker container. In the development of an application that includes the process of analyzing needs, planning, development, testing and deployment. Common problems that often occur are often centered in the development phase such as changing requirements or bug fixing. Another problem besides in the development phase is in the deployment phase. This phase is a crucial phase in the development of an application which is the last phase before returning to the development phase. Delays in the release of applications often occur due to dependence on the release of human input or difficulties that arise in implementing new versions of the application. Seeing this as a problem that can hinder the application release process, the CI / CD (Continuous Integration Continuous Deployment) implementation process is carried out. CI / CD (Continuous Integration Continuous Deployment) is a practice where the system will be automatically integrated with a third party Service with the aim of code testing or to combine several services into the application in a CI / CD pipeline that is run automatically if there is a triggered event that triggers it such as a push event to the repository. The CI/CD pipeline can be classified into stages to separate testing, integrating, approval and deployment. With CI/CD, automation takes place from coding, testing, to deployment. For example, when a customer Account application is about to release its latest version, changes or additions to the code made by developers will be integrated and then tested, functional testing will be carried out automatically. If it has passed the test, the deployment process will take place with the final result of the application that will later reach the user.*

Keywords : *CICD, Continuous Integration Continuous Deployment, DevOps, Docker*

LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI

KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIK

Sebagai sivitas akademik Universitas Bhayangkara Jakarta Raya, saya yang bertanda tangan di bawah ini :

Nama : Lutfy Farisal Aryandi
 NPM : 201810225192
 Program Studi : Informatika
 Fakultas : Ilmu Komputer
 Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Bhayangkara Jakarta Raya **Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty-Free Right*)**, atas karya ilmiah saya yang berjudul :

Implementasi Metode CI/CD (Continuous Integration / Continuous Deplyment) Dengan Docker Container

beserta perangkat yang ada (bila diperlukan). Dengan hak bebas royalti non-eksklusif ini, Universitas Bhayangkara Jakarta Raya berhak menyimpan, mengalihmediakan, mengelolanya dalam bentuk pangkalan data (*database*), mendistribusikannya dan mempublikasikannya di Internet atau media lain untuk kepentingan akademis tanpa perlu meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemilik hak cipta.

Segala bentuk tuntutan hukum yang timbul atas pelanggaran hak cipta dalam karya ilmiah ini menjadi tanggung jawab saya pribadi

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Jakarta
 Pada tanggal : 20 Januari 2025

Yang Menyatakan



 Lutfy Farisal Aryandi

KATA PENGANTAR

Dengan penuh rasa syukur, penulis panjatkan kepada Allah SWT berkat rahmat dan karunia-nya penulis dapat menyelesaikan skripsi ini dengan baik. Adapun judul skripsi yang penulis gunakan adalah “Implementasi Metode CI/CD (Continuous Integration/Continuous Deployment) dengan Docker Container”.

Penyusunan skripsi ini merupakan salah satu syarat untuk memperoleh gelar sarjana pada Fakultas Ilmu Komputer Universitas Bhayangkara Jakarta Raya. Dalam penulisan skripsi ini, penulis tidak lupa mengucapkan banyak terima kasih kepada semua pihak yang telah membantu dalam penulisan skripsi ini. Oleh karena itu penulis ingin menyampaikan rasa penghargaan dan terima kasih sebesar-besarnya kepada:

1. Bapak Irjen Pol. (Purn) Dr. Drs. Bambang Karsono, S.H., M.M. Selaku Rektor Universitas Bhayangkara Jakarta Raya.
2. Bapak Dr. Robertus Suraji, S.S., M.A. Selaku Dekan Fakultas Ilmu Komputer Universitas Bhayangkara Jakarta Raya.
3. Bapak Ahmad Fathurrozi, S.E., M.M.S.I. Selaku Ketua Program Studi Informatika Fakultas Ilmu Komputer Universitas Bhayangkara Jakarta Raya.
4. Bapak Rakhmat Purnomo, S.Pd., S.Kom., M.Kom. Selaku Dosen Pembimbing I, dalam penulisan skripsi di Universitas Bhayangkara Jakarta Raya yang telah banyak memberikan arahan dan membantu dalam penulisan skripsi.
5. Bapak Asep Ramdhani Mahbub, S.Kom., M.Kom. Selaku Dosen Pembimbing II, dalam penulisan skripsi di Universitas Bhayangkara Jakarta Raya yang telah banyak memberikan arahan dan membantu dalam penulisan skripsi.
6. Keluarga tercinta terutama ibu saya yang selalu memberikan doa, semangat serta dukungan dalam proses penulisan skripsi.
7. Teman-teman semua yang telah banyak membantu memberikan masukan dan motivasi, khususnya teman-teman dari Fakultas Ilmu Komputer yang selalu mendukung dalam melaksanakan penulisan skripsi ini.

Semoga Tuhan Yang Maha Esa memberikan balasan yang setimpal kepada semua pihak yang telah memberikan bimbingan, bantuan dan nasihat. Penulis

menyadari bahwa penyusunan skripsi ini belum sempurna baik penulisan maupun isi karena keterbatasan kemampuan penulis. Oleh karena itu, penulis mengharapkan kritik dan saran yang sifatnya membangun dari pembaca untuk penyempurnaan isi skripsi ini dan pengembangan aplikasi untuk dapat menjadi lebih baik lagi dikemudian hari.

Akhir kata penulis mengucapkan banyak terima kasih, semoga skripsi ini dapat berguna dan bermanfaat bagi semua pembaca dan semoga Allah SWT selalu melimpahkan rahmat dan karunia-Nya kepada kita semua.

