

**ANALISIS PERFORMA *WEB SERVER* RUST DAN GO PADA
PROTOKOL *HYPERTEXT TRANSFER PROTOCOL* (HTTP)**

SKRIPSI

Program Studi Sarjana Informatika
Jurusan Informatika

Oleh:
JANDER SUPARDI
NIM D1042181012



FAKULTAS TEKNIK
UNIVERSITAS TANJUNGPURA
PONTIANAK
2025

HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Jander Supardi

NIM : D1042181012

Menyatakan bahwa dalam skripsi yang berjudul “Analisis Performa *Web server* Rust Dan Go Pada Protokol *Hypertext Transfer Protocol* (HTTP)” tidak terdapat karya yang pernah diajukan untuk memperoleh gelar sarjana di suatu perguruan tinggi manapun. Sepanjang pengetahuan Saya, tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam Daftar Pustaka.

Demikian pernyataan ini dibuat dengan sebenar-benarnya. Saya sanggup menerima konsekuensi akademis dan hukum di kemudian hari apabila pernyataan yang dibuat ini tidak benar.

Pontianak, 31 Januari 2025

Jander Supardi
NIM D1042181012



KEMENTERIAN PENDIDIKAN TINGGI, SAINS,
DAN TEKNOLOGI
UNIVERSITAS TANJUNGPURA
FAKULTAS TEKNIK

Jl. Prof. Dr. H. Hadari Nawawi Pontianak 78124
Telepon (0561) 740186, WA: +6282152280907

Email : ft@untan.ac.id Website : <http://teknik.untan.ac.id>

HALAMAN PENGESAHAN

ANALISIS PERFORMA *WEB SERVER* RUST DAN GO PADA PROTOKOL
HYPertext TRANSFER PROTOCOL (HTTP)

Program Studi Sarjana Informatika
Jurusan Informatika

Oleh:

Jander Supardi
NIM D1042181012

Telah dipertahankan di depan Penguji Skripsi pada tanggal 31 Januari 2025
dan diterima sebagai salah satu persyaratan untuk memperoleh gelar sarjana.

Susunan Penguji Skripsi:

Ketua,

Heri Priyanto, S.T., M.T.
NIP 197504122003121001

Sekretaris,

Alfian Abdul Jalid, S.Kom., M.Cs
NIP 199106292022031003

Penguji Utama,

Haried Novriando, S.Kom., M.Eng.
NIP 198611132020121005

Penguji Pendamping,

Helfi Nasution, S.Kom, M.Cs
NIP 197104291998021002

Pontianak, 31 Januari 2025
Dekan,

Dr.-Ing. Ir. Slamet Widodo, M.T., IPM
NIP 196712231992031002

HALAMAN PERSEMBAHAN

Skripsi ini didedikasikan kepada kedua orang tuaku dan saudara-saudaraku. Apa yang mereka telah berikan melebihi daripada apa yang pernah aku inginkan.

KATA PENGANTAR

Puji syukur kepada Yesus Kristus karena kuasa-Nya, penulis dapat menyelesaikan skripsi yang berjudul "Analisis Performa *Web Server* Rust dan Go Pada Protokol *Hypertext Transfer Protocol* (HTTP)" dengan baik. Skripsi ini disusun untuk memenuhi salah satu syarat dalam memperoleh gelar Sarjana Informatika di Fakultas Teknik, Universitas Tanjungpura.

Penulisan skripsi ini tidak akan terwujud tanpa dukungan, bimbingan, dan bantuan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Heri Priyanto, S.T., M.T., selaku dosen pembimbing akademik sekaligus dosen pembimbing utama dan bapak Alfian Abdul Jalid, S.Kom., M.Cs selaku dosen pembimbing ke dua yang telah memberikan bimbingan, arahan, serta motivasi selama proses penyusunan skripsi ini.
2. Bapak Haried Novriando, S.Kom., M.Eng. dan Helfi Nasution, S.Kom, M.Cs, selaku dosen penguji pertama dan kedua yang telah memberikan masukan dan kritik dalam penyempurnaan skripsi ini.
3. Orang tua dan keluarga, yang senantiasa memberikan doa, dukungan moral, dan materiil selama masa studi dan penyusunan skripsi ini.
4. Rekan-rekan mahasiswa di Program Studi Informatika, yang telah berbagi ilmu dan pengalaman selama masa perkuliahan dan penelitian.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat kekurangan, baik dari segi isi maupun tata bahasa. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan demi penyempurnaan karya ini. Semoga skripsi ini dapat memberikan manfaat bagi perkembangan ilmu pengetahuan dan teknologi, khususnya di bidang pengembangan *web server*.

