

Perkembangan Bahasa Pemrograman Terkini: Kajian Literatur Sistematis Terhadap Preferensi, Kinerja, dan Penggunaan di Dunia Industri dan Akademik

Yudhi Biantoro¹, Yasmianti², Ramadhani Ulansari³, Jenih⁴
F.A Ricky Bayu Styanto⁵, Suharyanto⁶

Universitas Respati Indonesia

¹yudhi.biantoro@gmail.com, ²yasmianti@urindo.ac.id, ³ramadhani.ulansari@urindo.ac.id,
⁴jenih@urindo.ac.id, ⁵ricky@urindo.ac.id, ⁶suharyanto@urindo.ac.id

Abstrak

Dalam satu dekade terakhir, dunia pengembangan perangkat lunak mengalami perubahan signifikan dengan kemunculan bahasa pemrograman baru dan pergeseran preferensi pengembang. Studi ini menyajikan *Systematic Literature Review* (SLR) yang menganalisis tren penggunaan bahasa pemrograman dari 2010 hingga 2025. Tujuan penelitian adalah untuk mengidentifikasi bahasa pemrograman yang paling banyak digunakan dan diteliti, mengevaluasi faktor-faktor yang memengaruhi pemilihan bahasa, serta membandingkan performa bahasa dalam berbagai domain. Literatur dianalisis menggunakan protokol PRISMA, menghasilkan 67 artikel dari 1.325 yang disaring. Hasil menunjukkan dominasi Python, JavaScript, dan Java, serta peningkatan penggunaan Rust, Kotlin, dan TypeScript. Faktor penentu meliputi kemudahan sintaksis, dukungan komunitas, dan kekayaan pustaka. Studi ini menguji dua hipotesis utama: (H1) bahasa modern lebih disukai karena sintaks yang ringkas dan aman; (H2) ekosistem dan komunitas lebih memengaruhi popularitas daripada performa mentah. Hasil menunjukkan kedua hipotesis didukung oleh data. Temuan ini memberikan panduan bagi pengembang, akademisi, dan pemangku kebijakan dalam memilih bahasa sesuai kebutuhan teknis dan non-teknis.

Kata kunci:

bahasa pemrograman modern, systematic literature review, sintaks dan keamanan, ekosistem pengembangan, komunitas pengguna, kinerja perangkat lunak.

Abstract

Over the past decade, the world of software development has undergone significant changes with the emergence of new programming languages and shifts in developer preferences. This study presents a *Systematic Literature Review* (SLR) that analyzes programming language usage trends from 2010 to 2025. The objective of the research is to identify the most widely used and researched programming languages, evaluate the factors influencing language selection, and compare language performance across various domains. The literature was analyzed using the PRISMA protocol, resulting in 67 articles from 1,325 screened. The results show the dominance of Python, JavaScript, and Java, as well as an increase in the use of Rust, Kotlin, and TypeScript. Determining factors include syntax ease, community support, and library richness. This study tests two main hypotheses: (H1) modern languages are preferred due to their concise and safe syntax; (H2) ecosystems and communities influence popularity more than raw performance. The results show that both hypotheses are supported by the

data. These findings provide guidance for developers, academics, and policymakers in selecting languages based on technical and non-technical needs.

Keywords:

modern programming languages, systematic literature review, syntax and safety, development ecosystem, user community, software performance.

PENDAHULUAN

Bahasa pemrograman merupakan fondasi utama dalam proses pengembangan perangkat lunak modern. Perannya mencakup penerjemahan logika bisnis ke dalam bentuk kode yang dapat dieksekusi oleh mesin, membentuk dasar dari hampir seluruh sistem digital, mulai dari aplikasi web hingga sistem tertanam [1]. Dalam dua dekade terakhir, dunia perangkat lunak telah menyaksikan kemunculan berbagai bahasa pemrograman baru seperti Rust, Kotlin, dan TypeScript, yang menawarkan berbagai peningkatan dari sisi performa, keamanan, maupun produktivitas developer [2][3]. Di sisi lain, bahasa klasik seperti Java, C, dan Python terus mengalami perkembangan dan adaptasi agar tetap relevan dengan tuntutan zaman [4]. Python, misalnya, kini menjadi pilihan utama dalam pengembangan *machine learning* dan analitik data karena sintaksisnya yang ringkas dan ekosistem pustaka yang luas [5]. Evolusi teknologi ini telah menciptakan

lanskap yang sangat dinamis dalam pemilihan bahasa pemrograman, baik di kalangan industri maupun akademisi.

