

LAPORAN TUGAS

ANALISA PERBANDINGAN GO, NODE JS, PYTHON DAN LARAVEL DALAM PENGGUNAAN SEBAGAI LAYANAN BACKEND



Disusun oleh:

Nama: Rizki Shafara Adiyatma

NIM: A11.2023.15431

Program Studi: Teknik Informatika – S1

FAKULTAS ILMU KOMPUTER

UNIVERSITAS DIAN NUSWANTORO

SEMARANG

2025

BAB I

PENDAHULUAN

1.1. Latar Belakang

Layanan backend merupakan tulang punggung dari hampir semua aplikasi perangkat lunak, mulai dari aplikasi web hingga layanan mobile. Pemilihan teknologi backend yang tepat adalah salah satu keputusan paling krusial yang dapat mempengaruhi performa, skalabilitas, biaya pemeliharaan, dan kecepatan pengembangan sebuah proyek. Kesalahan dalam memilih tech stack dapat menyebabkan kesulitan teknis untuk pemeliharaan lebih lanjut.

Studi kasus yang diangkat dalam laporan ini adalah pembuatan layanan backend sederhana dengan fungsi CRUD (Create, Read, Update, Delete) untuk entitas produk. Dengan mengimplementasikan kasus yang sama pada empat platform yang berbeda Golang, Flask (Python), Laravel (PHP), dan Node.js (Express.js) kita dapat melakukan perbandingan yang adil dan relevan mengenai karakteristik masing-masing teknologi.

1.2. Rumusan Masalah

Penelitian ini bertujuan untuk menjawab pertanyaan-pertanyaan berikut:

- a. Bagaimana perbandingan performa, kecepatan eksekusi, dan konsumsi sumber daya antara layanan CRUD yang dibangun menggunakan Golang, Flask, Laravel, dan Node.js?
- b. Bagaimana perbandingan kemudahan dan kecepatan pengembangan (produktivitas developer) pada keempat teknologi tersebut untuk kasus implementasi layanan CRUD?
- c. Bagaimana perbandingan ekosistem, ketersediaan pustaka (library), dan dukungan komunitas untuk setiap teknologi dalam konteks pengembangan backend?

1.3. Ruang Lingkup

Analisis dalam laporan ini memiliki batasan sebagai berikut:

- a. Analisis terbatas pada implementasi dasar fungsionalitas CRUD (membuat produk, membaca daftar/detail produk, memperbarui produk, dan menghapus produk) tanpa logika bisnis yang kompleks.
- b. Analisis hanya mencakup Golang dengan pustaka standar net/http, Python dengan micro-framework Flask, PHP dengan full-featured framework Laravel, dan JavaScript dengan runtime Node.js dan framework Express.js.
- c. Perbandingan dilakukan secara kualitatif berdasarkan pengalaman implementasi dan studi literatur, bukan melalui benchmark kuantitatif yang terukur secara presisi.

BAB II

ANALISA

2.1. Go (net/http)

Golang, atau Go, adalah bahasa pemrograman yang dikompilasi dan diketik secara statis (statically typed) yang dikembangkan oleh Google. Go dirancang untuk menjadi sederhana, efisien, dan andal, dengan dukungan bawaan yang kuat untuk pemrograman konkuren.

a. Kelebihan:

- a. Performa Tinggi: Sebagai bahasa kompilasi, Go menghasilkan binary tunggal yang berjalan langsung di atas sistem operasi, menghasilkan kecepatan eksekusi yang sangat tinggi dan latensi rendah.
- b. Concurrency Bawaan: Goroutine dan channel memudahkan penanganan ribuan koneksi secara bersamaan dengan konsumsi memori yang efisien.
- c. Static Typing: Sistem tipe yang kuat membantu mendeteksi error pada saat kompilasi, bukan saat runtime, sehingga meningkatkan keandalan kode.
- d. Single Binary Deployment: Aplikasi dapat didistribusikan sebagai satu file eksekusi tanpa ketergantungan eksternal, menyederhanakan proses deployment.

b. Kekurangan:

- a. Verbosity: Untuk tugas sederhana seperti CRUD, Go bisa terasa lebih verbose (membutuhkan lebih banyak baris kode) dibandingkan framework dinamis.
- b. Manajemen Error: Pola penanganan error `if err != nil` yang repetitif dapat membuat kode terlihat berulang.
- c. Ekosistem: Meskipun berkembang pesat, ekosistem pustakanya belum sebesar Node.js (NPM) atau Python (PyPI).
- c. Analisis Implementasi: Mengembangkan service dengan net/http di Golang memberikan kontrol penuh atas setiap aspek permintaan dan respons HTTP. Prosesnya sangat transparan, namun membutuhkan penyiapan manual untuk routing, parsing JSON, dan koneksi basis data. Hasilnya adalah sebuah layanan yang sangat cepat dan efisien dalam penggunaan memori, tetapi memerlukan waktu pengembangan yang sedikit lebih lama dibandingkan menggunakan framework lain.