Pontianak, 31 Januari 2025
Penulis,

Jander Supardi

ABSTRAK

Web server memegang peran penting dalam melayani permintaan dari *browser client* dengan mengirimkan konten web kepada pengguna. Penelitian ini menganalisis performa dua *web server*, yaitu Rust dan Go, dalam menangani permintaan HTTP. Tujuan penelitian adalah membandingkan performa kedua *web server* dalam hal penggunaan sumber daya (CPU dan memori), kecepatan, stabilitas dan skalabilitas. Pengujian dilakukan menggunakan metode *load testing* dengan Apache JMeter, dengan simulasi beban pengguna mulai dari 200 hingga 1000 pengguna. Hasil penelitian menunjukkan bahwa Rust lebih rendah dalam pemanfaatan sumber daya dibanding Go. Pada beban rendah (200–400 pengguna) Go lebih rendah dalam penggunaan CPU, namun pada beban tinggi (600–1000 pengguna) Rust unggul dengan penggunaan CPU yang lebih rendah. Sebagai contoh, pada beban 600 pengguna Rust hanya menggunakan 0,47% CPU sementara Go mencapai 5,21%. Dalam semua skema pengujian Rust juga lebih hemat memori, dengan penggunaan 52,28 MB pada beban 800 pengguna, dibandingkan Go yang mengonsumsi 267,01 MB. Dari segi kecepatan Go memiliki performa lebih optimal dibanding Rust berdasarkan nilai *throughput* dan *processing time*. Perbedaan mencolok pada beban 600 pengguna Go memiliki rata-rata *throughput* 60,043 transaksi per detik (tps) sedangkan Rust hanya mencapai 8,047 tps. Go juga memiliki waktu pemrosesan yang lebih rendah dibanding Rust perbedaan menonjol pada beban 800 pengguna dimana Go memiliki nilai 173,47 ms sedangkan Rust 7.408,53 ms. Dari segi stabilitas, Go juga lebih unggul dari Rust ditunjukkan rata-rata nilai koefisien variasi (CV) yang lebih rendah. Sebagai contoh pada 200 pengguna nilai koefisien variasi Go sebesar 29,11% lebih rendah dibanding Rust yang mencapai 277,63%. Rust lebih unggul dalam skalabilitas, dengan peningkatan *throughput* sebesar 46,057% pada 1000 pengguna, dibandingkan Go yang hanya 2,074%. Kesimpulannya, Go lebih optimal untuk beban rendah dan aplikasi yang memerlukan kecepatan, sementara Rust lebih cocok untuk beban tinggi dan aplikasi yang memerlukan efisiensi sumber daya.

