

GENERATIVE AI SEBAGAI PENDAMPING PEMROGRAMAN: PELUANG, KETERBATASAN, DAN TANTANGAN ETIKA

Aditya Pratama Rahawarin
adityarahawarin5@gmail.com
Universitas Pamulang

ABSTRAK

Generative Artificial Intelligence (Generative AI) semakin berkembang sebagai teknologi pendamping dalam pengembangan perangkat lunak dan pembelajaran pemrograman. Teknologi berbasis Large Language Model (LLM) dapat membantu menghasilkan kode program, menjelaskan konsep pemrograman, memberikan saran debugging, membuat contoh pengujian, serta membantu menyelesaikan pekerjaan pemrograman yang bersifat berulang. Artikel non-penelitian ini bertujuan membahas peluang, keterbatasan, dan tantangan etika penggunaan Generative AI sebagai pendamping pemrograman. Pembahasan dilakukan melalui kajian literatur terhadap berbagai publikasi ilmiah mengenai Large Language Model, pemrograman berbantuan AI, rekayasa perangkat lunak, kualitas kode, keamanan, serta etika penggunaan kecerdasan buatan. Hasil kajian menunjukkan bahwa Generative AI berpotensi mempercepat pekerjaan pemrograman rutin dan mendukung proses pembelajaran melalui penjelasan serta contoh yang interaktif. Namun, kode yang dihasilkan AI tidak selalu benar, mudah dipelihara, atau aman sehingga tetap membutuhkan pemeriksaan dan pengujian manusia. Tantangan etika meliputi integritas akademik, kepengarangan, hak kekayaan intelektual, privasi, transparansi penggunaan AI, dan tanggung jawab terhadap hasil kode. Oleh karena itu, Generative AI sebaiknya diposisikan sebagai alat bantu, bukan pengganti kompetensi pemrograman. Penggunaan yang bertanggung jawab membutuhkan validasi manusia, pengujian, pemeriksaan keamanan, keterbukaan penggunaan AI sesuai kebutuhan, serta penguatan kemampuan dasar pemrograman.

Kata Kunci: Generative AI, Pemrograman, Large Language Model, Rekayasa Perangkat Lunak, Etika Teknologi.

ABSTRACT

Generative Artificial Intelligence (Generative AI) has increasingly become a companion technology in software development and programming education. Technologies based on Large Language Models (LLMs) can generate source code, explain programming concepts, suggest debugging strategies, create test cases, and assist developers in completing repetitive programming tasks. This conceptual article aims to discuss the opportunities, limitations, and ethical challenges of using Generative AI as a programming companion. The discussion uses a literature review approach by examining scientific publications concerning large language models, AI-assisted programming, software engineering, code quality, security, and the ethical use of artificial intelligence. The literature indicates that Generative AI can accelerate routine programming activities and support learning through interactive explanations and examples. However, AI-generated code is not always accurate, maintainable, or secure and therefore requires human verification and testing. Ethical challenges include academic integrity, authorship, intellectual property, privacy, transparency, and accountability for generated code. Therefore, Generative AI should be positioned as an assistive technology rather than a replacement for programming competence. Responsible use requires human review, software testing, security checks, appropriate disclosure, and continuous development of fundamental programming skills.

Keywords: Generative AI, Programming, Large Language Model, Software Engineering, Technology Ethics.

PENDAHULUAN

Perkembangan teknologi kecerdasan buatan telah membawa perubahan terhadap berbagai aktivitas manusia, termasuk dalam bidang teknologi informasi dan pengembangan perangkat lunak. Salah satu perkembangan yang semakin banyak digunakan adalah Generative Artificial Intelligence atau Generative AI. Teknologi ini mampu menghasilkan konten baru berdasarkan instruksi yang diberikan pengguna, termasuk teks, gambar, dan kode program. Dalam bidang pemrograman, perkembangan Large Language Model (LLM) membuat komputer mampu membantu pengguna memahami, membuat, mengubah, dan menjelaskan kode melalui interaksi menggunakan bahasa alami. Kajian Hou et al. (2024) terhadap 395 artikel menunjukkan bahwa LLM telah digunakan dalam berbagai aktivitas software engineering dan menjadi salah satu perkembangan penting dalam bidang rekayasa perangkat lunak.