2.2. Python (Flask)

Flask adalah micro-framework untuk pengembangan web dengan bahasa Python. Disebut "mikro" karena intinya dirancang agar tetap sederhana dan dapat diperluas (extensible), memberikan kebebasan kepada pengembang untuk memilih pustaka dan alat yang ingin mereka gunakan.

a. Kelebihan:

- a. **Fleksibilitas Tinggi:** Flask tidak memaksakan struktur proyek atau ketergantungan tertentu, memberikan kebebasan penuh kepada pengembang.
- b. **Mudah dipelajari:** Sintaksis Python yang bersih dan dokumentasi Flask yang sangat baik membuatnya mudah dipelajari oleh pemula.
- c. **Ekosistem Python:** Mendapat manfaat dari ekosistem Python yang sangat matang, terutama untuk analisis data, machine learning, dan komputasi ilmiah.
- b. **Kekurangan:**
 - a. **Memerlukan Ekstensi:** Untuk fungsionalitas yang lebih kompleks seperti ORM, autentikasi, atau migrasi basis data, pengembang perlu mengintegrasikan ekstensi pihak ketiga (misalnya, SQLAlchemy, Flask-Login).
 - b. **Performa:** Sebagai bahasa yang diinterpretasikan (interpreted), performa Python umumnya di bawah bahasa kompilasi seperti Golang.
- c. **Analisis Implementasi:** Implementasi CRUD dengan Flask terasa mudah. Mendefinisikan route dan handler sangat sederhana. Fleksibilitasnya memungkinkan integrasi yang mudah dengan pustaka ORM seperti SQLAlchemy untuk interaksi database. Proyek ini sangat cocok untuk pengembangan cepat, prototipe, atau layanan mikro yang tidak memerlukan semua fitur yang disediakan oleh framework besar.

2.3. PHP (Laravel)

Laravel adalah framework aplikasi web PHP dengan sintaksis yang ekspresif dan elegan. Laravel mengikuti pola arsitektur MVC (Model-View-Controller) dan menyediakan ekosistem yang sangat kaya dengan fitur bawaan untuk mempercepat proses pengembangan.

- a. **Kelebihan:**
 - a. **Pengembangan Cepat:** Fitur seperti Eloquent ORM, database migration, sistem templating Blade, dan routing yang sederhana memungkinkan pembuatan aplikasi fungsional dalam waktu yang sangat singkat.
 - b. **Ekosistem Terintegrasi:** Laravel menawarkan berbagai alat dan layanan resmi seperti Forge, Vapor, dan Nova yang menyederhanakan deployment dan manajemen aplikasi.
 - c. **Komunitas Besar:** Memiliki salah satu komunitas terbesar dan paling aktif, dengan dokumentasi yang sangat lengkap dan banyak tutorial.
- b. **Kekurangan:**
 - a. **Overhead:** Sebagai framework berfitur lengkap, Laravel memiliki overhead performa dan konsumsi memori yang lebih besar dibandingkan micro-framework atau pustaka standar.
 - b. **Kurva Belajar:** Meskipun dokumentasinya bagus, menguasai seluruh ekosistem dan konsep "Laravel way" membutuhkan waktu.

- c. Analisis Implementasi: Pengembangan layanan API dengan Laravel adalah pengalaman yang paling produktif. Dengan perintah artisan, model, controller, dan migrasi dapat dibuat secara otomatis. Eloquent ORM menyederhanakan interaksi dengan basis data menjadi sangat intuitif. Seluruh proses terasa terstruktur dan mengikuti "convention over configuration," yang meminimalkan kode boilerplate. Menurut Dewaweb (2023), pendekatan ini sangat efektif untuk proyek yang membutuhkan kecepatan pengembangan.

2.4. Node.js (Express.js)

Node.js adalah runtime environment JavaScript yang dibangun di atas mesin V8 Chrome. Express.js adalah framework aplikasi web minimalis dan fleksibel untuk Node.js, yang telah menjadi standar de facto untuk membangun API dan aplikasi web dengan Node.js.