Kata kunci: *web server*, Rust, Go, HTTP, Performa, *load testing*

ABSTRACT

Web servers play a crucial role in serving requests from client browsers by delivering web content to users. This study analyzes the performance of two web servers, Rust and Go, in handling HTTP requests. The aim of the research is to compare the performance of the two web servers in terms of resource usage (CPU and memory), speed, stability, and scalability. Testing was conducted using the load testing method with Apache JMeter, simulating user loads ranging from 200 to 1000 users. The results show that Rust is more efficient in resource utilization compared to Go. At low loads (200–400 users), Go has lower CPU usage, but at high loads (600–1000 users), Rust outperforms with significantly lower CPU usage. For example, at a load of 600 users, Rust only uses 0.47% CPU, while Go reaches 5.21%. In all test scenarios, Rust is also more memory-efficient, using 52.28 MB at a load of 800 users, compared to Go, which consumes 267.01 MB. In terms of speed, Go performs more optimally than Rust based on throughput and processing time metrics. A notable difference is observed at a load of 600 users, where Go achieves an average throughput of 60.043 transactions per second (tps), while Rust only reaches 8.047 tps. Go also has lower processing times compared to Rust, with a significant difference at a load of 800 users, where Go records 173.47 ms, while Rust reaches 7,408.53 ms. In terms of stability, Go also outperforms Rust, as indicated by a lower average coefficient of variation (CV). For example, at 200 users, Go's coefficient of variation is 29.11%, lower than Rust's 277.63%. Rust excels in scalability, with a throughput increase of 46.057% at 1000 users, compared to Go's 2.074%. In conclusion, Go is more optimal for low loads and applications requiring speed, while Rust is better suited for high loads and applications requiring resource efficiency.

Keywords: web server, Rust, Go, HTTP, performance, load testing

DAFTAR ISI

Halaman Pernyataan.....	ii
Halaman Pengesahan	iii
Halaman Persembahan	iv
Kata Pengantar	v
Abstrak	vi
Abstract	vii
Daftar Isi.....	viii
Daftar Tabel	x
Daftar Gambar.....	xi
Daftar Rumus	xiii
Daftar Lampiran	xiv
Bab I Pendahuluan.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Tujuan Penelitian.....	2
1.4 Pembatasan Masalah	2
1.5 Sistematika Penulisan.....	3
Bab II Tinjauan Pustaka	4
2.1 Penelitian Terkait.....	4
2.2 Dasar Teori	8
2.2.1 <i>Web Server</i>	8
2.2.2 <i>Go Web server</i>	8
2.2.3 <i>Rust Web server</i>	9
2.2.4 <i>HTTP (Hypertext Transfer Protocol)</i>	9
2.2.5 <i>Throughput</i>	10
2.2.6 <i>Performance Testing</i>	11
2.2.7 <i>Load Testing</i>	12
2.2.8 <i>Apache Jmeter</i>	13
2.2.9 <i>Analisis data</i>	14
Bab III Metodologi Penelitian	16
3.1 Perangkat Penelitian	16
3.1.1 Perangkat Keras.....	16
3.1.2 Perangkat Lunak.....	17
3.2 Data.....	18
3.3 Program Penelitian	18
3.3.1 Pengumpulan Data	20
3.3.2 Analisis Kebutuhan	20
3.3.3 Desain Skenario Pengujian.....	20
3.3.3.1 Setup Perangkat <i>Client-Server</i>	20
3.3.3.2 Perancangan Kasus Pengujian	21

3.3.4	Implementasi Pengujian	23
3.3.5	Analisis Hasil	26
3.3.5.1	Analisis Pemanfaatan CPU dan <i>Memory</i>	26
3.3.5.2	Analisis Kecepatan <i>Web server</i>	26
3.3.5.3	Analisis Stabilitas <i>Web server</i>	27
3.3.5.4	Analisis Skalabilitas <i>Web server</i>	28
3.3.6	Penarikan Kesimpulan.....	28
Bab IV Hasil dan Analisis.....		29
4.1	Implementasi	29
4.1.1	Setup Perangkat <i>Client-Server</i> Pengujian	29
4.1.2	Perancangan Kasus Pengujian.....	35
4.1.3	Implementasi Pengujian	38
4.1.3.1	Pengujian Beban 200 User.....	40
4.1.3.2	Pengujian Beban 400 User.....	42
4.1.3.3	Pengujian Beban 600 User.....	45
4.1.3.4	Pengujian Beban 800 User.....	47
4.1.3.5	Pengujian Beban 1000 User.....	50
4.2	Hasil.....	52
4.2.1	Throughput dan Processing time	52
4.2.2	Hasil Pemanfaatan CPU	55
4.2.3	Hasil Pemanfaatan Memori	57
4.3	Analisis Hasil Pengujian.....	58
4.3.1	Analisis Pemanfaatan CPU dan Memori.....	58
4.3.2	Analisis Kecepatan <i>Web Server</i>	61
4.3.3	Analisis Stabilitas <i>Web Server</i>	65
4.3.4	Analisis Skalabilitas <i>Web Server</i>	68
Bab V Kesimpulan dan Saran		70
5.1	Kesimpulan.....	70
5.2	Saran.....	71
Daftar Pustaka		72