Jakarta, 20 Januari 2025

Penulis,

Lutfy Farisal Aryandi



DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN PEMBIMBING	i
LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN BUKAN PLAGIASI	iii
ABSTRAK	iv
ABSTRACT	v
LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI	vi
KATA PENGANTAR.....	vii
DAFTAR TABEL	ix
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN.....	14
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	3
1.3 Rumusan Masalah	3
1.4 Batasan Masalah.....	3
1.5 Tujuan dan Manfaat Penelitian.....	4
1.5.1 Tujuan	4
1.5.2 Manfaat	4
1.6 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI	6
2.1 Tinjauan Pustaka	6
2.2 Landasan Teori	10
2.2.1 Aplikasi	10
2.2.2 CI/CD	10
2.2.3 Continuous Integration	11

2.2.4	<i>Continuous Deployment</i>	11
2.2.5	Gitlab.....	11
2.2.6	Docker	12
2.2.7	Manfaat Docker.....	12
2.2.8	Metodologi pengembangan aplikasi	13
2.2.9	<i>Unified Modelling Language (UML)</i>	14
BAB III METODOLOGI PENELITIAN		16
3.1	Tempat dan Waktu Penelitian	16
3.2	Kerangka Pikir Penelitian.....	16
3.3	Metode Penentuan Konsep	17
3.4	Metode Implementasi	17
3.5	Metode Pengembangan	19
3.6	Kebutuhan Perangkat Lunak	20
4.2	Kebutuhan Perangkat Keras	21
BAB IV HASIL DAN PEMBAHASAN.....		21
4.1	Hasil.....	21
4.2.1	<i>Defind requirement</i>	21
4.2.1	<i>System design</i>	22
4.2.2	<i>Git Flow</i>	23
4.2.3	<i>Activity Diagram</i> Alur Perilisan Aplikasi	24
4.2	Product Backlog	25
4.3	<i>Priority Backlog</i>	28
4.4	Backlog Child.....	30
4.5	<i>Sprint Pertama</i>	31
4.6	Penambahan <i>Backlog Sprint Pertama</i>	31
4.7	Review Sprint Pertama	32
4.8	Sprint Kedua.....	33
4.9	Penambahan <i>Backlog Sprint Kedua</i>	33
4.10	Review Sprint Kedua	34

4.11	Sprint Ketiga	35
4.12	Penambahan <i>Backlog Sprint</i> Ketiga	35
4.13	Review Sprint Ketiga.....	36
4.14	<i>Sprint</i> Keempat	37
4.15	Review Sprint Keempat	37
4.16	Perubahan <i>Product Backlog</i>	38
4.2.4	<i>Implementation system and Integration system</i>	40
4.2.5	<i>System testing result</i>	57
4.17	Pembahasan	59
BAB V PENUTUP		60
5.1	Kesimpulan.....	60
5.2	Saran	60
DAFTAR PUSTAKA		61
LAMPIRAN		46

DAFTAR TABEL

Tabel 2.1 <i>State of The Art</i>	8
Tabel 2.2 <i>Simbol-Simbol Activity Diagram</i>	15
Tabel 4.1 <i>Hasil Product Backlog</i>	26
Tabel 4.2 <i>Priority Backlog</i>	28
Tabel 4.3 <i>Back Child</i>	30
Tabel 4.4 <i>Sprint Pertama</i>	31
Tabel 4.5 <i>Penambahan Backlog Sprint Pertama</i>	31
Tabel 4.6 <i>Review Sprint Pertama</i>	32
Tabel 4.7 <i>Sprint Kedua</i>	33
Tabel 4.8 <i>Penambahan Backlog Sprint Kedua</i>	33
Tabel 4.9 <i>Review Sprint Kedua</i>	34
Tabel 4.10 <i>Sprint Ketiga</i>	35
Tabel 4.11 <i>Penambahan Backlog Sprint Ketiga</i>	35
Tabel 4.12 <i>Review Sprint Ketiga</i>	36
Tabel 4.13 <i>Sprint Keempat</i>	37
Tabel 4.14 <i>Penambahan Backlog Sprint keempat</i>	37
Tabel 4.15 <i>Review Sprint Keempat</i>	37
Tabel 4.16 <i>Perubahan Product Backlog</i>	38
Tabel 4.17 <i>Hasil Pengujian Sistem</i>	57

DAFTAR GAMBAR

Gambar 2.1 <i>Docker</i>	13
Gambar 2.2 Tahapan <i>Agile Model</i>	13
Gambar 3.1 Kerangka Pikir	16
Gambar 3.2 Metode Pengembangan.....	19
Gambar 4.1 <i>Defined requirement</i>	22
Gambar 4.2 <i>System Design</i>	23
Gambar 4.3 Alur Kerja pengembangan pada Git	24
Gambar 4.4 <i>Activity Diagram</i> Alur Perilisan Aplikasi.....	25
Gambar 4.5 Perubahan <i>Backlog</i> Keseluruhan	40
Gambar 4.6 Gitlab Runner Cofig File	41
Gambar 4.7 Tampilan Mulai	42
Gambar 4.8 Dockerfile	44
Gambar 4.9 <i>Variables</i>	44
Gambar 4.10 Gitlab-ci	52
Gambar 4.11 Tampilan <i>CI/CD pipeline</i>	53
Gambar 4.12 <i>Pipeline jobs</i>	54
Gambar 4.13 Hasil pengujian	55
Gambar 4.14 Hasil <i>deploy</i>	56
Gambar 4.15 <i>Backend API</i>	57

DAFTAR LAMPIRAN

LAMPIRAN 1 Plagiasi	47
LAMPIRAN 2 Biodata	48
LAMPIRAN 3 Kartu Bimbingan.....	49