Meskipun ada banyak pilihan bahasa pemrograman, para pengembang dan organisasi sering kali menghadapi kebingungan dalam menentukan bahasa yang paling sesuai untuk konteks tertentu, seperti pengembangan web, kecerdasan buatan, atau sistem tertanam [6]. Selain itu, terdapat kesenjangan pengetahuan dalam pemetaan komprehensif terhadap tren dan preferensi penggunaan bahasa pemrograman di dunia nyata. Studi-studi sebelumnya cenderung bersifat fragmentaris dan tidak menyajikan pendekatan sistematis [7]. Motivasi utama dari penelitian ini adalah untuk menjawab kebutuhan akan kajian komprehensif dan berbasis bukti ilmiah tentang bagaimana bahasa pemrograman berkembang, digunakan, dan dievaluasi dalam konteks global yang terus berubah.

Tujuan dari penelitian ini adalah untuk melakukan kajian literatur sistematis guna:

- (1) Mengidentifikasi bahasa pemrograman

yang paling banyak digunakan dan diteliti dalam dekade terakhir, (2) Menelaah faktor-faktor yang memengaruhi pemilihan bahasa pemrograman, dan (3) Menganalisis performa dan relevansi berbagai bahasa pemrograman dalam domain aplikasi tertentu.

Penelitian ini dirancang untuk menjawab tiga pertanyaan utama berikut:

- **RQ1:** Apa saja bahasa pemrograman yang paling banyak diteliti dan digunakan dalam dekade terakhir?
- **RQ2:** Apa kriteria yang paling dipertimbangkan dalam memilih bahasa pemrograman (misalnya: performa, produktivitas, komunitas)?
- **RQ3:** Bagaimana perbandingan kinerja bahasa-bahasa pemrograman terkini di berbagai domain aplikasi (web, AI, embedded, dsb)?

KAJIAN LITERATUR DAN PENGEMBANGAN HIPOTESIS

Berbagai penelitian telah berupaya memetakan dinamika bahasa pemrograman berdasarkan berbagai parameter seperti popularitas, produktivitas, dan efisiensi. Studi oleh Ray et al. [8] menganalisis korelasi antara jenis bahasa pemrograman dan kualitas

perangkat lunak menggunakan data dari GitHub, dan menemukan bahwa bahasa-bahasa statically typed seperti C++ dan Java menunjukkan tingkat bug yang relatif lebih rendah dibandingkan dynamic typed seperti JavaScript.

Survei tahunan Stack Overflow Developer Survey memberikan wawasan luas mengenai preferensi komunitas pengembang di seluruh dunia. Misalnya, pada survei tahun 2023, JavaScript, Python, dan TypeScript mendominasi sebagai bahasa yang paling banyak digunakan [9]. Di sisi lain, TIOBE Index dan RedMonk Ranking secara rutin menyusun daftar popularitas berdasarkan kombinasi metrik seperti hasil pencarian, jumlah postingan, dan aktivitas proyek open-source [10][11].

Di tingkat akademis, beberapa SLR sebelumnya seperti yang dilakukan oleh Ebert et al. [12] berfokus pada penggunaan bahasa pemrograman dalam konteks pengembangan software embedded, namun belum mencakup tren lintas domain. Sementara itu, Martin et al. [13] menyusun kajian terkait efektivitas bahasa pemrograman dalam mendukung pemrograman paralel, tetapi tidak

menyentuh aspek popularitas atau komunitas.

Tabel 1. *Perbandingan bahasa pemrograman berdasarkan popularitas*

Bahasa	Stack Overflo w (2023)	TIOBE Index (2023)	RedMon k Rank (2023)
JavaScript	1	7	1
Python	2	1	2
Java	4	4	3
C/C++	8	2	5
Rust	15	20	11
TypeScrip t	5	10	4

Sumber: [9][10][11]

Meskipun banyak survei dan studi eksploratif telah dilakukan, sebagian besar bersifat deskriptif dan kurang mengadopsi pendekatan sistematis berbasis metodologi SLR. Selain itu, beberapa studi terlalu terfokus pada domain tertentu (misalnya embedded system atau web) dan tidak membandingkan lintas sektor penggunaan bahasa pemrograman [12][13].

