

SKRIPSI
OPTIMASI *BUILD TIME CONTINUOUS INTEGRATION* PADA
***MULTI NODE JENKINS* DENGAN MENGGUNAKAN**
METODE *CACHE*



FEBRIYAN ADJI SAPUTRO

NIM : 195410236

PROGRAM STUDI INFORMATIKA
PROGRAM SARJANA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA

2024

SKRIPSI
OPTIMASI *BUILD TIME CONTINUOUS INTEGRATION* PADA
***MULTI NODE JENKINS* DENGAN MENGGUNAKAN**
METODE *CACHE*

Diajukan sebagai salah satu syarat untuk menyelesaikan studi



Disusun Oleh

FEBRIYAN ADJI SAPUTRO

NIM : 195410236

PROGRAM STUDI INFORMATIKA
PROGRAM SARJANA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA

2024

HALAMAN PERSETUJUAN

UJIAN SKRIPSI

Judul : **OPTIMASI *BUILD TIME CONTINUOUS INTEGRATION* PADA *MULTI NODE JENKINS* DENGAN MENGGUNAKAN METODE *CACHE***

Nama : **Febriyan Adji Saputro**

NIM : **195410236**

Program Studi : **Informatika**

Program : **Sarjana**


Semester : **Genap**

Tahun Akademik : **2023/2024**

Telah diperiksa dan disetujui untuk diujikan di hadapan Dewan Penguji
Skripsi

Yogyakarta, 29 Agustus 2024

Dosen Pembimbing,


M. Agung Nugroho, M.Kom.

NIDN : 0507078501

HALAMAN PENGESAHAN

SKRIPSI

**OPTIMASI BUILD TIME CONTINUOUS INTEGRATION PADA MULTI
NODE JENKINS DENGAN MENGGUNAKAN METODE CACHE**

Telah dipertahankan di depan Dewan Penguji Skripsi dan dinyatakan
diterima untuk memenuhi sebagian persyaratan guna memperoleh Gelar

Sarjana Komputer

Program Studi Informatika

Fakultas Teknologi Informasi

Universitas Teknologi Digital Indonesia

Yogyakarta

Yogyakarta, 29 Agustus 2024

Dewan Penguji

- | | NIDN | |
|---|------------|--|
| 1. Pius Dian Widi Anggoro, S.Si, M. Cs | 091157 | |
| 2. Dr. Bambang P.D.P., S.E., Ak., S.Kom.,
MMSI | 0525087201 | |
| 3. M. Agung Nugroho, M.Kom. | 0507078501 | |

Tanda Tangan



Mengetahui

Ketua Program Studi Informatika



Dipt. Lakta Sari, S.T., M.T.

NPP : 121172

PERNYATAAN KEASLIAN SKRIPSI

Dengan ini saya menyatakan bahwa naskah skripsi ini belum pernah diajukan untuk memperoleh gelar Sarjana Komputer di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara sah diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 28 Agustus 2024



Febriyan Adji Saputro

NIM : 195410236

HALAMAN PERSEMBAHAN

Skripsi ini penulis persembahkan kepada :

1. Kedua orang tua tercinta, Bapak Mustofa dan Ibu Zumaroh yang telah merawat, mendidik dengan baik dan senantiasa mendoakan kebaikan bagi penulis.
2. Dosen pembimbing skripsi M. Agung Nugroho, S.Kom., M.Kom. serta seluruh civitas akademika Universitas Teknologi Digital Indonesia yang telah membimbing dan membantu sehingga tugas akhir skripsi ini dapat terselesaikan.
3. Seluruh teman-teman kampus yang telah menemani, membantu dan mendoakan dalam rangka menyelesaikan tugas akhir ini

HALAMAN MOTTO

Bersungguh-sungguhlah untuk mendapatkan apa yang bermanfaat bagimu, dan mintalah pertolongan kepada Allah (dalam segala urusanmu) serta janganlah sekali-kali engkau merasa lemah. (HR. Muslim)

KATA PENGANTAR

Bismillahirrahmanirrahim

Segala puji bagi Allah *Subhanahu Wa Ta'ala* yang atas nikmat, pertolongan serta kehendak-Nya, penulis mampu menyelesaikan tugas akhir yang berjudul *OPTIMASI BUILD TIME CONTINUOUS INTEGRATION PADA MULTI NODE JENKINS DENGAN MENGGUNAKAN METODE CACHE* sebagai salah satu syarat menyelesaikan jenjang pendidikan Strata satu (S1) program studi Teknik informatika di Universitas Teknologi Digital Indonesia.

