MENGUJI PERFORMA DAN SKALABILITAS: STUDI PERBANDINGAN RUST DAN KOTLIN DALAM PENGEMBANGAN PERANGKAT LUNAK

Priyanto Tamami
Fungsional Pranata Komputer Badan Pendapatan Daerah Kabupaten Brebes
tamami.oka@gmail.com

Jurnal Ultras
Volume 8 Nomor 1 (hlm. 22 – 25)

Info Artikel:
Diterima: 13 Desember 2024
Disetujui: 13 Desember 2024

Kata Kunci :
transisi teknologi, rust, kotlin, *virtual machine*, pengambangan perangkat lunak, performa, keamanan, skalabilitas

Keywords :
technology transition, rust, kotlin, virtual machine, software development, performance, security, scalability

Abstrak

Dalam pengembangan perangkat lunak dewasa ini teknologi yang digunakan cukup beragam, mulai dari penggunaan teknologi *virtual machine*, bahasa terkompilasi, dan bahasa terinterpretasi. Pengembangan perangkat lunak yang secara khusus terjadi di lingkungan Badan Pendapatan Daerah Kabupaten Brebes, yang dilakukan di beberapa sistem secara swakelola oleh pegawai internal dilakukan menggunakan *virtual machine* (vm), dan seiring berjalannya waktu karena perangkat yang melakukan akses berasal dari kondisi kecepatan akses internet yang beragam, muncul beberapa kendala seperti latensi dan respon yang membutuhkan waktu lama untuk menampilkan informasi. Melihat akan proses yang cepat terhadap kebutuhan informasi, maka Badan Pendapatan Daerah membutuhkan kajian terhadap bahasa pemrograman tertentu yang mampu menyelesaikan permasalahan tersebut. Kriteria yang menjadi pertimbangan selain dari penggunaan respon dan latensi adalah kondisi performa dari masing-masing bahasa pemrograman dalam menangani banyak permintaan, serta faktor skalabilitas dari lingkungan pengembangannya. Maka dengan penelitian ini diharapkan mampu menjawab bahasa pemrograman apa yang mungkin menjadi pilihan dalam pengembangan perangkat lunak di masa mendatang.

Abstract

In contemporary software development, the technologies employed are highly diverse, ranging from the use of virtual machines, compiled languages, and interpreted languages. In the context of software development within Badan Pendapatan Daerah Kabupaten Brebes, several internally managed systems have been developed using virtual machines (VMs). Over time, challenges such as latency and delayed response times have arisen due to the varying internet access speeds of devices interacting with the systems. To address the need for rapid information access, Badan Pendapatan Daerah Kabupaten Brebes seeks to evaluate programming languages capable of resolving these issues. Key considerations include response time, latency reduction, the performance of various programming languages in handling high request volumes, and the scalability of the development environment. This study aims to identify

programming languages that might serve as optimal choices for future software development within the agency.

PENDAHULUAN

Bahasa pemrograman sangat erat kaitannya dengan proses data yang terjadi pada perangkat keras, semakin efisien dan efektif dalam melakukan eksekusi tiap perintah, semakin cepat respon yang dapat diberikan oleh perangkat lunak yang dikembangkan.

Proses yang efektif dan efisien dalam melakukan suatu baris perintah dalam sebuah bahasa pemrograman sangat erat kaitannya dengan konsep eksekusi kode program dengan basis terkompilasi, interpretasi, atau *virtual machines* (VM).

Beberapa contoh Perusahaan yang melakukan migrasi dari tingkat bahasa pemrograman salah satunya adalah Meta (Facebook) yang melakukan modifikasi penggunaan PHP dengan Hack (Verlaguet et al., 2014) dan mengadopsi Go (Hale, 2022) untuk layanan *backend* yang membutuhkan performa tinggi dan penanganan *concurrency* yang lebih baik.

