Basis Data: Jurnal Ilmiah Teknik Informatika
ISSN Cetak : 2614-722X ISSN Online : 2477-2673
Volume 03 Nomor 01, Maret 2026

KOMPARASI EFEKTIFITAS REKOMENDASI KODE GITHUB COPILOT VERSUS AMAZON CODEWHISPERER

[1]Dicky Pratama,[2]Figur Prasojo
[1,2]Universitas Muhammadiyah Madiun
[1]dp518@ummad.ac.id, [2]2555200009@ummad.ac.id

ABSTRACT

Advances in artificial intelligence (AI) technology have had a significant impact, particularly in the application development sector. GitHub Copilot and AWS Code Whisperer, which function as AI Code Generators, are examples of emerging solutions. This study aims to determine the effectiveness of using an AI code generator and the best results that can be generated between using GitHub Copilot and Amazon Code Whisperer. A systematic literature review was conducted by searching for and analyzing research journals related to the topic, applying inclusion and exclusion criteria. The study aims to identify differences in functionality, testing, and the influence of programming languages on the accuracy and validity of the code recommendations provided. It also aims to compare the effectiveness of code recommendations by GitHub Copilot and Amazon Code Whisperer based on previous research.

Keywords: GitHub Copilot, Amazon CodeWhisperer, AI Code Generator

ABSTRAK

Kemajuan teknologi kecerdasan buatan atau *artificial intelligence (AI)* memberikan dampak yang besar salah satunya pada sektor pengembangan aplikasi. *GitHub Copilot* dan *AWS Code Whisperer* yang memiliki fungsi sebagai *AI Code Generator* merupakan contoh yang muncul sebagai solusi. Studi ini bertujuan untuk mengetahui efektifitas dari pengunaan AI code generator juga hasil terbaik yang dapat dimunculkan antara penggunaan *GitHub Copilot dan Amazon Code Whisperer. Systematic literature review* dilakukan dengan mencari dan menganalisa jurnal penelitan yang berkaitan dengan topik dengan menerapkan kriteria inklusi dan eksklusi. Penelitian bertujuan untuk menemukan perbedaan fungsionalitas, pengujian dan pengaruh bahasa pemerograman terhadap akurasi dan validitas ketepatan rekomendasi kode yang diberikan dan untuk mengetahui pebandingan dari efektifitas kode yang direkomendasikan oleh *GitHub Copilot* dan *Amazon CodeWhisperer* berdasarkan penelitian yang sudah pernah dilakukan sebelumnya.

Kata Kunci: Kose *GitHub Copilot, Amazon CodeWhisperer, AI Code Generator*

Basis Data: Jurnal Ilmiah Teknik Informatika
ISSN Cetak : 2614-722X ISSN Online : 2477-2673
Volume 03 Nomor 01, Maret 2026

A. Pendahuluan

Kecerdasan buatan yang merupakan sebuah system dalam cabang bidang *computer science* yang berfokus dalam pembuatan sebuah mesin yang dapat melakukan tugas yang umumnya diperlukan kecerdasan manusia dalam proses eksekusinya saat ini mendapatkan perhatian yang sangat besar dalam komunitas penelitian ***[1]***. Perkembangan sistem kecerdasan buatan memberikan dampak pada berbagai bidang salah satunya pada proses pengembangan aplikasi. Tuntutan untuk membuat kode dengan waktu yang cepat melahirkan kecerdasan buatan berbasis *deep learning* yang berfungsi sebagai *code generator* ***[2]***. Kemudahan berkat bantuan *AI-Assistant* seperti GitHub Copilot dan Amazon CodeWhisperer memberikan dampak dalam mempermudah penulisan baris kode pada program yang telah ataupun sedang dibuat.

GitHub Copilot yang telah diumumkan sejak Juni 2021 merupakan sebuah AI Code generator buatan GitHub dengan kolaborasi bersama OpenAI yang dibuat secara berbayar ***[3]*** dan menjadi sebuah extension pada Visual Code Studio dan JetBrains IntelliJ IDE dengan menawarkan rekomendasi kode yang berbasis dari masalah yang dideskripsikan oleh pengembang aplikasi dikembangkan menggunakan codex berbasis model bahasa dan telah dilatih dengan sumber yang sangat besar ***[2], [4]***, ***[5]***. Amazon CodeWhisperer yang merupakan sebuah AI Code Generator yang dikembangkan oleh Amazon Web Service dan dirilis pada April 2023 merupakan sebuah AI Code Generator yang bersifat gratis untuk pengguna individual ***[6]***. Amazon CodeWhisperer dilatih dengan menggunakan milyaran baris kode dan mampu memberikan rekomendasi kode baik sebagai snippet atau secara keseluruhan juga diklaim dapat memberikan peningkatan terhadap keamanan kode ***[7]***.

