

BENCHMARK KINERJA BACKEND: STUDI KASUS PERBANDINGAN BAHASA KOTLIN DAN PHP DALAM MENANGANI PERMINTAAN WEB

Priyanto Tamami

Badan Pendapatan Daerah Kabupaten Brebes
tamami.oka@gmail.com

Jurnal Ultras

Volume 8 Nomor 2 (hlm. 68 – 71)

Info Artikel:

Diterima : 02 Juni 2025

Disetujui : 26 Juni 2025

Kata Kunci :

transisi teknologi, php, kotlin, *virtual machine*,
pengembangan perangkat lunak, performa,
keamanan, skalabilitas

Keywords :

technology transition, php, kotlin, virtual machine,
software development, performance, security,
scalability

Abstrak.

Pemilihan bahasa pemrograman erat kaitannya dengan bagaimana cara mereka dieksekusi oleh komputer, sehingga dalam prosesnya akan menentukan seberapa efektif dan efisien suatu perangkat lunak berjalan dalam suatu sistem operasi, terutama dalam keterbatasan sumber daya yang dimiliki oleh Badan Pendapatan Daerah Kabupaten Brebes. Dengan kebutuhan tersebut maka diharapkan dampak dari penelitian ini memberikan hasil perbandingan dua bahasa pemrograman populer yang biasa digunakan pemrogram dalam pengembangan aplikasi berbasis web. Melalui pendekatan eksperimental, pengujian yang dilakukan hanya terhadap lamanya waktu inisialisasi dan lamanya waktu respon terhadap permintaan yang masuk ke server secara paralel. Hasil penelitian menunjukkan bahwa Kotlin lebih unggul baik dari sisi inisialisasi sistem dan lamanya waktu respon. Diharapkan studi ini memberikan wawasan empiris bagi pengambil keputusan teknis dalam memilih teknologi yang sesuai dengan kebutuhan proyek perangkat lunak yang efektif dan efisien.

Abstract.

The choice of programming language is closely related to how it is executed by the computer, which in turn affects how effectively and efficiently software operates within an operating system—particularly under the resource constraints faced by Badan Pendapatan Daerah Kabupaten Brebes. Given these needs, this study aims to provide a comparative analysis of two popular programming languages commonly used by developers in web-based application development. Using an experimental approach, the study focuses solely on measuring system initialization time and the response time to parallel incoming requests to the server. The results indicate that Kotlin outperforms the alternative language in both system initialization and response time. This study is expected to offer empirical insights for technical decision-makers in selecting technologies that align with the goals of building effective and efficient software systems.

PENDAHULUAN

Memilih sebuah bahasa pemrograman bukan hal mudah, karena tidak ada satu bahasa pemrograman yang bagus untuk seluruh kondisi pengembangan perangkat lunak, masing-masing memiliki cara kerja yang khas dan bagaimana tiap lingkungan bahasa pemrograman berinteraksi terhadap sistem operasi.

Berdasarkan cara bahasa pemrograman dieksekusi oleh komputer, maka akan memiliki tiga pendekatan berbeda, yang pertama adalah *interpreter-based language* dimana bahasa pemrograman akan dieksekusi langsung tiap barisnya dan diterjemahkan ke dalam bahasa mesin, yang kedua adalah *compiler-based language* dimana bahasa pemrograman perlu dilakukan kompilasi terlebih dahulu ke dalam bahasa mesin sesuai Sistem Operasi yang digunakan

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).

TINJAUAN PUSTAKA

Penulis akan menggunakan istilah bahasa pemrograman adalah alat untuk mengekspresikan algoritma kepada mesin dengan cara yang dapat dibaca, dipahami, dan dieksekusi oleh komputer (Sebesta, R.W. 2022) atau sarana untuk menginstruksikan komputer melalui pernyataan-pernyataan yang terstruktur dan logis (Tanenbaum, A.S. 2012).

Penulis juga akan menggunakan istilah *Virtual Machine* yang mengacu pada definisi abstraksi dari perangkat keras komputer yang memungkinkan satu mesin fisik bertindak seperti beberapa mesin, masing-masing menjalankan sistem operasinya sendiri (Silberschatz, A. Galvin, P. B., Gagne, G. 2018).