- a. Kelebihan:
 - a. Non-Blocking I/O: Model event-driven dan non-blocking I/O membuatnya sangat efisien untuk menangani banyak koneksi bersamaan, ideal untuk aplikasi real-time dan I/O-intensif.
 - b. Ekosistem NPM: Node Package Manager (NPM) adalah registri perangkat lunak terbesar di dunia, menyediakan pustaka untuk hampir semua kebutuhan.
 - c. Satu Bahasa: Memungkinkan penggunaan JavaScript di sisi server dan klien, menyederhanakan proses pengembangan untuk tim full-stack.
- b. Kekurangan:
 - a. Manajemen Asinkron: Penggunaan callback, Promise, dan async/await bisa menjadi kompleks untuk dikelola, terkadang mengarah ke "callback hell" jika tidak ditangani dengan baik.
 - b. Single-Threaded: Node.js berjalan pada satu thread. Tugas yang memakan banyak CPU (CPU-bound) dapat memblokir event loop dan menurunkan performa seluruh aplikasi.
- c. Analisis Implementasi: Implementasi dengan Express.js mirip dengan Flask dalam hal minimalisme dan fleksibilitas. Menyiapkan server dan routing sangatlah mudah. Sifat asinkron Node.js sangat terasa saat berinteraksi dengan basis data, di mana penggunaan async/await menjadi esensial untuk menjaga kode tetap bersih dan mudah dibaca. Kecepatan respons untuk operasi I/O (seperti kueri basis data) terasa sangat cepat. Seperti yang dibahas oleh Niagahoster (2022), kekuatan utama Node.js terletak pada kemampuannya menangani operasi I/O secara efisien.

BAB III

PERBANDINGAN KOMPARATIF

Kriteria	Golang (net/http)	Flask (Python)	Laravel (PHP)	Node.js (Express.js)
Performa & Kecepatan Eksekusi	Sangat Tinggi	Sedang	Sedang	Tinggi (terutama untuk I/O)
Konsumsi Memori	Sangat Rendah	Rendah hingga Sedang	Sedang hingga Tinggi	Rendah hingga Sedang
Kemudahan & Kecepatan Pengembangan	Sedang (lebih <i>verbose</i>)	Sangat Cepat (untuk prototipe)	Sangat Cepat (untuk aplikasi lengkap)	Cepat
Skalabilitas	Sangat Tinggi (skalabilitas horizontal mudah)	Tinggi (dengan arsitektur yang tepat)	Tinggi (membutuhkan konfigurasi infrastruktur)	Sangat Tinggi (untuk I/O-bound)
Ekosistem & Ketersediaan Library	Baik dan terus berkembang	Sangat Luas (PyPI)	Sangat Luas (Packagist) dan terintegrasi	Paling Luas (NPM)
Kurva Pembelajaran	Sedang (konsep <i>concurrency & typing</i>)	Rendah	Sedang (banyak konsep <i>framework</i>)	Rendah ke Sedang (konsep asinkron)
Komunitas dan Dokumentasi	Aktif dan berkualitas tinggi	Sangat Besar dan beragam	Sangat Besar dan terfokus	Sangat Besar dan aktif

BAB IV

KESIMPULAN

Berdasarkan analisis yang telah dilakukan, dapat disimpulkan bahwa tidak ada satu teknologi pun yang menjadi solusi terbaik untuk semua kasus. Pilihan teknologi sangat bergantung pada prioritas dan kebutuhan spesifik proyek.

- a. Golang lebih cocok untuk aplikasi yang menuntut performa, efisiensi memori, dan konkurensi tinggi, seperti microservice berperforma tinggi
- b. Flask sangat ideal untuk prototipe cepat, microservice sederhana, API, atau proyek di mana fleksibilitas dan integrasi dengan ekosistem Machine Learning menjadi prioritas
- c. Laravel sangat baik untuk proyek yang membutuhkan pengembangan cepat dari awal hingga akhir dengan fitur yang lengkap, seperti aplikasi web monolitik, CMS, atau platform e-commerce
- d. Node.js (Express.js) adalah pilihan yang sangat kuat untuk aplikasi yang bersifat I/O-intensif, seperti aplikasi chat, streaming, dasbor real-time, atau API yang melayani banyak klien secara bersamaan.

DAFTAR PUSTAKA

1. Dewaweb. (2022). *Apa itu Laravel? Pengertian, Fitur dan Kelebihannya*. Diakses dari <https://www.dewaweb.com/blog/apa-itu-laravel>.
2. Hostinger. (2024). *Apa itu Laravel: fitur utama, contoh penggunaan, dan kelebihanannya*. Diakses dari <https://www.hostinger.com/id/tutorial/apa-itu-laravel>.
3. Niagahoster. (2022). *Apa itu Node.js: Memahami cara kerja dan cara menggunakannya*. Diakses dari <https://www.niagahoster.co.id/blog/node-js-adalah>.
4. Novalagung, M. (2022). *Dasar Pemrograman Golang: Web Service API Server*. Diakses dari <https://dasarpemrogramangolang.novalagung.com/A-web-service-api.html>.
5. Pallets Projects. (2023). *Flask Documentation (2.3.x)*. Diakses dari <https://flask.palletsprojects.com/en/2.3.x/>.