Tujuan dari penelitian ini adalah mengetahui pebandingan dari efektifitas kode yang direkomendasikan oleh GitHub Copilot dan Amazon CodeWhisperer berdasarkan penelitian yang sudah pernah dilakukan sebelumnya.

B. Metode Penelitian

Metode penelitian yang digunakan dalam penyusunan tinjauan Pustaka ini dimulai dengan melakukan formulasi pertanyaan pertanyaan, yaitu:

Basis Data: Jurnal Ilmiah Teknik Informatika
ISSN Cetak : 2614-722X ISSN Online : 2477-2673
Volume 03 Nomor 01, Maret 2026

RQ 1 : Apa perbedaan antara GitHub Copilot dan AWS Code Whisperer dalam hal fitur, fungsionalitas, dan pendekatan yang digunakan dalam menghasilkan rekomendasi kode?

RQ 2 : Bagaimana metode evaluasi efektivitas rekomendasi kode pada GitHub Copilot dan AWS Code Whisperer telah dilakukan dalam penelitian sebelumnya?

RQ 3 : Bagaimana perbandingan langsung antara GitHub Copilot dan AWS Code Whisperer dalam hal efektivitas, akurasi, dan kebergunaan rekomendasi kode?

RQ 4 : Bagaimana pengaruh bahasa pemrograman yang digunakan terhadap efektivitas rekomendasi kode pada GitHub Copilot dan AWS Code Whisperer?

penelitian kemudian melakukan pencarian literatur berkaitan dengan menggunakan kata kunci yang merujuk kepada topik.

("GitHub Copilot" OR "Amazon CodeWhisperer" OR "AI Code Generator")

Pencarian dilakukan tanpa mempertimbangkan sinonim mungkin berkaitan melalui 3 library jurnal berbeda yaitu Scopus, ScienceDirect, dan IEEE Explore. kemudian dilakukan skrining pada hasil pengujian untuk mendapatkan hasil yang mendekati yang dilanjutkan dengan analisis secara manual tiap jurnal.

Berdasarkan kata kunci yang digunakan didapatkan hasil sebagai berikut:

a. SCOPUS
    a. GitHub Copilot : 74
    b. Amazon CodeWhisperer : 1
    c. AI Code Generation : 911 (sampled 100)
b. ScienceDirect
    a. GitHub Copilot : 74
    b. Amazon CodeWhisperer : 0
    c. AI Code Generation : 60479 (sampled 100)
c. IEEE Xplore
    a. GitHub Copilot : 30
    b. Amazon CodeWhisperer : 1

Basis Data: Jurnal Ilmiah Teknik Informatika
ISSN Cetak : 2614-722X ISSN Online : 2477-2673
Volume 03 Nomor 01, Maret 2026

c. AI Code Generation : 911 (sampled 100)

Dari dokumen kemudian dilakukan inklusi dan eksklusi dengan ketentuan seperti berikut:

Tabel 1

Kriteria Inklusi dan Eksklusi

Inklusi	Eksklusi
Dari rentang 2018-2024	Duplikasi file
Memiliki Test Case	
Menggunakan GitHub Copilot dan/atau Amazon Code Whisperer sebagai AI Code Generation	
Merupakan jurnal / conference paper	

Dari proses inklusi dan eksklusi yang kemudian dilanjutkan dengan skiring manual didapatkan hasil 23 paper yang memenuhi kriteria dengan tambahan 2 paper pre-print yang didapat dari ArVix.menjadi total 25.

C.Hasil Penelitian dan Pembahasan

Hasil analisis dan sintesis berfokus untuk mengetahui perbandingan efektifitas penggunaan GitHub Copilot dan Amazon CodeWhisperer.

A. RQ1 : Apa perbedaan antara GitHub Copilot dan AWS Code Whisperer dalam hal fitur, fungsionalitas, dan pendekatan yang digunakan dalam menghasilkan rekomendasi kode?

