

BAB 1. PENDAHULUAN

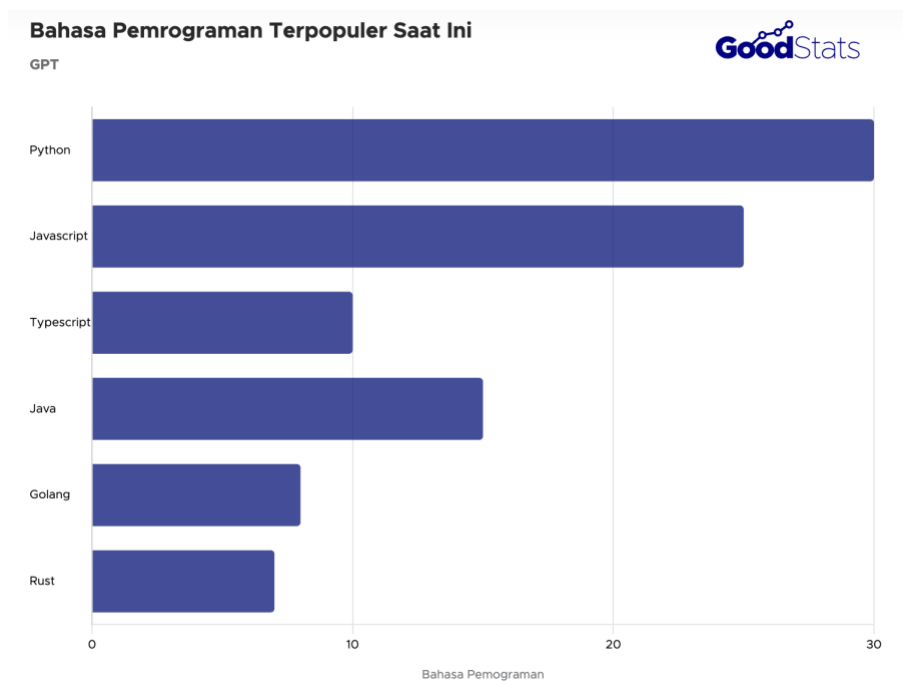
1.1 Latar Belakang

Berdasarkan data dari Asosiasi Penyelenggara Jasa Internet Indonesia (APJII), jumlah pengguna internet di Indonesia pada tahun 2024 mencapai 221.563.479 jiwa dari total populasi sebesar 278.696.200 jiwa penduduk Indonesia tahun 2023 (Haryanto, 2024). Peningkatan signifikan dalam jumlah pengguna internet ini menunjukkan semakin pentingnya kebutuhan akan aplikasi *web* yang mampu beroperasi dengan cepat dan stabil. Aplikasi modern saat ini sering kali dituntut untuk melayani ribuan hingga jutaan pengguna secara bersamaan, sehingga kecepatan respons serta penggunaan sumber daya *server* menjadi faktor yang sangat menentukan keberhasilan aplikasi *web service*.