Kemampuan tersebut menyebabkan Generative AI mulai digunakan sebagai pendamping programmer. Seorang programmer dapat memberikan deskripsi kebutuhan kepada sistem AI kemudian memperoleh contoh kode yang sesuai. Selain menghasilkan kode, AI juga dapat membantu menjelaskan sintaks, mencari kemungkinan penyebab kesalahan, memberikan alternatif algoritma, membuat dokumentasi, dan menghasilkan contoh pengujian. Kondisi tersebut dapat membantu mempercepat pekerjaan tertentu yang sebelumnya membutuhkan waktu lebih lama.

Dalam bidang pendidikan, Generative AI juga dapat dimanfaatkan sebagai pendamping mahasiswa dalam mempelajari pemrograman. Mahasiswa dapat meminta penjelasan mengenai konsep algoritma, struktur data, bahasa pemrograman, maupun penyebab suatu program mengalami error. Interaksi tersebut memberikan kesempatan bagi mahasiswa untuk memperoleh penjelasan secara lebih fleksibel. Namun, penggunaan AI dalam pendidikan juga membutuhkan perhatian terhadap proses belajar dan integritas akademik. UNESCO menekankan pentingnya pendekatan yang berpusat pada manusia, perlindungan privasi, penggunaan yang aman, serta kebijakan yang sesuai dalam pemanfaatan Generative AI di pendidikan dan penelitian.

Meskipun memberikan berbagai peluang, Generative AI tidak selalu menghasilkan kode yang benar. Sistem AI dapat menghasilkan kode yang memiliki kesalahan sintaks, kesalahan logika, ketidaksesuaian dengan kebutuhan, atau kelemahan keamanan. Penelitian mengenai keamanan kode yang dihasilkan LLM menunjukkan adanya berbagai jenis kerentanan yang perlu diperhatikan dalam penggunaan AI untuk pemrograman.

Selain masalah teknis, terdapat pula persoalan etika. Penggunaan kode yang dihasilkan AI dapat menimbulkan pertanyaan mengenai kepemilikan, hak cipta, lisensi, privasi kode sumber, transparansi, serta tanggung jawab pengguna. Dalam konteks pendidikan, penggunaan AI secara berlebihan juga dapat mengurangi proses berpikir dan latihan pemecahan masalah apabila mahasiswa hanya menyalin hasil yang diberikan AI.

Berdasarkan kondisi tersebut, Generative AI perlu dipahami sebagai alat pendukung pemrograman yang membutuhkan keterlibatan manusia. Artikel ini membahas tiga aspek utama, yaitu peluang pemanfaatan Generative AI dalam pemrograman, keterbatasan yang perlu diperhatikan, serta tantangan etika yang muncul dalam penggunaannya. Pembahasan diharapkan dapat memberikan pemahaman mengenai penggunaan Generative AI yang produktif sekaligus bertanggung jawab.

HASIL DAN PEMBAHASAN

Generative AI dalam Aktivitas Pemrograman

Generative AI dalam pemrograman merupakan teknologi yang dapat menghasilkan atau memodifikasi kode berdasarkan instruksi pengguna. Teknologi tersebut umumnya memanfaatkan model bahasa yang telah dilatih menggunakan data dalam jumlah besar sehingga mampu memahami pola bahasa pemrograman dan bahasa manusia. Dalam praktiknya, pengguna dapat memberikan perintah dalam bentuk bahasa alami seperti meminta pembuatan fungsi, menjelaskan kode, memperbaiki error, atau membuat contoh implementasi.

Penggunaan LLM dalam software engineering telah berkembang pada berbagai aktivitas. Hou et al. (2024) mengidentifikasi berbagai penggunaan LLM dalam software engineering, termasuk aktivitas yang berhubungan dengan pembuatan kode, pengujian, dokumentasi, pemeliharaan, dan berbagai tahapan pengembangan perangkat lunak.