GitHub Copilot menggunakan model pendekatan *auto-complete* untuk memberikan rekomendasi kode dengan cara menuliskan kode program yang ingin digunakan ataupun menggunakan *comment* dengan *Natural Language Processing (NLP)* ***[8]***. Amazon CodeWhisperer melakukan pendekatan dengan menganalisa *comment* dengan *Natural Language Processing (NLP)* yang dipadukan dengan baris kode yang sudah ada untuk memberikan rekomendasi kode. *Natural Language Processing* adalah sebuah media pemrosesan kecerdasan buatan dan

Basis Data: Jurnal Ilmiah Teknik Informatika
ISSN Cetak : 2614-722X ISSN Online : 2477-2673
Volume 03 Nomor 01, Maret 2026

linguistic yang ditujukan untuk membuat computer mampu memahami perintah ataupun kata yang ditulis dalam bahasa manusia ***[9]***. Amazon CodeWhisperer juga bisa digunakan untuk melakukan pemindaian secara keseluruhan untuk menemukan dan mendefinisikan permasalahan keamanan yang ada pada baris kode yang ada ***[10]***. GitHub Copilot dan Amazon CodeWhisperer mendukung banyak Bahasa pemerograman seperti Python, Java, JavaScript, C# dan banyak bahasa pemerograman lain ***[3], [7]***, namun Amazon CodeWhisperer mendukung lebih banyak IDE platform dari GitHub Copilot dimana Amazon CodeWhisperer mendukung VSCode, IntelliJ IDEA, PyCharm, AWS Cloud9, AWS Lambda Console ***[7]*** sementara GitHub Copilot mendukung Neovim, JetBrains IDE, Visual Studio, dan VS Code ***[3]***.

B. RQ2 : Bagaimana metode evaluasi efektivitas rekomendasi kode pada GitHub Copilot dan AWS Code Whisperer telah dilakukan dalam penelitian sebelumnya?

Proses pengujian dilakukan dengan menguji penggunaan dengan berbagai metode seperti *pair programming* antara AI dan manusia ***[4]*** untuk menemukan baris kode paling sedikit dan efektif, menggunakan SonarQube ***[5], [7]*** untuk menemukan *security rating*, jumlah bug, *jumlah code smell,* juga memunculkan matriks seperti *security reliability, maintainability* dan *correctness*. SonarQube adalah sebuah perangkat *open-source* yang digunakan untuk melakukan peninjauan terhadap kode yang ditulis untuk memberikan kode yang lebih bersih; kode yang dinilai aman, modular, dapat dengan mudah dibaca dan dikelola ***[11]***. juga pengamatan secara manual ***[6]*** untuk melihat hasil dari masalah yang diberikan apakah terpecahkan sesuai kebutuhan atau tidak.

Pada proses evaluasi menggunakan SonarQube, penulis menggunakan jenis dataset berupa Human Eval Dataset ***[12]*** dan pada pengujian lain menggunakan case yang didapat dari LeetCode. Pada pengujian *pair programming* antara AI dan manusia, proses dilakukan dengan menggunakan ndiff function dari difflib ***[13]***. Pengujian dilakukan dengan cara melakukan komparasi dari baris yang diberikan dengan banyak baris yang dihapus setelah dilakukan normalisasi pada proses percobaan. Percobaan lain yang dilakukan adalah menguji keberhasilan kode yang diberikan untuk melihat apakah memenuhi kriteria masalah yang diambil dari

Basis Data: Jurnal Ilmiah Teknik Informatika
ISSN Cetak : 2614-722X ISSN Online : 2477-2673
Volume 03 Nomor 01, Maret 2026

kerangka permasalah *Computer Science* 1 ***[14]***. Evaluasi lain secara manual ditunjukkan

Metode evaluasi lain yang digunakan adalah menggunakan CodeQL untuk menilai keamanan dari rekomendasi kode yang diberikan. CodeQL adalah sebuah alat analisis berbasis *open-source* yang digunakan untuk menganalisa keamanan kode, bugs dan jenis error lain dengan cara membuat query dan database dari kode yang ada ***[15]***.

