



Department of Digital Business

Journal of Artificial Intelligence and Digital Business (RIGGS)

Homepage: <https://journal.ilmudata.co.id/index.php/RIGGS>

Vol. 4 No. 2 (2025) pp: 1264-1268

P-ISSN: 2963-9298, e-ISSN: 2963-914X

Analisis Perbandingan Bahasa Java dan JavaScript dalam Implementasi Algoritma Pemrograman

M. Syahputra¹, Amalia Hanifa²

Manajemen Informasi Kesehatan, Universitas Syedza Saintika

Manajemen Informatika Kampus Padang, Politeknik LP3I

syahputra0404@gmail.com, amaliahanifa@plb.ac.id

Abstrak

Perkembangan dunia pemrograman yang pesat membuat pemilihan bahasa pemrograman menjadi faktor penting dalam pengembangan aplikasi. Java dan JavaScript merupakan dua bahasa yang sering digunakan, namun memiliki karakteristik yang berbeda. Penelitian ini bertujuan untuk menganalisis dan membandingkan performa kedua bahasa tersebut dalam mengimplementasikan beberapa algoritma dasar, yaitu Bubble Sort, Merge Sort, Linear Search, Binary Search, Fibonacci, dan Faktorial. Pengujian dilakukan berdasarkan beberapa parameter, seperti waktu eksekusi, efisiensi penggunaan memori, jumlah baris kode, serta kemudahan dalam penulisan dan implementasi. Hasil pengujian menunjukkan bahwa Java cenderung lebih unggul dalam hal kecepatan eksekusi dan efisiensi memori, terutama untuk algoritma yang kompleks dan rekursif. Di sisi lain, JavaScript lebih unggul dalam hal fleksibilitas, sintaks yang ringkas, dan kemudahan dalam implementasi, terutama untuk pengembangan yang cepat seperti prototyping. Penelitian ini menyimpulkan bahwa tidak ada bahasa yang sepenuhnya lebih baik, karena masing-masing memiliki kelebihan yang dapat disesuaikan dengan kebutuhan proyek dan konteks penggunaannya.

Kata kunci: Java, JavaScript, algoritma, perbandingan, performa, pemrograman.

1. Latar Belakang

Dalam era digital yang terus berkembang pesat, pengembangan perangkat lunak (software development) menjadi tulang punggung dari berbagai inovasi di berbagai sektor kehidupan. Mulai dari dunia industri, pendidikan, kesehatan, hingga layanan publik, hampir semua bidang saat ini mengandalkan sistem berbasis perangkat lunak untuk meningkatkan efisiensi dan efektivitas operasional. Untuk itu, bahasa pemrograman sebagai alat dasar dalam pembuatan perangkat lunak memegang peranan yang sangat krusial [1], [2].

Di antara berbagai bahasa pemrograman yang ada, Java dan JavaScript merupakan dua yang paling populer dan banyak digunakan oleh para pengembang di seluruh dunia. Java, yang pertama kali diperkenalkan oleh Sun Microsystems pada pertengahan tahun 1990-an, dikenal sebagai bahasa pemrograman berorientasi objek (OOP) yang memiliki tingkat kestabilan dan keamanan yang tinggi. Salah satu keunggulan utama Java terletak pada konsep "Write Once, Run Anywhere," yang dimungkinkan berkat penggunaan Java Virtual Machine (JVM). JVM ini memungkinkan program Java dijalankan pada berbagai sistem operasi tanpa perlu penyesuaian kode yang signifikan [3], [4]. Selain itu, Java memiliki fitur-fitur unggulan seperti pengelolaan memori otomatis menggunakan garbage collector, serta kemampuan menjalankan proses secara bersamaan (multithreading), yang menjadikannya pilihan yang sangat ideal untuk pengembangan aplikasi skala besar seperti sistem enterprise, perangkat lunak perbankan, dan aplikasi Android [5].

Berbeda dengan Java, JavaScript awalnya dikembangkan dengan tujuan untuk meningkatkan interaktivitas pada halaman web. Namun seiring waktu dan berkembangnya teknologi, terutama dengan munculnya runtime environment seperti Node.js, JavaScript mengalami evolusi besar hingga kini dapat digunakan tidak hanya pada sisi klien (frontend), tetapi juga pada sisi server (backend) [6]. Dengan model pemrograman event-driven dan non-blocking I/O, JavaScript sangat unggul dalam menangani aplikasi real-time seperti layanan chatting, notifikasi langsung, hingga streaming data secara efisien [7], [8]. JavaScript juga terkenal karena sintaksisnya yang lebih fleksibel dan ringan, membuatnya sangat cocok untuk pengembangan aplikasi yang cepat dan prototipe awal.