Penulisan tugas akhir ini juga tidak lepas dari bantuan dan dukungan dari berbagai pihak, sehingga pada kesempatan ini penulis ingin menyampaikan rasa terimakasih kepada semua pihak yang telah membantu dalam penulisan tugas akhir ini, antara lain :

1. Kedua orang tua yang senantiasa memberikan dukungan serta doa
2. Sri Redjeki, S. Si., M. Kom., Ph. D. selaku rektor Universitas Teknologi Digital Indonesia.
3. Dini Fakta Sari, S.T., M.T. selaku ketua program studi Informatika Universitas Teknologi Digital Indonesia.
4. Bapak M. Agung Nugroho, S.Kom., M.Kom selaku dosen pembimbing skripsi yang senantiasa memberikan dorongan serta masukan dalam menyelesaikan skripsi ini
5. Keluarga, teman serta semua orang yang mendukung dalam menyelesaikan tugas akhir skripsi ini hingga selesai .

Penulis menyadari bahwa tugas akhir ini masih jauh dari kata sempurna. Semoga tulisan ini dapat memberikan manfaat dan memberikan wawasan tambahan bagi para pembaca dan khususnya bagi penulis sendiri.

Yogyakarta, 28 Agustus 2024

Febriyan Adji Saputro

DAFTAR ISI

HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN.....	iv
PERNYATAAN KEASLIAN SKRIPSI.....	v
HALAMAN PERSEMBAHAN.....	vi
HALAMAN MOTTO.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xiii
INTISARI.....	xiv
ABSTRACT.....	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	3
1.3 Ruang Lingkup.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	6
2.1 Tinjauan Pustaka.....	6
2.2 Dasar Teori.....	8
2.2.1 <i>Jenkins</i>	8
2.2.2 <i>Continuous Integration, Delivery, dan Deployment (CI/CD)</i>	8
2.2.3 <i>Container</i>	9
2.2.4 <i>Docker</i>	9
2.2.5 <i>Amazon Web Service</i>	10
BAB III METODE PENELITIAN.....	11
3.1 Bahan.....	11
3.2 Peralatan.....	13
3.2.1 Kebutuhan Perangkat Keras.....	13

3.2.2	Kebutuhan perangkat lunak.....	14
3.3	Analisis dan Rancangan Sistem.....	14
3.4	Metode Pengujian.....	16
BAB IV IMPLEMENTASI DAN PEMBAHASAN.....		18
4.1	Implementasi.....	18
4.1.1	<i>Setup</i> repositori <i>GitHub</i>	18
4.1.2	Instalasi <i>Jenkins Master</i>	19
4.1.3	Instalasi <i>Jenkins Slave</i>	20
4.1.4	Instalasi <i>Docker Image Registry</i>	21
4.1.5	Membuat <i>S3 Bucket</i>	22
4.1.6	Membuat <i>Pipeline</i> pada <i>Jenkins</i>	22
4.2	Pengujian.....	25
4.2.1	Pengujian pada aplikasi <i>Python</i>	25
4.2.2	Pengujian pada aplikasi <i>Javascript</i>	27
4.2.3	Pengujian pada aplikasi <i>Ruby</i>	29
4.3	Pembahasan.....	30
4.3.1	<i>Dockerfile</i>	31
4.3.2	Arsitektur <i>Jenkins</i>	33
4.3.3	<i>Jenkins Pipeline</i>	34
4.3.4	<i>Docker Buildx</i>	35
4.3.5	Metode <i>cache</i>	35
BAB V KESIMPULAN DAN SARAN.....		38
5.1	Kesimpulan.....	38
5.2	Saran.....	38