C. RQ3 : Bagaimana perbandingan langsung antara GitHub Copilot dan Amazon Code Whisperer dalam hal efektivitas, akurasi, dan kebergunaan rekomendasi kode?

Pengujian dengan menggunakan SonarQube pada dataset LeetCode menunjukkan dari 33 pertanyaan yang diberikan memunculkan hasil sebesar 70% solusi yang diberikan oleh GitHub Copilot dapat diterima dan tidak melebihi batas waktu eksekusi yang diberikan untuk masing-masing bahasa pemerograman dengan kemungkinan error yang kecil sebesar 24% pada bahasa pemerograman C, sedangkan 0% error ditemukan pada bahasa pemerograman Java ***[16]***. Pengujian lain menggunakan SonarQube dengan Human Eval Dataset didapat hasil GitHub Copilot mampu memberikan baris kode yang dinilai valid sebesar 150 dari 164 masalah yang diberikan, sedangkan Amazon CodeWhisperer memberikan hasil valid sebesar 148 dari kesuluruhan 164 masalah ***[12]***.

Dari pengujian *pair-programming* dengan menggunakan Ndiff Function dari Diffub, dilakukan percobaan dengan 21 partisipan untuk mengembangkan permainan minesweeper berbasis teks dengan bahasa pemerograman Python. Pengujian dilakukan dengan 3 kondisi; *pair-programming* dengan Copilot, *pair-programming* dengan programmer lain sebagai driver, dan *pair-programming* dengan manusia lain sebagai navigator. Driver adalah orang yang melakukan ekseskusi dan menulis kode sedangkan navigator adalah orang yang mengamati kode yang dituliskan driver dan memberikan koreksi dan masukan kepada driver. Dari pengujian ini didapat hasil bahwa GitHub Copilot mendemonstrasikan produktivitas lebih tinggi daripada *humain pair-programming* yang bertindak sebagai driver dan navigator. Baris kode yang dibuat oleh GitHub Copilot dinilai memiliki kualitas lebih rendah ditunjukkan dengan lebih banyaknya baris yang

Basis Data: Jurnal Ilmiah Teknik Informatika
ISSN Cetak : 2614-722X ISSN Online : 2477-2673
Volume 03 Nomor 01, Maret 2026

dihapus setelah proses normalisasi ***[13]***. Pengujian serupa juga dilakukan oleh Madi ***[17]*** dimana penelitian di fokuskan pada *readability* dan hasil inspeksi secara visual dan didapatkan hasil yang sama bahwa *human pair-programming* memiliki hasil yang lebih baik baik sebagai driver maupun navigator namun programmer manusia kurang memperhatikan baris kode yang ditulis Ketika dilakukan pengamatan dengan mengimplementasikan teknologi *eye-tracking* untuk mengamati masing-masing programmer.

Pengujian dengan permasalah *Computer Science* 1 dengan CodeCheck dataset dan bahasa pemerograman Python memberikan hasil sebesar 47,6% permasalahan dapat diselesaikan dalam percobaan pertama dari keseluruhan 166 masalah. Sebesar 87 masalah tidak terselesaikan dan setelah dilakukan perubahan pada deskripsi masalah didapat 53 masalah atau 60,9% terselesaikan ***[14]***. Pengujian ini dievaluasi dengan cara manual untuk mengamati hasil yang bisa diterima. Pengujian lain yang menggunakan metode evaluasi manual adalah penelitian yang dilkaukan oleh asare, et al ***[18]*** untuk melihat kerentanan baris kode yang direkomendasikan. Pengujian dilakukan dengan menggunakan Big-Vul dataset dengan 3332 *commit* yang kemudian diamati dan dievaluasi secara manual oleh 3 orang yang berbeda untuk melihat apakah setiap hasil memunculkan tingkat kerentanan yang sama.

Penelitian lain untuk melihat tingkat keamanan dari kode yang direkomendasikan oleh GitHub Copilot menggunakan kerangka kasus yang dibuat berdasarkan 25 *Common Weakness Enunumeration (CWE)* yang kemudian hasilnya di identifikasi menggunakan CodeQL dan pengecekan manual menunjukkan bahwa dari keseluruhan kasus, terdapat 39.33% dari kasus yang dianggap tinggi dan 40.73% dari kesuluruhan uji coba menggunakan GitHub Copilot dinilai tidak aman sehingga programmer diharuskan lebih waspada dan meneliti kembali hasil yang direkomendasikan oleh GitHub Copilot ***[19]***.