DAFTAR TABEL

Tabel 2. 1 Perbandingan Penelitian Terkait.	6
Tabel 3. 1 Perangkat Keras.....	16
Tabel 3. 2 Data Hasil Pengujian <i>Throughput</i>	24
Tabel 3. 3 Tampilan Data Penggunaan CPU.....	25
Tabel 3. 4 Tampilan Data Memori	25
Tabel 4. 1 Konfigurasi Komponen <i>Test Plan</i> Apache Jmeter.	36
Tabel 4. 2 Hasil <i>Throughput</i> dan <i>Processing Time</i> <i>Web Server</i>	53
Tabel 4. 3 Hasil Pemanfaatan CPU oleh <i>Web Server</i>	55
Tabel 4. 4 Hasil Pemanfaatan <i>Memory</i> oleh <i>Web Server</i>	57
Tabel 4. 5 Tabel Perbandingan Pemanfaatan CPU	59
Tabel 4. 6 Tabel Perbandingan Pemanfaatan <i>Memory</i>	60
Tabel 4. 7 Tabel Perbandingan Rata – Rata <i>Processing Time</i>	62
Tabel 4. 8 Tabel Perbandingan <i>Throughput</i>	64
Tabel 4. 9 Tabel Perbandingan Koefisien Variasi pada <i>Web Server</i>	66
Tabel 4. 10 Perbandingan Peningkatan <i>Throughput</i> pada <i>Web Server</i>	69

DAFTAR GAMBAR

Gambar 2. 1 Cara Kerja <i>Web server</i>	8
Gambar 2. 2 Data yang dikirim dari <i>client</i> ke <i>server</i>	12
Gambar 3. 1 Program Penelitian.....	19
Gambar 3. 2 Topologi Jaringan	21
Gambar 3. 3 Konfigurasi Pengujian <i>Client-Server</i>	21
Gambar 3. 4 <i>Flowchart</i> Aplikasi pada <i>Web Server</i> untuk Pengujian.....	23
Gambar 4. 1 Komputer <i>Client</i>	30
Gambar 4. 2 Komputer <i>Server</i>	30
Gambar 4. 3 Hasil Ping dari <i>Client</i> ke <i>Server</i>	31
Gambar 4. 4 Instalasi Visual Studio Code 1.92.0.....	31
Gambar 4. 5 Instalasi Go (Golang) v0.42.0.....	32
Gambar 4. 6 Instalasi Rust v1.0.0.....	32
Gambar 4. 7 Instalasi Java JDK 22.0.1.....	33
Gambar 4. 8 Instalasi Apache Jmeter 5.6.3	33
Gambar 4. 9 Instalasi Perfmon Plugin 2.2.2.....	34
Gambar 4. 10 Instalasi <i>Server Agent</i> 2.2.3	34
Gambar 4. 11 Instalasi <i>Visual Studio Community</i> 2022 v17.10.1	35
Gambar 4. 12 <i>Test Plan</i> Pengujian <i>Web Server</i>	36
Gambar 4. 13 Proses Berjalannya <i>Web Server</i> Rust.....	39
Gambar 4. 14 Proses Berjalannya <i>Web Server</i> Go	39
Gambar 4. 15 Proses Berjalannya <i>Server Agent</i> Pada <i>Server</i>	39
Gambar 4. 16 Menunjukkan Respon <i>Web Server</i> Jika Proses Sukses	40
Gambar 4. 17 <i>Aggregat report</i> <i>Web Server</i> Go Pada Beban 200 <i>User</i>	40
Gambar 4. 18 Pemanfaatan CPU dan <i>Memory</i> Go Pada Beban 200 <i>User</i>	41
Gambar 4. 19 <i>Aggregat report</i> <i>web server</i> Rust Pada Beban 200 <i>User</i>	42
Gambar 4. 20 Pemanfaatan CPU dan <i>Memory</i> Rust Pada Beban 200 <i>User</i>	42
Gambar 4. 21 <i>Aggregat report</i> <i>web server</i> Go Pada Beban 400 <i>User</i>	43
Gambar 4. 22 Pemanfaatan CPU dan <i>Memory</i> Go Pada Beban 400 <i>User</i>	43
Gambar 4. 23 <i>Aggregat report</i> <i>web server</i> Rust Pada Beban 400 <i>User</i>	44
Gambar 4. 24 Pemanfaatan CPU dan <i>Memory</i> Rust Pada Beban 400 <i>User</i>	45
Gambar 4. 25 <i>Aggregat report</i> <i>web server</i> Go Pada Beban 600 <i>User</i>	45