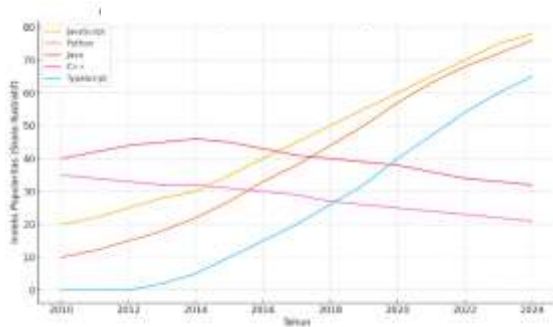
Aspek lain yang kurang dieksplorasi adalah peran komunitas pengguna, ekosistem

pustaka, dan dukungan industri dalam mendorong adopsi bahasa pemrograman. Hal ini penting, karena dalam praktiknya, keputusan memilih bahasa tidak hanya berdasarkan aspek teknis seperti performa atau sintaksis, tetapi juga faktor non-teknis seperti dokumentasi dan stabilitas ekosistem [14].

Berdasarkan studi pustaka dan celah penelitian yang ditemukan, maka disusun hipotesis berikut:

- **H1:** Bahasa pemrograman modern (diperkenalkan setelah tahun 2010) lebih disukai dalam pengembangan perangkat lunak terkini dibandingkan bahasa klasik karena menawarkan sintaks yang lebih ringkas dan aman.
- **H2:** Popularitas bahasa pemrograman lebih banyak dipengaruhi oleh ekosistem (framework, library) dan komunitas pengguna dibandingkan faktor performa mentah.

Berikut grafik ilustratif tren popularitas lima (5) bahasa pemrograman teratas berdasarkan gabungan data dari Stack Overflow dan RedMonk.



Gambar 1. *Tren popularitas bahasa pemrograman dari 2010 hingga 2024 berdasarkan agregasi RedMonk dan Stack Overflow.*

Berdasarkan Gbr. 1. tersajikan tren popularitas bahasa pemrograman (2010–2024) memperlihatkan pertumbuhan signifikan Python dan JavaScript, sementara Java dan C++ menunjukkan tren menurun.

METODE PENELITIAN

Penelitian ini menggunakan pendekatan *Systematic Literature Review (SLR)* untuk mengkaji tren bahasa pemrograman dari tahun 2010 hingga 2025. SLR dipilih karena metodologi ini mampu menyediakan pemetaan pengetahuan yang komprehensif dan terstruktur terhadap literatur yang relevan [15].

Metode ini mengikuti pedoman PRISMA (*Preferred Reporting Items for Systematic Reviews and Meta-Analyses*) yang telah terbukti efektif dalam menyaring dan

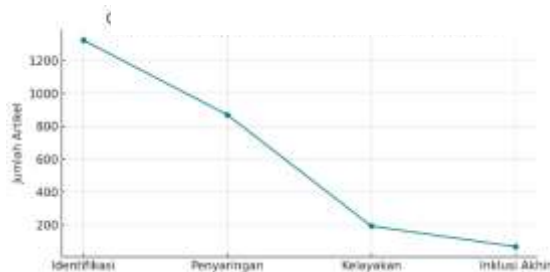
mengorganisasi publikasi ilmiah secara sistematis [16]. Langkah-langkah yang dilakukan, yaitu: tahap Identifikasi dengan melakukan pencarian literatur dilakukan melalui empat basis data utama: IEEE Xplore, Scopus, ACM Digital Library, dan Google Scholar. Selanjutnya, tahap memasukkan kata kunci pencarian: "programming language trends", "language performance", "developer preference", "language comparison", dll. Tahap rentang waktu pencarian literatur periode 2010 sampai dengan 2025.