Pengujian untuk menilai kualitas kode yang direkomendasikan oleh GitHub Copilot dilakukan oleh Wong, et al ***[20]*** dengan cara menguji 6 masalah yang dibuat secara acak dengan spesifikasi formal yang dibuat di Dafny, GitHub Copilot diminta memecahkan masalah dengan bahasa pemerograman Python yang kemudian hasilnya diterjemahkan secara manual dan meneliti hasil yang didapat

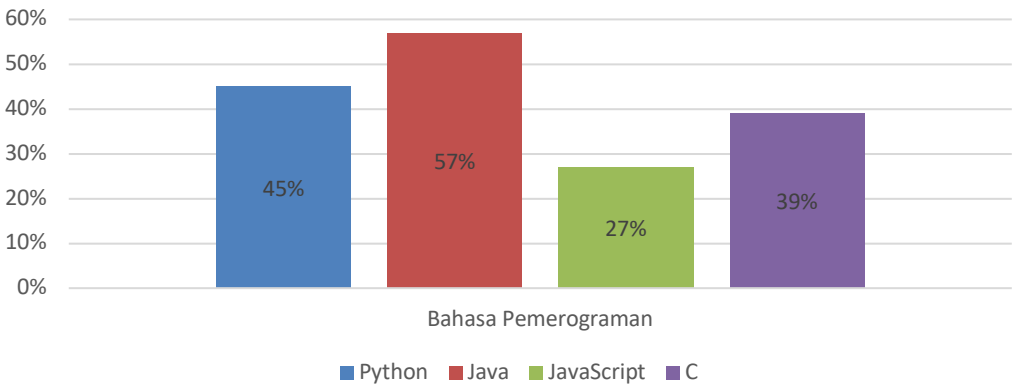
menggunakan Dafny. Dafny adalah sebuah alat yang berfungsi sebagai media verifikasi fungsionalitas dan validitas dari baris kode ***[21]***. Proses pengujian memberikan hasil dimana 4 dari 6 permasalahan bisa terpecahkan dengan kondisi dimana GitHub Copilot bisa memberikan performa yang lebih baik hanya pada masalah yang bersifat umum sehingga hasil rekomendasi harus ditinjau ulang lebih jauh untuk menghindari kesalahan. Penelitian serupa untuk menilai kualitas kode dengan permasalahan pemerograman untuk level pemula dengan bahasa pemerograman Python yang dilakukan oleh Dakhel, et al ***[22]*** memberikan hasil bahwa GitHub Copilot dapat merekomendasikan kode untuk hamper seluruh permasalahan fundamental namun pada perbandingan hasil antara programmer manusia dan AI didapat bahwa programmer manusia masih memberikan hasil yang lebih baik.

Pada pengujian *robustness* kode yang dihasilkan oleh GitHub Copilot ***[23]***, penulis menguji 892 method dalam bahasa Java untuk melihat hasilnya yang di verifikasi menggunakan CodeBLEU. CodeBLEU adalah sebuah metode untuk mengukur tingkat performa dari baris kode yang ditulis dengan cara memeriksa kode yang dituliskan dan struktur dari penulisan kodenya ***[24]***. Dari hasil penelitian didapat bahwa dari keseluruhan kasus yang dijalankan, setelah deskripsi pada *comment* yang digunakan diubah secara semantik, didapatkan 46% dari keseluruhan method berubah dengan 28% hasil valid yang dapat diterima.

D. RQ4 : Bagaimana pengaruh bahasa pemrograman yang digunakan terhadap efektivitas rekomendasi kode pada GitHub Copilot dan AWS Code Whisperer?

GitHub Copilot dan Amazon CodeWhisperer masing-masing mendukung banyak bahasa pemerograman yang dapat diselesaikan. Dari jurnal yang ditemukan, proses pengujian dilakukan dengan menggunakan bahasa pemerograman python ***[13]***, ***[16]***, ***[14]***, ***[12], [17], [20], [22]***, Java ***[12], [18], [23]***, JavaScript, dan C ***[12]***. Bisa dilihat pada grafik 1, dari hasil penelitian didapat rata-rata Python sebesar 45%, JavaScript sebesar 27%, C sebesar 39% dan dengan hasil *correctness* tertinggi adalah Java sebesar 57%.