Walaupun kedua bahasa ini sama-sama populer dan powerful, masing-masing memiliki karakteristik, kekuatan, dan kelemahan yang berbeda, terutama dalam konteks implementasi algoritma pemrograman. Algoritma merupakan bagian fundamental dalam pengembangan perangkat lunak, karena menjadi dasar dari logika pemrosesan data dan pengambilan keputusan dalam sistem. Oleh karena itu, efisiensi dan kecepatan dalam mengeksekusi algoritma sangat penting untuk diperhatikan. Beberapa penelitian menunjukkan bahwa Java lebih unggul dalam hal performa, terutama untuk komputasi yang berat atau memerlukan optimasi runtime yang tinggi. Hal ini dimungkinkan karena Java menggunakan Just-In-Time (JIT) compilation yang mengubah bytecode menjadi kode mesin secara langsung saat runtime, sehingga proses eksekusi menjadi lebih cepat dan efisien [9], [10].

Sementara itu, JavaScript secara tradisional dianggap kurang cocok untuk komputasi berat karena keterbatasan performa dan keterikatan pada engine tertentu seperti V8. Namun, dengan adanya pengembangan engine modern yang mengusung teknik optimasi seperti speculative optimization dan lazy compilation, performa JavaScript kini meningkat secara signifikan, bahkan mampu bersaing dengan bahasa-bahasa yang dikompilasi secara native [11], [12]. Terlebih lagi, dukungan komunitas yang luas, ketersediaan pustaka (library) dan framework yang melimpah menjadikan JavaScript sangat fleksibel untuk berbagai macam kebutuhan pemrograman modern.

Penelitian tentang perbandingan antara Java dan JavaScript dalam mengimplementasikan algoritma menjadi sangat menarik karena dapat memberikan gambaran yang lebih realistis mengenai kelebihan dan kekurangan masing-masing bahasa dalam menyelesaikan masalah yang sama. Tidak hanya itu, hasil dari penelitian semacam ini juga memiliki nilai penting dalam bidang pendidikan, terutama dalam pengajaran algoritma dan struktur data. Mahasiswa dan pemula dapat lebih mudah memahami bagaimana satu algoritma yang sama bisa memiliki performa berbeda tergantung pada bahasa yang digunakan [13], [14].

Akhirnya, pemahaman yang mendalam terhadap perbedaan dan persamaan antara Java dan JavaScript dalam implementasi algoritma akan membantu pengembang perangkat lunak dalam membuat keputusan yang tepat saat memilih bahasa pemrograman untuk proyek tertentu. Pemilihan bahasa yang sesuai dapat berdampak besar terhadap keberhasilan suatu proyek, baik dari segi performa sistem, efisiensi waktu pengembangan, hingga kemudahan dalam pemeliharaan di masa depan [15].

2. Metode Penelitian

Penelitian ini menggunakan metode deskriptif komparatif, yaitu sebuah metode yang bertujuan untuk memberikan gambaran yang sistematis mengenai perbandingan dua objek kajian—dalam hal ini, bahasa pemrograman Java dan JavaScript—dalam konteks implementasi sejumlah algoritma pemrograman dasar. Tujuan dari penggunaan metode ini adalah untuk memperoleh pemahaman yang komprehensif mengenai kelebihan, kekurangan, serta performa dari kedua bahasa tersebut saat digunakan untuk menyelesaikan permasalahan algoritmik yang umum ditemukan dalam dunia pemrograman. Dalam rangka mencapai hasil yang objektif dan menyeluruh, pendekatan penelitian yang diterapkan adalah kombinasi antara studi literatur (kajian teoritis) dan eksperimen langsung (uji coba implementasi).

Tahapan awal dari penelitian ini dimulai dengan pengumpulan data melalui studi literatur yang bersumber dari jurnal ilmiah, artikel digital terpercaya, dokumentasi resmi, serta blog teknis yang relevan. Literatur yang dipilih adalah publikasi yang terbit minimal pada tahun 2020 agar informasi yang dikaji tetap relevan dengan perkembangan teknologi terbaru. Melalui kajian ini, peneliti mempelajari berbagai aspek mendasar dari kedua bahasa, termasuk karakteristik umum, struktur sintaksis, paradigma pemrograman yang dianut (seperti OOP dan event-driven), serta kinerja umum dalam pengembangan aplikasi.