Pada penelitian dengan judul *AST vs. Bytecode: Interpreters in the Age of Meta Compilation* (LaRose, E., Marr, S., Gonzalez Boix, E., & De Meuter, W. 2017) melakukan implementasi terhadap beberapa interpreter, melakukan evaluasi kecepatan, penggunaan memori, dan *startup* time, namun penelitian ini hanya sebatas pada bahasa pemrograman interpreter yang terkompilasi saja dan hasil dari AST (*abstract-syntax-tree*) mampu menyaingi *bytecode* dalam hal fleksibilitas, namun *bytecode* lebih hemat memori dalam penggunaannya.

Pada penelitian *A Comparative Study of Programming Languages in Rosetta Code* (Nanz, S., & Furia, C.A. 2015) yang menguji 8 bahasa pemrograman (C, Go, C#, Java, F#, Haskell, Python, dan Ruby) untuk melakukan 745 tugas dalam Rosetta Code mendapatkan hasil :

1. Bahasa fungsional seperti F# dan Haskell, kode programnya lebih ringkas dibandingkan bahasa prosedural C dan Go serta bahasa berorientasi objek C# dan Java.
2. Bahasa C sangat unggul dalam hal kecepatan (*raw speed*) saat memproses input berukuran besar
3. Perbedaan performa menjadi kurang mencolok pada input dengan ukuran sedang, memungkinkan bahasa dengan basis interpretasi pun tetap kompetitif.
4. Bahasa yang dikompilasi dan bertipe kuat (*strongly-typed*), di mana lebih banyak kesalahan bisa ditangkap saat kompilasi, cenderung lebih sedikit mengalami kegagalan saat dijalankan dibandingkan dengan bahasa dengan basis interpretasi atau bertipe lemah.

Dari penelitian-penelitian tersebut belum memberikan hasil jika implementasi digunakan pada aplikasi berbasis web, terkhusus untuk layanan *back-end*. Untuk

kebutuhan layanan berbasis web, penelitian ini dilakukan.

Penelitian ini hanya menguji seberapa efektif dan efisien layanan yang dibangun dengan bahasa PHP dan Kotlin dalam menjawab permintaan yang disediakan dengan metode GET pada protokol REST dengan hasil teks "test", masing-masing bahasa pemrograman akan memanfaatkan *framework* yang sudah digunakan secara luas, dimana Kotlin akan menggunakan *framework* Spring Boot, dan PHP akan menggunakan *framework* Laravel.

METODE PENELITIAN

Penelitian ini dilakukan di lingkungan Badan Pendapatan Daerah Kabupaten Brebes pada tahun 2025.

Metode penelitian yang digunakan adalah studi kuantitatif dengan pendekatan *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.

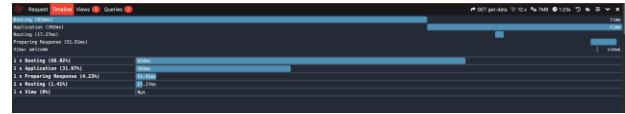
Alat yang digunakan untuk memicu beban kerja terhadap kedua layanan yang dibangun menggunakan bahasa pemrograman yang berbeda adalah JMeter.

Penelitian diawali dengan membangun *web service* sederhana yang menyediakan metode GET dari layanan Restful di kedua bahasa pemrograman. Setelah itu sistem layanan dijalankan di mesin yang sama, dan dilakukan pengukuran menggunakan aplikasi JMeter.

HASIL DAN DISKUSI

Hasil *benchmarking* terhadap waktu inisialisasi perangkat lunak yang dibangun menggunakan bahasa pemrograman PHP dengan Laravel seperti terlihat pada Gambar 1

dimana sistem yang dibangun membutuhkan waktu inisialisasi selama 834 milidetik.



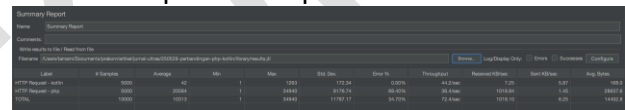
Gambar 1. Waktu inisialisasi PHP

Sedangkan waktu inisialisasi yang dibutuhkan oleh sistem perangkat lunak yang dibangun menggunakan Kotlin mencapai 22 milidetik sebagaimana terlihat pada Gambar 2.



