

**Perbandingan Performa pada Pembangunan API Sistem E-POK  
PT Pupuk Sriwidjaja Menggunakan *Echo Framework* Golang dan  
*Native Golang***

Diajukan Sebagai Syarat untuk Menyelesaikan  
Pendidikan Program Strata-1 pada  
Jurusan Teknik Informatika



Oleh:

Calvin Fadhil Mahendra

NIM: 09021381722115

JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS SRIWIJAYA  
2021

# LEMBAR PENGESAHAN TUGAS AKHIR

Perbandingan Performa pada Pembangunan API Sistem E-POK PT Pupuk  
Sriwidjaja Menggunakan Echo *Framework* Golang dan *Native* Golang

Oleh:

Calvin Fadhil Mahendra

NIM: 09021381722115

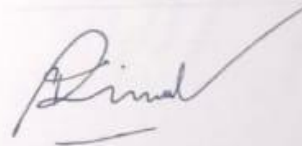
Palembang, 26 Juli 2021

Pembimbing I,



Samsuryadi, M.Kom., Ph.D.  
NIP. 197102041997021003

Pembimbing II,

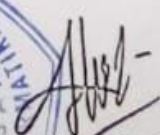


Mastura Diana Marieska, M.T.  
NIP. 198603212018032001

Mengetahui,

Ketua Jurusan Teknik Informatika



  
Alvi Syahrini Utami, M.Kom.  
NIP. 197812222006042003

## TANDA LULUS UJIAN SIDANG TUGAS AKHIR

Pada hari **Selasa** tanggal **6 Juli 2021** telah dilaksanakan ujian sidang tugas akhir oleh Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Sriwijaya.

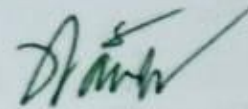
Nama : Calvin Fadhil Mahendra

NIM : 09021381722115

Judul : Perbandingan Performa pada Pembangunan API Sistem E-POK PT Pupuk Sriwidjaja Menggunakan Echo Framework Golang dan Native Golang.

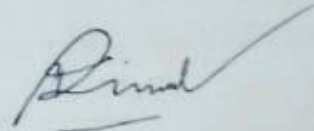
1. Pembimbing I

Samsuryadi, M.Kom., Ph.D.  
NIP. 197102041997021003



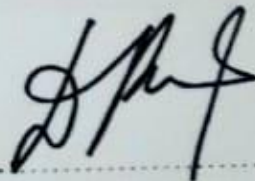
2. Pembimbing II

Mastura Diana Marieska, M.T.  
NIP. 198603212018032001



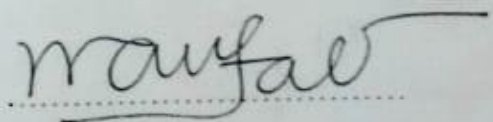
3. Penguji I

Dian Palupi Rini, M.kom., Ph.D  
NIP. 197802232006042002



4. Penguji II

Muhammad Naufal Rachmatullah, M.T  
NIP. 1671060112920006



Mengetahui,

Ketua Jurusan Teknik Informatika



Alvi Syahri Utami, M.Kom.  
NIP. 197812222006042003

## HALAMAN PERNYATAAN

Yang bertanda tangan dibawah ini:

Nama : Calvin Fadhil Mahendra

NIM : 09021381722115

Program Studi : Teknik Informatika Bilingual

Judul Skripsi : Perbandingan Performa pada Pembangunan API Sistem E-POK  
PT Pupuk Sriwidjaja Menggunakan Echo Framework Golang dan  
Native Golang.

Hasil Pengecekan *Software iThenticate/Turnitin* : 20%

Menyatakan bahwa Laporan proyek saya merupakan hasil karya sendiri dan bukan hasil penjiplakan/plagiat. Apabila ditemukan unsur penjiplakan/plagiat dalam laporan proyek ini, maka saya bersedia menerima sanksi akademik dari Universitas Sriwijaya sesuai dengan ketentuan yang berlaku.

Demikian, pernyataan ini saya buat dengan sebenarnya dan tidak ada paksaan oleh siapapun.



Palembang, Juli 2021



Calvin Fadhil Mahendra

NIM. 09021381722115

## MOTTO DAN PERSEMBAHAN

- *Sesungguhnya bersama kesulitan itu ada kemudahan.*
- *Dua musuh terbesar kesuksesan adalah penundaan dan alasan.*
- *Manusia bisa menunda, tetapi tidak dengan waktu.*

*Kupersembahkan karya tulis ini kepada :*

- *Allah SWT*
- *Orangtuaku tercinta*
- *Keluargaku*
- *Seseorang yang kusayangi dan kucintai*
- *Teman-teman yang sedang berjuang menghadapi skripsi*
- *Fakultas Ilmu Komputer Universitas Sriwijaya*

# **Performance Comparison on PT Pupuk Sriwidjaja's E-POK System API Development Using Echo *Framework* Golang and Native Golang**

**By:**  
**Calvin Fadhil Mahendra**  
**09021381722115**

## **ABSTRACT**

Frameworks are very helpful tools in building websites or applications. With all the conveniences that exist, users only need to think about the appearance and integration of the system, because the dependencies and data flows that have been neatly arranged have been prepared by the framework. Echo Framework in golang is a framework that moves on the server side which is used as an intermediary between the client and the server. In this study, the author wants to show the difference in execution time between the API route using the Echo Framework and without using the Framework or Native in the Golang programming language. By testing API routes using Jmeter concurrently with the same thread or user on both APIs 100,200,300,...,2000. On the results of studies with small threads or users, there is no significant time difference. However, for large threads or users, Echo Framework has a faster execution time than Native Golang.