Gambar 4. 26	Pemanfaatan CPU dan <i>Memory</i> Go Pada Beban 600 <i>User</i>	46
Gambar 4. 27	<i>Aggregat report web server</i> Rust Pada Beban 600 <i>User</i>	46
Gambar 4. 28	Pemanfaatan CPU dan <i>Memory</i> Rust Pada Beban 600 <i>User</i>	47
Gambar 4. 29	<i>Aggregat report web server</i> Go Pada Beban 800 <i>User</i>	48
Gambar 4. 30	Pemanfaatan CPU dan <i>Memory</i> Go Pada Beban 800 <i>User</i>	48
Gambar 4. 31	<i>Aggregat report web server</i> Rust Pada Beban 800 <i>User</i>	49
Gambar 4. 32	Pemanfaatan CPU dan <i>Memory</i> Rust Pada Beban 800 <i>User</i>	49
Gambar 4. 33	<i>Aggregat report web server</i> Go Pada Beban 1000 <i>User</i>	50
Gambar 4. 34	Pemanfaatan CPU dan <i>Memory</i> Go Pada Beban 1000 <i>User</i>	51
Gambar 4. 35	<i>Aggregat report web server</i> Rust Pada Beban 1000 <i>User</i>	51
Gambar 4. 36	Pemanfaatan CPU dan <i>Memory</i> Rust Pada Beban 1000 <i>User</i>	52
Gambar 4. 37	Diagram hasil <i>throughput</i> masing-masing <i>web server</i>	53
Gambar 4. 38	Grafik pemanfaatan CPU masing-masing <i>web server</i>	56
Gambar 4. 39	Grafik Pemanfaatan <i>Memory</i> Masing-Masing <i>Web Server</i>	57
Gambar 4. 40	Grafik Pemanfaatan CPU dan <i>Memory</i>	60
Gambar 4. 41	Grafik perbandingan kecepatan <i>web server</i> berdasarkan rasio.....	64
Gambar 4. 42	Grafik analisis stabilitas dari <i>web server</i>	67

DAFTAR RUMUS

Rumus (2.1) <i>Throughput</i>	11
Rumus (2.2) Pemanfaatan CPU	13
Rumus (2.3) Pemanfaatan <i>Memory</i>	13
Rumus (2.4) Rata-rata	15
Rumus (2.5) Standar Deviasi	15
Rumus (2.6) Koefisien Variasi.....	15

DAFTAR LAMPIRAN

Lampiran A	KODE PROGRAM GOLANG.....	A-1
Lampiran B	KODE PROGRAM RUST	B-1
Lampiran C	INDEX.HTML.....	C-1

BAB I PENDAHULUAN

1.1 Latar Belakang

Web server memegang peranan penting untuk melayani *request* dari *browser client* seperti google chrome, firefox, safari dan *software browser* lainnya lalu menampilkan hasil dari proses berupa data-data yang dibutuhkan *client* kembali ke *web browser* (Amrullah, Nugroho, and Ramadhan 2023). Hingga April 2024, terdapat 5,44 miliar pengguna internet di seluruh dunia, yang setara dengan 67,1 persen dari populasi global (Statista 2024). Berdasarkan survei yang dilakukan pada Juli 2024, terdapat lebih dari 1,1 miliar situs web yang aktif dan terhubung ke internet (Netcraft 2024). *Hypertext Transfer Protocol* (HTTP) merupakan salah satu protokol komunikasi yang paling banyak digunakan di dunia. Berdasarkan survei yang dilakukan juli 2024, protokol default http digunakan oleh 86,0% dari semua situs web (W3techs 2024).

