



EFEKTIVITAS PROMPT ENGINEERING PADA LARGE LANGUAGE MODELS SEBAGAI INSTRUMEN SCAFFOLDING DALAM PENDIDIKAN INFORMATIKA: TINJAUAN LITERATUR SISTEMATIS

Muhamad Rizki Ferdiansyah^{1,*}, Alia Agisna², Muhammad Tegar Yusuf Habibi³

^{1,3} Sekolah Tinggi Teknologi Wastukencana, Indonesia

² Universitas Pendidikan Indonesia, Indonesia

*Correspondence: E-mail: muhamadrizki9@wastukencana.ac.id

ABSTRACT

Integrasi Large Language Models (LLM) dalam pendidikan informatika memunculkan paradoks pedagogis: akses informasi menjadi tanpa batas, namun memicu fenomena "ilusi kompetensi" atau cognitive offloading, di mana siswa bergantung pada kode hasil generate mesin tanpa menguasai landasan algoritmiknya. Penelitian ini merupakan Systematic Literature Review (SLR) yang bertujuan untuk mengeksplorasi efektivitas Prompt Engineering sebagai instrumen scaffolding (perancah kognitif) untuk merestrukturisasi peran AI dari "penyedia solusi" menjadi "tutor pasif". Melalui protokol PRISMA pada basis data Scopus, sebanyak 59 dokumen teridentifikasi, dan 15 studi primer (2024-2026) diekstraksi untuk sintesis kualitatif. Hasil penelitian menunjukkan bahwa implementasi batasan prompt instruksional, khususnya Socratic Prompting dan evaluasi reflektif berbasis baris kode, berhasil memaksa siswa untuk berpikir deduktif dan mengurangi ketergantungan salin-tempel (blind copy-paste). Lebih lanjut, model perancah metakognitif adaptif terbukti mengaktifkan "rem kognitif" yang menjaga otonomi intelektual pembelajar. Kesimpulan dari studi ini menegaskan bahwa keahlian menyusun prompt instruksional telah bertransformasi dari sekadar keterampilan teknis operasi mesin menjadi kompetensi pedagogis esensial bagi pendidik ilmu komputer.

ARTICLE INFO

Article History:

Submitted/Received 23 Aug 2025

First Revised 15 Sep 2025

Accepted 30 Sep 2025

First Available online 04 Oct

2025

Publication Date 06 Oct 2025

Keyword:

Artificial Intelligence in Education,
Computer Science Education,
Large Language Models,
Prompt Engineering,
Scaffolding,
Socratic Method

1. PENDAHULUAN

Adopsi Generative Artificial Intelligence (GenAI) dan Large Language Models (LLM) dalam pendidikan vokasi dan informatika telah memicu disrupsi fundamental dalam lanskap pembelajaran pemrograman. Ketersediaan instrumen seperti ChatGPT, GitHub Copilot, dan model berbasis Transformer lainnya memberikan siswa kemampuan untuk menyelesaikan masalah pemrograman (debugging, generasi kode, optimasi sintaks) dalam hitungan detik. Namun, kemudahan akses ini melahirkan krisis pedagogis baru yang didefinisikan secara akademis sebagai "ilusi kompetensi" akibat cognitive offloading yang berlebihan (D. Liu et al., 2026). Mahasiswa kejuruan informatika kerap merasa telah menguasai kompetensi pemrograman, padahal secara faktual mereka hanya memindahkan kode mentah tanpa menginternalisasi struktur sintaksis maupun paradigma pemecahan masalah (problem-solving) yang mendasarinya (Kharrufa et al., 2026).

Dalam kerangka pedagogi tradisional, LLM sering kali difungsikan layaknya "joki ojek online". Pengguna hanya memasukkan instruksi tujuan akhir, dan AI mengambil alih seluruh proses komputasi, meninggalkan pengguna di kursi penumpang tanpa peta navigasi mental di kepalanya. Jika paradigma interaksi ini dibiarkan secara persisten, ekosistem pendidikan akan menghasilkan lulusan yang mengalami degradasi Computational Thinking.

Merespons ancaman tersebut, penelitian di bidang Computer Science Education (CSEDU) mulai mengintegrasikan teori Zone of Proximal Development (ZPD) dari Vygotsky dengan kapabilitas pemrosesan bahasa alami. Fokusnya bergeser pada rekayasa instruksi awal (Prompt Engineering) sebagai sarana memanipulasi parameter keluaran LLM. Melalui rekayasa prompt yang ketat, posisi AI ditransformasi menjadi "instruktur mengemudi". AI duduk di bangku asisten, membiarkan siswa memegang kendali penuh atas editor kode, dan hanya memberikan intervensi adaptif, pertanyaan pemantik, atau pengereman kognitif manakala siswa melakukan kesalahan struktural (Sun et al., 2026). Metode ini merepresentasikan esensi scaffolding konstruktivistik: bantuan epistemologis diberikan secara bertahap untuk memandu siswa memecahkan masalah di luar batas kemampuan mandirinya, lalu dikurangi (fading) seiring kematangan kompetensi (J. Liu et al., 2026).