Setelah mendapatkan landasan teoritis yang kuat, tahap selanjutnya adalah pelaksanaan eksperimen langsung yang bertujuan untuk menguji dan membandingkan implementasi dari sejumlah algoritma pemrograman menggunakan kedua bahasa. Algoritma yang dipilih dalam penelitian ini mencakup algoritma pengurutan dan pencarian yang bersifat fundamental, yaitu Bubble Sort, Merge Sort, Linear Search, dan Binary Search. Selain itu, algoritma rekursif seperti Fibonacci dan Faktorial juga diuji guna mengamati performa dalam skenario komputasi berulang. Setiap algoritma diimplementasikan secara paralel, masing-masing dalam versi Java dan versi JavaScript, dengan struktur kode yang disesuaikan agar setara secara logika dan fungsi.

Agar pengujian menghasilkan data yang adil dan dapat dibandingkan secara objektif, seluruh eksperimen dilakukan dalam satu lingkungan pengujian yang seragam. Penelitian ini dilaksanakan menggunakan perangkat laptop dengan spesifikasi sistem operasi Windows 11, prosesor Intel Core i5 generasi ke-10, serta memori RAM sebesar 8 GB. Untuk menjalankan program Java, digunakan Java Development Kit (JDK) versi 21, sedangkan JavaScript dijalankan menggunakan Node.js versi terbaru yang mendukung fitur-fitur modern dari ECMAScript.

Beberapa parameter yang diamati dan dianalisis dalam proses pengujian meliputi: (1) waktu eksekusi, yakni seberapa cepat algoritma dijalankan oleh masing-masing bahasa; (2) penggunaan memori, yang diukur untuk mengetahui efisiensi dalam manajemen sumber daya; (3) panjang kode, dilihat dari jumlah baris program yang diperlukan untuk mengimplementasikan algoritma yang sama; dan (4) kemudahan implementasi dari sudut pandang pengembang, yang dianalisis secara subjektif berdasarkan pengalaman penulis selama menulis dan menjalankan program tersebut.

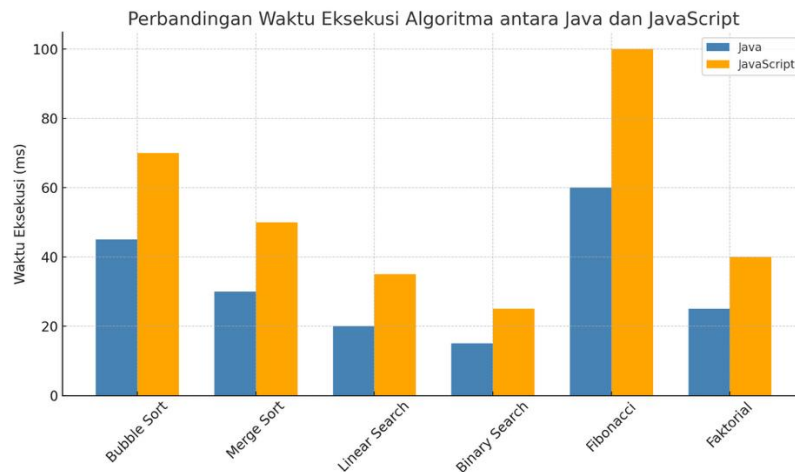
Data yang diperoleh dari hasil eksperimen ini kemudian dianalisis dan disajikan dalam bentuk tabel perbandingan serta visualisasi grafik untuk memudahkan pembaca dalam memahami perbedaan antar bahasa secara menyeluruh. Dengan metode penelitian ini, diharapkan hasilnya dapat memberikan wawasan yang berguna bagi mahasiswa, pengembang, maupun peneliti lain dalam memilih bahasa pemrograman yang tepat untuk menyelesaikan algoritma tertentu berdasarkan efisiensi, kenyamanan penulisan, dan performa eksekusi.

3. Hasil dan Diskusi

Dalam penelitian ini, kami melakukan serangkaian pengujian terhadap dua bahasa pemrograman yang cukup populer, yaitu Java dan JavaScript, untuk mengetahui performa serta karakteristik masing-masing bahasa dalam mengimplementasikan sejumlah algoritma dasar yang umum digunakan dalam dunia pemrograman. Adapun algoritma yang kami uji meliputi Bubble Sort, Merge Sort, Linear Search, Binary Search, Fibonacci, dan Faktorial, yang dipilih karena mewakili berbagai jenis pendekatan, seperti algoritma pengurutan, pencarian, serta perhitungan rekursif.