Keyword: API, JMETER, Golang, Echo *Framework*.

# **Perbandingan Performa pada Pembangunan API Sistem E-POK PT Pupuk Sriwidjaja Menggunakan Echo *Framework* Golang dan Native Golang**

**Oleh:  
Calvin Fadhil Mahendra  
09021381722115**

## **ABSTRAK**

*Framework* merupakan alat yang sangat membantu dalam membangun situs web ataupun aplikasi. Dengan segala kemudahan yang tersedia hingga pengguna hanya perlu memikirkan tampilan dan integrasi sistem, karena dependensi dan alur data yang telah tertata rapi telah disiapkan oleh framwork. Echo *Framework* pada golang merupakan salah satu framework yang bergerak pada sisi server yang digunakan sebagai perantara antara *client* dan *server*. Pada penelitian ini penulis ingin menunjukkan perbedaan waktu eksekusi pada rute API yang menggunakan Echo *Framework* dan tanpa menggunakan *Framework* atau *Native* pada bahasa pemrograman Golang. Dengan menguji rute API menggunakan Jmeter secara simultan atau bersamaan dengan *threads* atau *user* yang sama pada kedua API 100,200,300,...,2000. Pada hasil penelitian dengan *threads* atau *user* yang kecil tidak terdapat perbedaan waktu yang signifikan. Tetapi untuk *threads* atau *user* berukuran besar Echo *Framework* memiliki waktu eksekusi yang lebih cepat daripada *Native* Golang.

Kata Kunci: API, JMETER, Golang, Echo *Framework*.

## KATA PENGANTAR



Puji syukur kepada Allah SWT atas berkat dan rahmat-Nya yang telah diberikan kepada Penulis sehingga dapat menyelesaikan Tugas Akhir ini dengan baik. Tugas Akhir ini disusun untuk memenuhi salah satu syarat guna menyelesaikan pendidikan program Strata-1 pada Fakultas Ilmu Komputer Program Studi Teknik Informatika di Universitas Sriwijaya.

Dalam menyelesaikan Tugas Akhir ini banyak pihak yang telah memberikan bantuan dan dukungan baik secara langsung maupun secara tidak langsung. Pada kesempatan ini, penulis ingin menyampaikan ucapan terima kasih kepada pihak-pihak yang telah membantu penulis dalam menyelesaikan Tugas Akhir ini, yaitu kepada:

1. Orang tuaku, Dwi Iswanto Krisubagio dan Elok Sriwahyuni, dan mbak Ade Novie Calista dan semua keluarga yang memberikan cinta dan kasih sayangnya untuk selalu mendoakan serta memberikan dukungan, serta Astri Viona yang selalu ada untuk membantu dan memberikan semangat dalam menyelesaikan penelitian.
2. Bapak Jaidan Jauhari, S.Pd., M.T. selaku Dekan Fakultas Ilmu Komputer Universitas Sriwijaya.
3. Ibu Alvi Syahrini Utami, M.Kom. selaku Ketua Jurusan Teknik Informatika yang telah banyak memberikan nasehat dan informasi dalam perkuliahan.
4. Bapak Samsuryadi, M.Kom., Ph.D. selaku pembimbing Tugas Akhir yang telah membimbing, mengarahkan dan memberikan motivasi penulis dalam proses perkuliahan dan pengerjaan Tugas Akhir.
5. Ibu Mastura Diana Marieska, M.T. selaku pembimbing Tugas Akhir yang telah membimbing, mengarahkan dan memberikan motivasi penulis dalam proses perkuliahan dan pengerjaan Tugas Akhir.
6. Bapak M. Fachrurrozi, M.T. selaku dosen pembimbing akademik, yang telah membimbing, mengarahkan, dan memberikan motivasi penulis dalam proses



perkuliahan.

7. Ibu Dian Palupi Rini, M.kom., Ph.D. selaku dosen penguji I, yang telah memberikan masukan dan dorongan dalam proses pengerjaan Tugas Akhir.
8. Bapak Muhammad Naufal Rachmatullah, M.T. selaku dosen penguji II, yang telah memberikan masukan dan dorongan dalam proses pengerjaan Tugas Akhir.
9. Seluruh dosen Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Sriwijaya yang telah memberikan ilmunya.
10. Kak Ricy, Mbak Wiwin, dan Pak Toni serta seluruh staf tata usaha yang telah membantu dalam kelancaran proses administrasi dan akademik selama masa perkuliahan.
11. Teman-teman dari Angkatan IF 2017, kakak tingkat, adik tingkat, serta teman-teman lainnya yang telah berjuang selama menempuh Pendidikan di Fakultas Ilmu Komputer Universitas Sriwijaya.
12. Teman-teman seperjuangan dan tim dolol Dede, Isan, Vandi, Berlian, Kak Trikur, Naufal Naufal, dan teman-teman lain yang mengisi waktu luang, bermain game dan berjuang bersama penulis.

Penulis menyadari dalam penyusunan Tugas Akhir ini masih terdapat banyak kekurangan disebabkan keterbatasan pengetahuan dan pengalaman, oleh karena itu kritik dan saran yang membangun sangat diharapkan untuk kemajuan penelitian selanjutnya.

Akhir kata semoga Tugas Akhir ini dapat berguna dan bermanfaat bagi kita semua.