Penelitian ini bertujuan untuk melakukan Tinjauan Literatur Sistematis (Systematic Literature Review) guna membedah dan mengevaluasi efektivitas teknik Prompt Engineering, dengan penekanan pada Socratic Prompting dan perancah metakognitif adaptif, sebagai instrumen pendidikan pemrograman. Sintesis komprehensif ini ditujukan untuk memberikan panduan arsitektur interaksi manusia-AI bagi pendidik vokasi guna menstimulasi penalaran algoritmik yang kritis.

2. METODE PENELITIAN

Penelitian ini menerapkan metodologi Systematic Literature Review (SLR) dengan mengadopsi kerangka kerja Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) untuk menjamin transparansi, ketelitian, dan replikabilitas pencarian literatur.

Proses pencarian awal dilakukan pada bulan April 2026 menggunakan basis data akademik bereputasi, yakni Scopus. Strategi pencarian diformulasikan menggunakan operator Boolean multi-lapis untuk memaksimalkan presisi pada sub-disiplin pendidikan informatika: ("Prompt Engineering" OR "Large Language Models" OR "Generative AI") AND ("Scaffolding" OR "Socratic Prompting") AND ("Computer Science Education" OR "Programming" OR "Coding").

Kriteria inklusi ditetapkan dengan parameter berikut: (1) Artikel jurnal dan prosiding konferensi berbahasa Inggris yang melalui proses peer-review; (2) Rentang waktu publikasi dari tahun 2024 hingga 2026 untuk mengisolasi studi pasca-pematangan arsitektur LLM

modern; (3) Riset empiris atau teoretis yang menginvestigasi penerapan strategi intervensi pedagogis atau scaffolding menggunakan AI.

Berdasarkan Search String tersebut, teridentifikasi 77 dokumen awal. Setelah sistem Scopus memfilter batasan bahasa, rentang tahun, dan tipe dokumen, tersisa 59 dokumen empiris. Melalui tahap screening manual terhadap relevansi judul dan abstrak, 31 dokumen dieksklusi karena tidak spesifik membahas pemrograman atau CSEDU. Pada tahap uji kelayakan (eligibility), 28 teks penuh (full-text) dievaluasi, dan 13 artikel dieksklusi karena tidak membedah rekayasa prompt secara detail. Ekstraksi akhir menghasilkan 15 studi primer berkualitas tinggi yang disintesis secara kualitatif dalam riset ini.

3. HASIL DAN PEMBAHASAN

Hasil dan pembahasan merupakan bagian dimana memuat hasil-hasil dari penelitian dan pembahasan. Temuan yang didapatkan dari hasil penelitian harus dijelaskan dan digambarkan secara jelas dengan data-data yang memadai, sehingga dapat menjawab hipotesis dari penelitian yang dimaksud

3.1 Taksonomi Beban Kognitif: Scaffolding versus Offloading

Analisis literatur mengklasifikasikan interaksi pemelajar dengan LLM ke dalam sebuah spektrum kognitif. Pada satu kutub, AI bertindak sebagai "tongkat penyangga" (crutch) yang memicu atrofi analitis, dan di kutub lain sebagai tutor yang mengelevasi kognisi. Liu, Fan, dan Pan (2026) membangun sebuah teori dasar (grounded theory) yang memisahkan batas antara cognitive scaffolding dan offloading dalam pemrograman yang dibantu AI. Studi mereka membuktikan bahwa ketiadaan batasan prompt pedagogis secara absolut mendorong mahasiswa untuk melakukan offloading, menyerahkan beban kognitif penyusunan logika kepada mesin secara utuh (D. Liu et al., 2026).

Untuk membalikkan kecenderungan ini, Denny et al. (2024) merumuskan karakteristik esensial bagi asisten pengajar AI. Model AI harus dirancang (fine-tuned atau di-prompt secara sistemik) untuk menahan respons langsung (direct solution withholding) dan lebih mengutamakan penguraian masalah (problem decomposition) ke dalam unit-unit konseptual yang lebih kecil (Denny et al., 2024).