Konsep Generative AI sebagai “pendamping” menjadi penting karena AI tidak seharusnya dipandang sebagai pengganti programmer. AI dapat memberikan rekomendasi atau menghasilkan kode, tetapi programmer tetap perlu memahami kebutuhan sistem, menentukan pendekatan teknis, memeriksa kode, melakukan pengujian, dan bertanggung jawab terhadap hasil akhir.

Dengan demikian, penggunaan AI dalam pemrograman dapat digambarkan sebagai hubungan antara manusia dan sistem AI. Manusia memberikan tujuan, konteks, dan batasan, kemudian AI memberikan alternatif solusi. Setelah itu, manusia melakukan evaluasi dan menentukan apakah solusi tersebut dapat digunakan.

Peluang Generative AI dalam Pemrograman

Salah satu peluang utama Generative AI adalah meningkatkan efisiensi dalam menyelesaikan pekerjaan pemrograman yang bersifat rutin. Programmer dapat menggunakan AI untuk membuat struktur kode awal, fungsi sederhana, komentar kode, dokumentasi, maupun contoh pengujian. Pekerjaan tersebut dapat menjadi lebih cepat sehingga programmer dapat memusatkan perhatian pada permasalahan yang lebih kompleks.

Peluang berikutnya adalah membantu proses pembelajaran. Mahasiswa yang mengalami kesulitan memahami konsep tertentu dapat meminta AI memberikan penjelasan dengan bahasa yang lebih sederhana. Misalnya, mahasiswa dapat meminta penjelasan mengenai perbedaan antara perulangan `for` dan `while`, cara kerja fungsi, konsep object-oriented programming, atau proses debugging. Dengan demikian, AI dapat menjadi sumber pendamping belajar yang bersifat interaktif.

Generative AI juga dapat digunakan untuk membantu proses debugging. Ketika program menghasilkan pesan kesalahan, programmer dapat memberikan potongan kode dan informasi error kepada AI untuk memperoleh beberapa kemungkinan penyebab. Namun, saran tersebut tetap harus diverifikasi melalui proses pengujian karena AI tidak selalu mengetahui keseluruhan konteks aplikasi.

Selain itu, AI dapat membantu proses dokumentasi perangkat lunak. Dokumentasi sering kali membutuhkan waktu karena programmer harus menjelaskan fungsi dan struktur program. AI dapat membantu menghasilkan draft dokumentasi berdasarkan kode yang diberikan. Programmer kemudian dapat memeriksa dan menyesuaikan dokumentasi tersebut agar sesuai dengan sistem yang sebenarnya.

Peluang lainnya adalah membantu eksplorasi berbagai alternatif solusi. Programmer dapat meminta AI membandingkan beberapa algoritma atau pendekatan implementasi. Hasil perbandingan tersebut dapat digunakan sebagai bahan pertimbangan sebelum programmer menentukan solusi yang sesuai dengan kebutuhan sistem.

Keterbatasan Generative AI

Walaupun memiliki kemampuan yang semakin baik, Generative AI tetap memiliki keterbatasan. Salah satu keterbatasan utama adalah kemungkinan menghasilkan informasi atau kode yang tidak benar. Kode yang terlihat meyakinkan belum tentu sesuai dengan kebutuhan aplikasi. Hal tersebut membuat programmer tetap harus melakukan pemeriksaan terhadap setiap keluaran AI.

Keterbatasan lain berkaitan dengan pemahaman konteks. Sistem AI dapat mengalami kesulitan ketika harus memahami aplikasi yang sangat kompleks, terutama apabila aplikasi memiliki banyak dependensi, aturan bisnis, konfigurasi khusus, dan kode internal. Oleh karena itu, kualitas keluaran sangat dipengaruhi oleh informasi yang diberikan pengguna.