Gambar 2. Waktu inisialisasi Kotlin

Hasil *benchmark* dari kedua layanan tersebut dapat dilihat pada Gambar 3 berikut :



Gambar 3. Hasil pengukuran layanan

Dari kolom Label bisa dilihat bahwa ada proses permintaan ke masing-masing layanan, dimana label kotlin mengindikasikan proses permintaan yang terjadi ke layanan yang dibangun dengan bahasa Kotlin, dan label php adalah indikasi proses permintaan yang terjadi ke layanan yang dibangun dengan bahasa PHP.

Total permintaan yang terjadi ada 10.000 permintaan, yang masing-masing layanan akan menerima 5.000 permintaan yang muncul dalam kolom #Samples untuk menguji tingkat stres layanan.

Pada kolom Average menunjukkan bahwa rata-rata waktu respon dalam satuan milidetik menunjukkan layanan yang dibangun dengan bahasa pemrograman Kotlin unggul dengan mencapai 42 milidetik saja, sedangkan pada layanan yang dibangun dengan bahasa pemrograman PHP mencapai 20.584 milidetik, atau lebih dari 20 detik. Selisih waktu respon menunjukkan kesenjangan yang sangat jauh antara kedua sistem ini.

Pada kolom Min kedua layanan mampu memberikan waktu respon tercepat yang sama, yaitu 1 milidetik, namun pada kolom Max, layanan yang dibangun dengan bahasa Kotlin menempuh respon terlama dalam waktu 1.263 milidetik, atau sekitar dari 1 detik, sedangkan layanan yang dibangun dengan bahasa PHP menyentuh waktu 34.940 milidetik, atau sekitar 34 detik.

Pada kolom Error % adalah persentase permintaan yang mengalami kegagalan proses, hasilnya, layanan yang dibangun dengan bahasa Kotlin unggul dengan nilai 0,00%, artinya semua permintaan yang datang berhasil dijawab, sedangkan layanan yang dibangun dengan PHP mendapatkan nilai 69,4% yang artinya sistem dalam kondisi rawan bila menerima 5.000 permintaan data bertubi.

Pada kolom Throughput, adalah nilai yang menunjukkan berapa jumlah permintaan per detiknya. Pada layanan yang dibangun dengan Kotlin, mendapatkan nilai 44,2/sec yang artinya dalam 1 detik, layanan mampu menjawab sekitar 44 permintaan sekaligus, sedangkan layanan yang dibangun dengan PHP mendapatkan nilai 36,4/sec yang artinya dalam 1 detik, layanan mampu menangani sekitar 36 permintaan.

KESIMPULAN DAN SARAN

Hasil yang didapatkan pada penelitian kali ini ditemukan bahwa layanan yang dibangun dengan bahasa Kotlin jauh lebih unggul dari semua parameter pengukuran.

Peneliti meyakini memiliki keterbatasan pemahaman terhadap bahasa pemrograman PHP yang mungkin menyebabkan hasil pengukuran terlihat hasil yang jauh sekali perbedaannya sehingga penelitian lanjutan perlu dilakukan oleh peneliti atau bersama peneliti yang paham betul proses optimasi yang dapat dilakukan di lingkungan bahasa

pemrograman PHP sehingga dapat menghasilkan penelitian yang lebih objektif.

DAFTAR PUSTAKA

- Sebesta, R. W. (2022). *Concepts of Programming Languages* (12th ed.). Pearson Education.
- Tanenbaum, A. S. (2012). *Structured Computer Organization* (6th ed.). Pearson.
- Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts* (10th ed.). Wiley.
- LaRose, E., Marr, S., Gonzalez Boix, E., & De Meuter, W. (2017). AST vs. bytecode: Interpreters in the age of meta-compilation. *Proceedings of the ACM on Programming Languages*, 1(OOPSLA), 1–27. <https://doi.org/10.1145/3133877>
- Nanz, S., & Furia, C. A. (2015). A comparative study of programming languages in Rosetta Code. *Empirical Software Engineering*, 21(2), 590–632. <https://doi.org/10.1109/ICSE.2015.90>