3.2 Socratic Prompting dan Evaluasi Kode Reflektif

Sintesis data menunjukkan bahwa Socratic Prompting merupakan varian Prompt Engineering yang paling dominan dan terbukti efektif untuk pendidikan informatika. Alih-alih memberikan jawaban deklaratif, teknik ini merekayasa sistem untuk merespons kesalahan dengan pertanyaan interogatif yang terkalibrasi. Sun et al. (2026) menginvestigasi dinamika pembelajaran ketika metode Socrates diinjeksikan ke dalam agen GenAI. Pemelajar yang dihadapkan pada pertanyaan elisitasi dari AI menunjukkan kualitas interaksi yang jauh lebih mendalam, di mana mereka dipaksa menjustifikasi blok kode yang mereka tulis sendiri [3].

Pendekatan ini diekspansi oleh Qiu (2026) melalui perancah tingkat baris kode (One Line at a Time). Alih-alih meminta AI untuk mengevaluasi keseluruhan program yang gagal dikompilasi (compile error), prompt direkayasa agar AI memandu siswa menelusuri alur eksekusi (trace flow) secara sekuensial. Metode bantuan terstruktur ini secara signifikan menekan kebiasaan modifikasi tebak-tebakan (trial-and-error) dan memaksa siswa membangun representasi mental yang utuh atas sintaksis program (Qiu, 2026). Pengembangan Socratic Tutor for Adaptive Programming (STAP) oleh Xie et al. (2025) lebih jauh membuktikan bahwa penundaan gratifikasi instan melalui scaffolding metakognitif membangun resiliensi (debugging resilience) pada pemrogram pemula (Xie et al., 2025).

Tabel 1. Sintesis Studi Kunci Pemetaan Scaffolding AI pada Pendidikan Pemrograman

Peneliti (Tahun)	Fokus Intervensi / Pendekatan Prompting	Temuan Utama / Dampak Pedagogis
D. Liu et al. (2026)	<i>Cognitive scaffolding</i> vs <i>offloading</i> .	Ketiadaan batas instruksional mendorong siswa menggunakan AI sebagai "tongkat", mereduksi otonomi belajar.
Sun et al. (2026)	GenAI dengan Metode Socrates.	Dialog elisitasi memaksa penalaran deduktif tingkat tinggi dan mengoptimalkan pemahaman alur kontrol kode.
Qiu (2026)	Evaluasi Reflektif (<i>One Line at a Time</i>).	<i>Scaffolding</i> sekuensial pada level baris kode efektif meruntuhkan kebiasaan <i>blind copy-paste</i> .
Xie et al. (2025)	<i>Socratic Tutor for Adaptive Programming</i> .	Intervensi Socratic terstruktur membangun resiliensi siswa saat menghadapi <i>compiler errors</i> .
Hou et al. (2024)	<i>Parsons Problems</i> + Penjelasan LLM.	Eksplanasi tekstual bertingkat lebih mendongkrak keterlibatan penyelesaian soal dibanding menampilkan hasil akhir.
D. Liu et al. (2026)	<i>Adaptive vs Planned Metacognitive Scaffolding</i> .	Perancah adaptif yang membaca level kompetensi siswa (<i>state-aware</i>) mendongkrak performa <i>Computational Thinking</i> .
Zou et al. (2026)	Rem Kognitif (<i>Cognitive Brake</i>) & Kesadaran Etis.	Desain pedagogi yang memaksa siswa memverifikasi luaran AI dapat meringem ketergantungan absolut terhadap otomatisasi.

3.3 Adaptabilitas dan Pengaktifan "Rem Kognitif"

Tantangan sentral dalam mengelola interaksi siswa-LLM adalah tingkat intrusivitas bantuan. Liu et al. (2026) membedah komparasi antara perancah metakognitif yang direncanakan secara statis (*planned*) dengan yang bersifat adaptif. AI yang dikalibrasi untuk membaca konteks (*context-aware*) melalui prompt dinamis terbukti lebih ampuh dalam mempertahankan flow state pembelajar dasar di level pendidikan menengah dan tinggi (J. Liu et al., 2026). Di sisi lain, Tye Dobson & Karkalas (2026) menekankan perlunya minimally intrusive GenAI scaffolding, di mana intervensi AI tidak mendominasi layar atau membanjiri fungsi kognitif (*cognitive overload*), melainkan hadir secara tersembunyi sebagai alat evaluasi post-execution (Tye Dobson & Karkalas, 2026).