Palembang, 31 Juli 2021



Calvin Fadhil Mahendra

## DAFTAR ISI

Halaman

<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>LEMBAR PENGESAHAN TUGAS AKHIR .....</b>	<b>ii</b>
<b>TANDA LULUS UJIAN SIDANG TUGAS AKHIR .....</b>	<b>iii</b>
<b>HALAMAN PERNYATAAN.....</b>	<b>iv</b>
<b>MOTTO DAN PERSEMBAHAN.....</b>	<b>v</b>
<b>ABSTRACT .....</b>	<b>vi</b>
<b>ABSTRAK .....</b>	<b>vii</b>
<b>KATA PENGANTAR.....</b>	<b>viii</b>
<b>DAFTAR ISI.....</b>	<b>x</b>
<b>DAFTAR TABEL.....</b>	<b>xv</b>
<b>DAFTAR GAMBAR.....</b>	<b>xvii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>I-1</b>
1.1 Pendahuluan .....	I-1
1.2 Latar Belakang .....	I-1
1.3 Rumusan Masalah .....	I-4
1.4 Tujuan Penelitian.....	I-4
1.5 Manfaat Penelitian.....	I-5
1.6 Batasan Masalah .....	I-5
1.7 Sistematika Penulisan.....	I-5
1.8 Kesimpulan.....	I-7
<b>BAB II KAJIAN LITERATUR .....</b>	<b>II-1</b>

2.1	Pendahuluan .....	I-1
2.2	Landasan Teori .....	II-1
2.2.1	API (Application Programming Interface) .....	II-1
2.2.2	Golang .....	II-2
2.2.3	Echo <i>Framework</i> .....	II-3
2.2.4	Agile.....	II-3
2.2.5	GoLand.....	II-4
2.2.6	Postman .....	II-4
2.2.7	Version Control.....	II-5
2.2.8	Draw.io.....	II-5
2.2.9	Tuleap.....	II-6
2.2.10	Google SpreadSheets .....	II-6
2.2.11	Zoom Meeting.....	II-7
2.2.12	JSON .....	II-7
2.2.13	Apache JMeter .....	II-8
2.3	Penelitian Terkait .....	II-8
2.4	Kesimpulan.....	II-9
<b>BAB III METODOLOGI PENELITIAN .....</b>		<b>III-1</b>
3.1	Pendahuluan .....	III-1
3.2	Unit Penelitian .....	III-1
3.3	Pengumpulan Data .....	III-1
3.3.1	Jenis Data .....	III-1
3.3.2	Sumber Data.....	III-1
3.3.3	Metode Pengumpulan Data .....	III-2

34 Tahapan Penelitian .....	I-2
3.4.1 Kerangka Kerja .....	III-2
3.4.2 Menentukan Ruang Lingkup dan Unit Penelitian.....	III-3
3.4.3 Menemukan Dasar Teori yang Berkaitan dengan Penelitian.....	III-3
3.4.4 Menetapkan Kriteria Pengujian .....	III-4
3.4.5 Menentukan Alat yang Digunakan dalam Penelitian.....	III-4
3.4.6 Menetapkan Format Data Pengujian.....	III-5
3.4.7 Pengujian Penelitian.....	III-6
3.4.7.1 Proses Pengujian Peforma.....	III-6
3.4.7.2 Melakukan Analisis Hasil Pengujian dan Membuat .....	III-7
3.4.8 Kebutuhan Fungsional dan Non Fungsional.....	III-7
3.5 Tahapan Pengujian .....	III-8
3.5.1 Menentukan Apa yang Akan Diukur Melalui Pengujian.....	III-8
3.5.2 Menentukan Rute yang Akan Diuji .....	III-8
3.5.3 Menentukan Bagaimana Pengujian akan Dilaksanakan .....	III-8
3.6 Metode Pengembangan Perangkat Lunak.....	III-9
3.6.1 Fase Analisis .....	III-9
3.6.2 Fase Pengembangan Perangkat Lunak.....	III-10
3.6.3 Fase Pengujian Perangkat Lunak .....	III-12
3.6.4 Fase Peluncuran Perangkat Lunak .....	III-13
3.7 Manajemen Proyek Penelitian.....	III-14
3.8 Kesimpulan.....	III-13
<b>BAB IV PENGEMBANGAN PERANGKAT LUNAK .....</b>	<b>IV-1</b>
4.1 Pendahuluan .....	IV-1

4.2 Agile .....	V-1
4.2.1 Fase Analisis .....	IV-1
4.2.1.1 <i>Use Case Diagram</i> .....	IV-1
4.2.1.2 Skenario <i>Use Case</i> .....	IV-10
4.2.1.3 <i>Sequence Diagram</i> .....	IV-27
4.2.1.4 <i>Class Diagram</i> .....	IV-43
4.2.1.5 <i>Entity Relationship Diagram</i> .....	IV-58
4.2.2 Fase Pengembangan Perangkat Lunak.....	IV-59
4.2.2.1 Implementasi Paket.....	IV-59
4.2.3 Fase Pengujian Perangkat Lunak .....	IV-60
4.2.3.1 Rencana Pengujian.....	IV-60
4.2.3.2 Pengujian Kasus.....	IV-63
4.2.4 Fase Peluncuran Perangkat Lunak .....	IV-67
4.3 Kesimpulan.....	IV-68
<b>BAB V HASIL DAN ANALISIS PENELITIAN.....</b>	<b>V-1</b>
5.1 Pendahuluan .....	V-1
5.2 Hasil Pengujian.....	V-1
5.2.1 Performa API .....	V-1
5.2.1.1 Performa Rute API Metode POST.....	V-2
5.2.1.2 Performa Rute API Metode GET.....	V-6
5.3 Analisis Hasil Uji .....	V-11
5.4 Kesimpulan.....	V-11
<b>BAB VI KESIMPULAN DAN SARAN.....</b>	<b>VI-1</b>
6.1 Kesimpulan.....	VI-1