DAFTAR GAMBAR

Gambar 3.1 Skema <i>Pipeline Continuous Integration</i> dengan metode <i>cache</i>	15
Gambar 3.2 Rancangan arsitektur <i>AWS</i>	16
Gambar 3.3 Diagram alir proses pengujian.....	17
Gambar 4.1 Hasil pembuatan 3 repositori GitHub.....	18
Gambar 4.2 Contoh isi repositori GitHub.....	19
Gambar 4.3 Halaman dashboard jenkins master.....	20
Gambar 4.4 Halaman untuk melihat daftar <i>jenkins nodes</i>	21
Gambar 4.5 Pengujian instalasi Docker Image Registry.....	22
Gambar 4.6 AWS S3 Bucket untuk penyimpanan <i>cache</i>	22
Gambar 4.7 Grafik <i>build time</i> aplikasi <i>python</i>	26
Gambar 4.8 Grafik ringkasan <i>build time</i> aplikasi javascript.....	28
Gambar 4.9 Grafik <i>build time</i> aplikasi <i>ruby</i>	29
Gambar 4.10 <i>Cache</i> pada <i>docker image layer</i>	31
Gambar 4.11 <i>Log build</i> aplikasi <i>javascript</i> menggunakan metode <i>cache S3</i>	32
Gambar 4.12 <i>Console output</i> lokasi <i>node job jenkins</i>	33
Gambar 4.13 Hasil eksekusi <i>jenkins pipeline</i>	34
Gambar 4.14 Isi folder penyimpanan <i>cache</i> secara lokal di <i>server</i>	36

DAFTAR TABEL

Tabel 2.1 Tinjauan Pustaka.....	6
Tabel 3.1 Daftar perangkat lunak untuk pengujian.....	12
Tabel 3.2 Daftar perangkat keras.....	13
Tabel 4.1 Hasil pengujian build time aplikasi <i>python</i>	25
Tabel 4.2 Penambahan kecepatan <i>build time</i> aplikasi <i>Python</i>	26
Tabel 4.3 Hasil pengujian <i>build time</i> aplikasi <i>javascript</i>	27
Tabel 4.4 Penambahan kecepatan rata-rata <i>build time</i> aplikasi <i>Javascript</i>	28
Tabel 4.5 Hasil pengujian <i>build time</i> aplikasi <i>ruby</i>	29
Tabel 4.6 Penambahan kecepatan rata-rata <i>build time</i> aplikasi <i>ruby</i>	30

INTISARI

Dalam era pengembangan perangkat lunak yang modern, kecepatan dan efisiensi proses *build* aplikasi menjadi faktor penting untuk meningkatkan produktivitas tim *developer*. *Jenkins* sebagai alat *CI/CD* yang populer dapat dimanfaatkan untuk menjalankan proses *build* secara terdistribusi di beberapa *server*. Namun, *build time* yang tinggi tetap menjadi tantangan, terutama untuk aplikasi berbasis *container* yang memerlukan banyak dependensi dan tahap kompilasi.

Penelitian ini mengusulkan metode optimasi *build time* dengan menggunakan *cache*. Metode ini memanfaatkan *cache* untuk menyimpan artefak *build* yang sering digunakan, sehingga mengurangi waktu yang diperlukan untuk proses kompilasi ulang pada *build* berikutnya. Dalam penelitian ini dilakukan eksperimen dengan membandingkan *build time* aplikasi berbasis *container* sebelum dan sesudah penerapan metode *caching*.

Hasil penelitian ini menunjukkan bahwa penggunaan metode *cache* pada aplikasi berbasis *container* dapat meningkatkan kecepatan *build time* hingga 76,31%. Hal ini menunjukkan bahwa metode *cache* sangat efektif dalam meningkatkan efisiensi *build time* pada arsitektur multi-node *Jenkins*. Dengan demikian, optimasi ini tidak hanya meningkatkan kecepatan proses *build*, melainkan juga dapat mengurangi konsumsi sumber daya komputasi, sehingga mendukung praktik *CI/CD* yang lebih efisien dan hemat biaya.

Kata Kunci: *Build, Cache, CI/CD, Docker, Jenkins*

ABSTRACT

In the era of modern software development, the speed and efficiency of the application build process are critical factors in enhancing the productivity of development teams. Jenkins, a popular CI/CD tool, can be utilized to run build processes distributed across multiple servers. However, high build times remain a challenge, particularly for container-based applications that require numerous dependencies and compilation stages.

This study proposes a method to optimize build time using caching. This method leverages cache to store frequently used build artifacts, thereby reducing the time required for recompilation in subsequent builds. In this research, experiments were conducted by comparing the build times of container-based applications before and after the application of the caching method.

The results of this study indicate that the use of caching in container-based applications can improve build time speed by up to 76,31%. This demonstrates that caching is highly effective in enhancing build time efficiency in a multi-node Jenkins architecture. Consequently, this optimization not only accelerates the build process but also reduces computational resource consumption, supporting more efficient and cost-effective CI/CD practices.

Keywords: *Build, Cache, CI/CD, Docker, Jenkins*