Lebih jauh, literatur memunculkan konsep "Rem Kognitif" (Cognitive Brake). Zou et al. (2026) menemukan bahwa kebaruan pedagogis dalam adopsi AI harus diiringi dengan pengembangan kesadaran etis (Zou et al., 2026). Siswa harus disadarkan akan sifat probabilistik LLM (yang rentan terhadap halusinasi logis) (Zhang, 2026). Pendidik memiliki tanggung jawab untuk menyematkan instruksi system prompt yang tidak hanya membatasi jawaban, namun secara eksplisit memerintahkan model AI untuk mendesak pengguna melakukan verifikasi silang terhadap setiap saran kode yang diberikan. Desain prompt ini secara inheren melatih skeptisisme akademis.

4. PENUTUP

Integrasi LLM secara plug-and-play tanpa mitigasi pedagogis di lingkungan pendidikan kejuruan informatika memicu efek degradasi pada kompetensi pemecahan masalah algoritma. Tinjauan literatur sistematis ini menyimpulkan bahwa instrumen scaffolding yang direkayasa melalui Prompt Engineering mutlak diperlukan untuk membendung ilusi kompetensi. Penerapan Socratic Prompting, dekomposisi masalah bertahap (line-by-line reflective evaluation), dan perancah metakognitif terbukti secara empiris berhasil menempatkan AI kembali pada kodrat akademisnya: sebagai tutor fasilitator, bukan pengerja tugas eksklusif. Konsekuensi logis dari pergeseran ini menuntut para dosen dan pengembang kurikulum informatika untuk tidak lagi sekadar menjadi ahli sintaksis bahasa pemrograman, melainkan berevolusi menjadi arsitek interaksi (melalui Prompt Engineering) yang mampu mendesain parameter kognitif presisi tinggi pada mesin kecerdasan buatan. Arah penelitian selanjutnya harus difokuskan pada studi komparatif jangka panjang untuk memetakan kurva retensi kemampuan arsitektural perangkat lunak pada kohort siswa yang terpapar Socratic AI tutor.

6. REFERENCES

- Denny, P., Macneil, S., Savelka, J., Porter, L., & Luxton-Reilly, A. (2024). Desirable Characteristics for AI Teaching Assistants in Programming Education. *Annual Conference on Innovation and Technology in Computer Science Education*, 408–414.
- Hou, X., Ericson, B. J., & Wang, X. (2024). Integrating Personalized Parsons Problems with Multi-Level Textual Explanations to Scaffold Code Writing. *ACM Technical Symposium on Computer Science Education*, 1686–1687.
- Kharrufa, A., Alghamdi, S., Aziz, A., & Bull, C. (2026). LLMs Integration in Software Engineering Team Projects: Roles, Impact, and a Pedagogical Design Space for AI Tools in Computing Education. *ACM Transactions on Computing Education*, 26(2), 20.
- Liu, D., Fan, G., & Pan, L. (2026). Tool, tutor, or crutch?: A grounded theory of cognitive scaffolding and offloading in AI-assisted programming education. *International Journal of STEM Education*, 13(1), 10.
- Liu, J., Zhang, Y., Li, W., Niu, P., & Zhang, X. (2026). Adaptive vs. planned metacognitive scaffolding for computational thinking: Evidence from generative AI-supported programming in elementary education. *Computers and Education*, 241, 105473.
- Qiu, V. (2026). One Line at a Time: Scaffolding Reflective Code Evaluation through Structured AI Assistance. *Australasian Computing Education Conference*, 102–111.

- Sun, D., Zheng, Y., Xu, J., & Yang, Z. (2026). When Generative AI Meets Socratic Method: Investigating Programming Learning Dynamics Through Behaviours, Interaction Qualities and Perceptions. *Journal of Computer Assisted Learning*, 42(2), e70210.
- Tye Dobson, L.-R., & Karkalas, S. (2026). Exploring Minimally Intrusive GenAI Scaffolding for Introductory Programming Education. *International Conference on Software and Information Engineering*, 26–34.
- Xie, W., Shi, H., Li, Y., & Liu, J. (2025). STAP: A Socratic Tutor for Adaptive Programming with Pedagogical Scaffolding. *International Symposium on Artificial Intelligence for Education*, 584–587.
- Zhang, S. (2026). Examining Students' Code Comprehension with LLMs in Block- and Text-Based Programming. *ACM Technical Symposium on Computer Science Education*, 1601–1602.
- Zou, H., Chan, K. I., Pang, P., Manditereza, B., & Shih, Y.-H. (2026). To Use but Not to Depend: Pedagogical Novelty and the Cognitive Brake of Ethical Awareness in Computer Science Students' Adoption of Generative AI. *Education Sciences*, 16(2), 311.