6.2 Saran.....I-1

**DAFTAR PUSTAKA..... xvii**

## DAFTAR TABEL

	Halaman
Tabel III-1: Rancangan Hasil Pengujian Performa <i>Framework</i> .....	I-5
Tabel III-2: Rancangan Hasil Pengujian Performa Native API.....	III-6
Tabel IV-1: Definisi Aktor Pada Use-Case .....	IV-4
Tabel IV-2: Definisi Dari <i>Use-Case</i> Pada Aktor Unit Kerja Lain.....	IV-6
Tabel IV-3: Definisi Dari <i>Use-Case</i> Oleh Aktor Manager.....	IV-7
Tabel IV-4: Definisi Dari <i>Use-Case</i> Oleh Aktor TU RMP .....	IV-8
Tabel IV-5: Definisi Dari <i>Use-Case</i> Oleh Aktor TU RBP .....	IV-9
Tabel IV-6: Definisi Dari <i>Use-Case</i> Oleh Aktor Superintendent RMP .	IV-10
Tabel IV-7: Aktor Unit Kerja Lain <i>Use Case</i> Skenario Login.....	IV-10
Tabel IV-8: Aktor Unit Kerja Lain <i>Use Case</i> Membuat POK .....	IV-11
Tabel IV-9: Aktor Unit Kerja Lain <i>Use Case</i> Melihat Seluruh POK.....	IV-11
Tabel IV-10: Aktor Unit Kerja Lain <i>Use Case</i> Melihat Surat Balasan ..	IV-12
Tabel IV-11: Aktor Unit Kerja Lain <i>Use Case</i> Melihat POK Selesai....	IV-12
Tabel IV-12: Aktor Unit Kerja Lain <i>Use Case</i> Memberikan Feedback.	IV-13
Tabel IV-13: Aktor Unit Kerja Lain <i>Use Case</i> Melihat POK Dalam.....	IV-14
Tabel IV-14: Aktor Unit Kerja Lain <i>Use Case</i> Melihat Surat Balasan ..	IV-14
Tabel IV-15: Aktor Manager RBP <i>Use Case</i> Login.....	IV-15
Tabel IV-16: Aktor Manager RBP <i>Use Case</i> Melihat Seluruh POK .....	IV-15
Tabel IV-17: Aktor Manager RBP <i>Use Case</i> Melihat Surat Masuk .....	IV-16
Tabel IV-18: Aktor Manager RBP <i>Use Case</i> Melihat Surat Keluar .....	IV-16
Tabel IV-19: Aktor Manager RBP <i>Use Case</i> Menolak POK .....	IV-17
Tabel IV-20: Aktor Manager RBP <i>Use Case</i> Disposisi Surat Masuk....	IV-17
Tabel IV-21: Aktor Manager RBP <i>Use Case</i> Konfirmasi Surat Keluar.	IV-18
Tabel IV-22: Aktor TU RMP <i>Use Case</i> Login.....	IV-19
Tabel IV-23: Aktor TU RMP <i>Use Case</i> Melihat Seluruh POK .....	IV-19
Tabel IV-24: Aktor TU RMP <i>Use Case</i> Melihat Surat Masuk POK .....	IV-20
Tabel IV-25: Aktor TU RMP <i>Use Case</i> Mengisi Nomor POK.....	IV-20
Tabel IV-26: Aktor TU RMP <i>Use Case</i> Melihat Surat Keluar POK .....	IV-21
Tabel IV-27: Aktor TU RMP <i>Use Case</i> Mengisi Nomor POK Keluar..	IV-21
Tabel IV-28: Aktor TU RBP <i>Use Case</i> Login.....	IV-22

Tabel IV-29: Aktor TU RBP <i>Use Case</i> Melihat Seluruh POK .....	V-22
Tabel IV-30: Aktor TU RBP <i>Use Case</i> Membuat Surat Masuk POK ...	IV-23
Tabel IV-31: Aktor TU RBP <i>Use Case</i> Melihat Surat POK Balasan.....	IV-24
Tabel IV-32: Aktor TU RBP <i>Use Case</i> Membuat Surat POK Balasan..	IV-24
Tabel IV-33: Aktor Superintendent RMP <i>Use Case</i> Login.....	IV-25
Tabel IV-34: Aktor Superintendent RMP <i>Use Case</i> Melihat Seluruh ...	IV-25
Tabel IV-35: Aktor Superintendent RMP <i>Use Case</i> Melihat Surat .....	IV-26
Tabel IV-36: Aktor Superintendent RMP <i>Use Case</i> Disposisi Surat .....	IV-26
Tabel IV-37: Daftar Implementasi Paket.....	IV-59
Tabel IV-38: Rencana Pengujian.....	IV-60
Tabel IV-39: Pengujian Rute API.....	IV-61
Tabel V-1: Hasil Uji Rute API Golang Dengan Menggunakan Echo .....	V-2
Tabel V-2: Hasil Uji Rute API Golang Dengan Menggunakan Echo .....	V-3
Tabel V-3: Perbandingan Hasil Uji Rute API Echo Framework Golang..	V-4
Tabel V-4: Hasil Uji Rute API Golang Dengan Menggunakan Native ....	V-6
Tabel V-5: Hasil Uji Rute API Golang Dengan Menggunakan Native.....	V-8
Tabel V-6: Perbandingan Hasil Uji Rute API Echo Framework Golang..	V-9