Aspek keamanan juga menjadi perhatian. Studi mengenai keamanan kode yang dihasilkan LLM menemukan berbagai jenis kerentanan yang dapat muncul dalam kode yang dibuat oleh model. Penelitian tersebut menunjukkan perlunya praktik mitigasi dan penggunaan alat keamanan tambahan ketika kode yang dihasilkan AI digunakan dalam perangkat lunak.

Laporan Center for Security and Emerging Technology juga mengidentifikasi risiko keamanan yang berkaitan dengan AI-generated code, termasuk kemungkinan model menghasilkan kode yang tidak aman serta risiko terhadap sistem AI itu sendiri.

Keterbatasan lainnya adalah potensi ketergantungan pengguna. Mahasiswa atau programmer yang terlalu bergantung pada AI dapat mengalami kesulitan ketika harus menyelesaikan masalah tanpa bantuan AI. Dalam pembelajaran pemrograman, kondisi tersebut dapat menghambat perkembangan kemampuan algoritmik dan kemampuan berpikir kritis.

Oleh karena itu, penggunaan Generative AI sebaiknya disertai dengan kemampuan membaca kode, memahami algoritma, melakukan debugging, dan menguji perangkat lunak secara mandiri.

Generative AI dan Produktivitas Programmer

Produktivitas merupakan salah satu alasan utama berkembangnya penggunaan AI dalam pemrograman. Dengan bantuan AI, programmer dapat mengurangi waktu untuk pekerjaan tertentu seperti membuat kode berulang, mencari contoh sintaks, atau membuat dokumentasi awal.

Namun, produktivitas tidak seharusnya hanya diukur dari jumlah kode yang dihasilkan. Perangkat lunak yang baik harus memenuhi kebutuhan pengguna, dapat dipelihara, aman, dan dapat diuji. Apabila programmer menghasilkan banyak kode tetapi harus menghabiskan waktu yang besar untuk memperbaiki kesalahan yang dihasilkan AI, maka keuntungan produktivitas dapat berkurang.

Kajian sistematis Hou et al. (2024) menunjukkan bahwa penelitian mengenai LLM untuk software engineering berkembang pesat, tetapi efektivitasnya tetap bergantung pada tugas, model, data, metode evaluasi, dan cara teknologi tersebut digunakan.

Dengan demikian, Generative AI sebaiknya digunakan pada bagian pekerjaan yang memang dapat dibantu secara efektif. Programmer tetap perlu menentukan bagian mana yang membutuhkan kreativitas, penalaran, pengalaman, dan keputusan teknis manusia.

Tantangan Etika dalam Penggunaan Generative AI

Salah satu tantangan etika yang penting adalah integritas akademik. Dalam lingkungan perguruan tinggi, mahasiswa dapat menggunakan AI untuk membantu mengerjakan tugas pemrograman. Penggunaan tersebut perlu disesuaikan dengan aturan dosen atau institusi. Apabila mahasiswa menyerahkan seluruh kode yang dibuat AI sebagai hasil pemikiran sendiri tanpa memahami kode tersebut, proses pembelajaran menjadi tidak optimal.

UNESCO menekankan pentingnya penggunaan Generative AI yang berpusat pada manusia, aman, adil, dan bermakna dalam pendidikan dan penelitian. Perlindungan data dan kesiapan institusi juga menjadi bagian penting dari penerapan teknologi tersebut.

Tantangan berikutnya berkaitan dengan hak kekayaan intelektual. Kode yang dihasilkan AI dapat berhubungan dengan berbagai sumber kode yang digunakan dalam proses pelatihan model. Oleh sebab itu, pengguna perlu memperhatikan lisensi perangkat lunak dan aturan yang berlaku sebelum menggunakan kode AI dalam proyek komersial maupun akademik.

Privasi juga perlu diperhatikan. Programmer sebaiknya tidak memasukkan data rahasia, password, API key, data pribadi, atau kode sumber perusahaan ke dalam layanan AI tanpa memahami kebijakan privasi dan penggunaan datanya.