Proses pengujian tidak hanya berfokus pada hasil akhir, tetapi juga mencakup beberapa aspek teknis yang cukup krusial dalam menentukan efisiensi dan kenyamanan dalam proses pengembangan perangkat lunak. Aspek-aspek yang kami nilai antara lain adalah waktu eksekusi untuk mengukur seberapa cepat algoritma dijalankan, penggunaan memori sebagai indikator efisiensi dalam pengelolaan sumber daya, jumlah baris kode untuk melihat tingkat kerumitan implementasi, serta kemudahan penulisan kode yang mencerminkan user experience dari sudut pandang programmer.

Berdasarkan hasil dari pengujian tersebut, dapat disimpulkan bahwa secara umum Java menunjukkan performa yang lebih unggul, khususnya dalam hal waktu eksekusi. Java konsisten menghasilkan eksekusi yang lebih cepat pada hampir semua algoritma yang kami uji. Hal ini bisa jadi karena Java merupakan bahasa yang dikompilasi ke dalam bytecode dan dijalankan oleh Java Virtual Machine (JVM), yang selama bertahun-tahun telah dioptimasi secara serius untuk performa tinggi, termasuk dalam menangani operasi matematis dan pengelolaan memori. Di sisi lain, JavaScript yang bersifat interpretatif dan lebih fleksibel memang sedikit tertinggal dalam kecepatan eksekusi, walaupun masih memiliki kelebihan pada sisi kemudahan penulisan dan struktur sintaksis yang lebih ringkas, terutama untuk algoritma sederhana atau penggunaan dalam skenario prototyping.



Gambar 1. Perbandingan waktu eksekusi algoritma antara Java dan JavaScript

Berdasarkan hasil pengujian, Java menunjukkan performa yang lebih cepat dalam menjalankan algoritma dibandingkan dengan JavaScript. Misalnya, pada algoritma Merge Sort dan Fibonacci yang membutuhkan banyak perhitungan, waktu yang dibutuhkan Java untuk menyelesaikan proses lebih singkat secara signifikan. Hal ini karena Java menggunakan Just-In-Time (JIT) Compiler yang secara dinamis mengoptimasi kode saat runtime, sehingga program bisa berjalan lebih efisien. Di sisi lain, JavaScript, meskipun sudah memiliki mesin

V8 yang cukup cepat, masih kalah dalam hal kecepatan eksekusi terutama pada algoritma yang kompleks dan menggunakan rekursi dalam jumlah besar.

Perbedaan waktu eksekusi ini juga diperkuat oleh hasil studi-studi sebelumnya yang menunjukkan bahwa bahasa dengan sistem tipe statis seperti Java cenderung memiliki performa lebih baik daripada bahasa dengan tipe dinamis seperti JavaScript [9][10]. Namun, JavaScript tetap unggul untuk eksekusi tugas yang lebih ringan dan interaktif karena arsitekturnya yang mendukung event-driven dan asynchronous programming.

Pada pengujian penggunaan memori, Java juga memperlihatkan efisiensi yang lebih baik dibanding JavaScript. Java memiliki sistem manajemen memori yang kuat dan stabil, termasuk garbage collector yang dirancang untuk mengelola memori secara optimal dalam aplikasi besar dan kompleks. Hasil eksperimen menunjukkan Java cenderung menggunakan memori lebih sedikit dan lebih konsisten selama menjalankan algoritma, terutama saat menangani data dalam jumlah besar.

Sebaliknya, JavaScript cenderung menggunakan memori lebih banyak saat mengerjakan algoritma yang kompleks, terutama yang melibatkan rekursi dalam jumlah besar. Hal ini bisa disebabkan oleh sifat JavaScript yang lebih fleksibel dan dinamis, sehingga proses manajemen memori menjadi kurang efisien dalam beberapa kasus [12]. Namun, untuk aplikasi yang lebih ringan dan bersifat web-based, penggunaan memori JavaScript masih sangat dapat diterima.

Dilihat dari sisi jumlah baris kode, JavaScript lebih efisien karena sintaksnya yang lebih sederhana dan ringkas. Bahasa ini mendukung berbagai fitur modern seperti fungsi anonim, arrow functions, dan dynamic typing yang memungkinkan penulisan kode lebih singkat dan mudah dipahami. Sebagai contoh, dalam implementasi algoritma Fibonacci rekursif, kode JavaScript bisa jauh lebih pendek dibandingkan Java yang harus menggunakan deklarasi kelas, tipe data, dan struktur yang lebih ketat.