## DAFTAR GAMBAR

	Halaman
Gambar II-1: Tahapan Agile.....	I-4
Gambar III-1: Kerangka Kerja.....	III-3
Gambar III-2: Fase Analisis sistem E-POK .....	III-9
Gambar III-3: Fase Pengembangan sistem E-POK .....	III-10
Gambar III-4: Fase Pengujian API sistem E-POK.....	III-12
Gambar III-5: Fase Peluncuran sistem E-POK.....	III-13
Gambar III-6: WBS sistem E-POK .....	III-14
Gambar III-7: GANTT CHART sistem E-POK bagian 1 .....	III-15
Gambar III-8: GANTT CHART sistem E-POK bagian 2.....	III-16
Gambar IV-1: Penggunaan Diagram Kasus Oleh Unit Kerja Lain .....	IV-2
Gambar IV-2: Penggunaan Diagram Kasus Oleh TU RBP Aktor .....	IV-2
Gambar IV-3: Penggunaan Diagram Kasus Oleh Manager RBP .....	IV-3
Gambar IV-4: Penggunaan Diagram Kasus Oleh TU RMP Aktor.....	IV-3
Gambar IV-5: Penggunaan Diagram Kasus Oleh Superintendent .....	IV-4
Gambar IV-6: Aktor Unit Kerja Lain <i>Sequence Diagram Use Case</i> .....	IV-28
Gambar IV-7: Aktor Unit Kerja Lain <i>Sequence Diagram Use Case</i> .....	IV-28
Gambar IV-8: Aktor Unit Kerja Lain <i>Sequence Diagram Use Case</i> .....	IV-29
Gambar IV-9: Aktor Unit Kerja Lain <i>Sequence Diagram Use Case</i> .....	IV-29
Gambar IV-10: Aktor Unit Kerja Lain <i>Sequence Diagram Use Case</i> ...	IV-30
Gambar IV-11: Aktor Unit Kerja Lain <i>Sequence Diagram Use Case</i> ...	IV-30
Gambar IV-12: Aktor Unit Kerja Lain <i>Sequence Diagram Use Case</i> ...	IV-31
Gambar IV-13: Aktor Unit Kerja Lain <i>Sequence Diagram Use Case</i> ...	IV-31
Gambar IV-14: Aktor TU RBP <i>Sequence Diagram Use Case</i> Login ....	IV-32
Gambar IV-15: Aktor TU RBP <i>Sequence Diagram Use Case</i> Melihat .	IV-32
Gambar IV-16: Aktor TU RBP <i>Sequence Diagram Use Case</i> Entry.....	IV-33
Gambar IV-17: Aktor TU RBP <i>Sequence Diagram Use Case</i> Melihat .	IV-33
Gambar IV-18: Aktor TU RBP <i>Sequence Diagram Use Case</i> .....	IV-34
Gambar IV-19: Aktor Manager RBP <i>Sequence Diagram Use Case</i> .....	IV-34
Gambar IV-20: Aktor Manager RBP <i>Sequence Diagram Use Case</i> .....	IV-35

Gambar IV-21: Aktor Manager RBP <i>Sequence Diagram Use Case</i>	V-35
Gambar IV-22: Aktor Manager RBP <i>Sequence Diagram Use Case</i> .....	IV-36
Gambar IV-23: Aktor Manager RBP <i>Sequence Diagram Use Case</i> .....	IV-36
Gambar IV-24: Aktor Manager RBP <i>Sequence Diagram Use Case</i> .....	IV-37
Gambar IV-25: Aktor Manager RBP <i>Sequence Diagram Use Case</i> .....	IV-37
Gambar IV-26: Aktor TU RMP <i>Sequence Diagram Use Case Login</i> ...	IV-38
Gambar IV-27: Aktor TU RMP <i>Sequence Diagram Use Case Melihat</i>	IV-38
Gambar IV-28: Aktor TU RMP <i>Sequence Diagram Use Case Melihat</i>	IV-39
Gambar IV-29: Aktor TU RMP <i>Sequence Diagram Use Case</i> .....	IV-39
Gambar IV-30: Aktor TU RMP <i>Sequence Diagram Use Case Melihat</i>	IV-40
Gambar IV-31: Aktor TU RMP <i>Sequence Diagram Use Case</i> .....	IV-40
Gambar IV-32: Aktor Superintendent RMP <i>Sequence Diagram</i> .....	IV-41
Gambar IV-33: Aktor Superintendent RMP <i>Sequence Diagram</i> .....	IV-41
Gambar IV-34: Aktor Superintendent RMP <i>Sequence Diagram</i> .....	IV-42
Gambar IV-35: Aktor Superintendent RMP <i>Sequence Diagram</i> .....	IV-42
Gambar IV-36: Aktor Unit Kerja Lain <i>Class Diagram Use Case</i> .....	IV-43
Gambar IV-37: Aktor Unit Kerja Lain <i>Class Diagram Use Case</i> .....	IV-43
Gambar IV-38: Aktor Unit Kerja Lain <i>Class Diagram Use Case</i> .....	IV-44
Gambar IV-39: Aktor Unit Kerja Lain <i>Class Diagram Use Case</i> .....	IV-44
Gambar IV-40: Aktor Unit Kerja Lain <i>Class Diagram Use Case</i> .....	IV-45
Gambar IV-41: Aktor Unit Kerja Lain <i>Class Diagram Use Case</i> .....	IV-45
Gambar IV-42: Aktor Unit Kerja Lain <i>Class Diagram Use Case</i> .....	IV-46
Gambar IV-43: Aktor Unit Kerja Lain <i>Class Diagram Use Case</i> .....	IV-46
Gambar IV-44: Aktor TU RBP <i>Class Diagram Use Case Login</i> .....	IV-47
Gambar IV-45: Aktor TU RBP <i>Class Diagram Use Case Melihat</i> .....	IV-47
Gambar IV-46: Aktor TU RBP <i>Class Diagram Use Case Membuat</i> ....	IV-48
Gambar IV-47: Aktor TU RBP <i>Class Diagram Use Case Melihat</i> .....	IV-48
Gambar IV-48: Aktor TU RBP <i>Class Diagram Use Case Balasan</i> .....	IV-49
Gambar IV-49: Aktor Manager RBP <i>Class Diagram Use Case Login</i> .	IV-49
Gambar IV-50: Aktor Manager RBP <i>Class Diagram Use Case</i> .....	IV-50
Gambar IV-51: Aktor Manager RBP <i>Class Diagram Use Case</i> .....	IV-50
Gambar IV-52: Aktor Manager RBP <i>Class Diagram Use Case</i> .....	IV-51
Gambar IV-53: Aktor Manager RBP <i>Class Diagram Use Case</i> .....	IV-51
Gambar IV-54: Aktor Manager RBP <i>Class Diagram Use Case</i> .....	IV-52