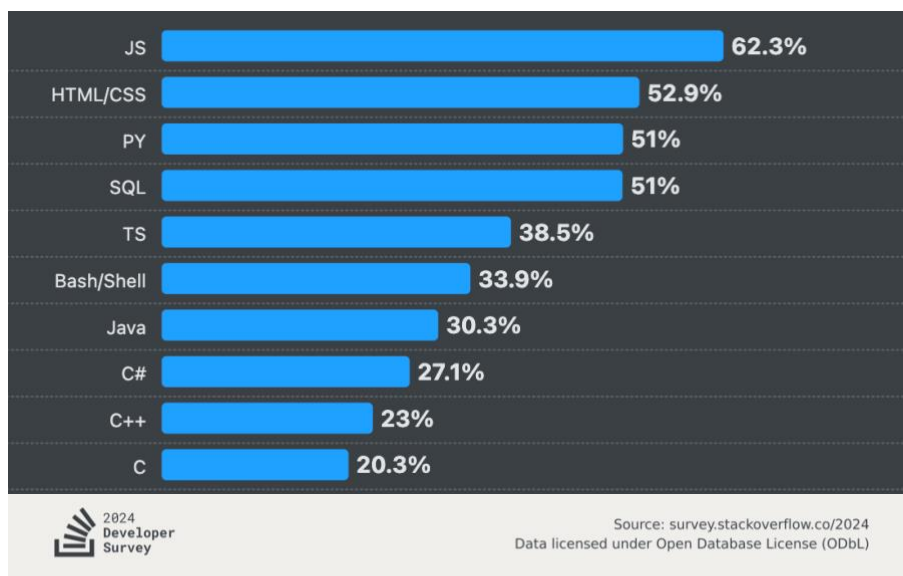
Untuk memenuhi kebutuhan tersebut, aplikasi *web service* dapat dikatakan berkinerja baik apabila mampu menangani sejumlah besar *request* dari pengguna tanpa mengonsumsi sumber daya *server* secara berlebihan. Hal ini sangat penting untuk mengurangi biaya operasional yang harus ditanggung oleh pengelola aplikasi. Salah satu komponen utama yang mendukung kinerja aplikasi *web service* adalah *Application Programming Interface* (API), yang berfungsi sebagai penghubung antara berbagai layanan atau sistem melalui protokol HTTP (Ardiansyah & Fatwanto, 2022). API berperan sebagai jembatan antara aplikasi *client* dan *server*, memfasilitasi pertukaran data, serta dirancang untuk menangani ribuan hingga jutaan *request* dari berbagai *client* secara simultan (Mubariz dkk., 2020).

Namun, keberhasilan implementasi API sangat bergantung pada peran *backend framework* yang digunakan. Dalam pengembangan aplikasi *web service*, *backend framework* memiliki peran penting dalam menentukan kecepatan respons aplikasi serta penggunaan sumber daya seperti CPU dan RAM (Mulana dkk., 2022). Dengan kemampuannya dalam *code reusability*, *backend framework* memungkinkan pengembang untuk mempercepat proses pengembangan aplikasi dengan memfokuskan lebih banyak waktu pada implementasi logika bisnis daripada membangun infrastruktur dari awal (Ghimire, 2020). Selain itu, *backend*

framework umumnya dilengkapi dengan berbagai fitur, seperti pengelolaan *routing*, *middleware*, dan integrasi dengan basis data. Hampir setiap bahasa pemrograman modern memiliki *backend framework* yang dirancang untuk membantu pengembang menciptakan aplikasi dengan lebih terstruktur.



Gambar 1.1 Survei GoodStats Bahasa Pemrograman Terpopuler 2024



Gambar 1.2 Survei *Stack Overflow* Teknologi Terpopuler 2023

Golang, Node.js, dan Python merupakan tiga bahasa pemrograman yang populer dalam pengembangan *web service*. Golang dipilih karena tren penggunaannya di industri yang semakin meningkat. Meskipun tergolong baru, berdasarkan survei yang dilakukan oleh GoodStats tahun 2024 pada gambar 1.1, Golang telah menempati posisi kelima sebagai bahasa pemrograman terpopuler dan telah banyak diadopsi oleh industri sebagai bagian dari *backend* stack mereka (Gunawan, 2024). Di sisi lain, Node.js dan Python dipilih karena tingkat popularitasnya yang tinggi, sebagaimana ditunjukkan dalam survei Stack Overflow tahun 2023 pada gambar 1.2, di mana kedua bahasa ini termasuk dalam tiga teknologi yang paling banyak digunakan, bersama dengan HTML/CSS. Javascript yang menjadi dasar Node.js digunakan oleh 62,3% responden, sementara Python digunakan oleh 51% dari total 60.171 partisipan (Yepis, 2025).