Grafik 1
Perbandingan hasil pengujian berdasarkan bahasa pemerograman



D. Kesimpulan

Sebagai kesimpulan, Systematic literature review ini bertujuan untuk menkomparasi afektifitas dari AI Code Generator yaitu GitHub Copilot dan Amazon CodeWhisperer. Analisis dari berbagai paper menghasilkan beberapa temuan. Pertama pada fitur, fungsionalitas dan pendekatan, GitHub Copilot menggunakan model *auto-complete* dan *Natural Language Processing (NLP),* sedangkan Amazon CodeWhisperer menggunakan *Natural Language Processing (NLP)* untuk menganalisa *comment* yang digunakan untuk memberikan rekomendasi kode. GitHub Copilot dan Amazon CodeWhisperer mendukung berbagai bahasa pemerograman yang berbeda, namun CodeWhisperer memiliki dukungan *platform* IDE yang lebih banyak dibanding dengan GitHub Copilot.

Terdapat berbagai metode evaluasi yang digunakan. SonarQube untuk menilai tingkat keamanan, matriks kualitas kode dan aspek lain. CodeQL dan pengamatan visual untuk melihat akurasi, kelemahan keamanan dan keterbacaan kode, juga pengujian secara manual. Hasil pengujian yang didapat menunjukkan bahwa GitHub Copilot memiliki tingkat validitas lebih tinggi dimana pada pengujian dengan menggunakan SonarQube pada LeetCode dataset menunjukkan 70% hasil dengan minimal error. Pada evaluasi *pair-programming* didapat bahwa GitHub Copilot menunjukkan bahwa GitHub copilot menghasilkan produktifitas lebih tinggi dari programmer manusia. Hasil pengujian pada permasalahan Computer Science 1 menunjukkan hasil yang beragam sehingga diperlukan evaluasi lebih mendalam pada hasil rekomendasi kode yang diberikan.

Basis Data: Jurnal Ilmiah Teknik Informatika
ISSN Cetak : 2614-722X ISSN Online : 2477-2673
Volume 03 Nomor 01, Maret 2026

Keterbatasan paper pada topik Amazon CodeWhisperer menjadikan perlu adanya investigasi lebih lanjut untuk mendapatkan hasil perbandingan yang lebih komprehensif. Penelitian lanjutan bisa dilakukan dengan pengembangan pada metode evaluasi, dataset, dan dampak penggunaan bahasa pemerograman yang berbeda dimana penelitian ini bisa berkontribusi untuk mendapatkan hasil pengujian dan efektifitas yang lebih baik dalam proses pengembangan sistem AI Code Generator.

DAFTAR PUSTAKA

[1] C. Collins, D. Dennehy, K. Conboy, and P. Mikalef, “Artificial intelligence in information systems research: A systematic literature review and research agenda,” *International Journal of Information Management*, vol. 60, p. 102383, Oct. 2021, doi: 10.1016/j.ijinfomgt.2021.102383.

[1] Jaworski, M. and Piotrkowski, D. (2023) 'Study of software developers' experience using the Github Copilot Tool in the software development proces.'

[3] “GitHub Copilot · Your AI pair programmer,” *GitHub*. https://github.com/features/copilot (accessed Jul. 13, 2023).

[3] Sobania, D., Briesch, M. and Rothlauf, M. (2021) 'Choose Your Programming Copilot A Comparison of the Program Synthesis Performance of GitHub Copilot and Genetic Programming.'

[4] Torrres Ovalle, B.S. (2022) ‘GitHub copilot’, *Encuentro Internacional de Educación en Ingeniería* [Preprint]. doi:10.26507/paper.2300.

[6] “Amazon CodeWhisperer is now generally available,” *Amazon Web Services, Inc.* https://aws.amazon.com/about-aws/whats-new/2023/04/amazon-codewhisperer-generally-available/ (accessed Jul. 12, 2023).

[7] “AI Code Generator - Amazon CodeWhisperer - AWS,” *Amazon Web Services, Inc.* https://aws.amazon.com/codewhisperer/ (accessed Jul. 12, 2023).

[8] "About GitHub Copilot for Individuals," *GitHub Docs*. https://ghdocs-prod.azurewebsites.net/en/copilot/overview-of-github-copilot/about-github-copilot-for-individuals (accessed Jul. 07, 2023).