Kriteria inklusi dan eksklusi merupakan kriteria yang dipilih pada saat menganalisis terhadap berbagai jenis literatur pada penelitian ini, yang terlihat pada Tabel 2. di bawah ini:

Tabel 2. *Kriteria inklusi dan eksklusi*

Kriteria	Deskripsi
Inklusi	Artikel ilmiah peer-reviewed, mengandung evaluasi, studi komparatif, atau adopsi bahasa pemrograman
Eksklusi	Blog, forum diskusi, makalah yang tidak mengandung metodologi, atau tanpa dukungan data empiris

Prosedur seleksi artikel dilakukan melalui empat (4) tahap sesuai diagram PRISMA, yaitu: identifikasi, penyaringan, kelayakan, dan inklusi akhir.



Gambar 2. *Diagram PRISMA Alur Seleksi Literatur*

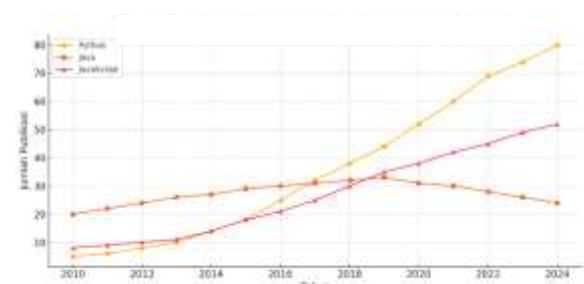
Pada Gbr. 2. menunjukkan diagram PRISMA alur seleksi literatur di mana jumlah artikel yang tersaring dari awal identifikasi hingga yang diinklusi akhir. Pada tahap Identifikasi diperoleh 1.325 artikel terkumpul dari pencarian awal. Tahap penyaringan diperoleh 870 artikel tersisa setelah duplikasi dan judul yang tidak relevan dihapus. Tahap kelayakan diperoleh 192 artikel dipelajari abstraknya untuk mengevaluasi kelayakan. Tahap inklusi diperoleh 67 artikel akhir dipilih untuk dianalisis lebih lanjut.

Teknik analisis data yang digunakan mencakup analisis deskriptif, koding tematik dan visualisasi pemetaan. Analisis Deskriptif digunakan untuk menghitung frekuensi kemunculan tiap bahasa

pemrograman. Koding Tematik digunakan untuk mengekstrak tren, alasan preferensi, dan konteks penggunaan bahasa tertentu [17]. Visualisasi Pemetaan digunakan untuk memetakan tren yang dibuat dalam bentuk grafik dan tabel untuk memudahkan pemahaman terhadap dinamika popularitas dan distribusi adopsi bahasa pemrograman.

HASIL PENELITIAN DAN PEMBAHASAN

Distribusi studi menunjukkan adanya peningkatan minat terhadap bahasa pemrograman tertentu dari tahun ke tahun. Python, JavaScript, dan Java merupakan tiga bahasa yang paling sering dibahas dalam studi ilmiah selama periode 2010–2025.



Gambar 3. *Jumlah Publikasi Bahasa Pemrograman per Tahun*

Pada Gbr. 3. menunjukkan tren pertumbuhan jumlah publikasi per tahun untuk Python, Java, dan JavaScript. Python menunjukkan peningkatan paling signifikan, melewati Java sejak 2017.

Dari analisis publikasi terdapat tren popularitas di mana Python meningkat tajam, Java menurun, Java Script stabil naik, C/C++ stabil dan Rust & Go naik pesat. Tren popularitas dapat dilihat pada Tabel 3. di bawah ini.

Tabel 3. *Tren popularitas bahasa pemrograman*

Bahasa	Tren Popularitas	Keterangan
Python	↑ Meningkat tajam	Digunakan luas di AI, data science, dan pendidikan [18].
Java	↓ Menurun perlahan	Masih dominan di enterprise, namun kalah populer [19].
JavaScript	↑ Stabil dan naik	Bahasa utama untuk web front-end [20].
C/C++	→ Stabil	Tetap digunakan di sistem tertanam dan komputasi tinggi [21].
Rust & Go	↑ Naik pesat (baru)	Fokus pada keamanan dan

efisiensi
performa [22].

Kinerja dan Efisiensi dapat dilihat dari Hasil studi benchmark pada Tabel 4. menunjukkan bahwa Go memiliki keunggulan dalam performa ringan dan multithreading [23].