Transparansi menjadi aspek penting lainnya. Dalam konteks akademik atau profesional, pengguna perlu mengikuti aturan mengenai pengungkapan penggunaan AI. Transparansi membantu memastikan bahwa proses pembuatan perangkat lunak dapat dipertanggungjawabkan.

Tanggung Jawab Programmer dalam Penggunaan AI

Walaupun kode dibuat dengan bantuan AI, tanggung jawab terhadap perangkat lunak tetap berada pada manusia yang menggunakan dan menerapkannya. Programmer tidak dapat menjadikan AI sebagai alasan untuk menghindari tanggung jawab ketika program memiliki kesalahan.

Tanggung jawab tersebut dapat dilakukan melalui beberapa langkah. Pertama, programmer harus membaca dan memahami kode yang dihasilkan. Kedua, kode harus diuji menggunakan berbagai kondisi input. Ketiga, programmer perlu memeriksa keamanan kode. Keempat, penggunaan library atau komponen tambahan harus diperiksa berdasarkan dokumentasi dan lisensinya.

Selain itu, programmer perlu menjaga keamanan data ketika berinteraksi dengan layanan AI. Informasi sensitif tidak seharusnya diberikan kepada sistem AI secara sembarangan. Hal tersebut sejalan dengan prinsip perlindungan privasi dalam penggunaan Generative AI yang ditekankan UNESCO.

Pendekatan yang dapat digunakan adalah human-in-the-loop. Dalam pendekatan ini, AI digunakan untuk memberikan bantuan atau rekomendasi, sedangkan manusia melakukan pemeriksaan dan mengambil keputusan akhir. Pendekatan tersebut membuat penggunaan AI lebih terkendali dan dapat dipertanggungjawabkan.

Penerapan Generative AI bagi Mahasiswa Teknik Informatika

Mahasiswa Teknik Informatika dapat menggunakan Generative AI sebagai alat pendukung pembelajaran dengan cara yang terarah. Misalnya, AI dapat digunakan untuk meminta penjelasan mengenai konsep pemrograman, membuat contoh kode sederhana, memberikan latihan debugging, atau membandingkan beberapa algoritma.

Penggunaan AI juga dapat diarahkan menjadi latihan berpikir kritis. Mahasiswa tidak hanya meminta AI membuat kode, tetapi dapat diminta untuk menemukan kesalahan pada kode yang dihasilkan AI. Mahasiswa kemudian menjelaskan kesalahan tersebut dan memperbaikinya.

Contoh lainnya adalah mahasiswa diberikan suatu program yang dihasilkan AI kemudian diminta menganalisis kompleksitas algoritma, keamanan, efisiensi, dan kemungkinan masalahnya. Pendekatan tersebut membuat AI tidak menggantikan proses belajar, tetapi menjadi bahan latihan untuk mengembangkan kemampuan analisis.

Dalam pembelajaran pemrograman, kemampuan dasar tetap penting. Mahasiswa perlu memahami algoritma, struktur data, sintaks bahasa pemrograman, basis data,

jaringan, sistem operasi, dan rekayasa perangkat lunak. Generative AI dapat membantu proses tersebut, tetapi tidak seharusnya menjadi satu-satunya sumber pengetahuan.

Prinsip Penggunaan Generative AI yang Bertanggung Jawab

Penggunaan Generative AI dalam pemrograman dapat dilakukan melalui beberapa prinsip. Pertama, gunakan AI sebagai alat bantu dan bukan sebagai pengganti kemampuan berpikir. Kedua, pahami kode sebelum menggunakannya. Ketiga, lakukan pengujian terhadap kode yang dihasilkan.

Keempat, periksa aspek keamanan, khususnya pada sistem yang berhubungan dengan autentikasi, database, pembayaran, dan data pribadi. Kelima, jangan memasukkan informasi rahasia ke layanan AI tanpa izin. Keenam, periksa dokumentasi dan lisensi library atau kode yang disarankan AI.