Gambar IV-55: Aktor Manager RBP <i>Class Diagram Use Case</i> .....	V-52
Gambar IV-56: Aktor TU RMP <i>Class Diagram Use Case Login</i> .....	IV-53
Gambar IV-57: Aktor TU RMP <i>Class Diagram Use Case Melihat</i> .....	IV-53
Gambar IV-58: Aktor TU RMP <i>Class Diagram Use Case Melihat</i> .....	IV-54
Gambar IV-59: Aktor TU RMP <i>Class Diagram Use Case</i> .....	IV-54
Gambar IV-60: Aktor TU RMP <i>Class Diagram Use Case Melihat</i> .....	IV-55
Gambar IV-61: Aktor TU RMP <i>Class Diagram Use Case</i> .....	IV-55
Gambar IV-62: Aktor Superintendent RMP <i>Sequence Diagram Use</i> ....	IV-56
Gambar IV-63: Aktor Superintendent RMP <i>Sequence Diagram Use</i> ....	IV-56
Gambar IV-64: Aktor Superintendent RMP <i>Sequence Diagram Use</i> ....	IV-57
Gambar IV-65: Aktor Superintendent RMP <i>Sequence Diagram Use</i> ....	IV-57
Gambar IV-66: <i>Entity Relationship Diagram</i> Sistem E-POK.....	IV-58
Gambar V-1: Grafik Hasil Uji Echo <i>Framework</i> Golang Metode .....	V-3
Gambar V-2: Grafik Hasil Uji <i>Native</i> Golang Metode POST.....	V-4
Gambar V-3: Grafik Perbandingan Hasil Uji Echo <i>Framework</i> Dengan..	V-5
Gambar V-4: Grafik Hasil Uji Echo <i>Framework</i> Golang Metode GET ...	V-7
Gambar V-5: Grafik Hasil Uji <i>Native</i> Golang Metode GET.....	V-9
Gambar V-6: Grafik Perbandingan Hasil Uji Echo <i>Framework</i> Dengan	V-10

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Pendahuluan**

Pada bab I pendahuluan akan berisi tentang latar belakang masalah, rumusan masalah, tujuan dan manfaat dari penelitian ini, hingga batasan dari permasalahan. Semua bagian dari bab ini menjelaskan keseluruhan bagaimana gambaran penelitian ini secara umum.

Dimulai dengan adanya *Echo Framework* pada bahasa pemrograman GO yang digunakan dalam membangun API untuk mendapatkan kenyamanan dan kemudahan dalam membangun sebuah arsitektur API dan juga untuk mendapatkan performa yang diharapkan lebih baik daripada membangun sebuah API dengan tanpa menggunakan *framework* atau dengan *native* dari pemrograman GO.

#### **1.2 Latar Belakang**

Pada awal pengembangan web *development* semua aplikasi ataupun situs web dibangun dan dibuat *code*-nya secara manual. Pembuatan *code* yang manual tersebut menimbulkan banyak kesalahan dan membuat pengembang mengerjakan lebih banyak pekerjaan yang harus dilakukan. Untuk mengatasi masalah ini *web framework* diperkenalkan di sekitar awal tahun 2000 (Curie *et al.*, 2019). *Web framework* merupakan sebuah alat yang membantu untuk membangun sebuah situs web yang dapat menghindari *bugs* atau kesalahan dan juga mempersingkat waktu dalam pengerjaannya. Untuk situs web statis maupun dinamis dapat menggunakan

*frameworks*. Terdapat banyak macam *framework* yang tersedia untuk berbagai macam kebutuhan. *Framework* dapat diklasifikasikan menjadi dua kategori yaitu *client side* dan *server side*. *Client side* yang berarti terdapat pada sisi *client* atau pengguna terdapat beberapa contoh seperti Angular, Vue dan React. Lalu pada sisi server atau *server side* terdapat salah satu contoh *Framework* yaitu Echo pada bahasa pemrograman golang yang akan digunakan pada penelitian ini.

Curite *et al.*, (2019) mengusulkan untuk mempertimbangkan penggunaan *web framework* yang akan bertindak sebagai kerangka kerja. *Framework* yang diusulkan dapat digunakan untuk membangun atau merancang situs web untuk berbagai Lembaga pemerintah, *non*-pemerintah atau swasta. Kerangka kerja yang digunakan untuk merancang situs web dengan tingkat kegunaan yang tinggi.

API atau *Application Programming Interface* adalah layanan yang digunakan beragam institusi atau perusahaan di seluruh dunia oleh agen prakiraan cuaca, pemerintah, browser, dan lainnya untuk melayani serangkaian prosedur untuk mengakses fitur atau data dari sistem atau aplikasi tertentu (Domingues, 2017). API bertugas sebagai *server side* program yang bertransaksi dengan database secara langsung. API mengembalikan response berbentuk JSON atau Javascript Object Notation merupakan format data yang dapat diproses oleh banyak jenis bahasa pemrograman dan juga mudah dipahami oleh manusia.