[9] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: state of the art, current trends and challenges," *Multimed Tools Appl*, vol. 82, no. 3, pp. 3713–3744, Jan. 2023, doi: 10.1007/s11042-022-13428-4.

[10] "Amazon CodeWhisperer Documentation." https://docs.aws.amazon.com/codewhisperer/index.html (accessed Jul. 07, 2023).

[11] "SonarQube 10.1." https://docs.sonarsource.com/sonarqube/latest/?_gl=1*1qcf8ss*_gcl_au*MTMwNTUzODgxMC4xNjg5MjE5MzEx*_ga*ODMwOTM2NDUwLjE2ODkyMTkzMTE.*_ga_9JZ0GZ5TC6*MTY4OTIzNDcyNi4yLjAuMTY4OTIzNDcyNi42MC4wLjA. (accessed Jul. 13, 2023).

[10] Yetistiren, B., Ozson I., Ayerdem, M. and Tuzun, E. (2023). 'Evaluating the Code Quality of AI Code Generation tools: An Empirical Study on GitHub Copilot, Amazon CodeWhisperer, and ChatGPT' [pre print]

[11] Imai, S. (2022) 'Is github copilot a substitute for human pair-programming?', *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings* [Preprint]. doi:10.1145/3510454.3522684.

[12] Denny, P., Kumar, V. and Giacaman, N. (2023) 'Conversing with copilot: Exploring prompt engineering for solving CS1 problems using natural language', Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 [Preprint]. doi:10.1145/3545945.3569823.

[15] "About CodeQL — CodeQL." https://codeql.github.com/docs/codeql-overview/about-codeql/ (accessed Jul. 14, 2023).

[14] Nguyen, N. and Nadi, S. (2022) 'An empirical evaluation of github copilot's code suggestions', Proceedings of the 19th International Conference on Mining Software Repositories [Preprint]. doi:10.1145/3524842.3528470.

[17] N. Al Madi, "How Readable is Model-generated Code? Examining Readability and Visual Inspection of GitHub Copilot," in *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, Rochester MI USA: ACM, Oct. 2022, pp. 1–5. doi: 10.1145/3551349.3560438.

[18] O. Asare, M. Nagappan, and N. Asokan, "Is GitHub's Copilot as Bad as Humans at Introducing Vulnerabilities in Code?" arXiv, Jun. 05, 2023. Accessed: Jul. 15, 2023. [Online]. Available: http://arxiv.org/abs/2204.04741

[16] Pearce, H. *et al.* (2022) 'Asleep at the keyboard? assessing the security of github copilot's code contributions', *2022 IEEE Symposium on Security and Privacy (SP)* [Preprint]. doi:10.1109/sp46214.2022.9833571.

[20] D. Wong, A. Kothig, and P. Lam, "Exploring the Verifiability of Code Generated by GitHub Copilot." arXiv, Oct. 27, 2022. Accessed: Jul. 14, 2023. [Online]. Available: http://arxiv.org/abs/2209.01766

[21] K. R. M. Leino, "Dafny: An Automatic Program Verifier for Functional Correctness," in *Logic for Programming, Artificial Intelligence, and Reasoning*, E. M. Clarke and A. Voronkov, Eds., in Lecture Notes in Computer Science, vol. 6355. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 348–370. doi: 10.1007/978-3-642-17511-4_20.

[22] A. Moradi Dakhel, V. Majdinasab, A. Nikanjam, F. Khomh, M. C. Desmarais, and Z. M. (Jack) Jiang, "GitHub Copilot AI pair programmer: Asset or Liability?," *Journal of Systems and Software*, vol. 203, p. 111734, Sep. 2023, doi: 10.1016/j.jss.2023.111734.

[23] A. Mastropaolo *et al.*, "On the Robustness of Code Generation Techniques: An Empirical Study on GitHub Copilot." arXiv, Feb. 01, 2023. Accessed: Jul. 14, 2023. [Online]. Available: http://arxiv.org/abs/2302.00438

[24] S. Ren *et al.*, "CodeBLEU: a Method for Automatic Evaluation of Code Synthesis." arXiv, Sep. 27, 2020. Accessed: Jul. 14, 2023. [Online]. Available: http://arxiv.org/abs/2009.10297