Rust dan Go adalah dua bahasa pemrograman yang relatif baru dan sedang naik daun dalam pengembangan *web server*. Go pada juli 2023 berada pada peringkat ke-13, sekarang pada juli 2024 berada pada peringkat ke-7. Rust melonjak dari posisi 17 pada juli 2023 ke posisi 13 pada juli 2024 (TIOBE 2024). Sebagai *web server* golang memiliki keunggulan dari segi kecepatan, skalabilitas dan pemanfaatan *memory* (Pragestu, Sujaini, and Faja 2023). Rust merupakan proyek *open source*, yang berarti dapat digunakan secara gratis dan diubah sesuai kebutuhan. Dari hal perfoma Rust menawarkan layanan yang cepat dan optimal dalam pengelolaan memori sehingga sangat cocok untuk aplikasi yang berkinerja tinggi. Rust sering digunakan untuk pengembangan sistem dan perangkat lunak yang memerlukan kontrol tinggi terhadap sumber daya perangkat keras (Steve Klabnik 2023).

Tujuan dari penelitian ini adalah untuk membandingkan performa *web server* Go dan Rust dalam menangani permintaan klien pada protokol HTTP, termasuk kecepatan, penggunaan sumber daya, stabilitas dan skalabilitas.

1.2 Perumusan Masalah

Adapun rumusan masalah yang diperoleh dari latar belakang permasalahan ini adalah:

1. Bagaimana perbandingan performa antara *web server* Go dan Rust dalam hal pemanfaatan sumber daya (CPU dan Memori), kecepatan, stabilitas, serta skalabilitas?
2. Bagaimana pengaruh penerapan protokol HTTP terhadap performa *web server* Go dan Rust?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Membandingkan performa *web server* Go dan Rust pada protokol HTTP dalam segi penggunaan sumber daya (CPU dan Memori), kecepatan, stabilitas dan skalabilitas.
2. Menganalisis performa *web server* Go dan Rust pada protokol HTTP.

1.4 Pembatasan Masalah

Penelitian ini memiliki batasan masalah sebagai berikut :

1. Penelitian ini tidak mencakup pengaruh perangkat keras, sistem operasi dan kondisi jaringan terhadap performa *web server*.
2. Penelitian ini di batasi pada penggunaan topologi jaringan peer-to-peer dengan koneksi kabel LAN *straight-through*.
3. Penelitian ini menggunakan protokol HTTP (*Hypertext Transfer Protocol*).

1.5 Sistematika Penulisan

Sistematika penulisan penelitian ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan.

BAB I : PENDAHULUAN

Berisi latar belakang penelitian, perumusan masalah, tujuan penelitian, pembatasan masalah, dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Bagian ini memuat landasan teori yang berkaitan dengan penelitian yang akan dilakukan. Beberapa teori yang terkait adalah *web server*, *Go*, *Rust*, *HTTP*, *throughput*, *performance testing*, Pengujian beban, Apache Jmeter, penelitian terkait yang dilakukan oleh peneliti-peneliti sebelumnya dan penjelasan dari berbagai istilah yang terkait dengan topik penelitian.

BAB III : METODOLOGI PENELITIAN

Berisi tentang alat dan perangkat yang digunakan dan langkah penelitian sebagai rincian kegiatan yang dilakukan selama penelitian.

BAB IV : HASIL DAN ANALISIS

Berisi penjelasan mengenai hasil dari implementasi sistem yang analisis dan dirancang pada Bab III dan analisis hasil pengujian sistem untuk dilakukan penarikan kesimpulan.

BAB V : PENUTUP

Berisi kesimpulan dari penelitian yang telah dilakukan dan saran/rekomendasi untuk perbaikan, pengembangan maupun kesempurnaan dan kelengkapan penelitian yang telah dilakukan.