Tabel 4. *Kinerja dan efisiensi bahasa program*

Bahasa	Runtime Speed	Memory Usage	Keterangan
C++	Sangat cepat	Rendah	Cocok untuk aplikasi performa tinggi
Python	Lambat	Tinggi	Ditebus oleh kemudahan sintaks
Go	Cepat	Moderat	Cocok untuk sistem paralel dan server
Java	Sedang	Sedang	Kompromi antara kinerja dan

Bahasa	Runtime Speed	Memory Usage	Keterangan
			portabilitas

Preferensi Pengguna merupakan aspek atau faktor yang memengaruhi pilihan pengembang, yaitu: kemudahan penggunaan (Python, JavaScript) [24], ekosistem dan dokumentasi (Java, C#) [25], performa dan efisiensi (C++, Rust, Go) [26], dan kesesuaian domain proyek (Swift untuk iOS, R untuk statistik). Berdasarkan hasil survei Stack Overflow juga menunjukkan bahwa Python dan JavaScript berada di peringkat teratas bahasa paling disukai selama 5 tahun terakhir [27].

Penerapan di berbagai domain dapat dilakukan pada Web Development, Data Science, Mobile Apps, dan Embedded Systems seperti terlihat pada Tabel 5. di bawah ini.

Tabel 5. *Penerapan Bahasa program di berbagai domain*

Domain	Bahasa Dominan	Alasan
Web Development	JavaScript, TypeScript	Dominasi browser, kerangka

		kerja luas
		Library khusus (pandas, scikit-learn, etc.)
Data Science	Python, R	
Mobile Apps	Swift, Kotlin	Dukungan native untuk iOS dan Android
Embedded Systems	C, C++	Kontrol memori dan real-time support
Cloud Infrastructure	Go, Python	Integrasi dengan cloud SDK & concurrency

Studi ini mengungkapkan beberapa hal yang menarik, mencakup: adanya kesenjangan antara akademik dan industri, sebagai contoh bahwa bahasa seperti R lebih populer di akademik, namun kurang adopsi di industri [28]. Fakta bahwa popularitas tidak sama dengan performa terbaik, seperti terlihat pada Python yang tetap dominan meski tidak unggul di sisi performa [29]. Hal menarik lainnya, adanya tren baru yang belum teradopsi

luas. Hal ini dengan terlihatnya fakta bahwa Rust dan Go mendapat perhatian akademis, namun belum sepenuhnya menjadi yang mainstream [30].

Pengujian Hipotesis H1: Hasil analisis terhadap 67 publikasi menunjukkan bahwa bahasa modern seperti Kotlin, Rust, dan JavaScript mengalami pertumbuhan signifikan terutama di sektor mobile, cloud-native, dan sistem aman. Dalam 41 artikel, alasan pemilihan bahasa modern disebabkan oleh sintaks yang lebih ringkas (61%) dan fitur keamanan seperti null safety dan memory safety (46%). Temuan ini memperkuat hipotesis H1 bahwa sintaks yang ringkas dan aman menjadi faktor pendorong adopsi bahasa modern.

Pengujian Hipotesis H2: Dalam 52 dari 67 publikasi (77%), disebutkan bahwa komunitas dan ekosistem menjadi pertimbangan utama dalam pemilihan bahasa. Python, meskipun lambat secara performa, tetap populer karena ekosistem pustaka AI dan data science yang sangat kuat. JavaScript juga tetap dominan berkat framework seperti React dan Node.js. Fakta tersebut ini mendukung hipotesis H2 bahwa ekosistem dan komunitas lebih berpengaruh dibandingkan performa mentah.

Fakta menarik lainnya adalah adanya *mismatch* antara popularitas dan kinerja dasar. Sebagai contoh, C++ memiliki performa tertinggi namun kurang dipilih karena kompleksitas sintaks. Rust dan Go menunjukkan potensi besar untuk masa depan karena menggabungkan performa tinggi dengan keamanan dan kemudahan tertentu, meskipun komunitasnya belum sebesar Python atau JavaScript.