TINJAUAN PUSTAKA

Pada penelitian berjudul Analisis Perbandingan Performa Web Service Menggunakan Bahasa Pemrograman Python, PHP, dan Perl pada Client Berbasis Android (Harismawan et al. 2017) mendapatkan hasil bahwa setiap bahasa pemrograman memiliki perbedaan performa dalam melakukan respon terhadap permintaan data.

Pada penelitian dengan judul Pengujian API Website untuk Perbaikan Performansi Aplikasi Ditenun (Arlinta et al. 2021) mendapatkan hasil bahwa penggunaan *Load Test* mampu mengukur dan melihat penurunan performa API berdasarkan jumlah *user* yang melakukan permintaan data.

Untuk kebutuhan pengujian, penelitian ini akan bersumber dari kode program yang disajikan dalam artikel *Rust CRUD Rest API with Docker* (Ciulla, 2023) dimana layanan ini mampu menyediakan data yang ditampilkan dari basis data PostgreSQL melalui protokol REST, sedangkan kode program yang dibangun dengan Kotlin akan melakukan respon yang sama berdasarkan kode yang dibangun dengan Rust.

METODE PENELITIAN

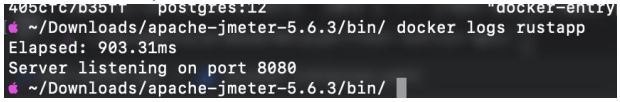
Penelitian ini dilakukan di lingkungan Badan Pendapatan Daerah Kabupaten Brebes pada tahun 2024.

Metode penelitian yang digunakan adalah *benchmarking* terhadap waktu inisialisasi perangkat lunak yang dibangun, dan metode simulasi beban kerja, dimana kedua bahasa pemrograman akan dilihat bagaimana proses perangkat lunak yang dibangun mampu menangani stres.

Alat yang digunakan untuk memicu beban kerja terhadap kedua layanan yang dibangun menggunakan bahasa pemrograman yang berbeda adalah JMeter.

HASIL DAN DISKUSI

Hasil *benchmarking* terhadap waktu inisialisasi perangkat lunak yang dibangun menggunakan bahasa pemrograman Rust seperti terlihat pada Gambar 1 dimana sistem yang dibangun membutuhkan waktu inisialisasi selama 903,31 milidetik.



Gambar 1. Waktu inisialisasi Rust

Sedangkan waktu inisialisasi yang dibutuhkan oleh sistem perangkat lunak yang dibangun menggunakan Kotlin mencapai 31,952 detik sebagaimana terlihat pada Gambar 2.

Gambar 2. Waktu inisialisasi Kotlin

Hasil *benchmark* tersebut menunjukkan bahwa sistem perangkat lunak yang dibangun menggunakan Rust jauh lebih unggul pada saat inisialisasi sistem karena bahasa pemrograman telah terkompilasi ke bahasa *native* dari sistem operasi yang berjalan, sehingga komunikasi data antar perangkat keras terjadi lebih cepat.

Sedangkan sistem perangkat lunak yang dibangun menggunakan Kotlin membutuhkan waktu yang lebih lama karena sebelum menjalankan perangkat lunak yang dibangun menggunakan bahasa pemrograman, sistem perlu menjalankan *virtual machines* terlebih dahulu sebagai jembatan penghubung antara bahasa pemrograman yang telah dikompilasi dengan perangkat keras, proses inilah yang menjadikan perangkat lunak yang dibangun menggunakan teknologi *virtual machines* memiliki waktu inisialisasi yang lebih lama.

Pada metode simulasi beban kerja menggunakan aplikasi JMeter, parameter yang digunakan untuk melakukan *stress-test* adalah dengan 100 koneksi aktif dengan 10 kali iterasi di masing-masing koneksinya.