Tanpa kerangka kerja atau ketentuan suatu kode dari API mengakibatkan penulisan kode dari API tersebut menjadi tidak teratur karena tidak memiliki kerangka atau landasan untuk menjadi pondasi dari API. Dapat berakibat kepada tidak efisiennya suatu kode yang menjadikan waktu eksekusi dan performa dari API

menjadi tidak efisien. Jika kode tidak teratur maka berdampak juga pada pengembangan dan pemeliharaan API jika suatu saat ada penambahan fitur atau pembenaran *bug*.

Sistem Elektronik Pengajuan Order Kerja atau E-POK merupakan sistem yang akan dibangun pada PT. Pupuk Sriwidjaja yang ditujukan untuk departemen Rancang Bangun Perekayasaan Atau RBP. Sistem E-POK dibangun dengan dua macam *platform* yaitu android dan website, maka dari itu dibutuhkan sebuah API atau *Application Programming Interface* yang bertugas sebagai penghubung antara sisi *client* yaitu android dan website dengan sisi *server* yaitu database.

Golang merupakan bahasa pemrograman yang cepat dan terkompilasi, yang telah banyak digunakan dalam dunia industri, dikarenakan kinerjanya yang sangat baik pada *concurrent programming* atau komputasi konkuren yang dieksekusi secara bersamaan selama periode waktu yang tumpang tindih (Wang *et al.*, 2019). *Framework Echo* diklaim menjadi salah satu dari 3 teratas *open-sorce Framework* pada bahasa pemrograman golang dengan hasil kinerja yang baik (Domingues, 2017). Performa waktu eksekusi sebuah API dapat berpengaruh besar untuk user yang menggunakan sebuah sistem yang terdapat API didalamnya. Diputuskan pada sisi API sistem E-POK PT. Pupuk Sriwidjaja menggunakan *Framework Echo* pada bahasa pemrograman golang dan menghitung perbedaan waktu eksekusi dengan *native* golang.

Berdasarkan pertimbangan di atas, maka penulis akan membuat penelitian yang berjudul Perbandingan Performa pada Pembangunan API sistem E-POK PT Pupuk Sriwidjaja menggunakan *Echo Framework* Golang dan *Native* Golang.

### 1.3 Rumusan Masalah

Rumusan masalah pada penelitian ini disebabkan karena belum adanya API (*Application Program Interface*) sistem E-POK PT Pupuk Sriwidjaja dan pengukuran perbandingan antara *native* Golang dan *Echo Framework* Golang. Untuk menjawab rumusan masalah utama tersebut maka menimbulkan beberapa aspek pertanyaan atau yang biasa disebut dengan *Research Question* di bawah ini:

1. Bagaimana cara mengembangkan API (*Application Programming Interface*) pada sistem E-POK PT Pupuk Sriwidjaja dengan menggunakan *Echo Framework* golang?
2. Bagaimana mengukur performa *execution time* pada API (*Application Programming Interface*) yang dibangun dengan *Echo Framework* dan *native* golang pada sistem E-POK PT Pupuk Sriwidjaja?

### 1.4 Tujuan Penelitian

Tujuan penelitian ini adalah :

1. Untuk mengetahui cara mengembangkan API (*Application Programming Interface*) pada sistem E-POK PT Pupuk Sriwidjaja dengan menggunakan *Echo Framework* golang.
2. Untuk mengetahui cara mengukur dan menentukan pengukuran performa atau *execution time* API (*Application Programming Interface*) yang lebih efisien antara *Echo framework* atau *native Golang*, dari segi performa *execution time*.

## 1.5 Manfaat Penelitian

Manfaat penelitian ini adalah :

1. API dimanfaatkan untuk sistem E-POK PT Pupuk Sriwidjaja.
2. Dapat mengetahui efisiensi antara API (*Application Programming Interface*) dengan menggunakan *native Golang* dan API (*Application Programming Interface*) dengan menggunakan *Echo Framework*.
3. Dapat menjadi referensi bagi peneliti dan pembaca, mengenai API (*Application Programming Interface*) dan *Framework* pada bahasa pemrograman Golang.

## 1.6 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut :

1. API dibangun dengan dataflow yang telah dianalisa.
2. Menggunakan *Echo Framework* dengan bahasa pemrograman golang.
3. Performa yang dihitung yaitu execution time dan hanya pada salah satu rute dari dua metode API yang ada pada sistem dengan jumlah data dan pemrosesan data yang dikirim dan diterima paling banyak.

## 1.7 Sistematika Penulisan

Sistematika penulisan Tugas Akhir ini mengikuti standar penulisan Tugas Akhir Fakultas Ilmu Komputer Universitas Sriwijaya yaitu sebagai berikut.



## **BAB I PENDAHULUAN**

Bab ini akan membahas tentang latar belakang, rumusan masalah, tujuan dan manfaat penelitian, batasan atau ruang lingkup masalah, metodologi penelitian, dan sistematika penulisan.

## **BAB II KAJIAN LITERATUR**

Bab ini membahas semua dasar-dasar teori yang digunakan mulai dari definisi sistem, informasi tentang domain, dan segala sesuatu yang digunakan dalam tahap analisis, perancangan, dan implementasi.

## **BAB III METODOLOGI PENELITIAN**

Bab ini membahas tahapan-tahapan yang akan diterapkan pada penelitian. Setiap rencana tahap penelitian dijelaskan secara rinci berdasarkan kerangka kerjanya. Diikuti dengan perancangan manajemen proyek dalam pelaksanaan penelitian.