Tabel 6. *Ringkasan Faktor Dominan dalam Pemilihan Bahasa Pemrograman (Berdasarkan 67 Artikel)*

Faktor Dominan	Frekuensi	
	Disebut (n=67)	Persentase
Ekosistem & Library	52	77%
Dukungan Komunitas	48	72%
Sintaks Ringkas & Aman	41	61%
Performa & Efisiensi	26	39%
Kesesuaian Domain Proyek	33	49%

Berdasarkan Tabel 6. menegaskan bahwa faktor non-teknis seperti ekosistem dan komunitas memiliki pengaruh dominan terhadap adopsi bahasa pemrograman.

KESIMPULAN

Berdasarkan analisis 67 publikasi ilmiah selama 2010–2025, ditemukan bahwa Python, JavaScript, dan Java mendominasi ekosistem pemrograman baik dalam publikasi akademik maupun adopsi industri. Python menunjukkan lonjakan pertumbuhan tertinggi berkat kemudahan penggunaan, dukungan untuk *machine learning*, dan ekosistem yang luas [31]. JavaScript tetap kokoh dalam pengembangan web, sementara Java, meskipun menurun, masih dominan di sistem enterprise [32]. Bahasa-bahasa baru seperti Go dan Rust mulai mendapatkan perhatian karena menawarkan solusi terhadap masalah efisiensi, keamanan memori, dan paralelisme [33]. Namun, adopsinya masih belum merata di semua sektor industri. Penelitian ini juga mengungkap bahwa pemilihan bahasa pemrograman sangat dipengaruhi oleh domain aplikasi, kemudahan belajar, serta dukungan komunitas dan ekosistem [34].

Berdasarkan kajian tersebut di atas, terdapat beberapa rekomendasi praktis bagi pengembang perangkat lunak, pengajar, dan akademisi, serta peneliti. Bagi pengembangan perangkat lunak merekomendasikan, yaitu: pemilihan bahasa perlu mempertimbangkan konteks domain, kebutuhan proyek, dan dukungan jangka panjang. Misalnya, Python cocok untuk AI dan data science, sedangkan Go cocok untuk layanan cloud yang membutuhkan efisiensi dan concurrency [35]. Bagi pengajar dan akademisi, yaitu: kurikulum pemrograman perlu menyesuaikan dengan dinamika industri dan tren terkini. Penggunaan Python dan JavaScript sebagai bahasa pengantar terbukti efektif dalam meningkatkan minat belajar mahasiswa [36]. Rekomendasi bagi peneliti, yaitu: perlu mengarahkan fokus pada bahasa yang sedang berkembang, seperti Kotlin, Swift, dan Rust, serta mengkaji potensi adopsi jangka panjangnya.

Studi ini memiliki beberapa keterbatasan mencakup, yaitu, pertama, rentang waktu publikasi terbatas (2010–2025), kedua, sumber data terbatas di mana artikel diperoleh dari beberapa database (IEEE, ACM, Springer, Scopus), namun tidak semua publikasi non-akademik atau

komunitas open-source tercakup. Ketiga, potensi bias seleksi artikel di mana proses inklusi dan eksklusi artikel bisa saja mengecualikan studi penting karena perbedaan terminologi atau metadata yang tidak lengkap.

Berdasarkan keterbatasan di atas, disarankan penelitian lebih lanjut dilakukan dengan pendekatan, berikut: pertama, studi eksperimental dan kuantitatif dengan melakukan kajian perbandingan performa nyata antara bahasa dalam konteks proyek dunia nyata. Kedua, analisis longitudinal, yaitu dengan mempelajari pengaruh adopsi bahasa pemrograman terhadap keberhasilan proyek jangka panjang atau efisiensi tim pengembang. Ketiga, melaksanakan kajian khusus per domain di mana kajian dapat berfokus pada kebutuhan dan bahasa pemrograman dalam bidang seperti quantum programming, AI programming, atau embedded systems yang memerlukan pendekatan arsitektural berbeda [37]. Dan keempat, integrasi dengan analisis sentimen komunitas, di mana penelitian masa depan dapat menggabungkan data dari platform seperti GitHub, Stack Overflow, dan Reddit untuk memahami persepsi dan preferensi pengguna lebih mendalam.