Saat ini, tidak sedikit pengembang *web service* cenderung memilih bahasa pemrograman yang sudah dikuasai untuk mengembangkan *web service* tanpa mempertimbangkan performa, penggunaan sumber daya *server*, atau jumlah pengguna yang akan mengakses aplikasi. Pendekatan seperti ini dapat menyebabkan aplikasi kurang optimal, terutama dalam hal kecepatan respon dan penggunaan sumber daya *server*. Padahal, kecepatan respons dan kemampuan aplikasi dalam menangani jumlah pengguna yang besar secara simultan sangat penting untuk memberikan pengalaman pengguna yang baik. Oleh karena itu, penting bagi pengembang untuk memahami keunggulan dan keterbatasan berbagai bahasa pemrograman atau *backend framework*, guna memilih teknologi yang paling sesuai dengan aplikasi yang dikembangkan (Winardi dkk., 2023)

Berdasarkan pertimbangan yang ada, penelitian ini bertujuan untuk menganalisis performa kecepatan respons dan penggunaan sumber daya *server* dari penggunaan bahasa pemrograman Golang, Node.js, dan Python dalam pengembangan *web service*. Untuk itu, akan dibuat tiga aplikasi *web service* yang masing masing menggunakan *framework* Gin untuk Golang, Express untuk Node.js, dan Flask untuk Python. Ketiga aplikasi tersebut akan dibangun dengan arsitektur dan basis data yang serupa guna memastikan konsistensi hasil pengujian. Pengujian akan dilakukan menggunakan Apache Jmeter dengan metode *load*

testing untuk melihat apakah aplikasi dapat merespons dengan cepat dan tetap stabil saat menghadapi berbagai tingkat beban pengguna. Selain itu, selama pengujian penggunaan sumber daya CPU dan RAM akan dicatat secara terperinci. Hasil pengujian ini akan dianalisis untuk mengidentifikasi keunggulan dan keterbatasan masing-masing bahasa pemrograman, sehingga dapat ditentukan bahasa apa yang paling tepat untuk *web service* yang akan dikembangkan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah disampaikan, masalah yang akan dibahas dalam penelitian ini dapat dirumuskan sebagai berikut:

- a. Bagaimana perbandingan kecepatan respons antara *backend framework* Gin, Express, dan Flask dalam API *web service*?
- b. Bagaimana perbandingan penggunaan sumber daya *server*, yaitu CPU dan RAM, antara *backend framework* Gin, Express, dan Flask dalam API *web service*?
- c. Bagaimana karakteristik keunggulan dan keterbatasan *backend framework* Gin, Express, dan Flask dalam API *web service* berdasarkan kecepatan respons dan penggunaan sumber daya *server*?

1.3 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut:

- a. Menganalisis dan membandingkan kecepatan respons antara *backend framework* Gin, Express, dan Flask, dalam API *web service*.
- b. Menganalisis dan membandingkan penggunaan sumber daya *server*, yaitu CPU dan RAM, antara *backend framework* Gin, Express, dan Flask, dalam API *web service*.
- c. Mengidentifikasi keunggulan dan keterbatasan *backend framework* Gin, Express, dan Flask dalam pengembangan API *web service* berdasarkan hasil analisis kecepatan respons dan penggunaan sumber daya *server*.

1.4 Manfaat

Adapun manfaat yang diperoleh dari penelitian ini adalah sebagai berikut:

- a. Sebagai bahan pertimbangan bagi pengembang *web service* dalam memilih *backend framework* yang sesuai dengan kebutuhan aplikasi berdasarkan hasil analisis kecepatan respons dan penggunaan sumber daya *server* dalam penelitian ini.
- b. Dari sisi akademik, penelitian ini menambah wawasan dalam bidang pemrograman *backend* serta dapat menjadi referensi untuk penelitian selanjutnya.

1.5 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

- a. Penelitian ini hanya berfokus pada satu *backend framework* masing-masing dari Golang, Node.js, dan Python untuk membangun aplikasi *web service* dengan arsitektur dan basis data yang serupa.
- b. Penelitian ini tidak membahas aspek lain seperti keamanan dan skalabilitas dari aplikasi *web service*.