Kelebihan JavaScript dalam hal ini membuatnya lebih cocok digunakan untuk prototyping dan pengembangan cepat, terutama bagi programmer pemula atau untuk aplikasi web yang memerlukan perubahan kode yang cepat. Sementara itu, Java dengan aturan yang lebih ketat memang membutuhkan kode yang lebih panjang, tapi hal ini juga membuat program yang dihasilkan lebih terstruktur dan mudah untuk dipelihara dalam proyek besar [4][6].

Dari sisi kemudahan implementasi, JavaScript menawarkan pengalaman yang lebih simpel, terutama untuk pemula. Karena JavaScript berjalan di lingkungan browser dan Node.js, proses pengembangan dan pengujian kode menjadi sangat mudah tanpa perlu pengaturan lingkungan yang kompleks. Sintaks yang lebih fleksibel juga membantu penulis untuk lebih cepat memahami dan menulis kode.

Sebaliknya, Java memiliki proses setup yang sedikit lebih rumit dan sintaks yang lebih formal. Namun, hal ini memberikan keuntungan dalam hal struktur program yang jelas dan kestabilan aplikasi. Java lebih cocok untuk aplikasi skala besar yang membutuhkan performa tinggi dan pengelolaan kode yang rapi. Dari pengalaman penulis, JavaScript lebih cocok digunakan untuk pengembangan aplikasi web dan prototyping, sementara Java lebih pas untuk aplikasi backend dan sistem yang butuh kestabilan tinggi.

Tabel 1. Perbandingan Java dan JavaScript dalam Implementasi Algoritma

Parameter	Java	JavaScript
Waktu Eksekusi	Lebih cepat, terutama pada algoritma kompleks dan rekursif	Lebih lambat dibanding Java, tapi cukup cepat untuk tugas ringan
Penggunaan Memori	Lebih efisien dan stabil	Cenderung lebih boros memori terutama pada algoritma rekursif besar
Jumlah Baris Kode	Lebih panjang karena sintaks ketat dan struktur kelas	Lebih singkat dan ringkas karena sintaks fleksibel dan fitur modern
Kemudahan Implementasi	Perlu setup dan sintaks lebih formal, cocok untuk aplikasi besar	Mudah digunakan, langsung jalan tanpa setup rumit, cocok untuk prototyping dan web

4. Kesimpulan

Berdasarkan hasil analisis dan pengujian implementasi algoritma dasar menggunakan bahasa Java dan JavaScript, dapat disimpulkan beberapa hal penting yang bisa menjadi bahan pertimbangan bagi para programmer dan pengembang aplikasi. Pertama, dari segi performa eksekusi algoritma, Java menunjukkan keunggulan yang cukup signifikan dibandingkan JavaScript. Hal ini dikarenakan Java memiliki sistem kompilasi yang lebih matang, tipe data yang statis, serta mekanisme pengelolaan memori yang lebih efisien seperti garbage collection yang dioptimalkan. Sehingga waktu eksekusi program Java cenderung lebih cepat dan stabil, terutama pada