Hasil yang didapat untuk sistem perangkat lunak yang dibangun menggunakan bahasa pemrograman Rust sebagaimana terlihat pada Tabel 1, dimana untuk menyelesaikan proses secara keseluruhan membutuhkan waktu 21 detik dengan waktu rata-rata yang dibutuhkan 48,2 proses per detik.

koneksi	waktu proses	kecepatan
1 - 395	9 detik	41,9 / detik
1 - 605	11 detik	53,4 / detik
1 - 1000	21 detik	48,2 / detik

Tabel 1. *Stress-test* Rust

Sedangkan untuk sistem perangkat lunak yang dibangun menggunakan bahasa pemrograman Kotlin dapat dilihat pada Tabel 2, dimana untuk menyelesaikan seluruh proses *request* membutuhkan waktu total 13 detik dengan kecepatan 74,5 proses per detik, kondisi ini menggambarkan bahwa perangkat lunak yang dibangun dengan bahasa pemrograman Kotlin lebih cepat dan stabil untuk menangani beberapa *request* sekaligus pada waktu yang hampir bersamaan

koneksi	waktu proses	kecepatan
1	3 detik	0,3 / detik
1 - 999	10 detik	99,9 / detik
1 - 1000	13 detik	74,5 / detik

Tabel 2. *Stress-test* Kotlin

Kedua sistem perangkat lunak yang dibangun sudah menggunakan sistem virtualisasi Docker, sehingga kedua sistem sudah sama-sama mampu melakukan penyesuaian skalabilitas.

KESIMPULAN DAN SARAN

Hasil yang didapatkan penelitian kali ini yaitu, bila membutuhkan waktu inisialisasi yang cepat, maka pilihan bahasa pemrograman Rust lebih tepat, namun perlu dipertimbangkan waktu respon saat menangani beberapa koneksi secara bersamaan yang tentunya akan berdampak pada lamanya informasi yang ditampilkan di aplikasi pada sisi pengguna.

Sedangkan bila prioritas yang dipilih adalah waktu respon yang cepat terhadap beberapa permintaan data yang datang bersamaan, pilihan sistem yang dibangun menggunakan Kotlin tentu lebih tepat.

Karena keterbatasan pemahaman peneliti terhadap bahasa pemrograman Rust, tentunya hasil yang didapat mungkin terlihat tidak sebanding pada penelitian kali ini, di masa mendatang mungkin perlu dilakukan penelitian pembanding dimana sistem yang dibangun menggunakan bahasa Rust perlu menggunakan *framework* tertentu yang dapat melakukan optimasi terhadap koneksi dengan sistem basis data, sebagaimana yang telah dilakukan pada pengembangan sistem perangkat lunak dengan

bahasa pemrograman Kotlin yang menggunakan Spring Data JPA untuk melakukan koneksi ke sistem basis data.

DAFTAR PUSTAKA

Harismawan, A. F., Kharisma, A. P., & Afirianto, T. (2017). Analisis Perbandingan Performa Web Service Menggunakan Bahasa Pemrograman Python, PHP, dan Perl pada Client Berbasis Android. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *2*(1), 237–245.

Barus, A. C., Harungguan, J., & Manulu, E. (2021). Pengujian api website untuk performansi aplikasi ditenun. *Journal of Applied Technology and Informatics Indonesia*, 1(2), 14-21.

Verlaguet, J., Menghrajani, A. (2014). Hack: a new programming language for HHVM. Daring dari alamat https://engineering.fb.com/2014/03/20/developer-tools/hack-a-new-programming-language-for-hhvm/. Diakses pada 8 Oktober 2024.

Hale, C. (2022). Meta Developers will be asked to program almost exclusively in these four programming languages. Daring dari alamat https://www.techradar.com/news/meta-developers-will-be-asked-to-program-almost-exclusively-in-these-four-programming-languages. Diakses pada 8 Oktober 2024.

Ciulla, Francesco. Rust CRUD Rest API with Docker. Daring dari alamat https://dev.to/francescoxx/rust-crud-rest-api-3n45. Diakses pada 10 Oktober 2024.