Ketujuh, ikuti aturan akademik dan organisasi mengenai penggunaan AI. Kedelapan, apabila diperlukan, cantumkan penggunaan AI secara transparan. Kesembilan, terus tingkatkan kemampuan pemrograman dasar agar penggunaan AI tidak menimbulkan ketergantungan.

Dengan prinsip tersebut, penggunaan Generative AI dapat diarahkan untuk meningkatkan produktivitas sekaligus mempertahankan kualitas dan tanggung jawab manusia dalam pengembangan perangkat lunak.

KESIMPULAN

Generative AI memiliki potensi besar sebagai pendamping pemrograman karena dapat membantu menghasilkan kode, menjelaskan konsep, memberikan saran debugging, membuat dokumentasi, serta membantu pekerjaan pemrograman yang bersifat rutin. Perkembangan penelitian mengenai LLM dalam software engineering menunjukkan bahwa teknologi tersebut telah digunakan dalam berbagai aktivitas pengembangan perangkat lunak.

Namun, Generative AI tidak selalu menghasilkan kode yang benar, aman, dan sesuai dengan kebutuhan. Kode yang dihasilkan tetap membutuhkan pemeriksaan, pengujian, dan evaluasi manusia. Selain keterbatasan teknis, terdapat tantangan etika berupa integritas akademik, hak kekayaan intelektual, privasi, transparansi, dan tanggung jawab.

Oleh karena itu, Generative AI sebaiknya digunakan sebagai alat pendamping, bukan pengganti kemampuan programmer. Penggunaan yang bertanggung jawab dapat dilakukan melalui pendekatan human-in-the-loop, pemeriksaan kode, pengujian, pemeriksaan keamanan, perlindungan data, dan kepatuhan terhadap aturan akademik maupun profesional. Bagi mahasiswa Teknik Informatika, kemampuan memahami algoritma dan melakukan pemrograman secara mandiri tetap menjadi dasar yang penting meskipun teknologi AI semakin berkembang.

DAFTAR PUSTAKA

- Alaoui, A. E., et al. (2024). A systematic literature review on the impact of AI models on the security of code generation. *Frontiers in Big Data*. <https://doi.org/10.3389/fdata.2024.1386720>
- Hou, X., Zhao, Y., Liu, Y., Yang, Z., Wang, K., Li, L., Luo, X., Lo, D., Grundy, J., & Wang, H. (2024). Large language models for software engineering: A systematic literature review. *ACM Transactions on Software Engineering and Methodology*, 33(8), Article 220, 1–79. <https://doi.org/10.1145/3695988>
- Idrisov, B., & Schlippe, T. (2024). Program code generation with generative AIs. *Algorithms*, 17(2), 62. <https://doi.org/10.3390/a17020062>
- IEEE. (2024). State of the art of the security of code generated by LLMs: A systematic literature review. 2024 12th International Conference in Software Engineering Research and

- Innovation (CONISOFT). <https://doi.org/10.1109/CONISOFT63288.2024.00050>
- Ji, J., Jun, J., Wu, M., & Gelles, R. (2024). Cybersecurity risks of AI-generated code. Center for Security and Emerging Technology, Georgetown University.
- Karlovs-Karlovskis, U. (2024). Generative artificial intelligence use in optimising software engineering process: A systematic literature review. *Applied Computer Systems*, 29(1), 68–77. <https://doi.org/10.2478/acss-2024-0009>
- Sauvola, J., Tarkoma, S., Klemettinen, M., Riekk, J., & Doermann, D. (2024). Future of software development with generative AI. *Automated Software Engineering*, 31, Article 26. <https://doi.org/10.1007/s10515-024-00426-z>
- Tihanyi, N., Bisztray, T., Ferrag, M. A., Jain, R., & Cordeiro, L. C. (2025). How secure is AI-generated code: A large-scale comparison of large language models. *Empirical Software Engineering*, 30, Article 47. <https://doi.org/10.1007/s10664-024-10590-1>
- UNESCO. (2023). Guidance for generative AI in education and research. UNESCO.