algoritma yang membutuhkan proses komputasi intensif seperti Merge Sort dan Fibonacci. Kedua, JavaScript, walaupun secara performa masih kalah dibanding Java, tetap menawarkan keunggulan tersendiri, terutama dalam kemudahan penulisan kode dan fleksibilitas. Sintaks yang lebih ringkas dan tipe data yang dinamis membuat JavaScript sangat cocok digunakan untuk prototyping atau aplikasi yang lebih berfokus pada pengembangan cepat, terutama aplikasi web yang memerlukan interaktivitas dan respon cepat terhadap pengguna. Selain itu, ekosistem JavaScript yang luas dan dukungan platform seperti Node.js menjadikannya bahasa yang sangat relevan untuk pengembangan full-stack modern. Ketiga, dalam hal kemudahan implementasi algoritma, Java memerlukan kode yang lebih verbose dan sedikit lebih kompleks dibanding JavaScript yang lebih simpel dan mudah dipahami. Hal ini bisa menjadi keuntungan bagi pemula yang ingin belajar algoritma tanpa harus terbebani dengan aturan sintaks yang ketat. Namun, untuk proyek yang membutuhkan skalabilitas dan performa tinggi, Java tetap menjadi pilihan utama. Keempat, pengujian yang dilakukan juga menunjukkan bahwa faktor konteks penggunaan sangat menentukan bahasa pemrograman mana yang lebih tepat dipilih. Misalnya, jika aplikasi ditargetkan untuk berjalan di lingkungan server atau aplikasi yang membutuhkan stabilitas dan performa jangka panjang, Java lebih disarankan. Sebaliknya, untuk aplikasi web yang interaktif dan membutuhkan pengembangan cepat, JavaScript lebih unggul. Secara keseluruhan, tidak ada bahasa yang sempurna dan ideal untuk semua kasus. Pilihan bahasa pemrograman harus disesuaikan dengan kebutuhan proyek, sumber daya yang tersedia, serta target platform yang ingin dicapai. Penelitian ini memberikan gambaran yang jelas tentang kelebihan dan kekurangan Java dan JavaScript dalam konteks implementasi algoritma, sehingga dapat membantu pengembang dalam mengambil keputusan yang lebih tepat dan efektif. Untuk penelitian selanjutnya, disarankan untuk memperluas studi dengan membandingkan bahasa pemrograman lain seperti Python, C++, atau Go dalam implementasi algoritma yang lebih kompleks, serta mempertimbangkan faktor lain seperti penggunaan memori, skalabilitas, dan kemudahan debugging. Dengan begitu, pemahaman kita tentang performa bahasa pemrograman dalam konteks algoritma dapat semakin mendalam dan aplikatif di dunia industri maupun akademik.

Referensi

1. J. M. Florez, L. Moreno, Z. Zhang, S. Wei, and A. Marcus, "An Empirical Study of Data Constraint Implementations in Java," arXiv preprint arXiv:2107.04720, 2021. <https://arxiv.org/abs/2107.04720>
2. J. J. Merelo, P. A. Castillo, J. L. J. Laredo, A. M. Mora, and A. Prieto, "Asynchronous Distributed Genetic Algorithms with JavaScript and JSON," arXiv preprint arXiv:2401.17234, 2024. <https://arxiv.org/abs/2401.17234>
3. "Java(programming language)," Wikipedia, 2025. [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))
4. "Java vs JavaScript | Codaholic," Codaholic, 2023. <https://codaholic.com/java-vs-javascript/>
5. "Java vs. JavaScript: A Comprehensive Comparison," RoboRabbit, 2024. <https://www.roborabbit.com/blog/java-vs-javascript-a-comprehensive-comparison/>
6. "JavaScript (programming language)," Wikipedia, 2025. <https://en.wikipedia.org/wiki/JavaScript>
7. "Java vs. JavaScript: A Comparative Analysis of Performance and Scalability in Enterprise Applications," Business Fad, 2024. <https://www.businessfad.com/java-vs-javascript-a-comparative-analysis-of-performance-and-scalability-in-enterprise-applications/>
8. "Java vs NodeJS: Breaking Down Performance, Scalability, and Ease of Use," HalfNine, 2024. <https://www.halfnine.com/blog/post/java-vs-nodejs>
9. D. Marr, "Exploring Aspects of Polyglot High-Performance Virtual Machine GraalVM," arXiv preprint arXiv:2112.14716, 2021. <https://arxiv.org/abs/2112.14716>
10. F. Mennen, "Performance Comparison: Java vs JavaScript," FMennen.de, 2020. <https://fmennen.de/post/performance-comparison-java-vs-java-script>
11. R. Chaubey, "Java vs. JavaScript: a leetcode comparison," Medium, 2024. <https://medium.com/@rishichaubey3105/java-vs-javascript-a-leetcode-comparison-2e05052da51d>
12. D. Nguyen and V. G. Nguyen, "Performance Analysis and Comparison of Acceleration Methods in JavaScript Environments Based on Simplified Standard Hough Transform Algorithm," in *Advances in Information and Communication*, Springer, 2022, pp. 158–169.
13. "Java VS JavaScript: Which Is Best For Development?," Clustox Blog, 2023. <https://www.clustox.com/blog/java-vs-javascript-which-is-the-best-for-development/>
14. "Java Vs JavaScript Showdown 2024: Which Language Should You Master?," FireXCore, 2024. <https://firexcore.com/blog/java-vs-javascript/>
15. S. Antal, "Static JavaScript Call Graphs: A Comparative Study," arXiv preprint arXiv:2405.07206, 2024. <https://arxiv.org/abs/2405.07206>