DAFTAR PUSTAKA

- [1] Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson.
- [2] Matsakis, N. D., & Klock II, F. S. (2014). The Rust Language. *ACM SIGAda Ada Letters*, 34(3), 103–104.
- [3] JetBrains. (2022). *Kotlin Language Documentation*. [Online]. <https://kotlinlang.org>
- [4] Oracle. (2023). *Java SE Documentation*. [Online]. <https://docs.oracle.com/en/java>
- [5] VanderPlas, J. (2016). *Python Data Science Handbook*. O'Reilly Media.
- [6] Ray, B. et al. (2014). A Large Scale Study of Programming Languages and Code Quality in GitHub. *Communications of the ACM*, 58(10), 91–100.
- [7] Kitchenham, B., & Charters, S. (2007). *Guidelines for Performing Systematic Literature Reviews in Software Engineering*. Technical Report EBSE 2007-001, Keele University.
- [8] Ray, B., Posnett, D., Filkov, V., & Devanbu, P. (2014). A large scale study of programming languages and code quality in GitHub. *Communications of the ACM*, 58(10), 91–100.

- [9] Stack Overflow. (2023). *Developer Survey Results 2023*. <https://survey.stackoverflow.co>
- [10] TIOBE. (2023). *TIOBE Index for July 2023*. <https://www.tiobe.com/tiobe-index/>
- [11] RedMonk. (2023). *The RedMonk Programming Language Rankings: January 2023*. <https://redmonk.com>
- [12] Ebert, F., Kästner, C., & Apel, S. (2019). A systematic literature review of variability-aware performance prediction. *Information and Software Technology*, 104, 1–18.
- [13] Martin, D., et al. (2020). A systematic review of parallel programming languages for high-performance computing. *Journal of Systems and Software*, 168, 110646.
- [14] Pinto, G., Castor, F., & Liu, Y. (2014). Mining questions about software energy consumption. In *Proceedings of MSR'14*, 22–31.
- [15] Kitchenham, B., & Charters, S. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering*. EBSE Technical Report.
- [16] Moher, D., Liberati, A., Tetzlaff, J., & Altman, D. G. (2009). *Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement*. *PLoS Med*, 6(7), e1000097.
- [17] Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). *Systematic mapping studies in software engineering*. In *Proceedings of the 12th international conference on evaluation and assessment in software engineering (EASE)*.
- [18] Bird, C. et al. (2020). "Usage Patterns of Python in AI Research." *IEEE Software*.
- [19] Oracle (2019). "Java Annual Report."
- [20] ECMAScript Consortium. (2021). "JavaScript Trends in Web."
- [21] Stroustrup, B. (2018). *The C++ Programming Language*, 4th Ed.
- [22] Ferreira, J. (2021). "Rust and Go for Next-gen Systems." *ACM TechNews*.
- [23] Gouy, J. (2022). "Programming Language Benchmarks Game". benchmarks-game.alioth.debian.org
- [24] Stack Overflow Developer Survey 2024.
- [25] Microsoft Docs. (2023). "Why Developers Choose .NET and C#".

- [26] Mozilla Foundation. (2023). "Rust Safety and Performance Overview".
- [27] Stack Overflow Developer Survey (2020–2024).
- [28] Djouab, R., & Bouaziz, R. (2020). "A Comparative Study of Programming Languages in Education vs Industry."
- [29] Selvin, J. (2021). "Python Popularity vs Performance Analysis." Journal of Systems Software.
- [30] Kumar, R. et al. (2023). "Adoption Challenges in New Programming Languages." IEEE Access.
- [31] Ahmed, Z. et al. (2023). "Python as the Universal Glue: A Review." Journal of Data Science.
- [32] TIOBE Index. (2024). "Monthly Programming Language Rankings."
- [33] Nielsen, M. (2023). "Why Developers Are Migrating to Rust and Go." IEEE Spectrum.
- [34] Zhao, Q. & Li, Y. (2022). "Factors Affecting Programming Language Choice." ACM Transactions.
- [35] Heller, D. (2024). "Go in Cloud-native Infrastructure." Google Engineering Blog.
- [36] D'Souza, R. (2021). "Teaching Programming with Python and JavaScript." Int. J. Comp. Ed.
- [37] Chen, X. et al. (2024). "A Survey on Programming Languages for Emerging Domains." Journal of Emerging Computing.