## **BAB IV PENGEMBANGAN PERANGKAT LUNAK**

Bab ini membahas perancangan dan lingkungan implementasi, berupa analisis dari sistem E-POK PT Pupuk Sriwidjaja, serta perancangan perangkat lunak yang akan digunakan agar sistem E-POK PT Pupuk Sriwidjaja berjalan dengan baik, yang akan digunakan sebagai alat penelitian.

## **BAB V HASIL DAN ANALISIS PENELITIAN**

Bab ini membahas implementasi hasil analisis dan perancangan sebelumnya. Hasil analisis berupa kesimpulan yang dapat diambil dari hasil penelitian. Lakukan pengujian perangkat lunak dan pengujian data penelitian.

## **BAB VI KESIMPULAN DAN SARAN**

Bab ini memuat semua kesimpulan dari uraian yang telah dibahas sebelumnya, dan saran-saran yang diharapkan dapat bermanfaat untuk pengembangan penelitian selanjutnya.

### **1.8 Kesimpulan**

Terdapat enam bab yang akan dibahas dalam penelitian ini. Bab 1 membahas tentang gagasan dasar yang diajukan mengenai pengenalan *Echo Framework* untuk membangun API dengan menggunakan bahasa pemrograman golang. Hal ini penting untuk memahami konsep dasar tentang penelitian yang akan dikerjakan. Latar belakang masalah dibahas pada bagian 1.2. Pernyataan masalah telah dijelaskan sehingga solusi untuk memecahkan masalah dapat diidentifikasi. Selain itu, terdapat dua tujuan yang ingin dicapai dalam mencapai tujuan penelitian. Ruang lingkup penelitian ini juga diberikan dengan jelas, Pada bagian akhir, berisi alasan mengapa penting untuk melakukan penelitian ini.

Bab 2 akan membahas kajian literatur yang berkaitan dengan penelitian. Bab 3 berisi tentang metodologi penelitian keseluruhan untuk merinci kerangka penelitian dan untuk mengembangkan sistem. Bab 4 melanjutkan tahapan

penelitian ini yaitu memberikan gambaran tentang bagaimana perangkat lunak dikembangkan sehingga dapat memberikan hasil dan analisis yang akurat seperti yang akan diuji pada Bab 5 di bagian akhir yaitu Bab 6 merupakan kesimpulan dari penelitian yang diusulkan, juga menerima saran dari semua pihak terkait dengan penelitian ini sehingga dapat ditingkatkan dan dikembangkan lebih lanjut.

## DAFTAR PUSTAKA

- Alder, G. (2020, December 6). *drawio*. Retrieved from drawio: [app.diagrams.net](http://app.diagrams.net)
- Curie, D. H., Jaison, J., Yadav, J., & Fiona, J. R. (2019). Analysis on Web Frameworks. *Journal of Physics: Conference Series*, 1.
- Domingues, M. A. (2017). Performance testing of open-source HTTP web. *Proceedings of the 12th Doctoral Symposium in Informatics Engineering - DSIE'17* (p. 1). Porto: Faculty of Engineering, University of Porto.
- Go. (2020, December 6). *Go. Documentation*. Retrieved from Go: <https://golang.org/doc/>
- Google. (2020, December 6). *Sheets*. Retrieved from Google. Sheets: <https://docs.google.com/spreadsheets/>
- Hoque, M. M. (2017). Scalable Micro-Service based Approach to FHIR Server with Golang and No-SQL. *International Conference of Computer and Information Technology (ICCIT)*, 2.
- JETBRAINS. (2020, December 6). *GoLand*. Retrieved from Jet Brains: <https://www.jetbrains.com/go/>
- JSON. (2020, December 6). *Introducing JSON*. Retrieved from JSON: <https://www.json.org/json-en.html>
- O'Sullivan, B. (2009). *Mercurial: the Definitive Guide*. Sebastopol: O'Reilly Media, Inc. ISBN 9780596555474.
- Permatasari, D. I., Ardani, M., Ma'ulfa, A. Y., Ilhami, N., Pratama, S. G., Astuti, S. R., & Naufalita, N. W. (2020). Pengujian Aplikasi Menggunakan Metode Load Testing dengan Apache Jmeter pada Sistem Informasi Pertanian. *Jurnal sistem dan teknologi informasi*, 2.
- Postman. (2020, December 6). *Postman*. Retrieved from Postman: <https://www.postman.com/>

- Prayogo, N. A. (2020, December 06). *Dasar Pemrograman Golang*. Retrieved from Dasar Pemrograman Golang: <https://dasarpemrogramangolang.novalagung.com/>
- Rompis, A. C., & Aji, R. F. (2018). Perbandingan Performa Kinerja Node.js, PHP, dan Python . *Cogito Smart Journal/VOL. 4/NO.1/JUNE 2018*, 16.
- Tuleap. (2020, December 6). *Tuleap*. Retrieved from Tuleap: <https://www.tuleap.org/>
- Wahlström, S. (2019). Comparing Scaling Benefits of Monolithic and Microservice Architectures Implemented in Java and Go. *Faculty of Computing Blekinge Institute of Technology*, 9.
- Wang, C., Gao, J., Jiang, Y., Xing, Z., Zhang, H., Yin, W., . . . Sun, J. (2019). Go-clone: graph-embedding based clone detector for Golang. *ISSTA 2019: Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 374.
- Wickramage, N., & Weerawarana, S. (2005). A Benchmark for Web Service Frameworks. *Proceedings of the 2005 IEEE International Conference on Services Computing*, 8.
- Zoom. (2020, December 6). *Zoom*. Retrieved from Zoom: <https://zoom.us/>