

Full Paper

Comparative Analysis of Vibe Coding Usage, Perception of Programming Ability, and Prompt Quality of Software Engineering Students

Novita Syahwa Tri Hapasari ¹,

Telkom University, Banyumas, Indonesia

novitasyahwa@student.telkomuniversity.ac.id

Rifki Adhitama ^{2*},

Telkom University, Banyumas, Indonesia

rifkiadhitama@telkomuniversity.ac.id

Yudha Islami Sulistya ³,

Telkom University, Banyumas, Indonesia

yudhaislamisulistya@telkomuniversity.ac.id

ABSTRACT

The growing use of generative AI has introduced vibe coding as a new programming approach in higher education, but its varying impacts on students' experience levels have not yet been extensively explored in the Indonesian context. This study investigates whether vibe coding is differently associated with programming competencies between early-stage and advanced-stage students in the Software Engineering Program at Telkom University Purwokerto. A comparative design with a mixed-methods approach was used, involving 30 respondents: 15 early-stage students (Semester 4) and 15 advanced-stage students (Semester 6). Data were collected via a five-point Likert-scale questionnaire measuring the intensity of vibe coding (Section B) and perceptions of programming competence (Section C), supplemented by a prompt quality rubric validated by faculty members based on a case study of a circular single-linked list. Quantitative analysis employed Spearman's rank correlation and the Mann-Whitney U test, while qualitative findings from the rubric were triangulated with questionnaire results. The results indicate that advanced-level students have a significantly higher coding vibe intensity ($U = 57.5$; $p = 0.023$), while beginner-level students report a significantly higher perceived ability ($U = 39.0$; $p = 0.002$). The Spearman test yielded a non-significant negative correlation in both groups, with the final-year group showing a moderate trend ($r_s = -0.50$). No significant differences were found in prompt output quality between groups ($U = 92.0$; $p = 0.400$), and triangulation revealed inconsistencies between perceived ability and actual output quality. These findings indicate that vibe coding without structured instructional guidance has the potential to widen the gap between perceived and actual programming competence, underscoring the need for the explicit integration of prompting literacy into the programming curriculum.

KEYWORDS

vibe coding; programming competency; prompting literacy; Cognitive Load Theory; mixed methods; software engineering

Tri Hapasari. Novita. Adhitama. Rifki. Sulistya. Yudha Islami. (2026). Comparative Analysis of Vibe Coding Usage, Perception of Programming Ability, and Prompt Quality of Software Engineering Students. *jasmed*, 4(1), pp. 34-48.
<https://doi.org/10.20895/jasmed.v4i1.11260>

Article Submitted 07/06/2026. Revision uploaded 30/03/2026. Accepted 02/07/2026.

© 2026 by the authors of this article. Published under CC-BY 

1. INTRODUCTION

Kemajuan kecerdasan buatan generatif dalam beberapa tahun terakhir telah mendorong munculnya paradigma pemrograman baru yang dikenal sebagai *vibe coding*. Berbeda dengan pendekatan konvensional yang mengharuskan pengembang menulis kode secara langsung baris per baris, *vibe coding* memungkinkan pengguna menyampaikan tugas atau kebutuhan dalam bentuk bahasa alami, yang kemudian secara otomatis diterjemahkan menjadi kode oleh model kecerdasan buatan. *Vibe coding* didefinisikan sebagai paradigma pemrograman yang sedang berkembang, di mana proses pengembangan perangkat lunak dilakukan terutama melalui interaksi percakapan dengan *large language model* (LLM) penghasil kode, bukan dengan menulis kode secara langsung [1]. Istilah *vibe coding* ini diperkenalkan oleh Karpathy pada awal Februari 2025 pendekatan ini berkembang pesat melalui berbagai platform seperti Claude Code, ChatGPT, GitHub Copilot, dan Google Antigravity mendukung pendekatan ini dengan cepat. Efektivitas dari berbagai platform tersebut didukung oleh bukti empiris secara global. Dalam analisis kualitatif terhadap 57 video praktisi yang dipublikasikan antara akhir tahun 2024 hingga 2025, [2] menemukan bahwa *vibe coding* dapat mempercepat linimasa pengembangan perangkat lunak dan mengurangi hambatan masuk, memungkinkan bahkan orang yang tidak ahli untuk membuat aplikasi fungsional dengan prompt deskriptif. Semakin banyak mahasiswa Program Studi Rekayasa Perangkat Lunak termasuk di Telkom University Purwokerto menggunakan teknologi ini, yang pada akhirnya memengaruhi industri ini. Fenomena ini menunjukkan bahwa *vibe coding* bukan sekadar tren teknologi, melainkan telah menjadi bagian dari ekosistem pembelajaran pemrograman secara global.

Kemunculan *vibe coding* dalam ekosistem pendidikan memunculkan ekspektasi positif terhadap kemampuan pemrograman mahasiswa. AI code generator memiliki potensi untuk membantu pemrogram pemula dengan menghasilkan kode dari deskripsi bahasa alami, sehingga mahasiswa dapat lebih berfokus pada pemahaman konsep daripada terhambat oleh persoalan sintaks [3]. Sejalan dengan itu, alat bantu berbasis AI generatif mampu mendukung pengembangan kompetensi komputasional mahasiswa dengan menyediakan umpan balik otomatis, mendorong eksplorasi solusi alternatif, dan mengurangi beban kognitif dalam penyelesaian tugas pemrograman [4]. Ekspektasi ini mendorong optimisme bahwa *vibe coding* akan membantu mahasiswa menghasilkan kode lebih cepat, berkonsentrasi pada pemahaman logika, dan memperluas aksesibilitas pemrograman bagi mereka yang sebelumnya terkendala hambatan teknis.

Namun, penelitian terbaru menunjukkan bahwa dampak *vibe coding* tidak bersifat seragam dan bergantung pada konteks penggunaannya. Ketergantungan berlebihan pada AI berisiko menghambat pertumbuhan kognitif mahasiswa dan melemahkan kemampuan berpikir kritis, khususnya ketika mahasiswa menggunakan AI tanpa disertai panduan instruksional yang tepat [5]. Ketergantungan berlebihan pada AI code generator berpotensi memperburuk keterampilan metakognitif mahasiswa pemula, di mana mahasiswa yang menggunakan GitHub Copilot mengalami kesulitan dalam memverifikasi dan memahami kode yang dihasilkan AI [6]. Penelitian lebih lanjut menemukan bahwa meskipun *vibe coding* terbukti meningkatkan efisiensi pengembangan dibandingkan pemrograman konvensional, pendekatan ini menimbulkan kekhawatiran serius terkait maintainability, keamanan kode, dan kepercayaan pengguna terhadap output yang dihasilkan, sehingga belum dapat menggantikan praktik konvensional secara menyeluruh [7]. Temuan-temuan ini mengindikasikan bahwa pemanfaatan *vibe coding* dalam pembelajaran memerlukan pendampingan pedagogis yang terstruktur agar tidak justru merugikan perkembangan kompetensi mahasiswa.

Kesenjangan dampak tersebut semakin jelas ketika dikaji berdasarkan jenjang pengalaman pemrograman mahasiswa. Terdapat kesenjangan yang luas antara keuntungan dan kerugian AI generatif bagi pemrogram pemula, di mana sebagian mahasiswa mampu memanfaatkannya untuk mempercepat progres sementara sebagian lain justru mengalami kerugian akibat rendahnya kesadaran metakognitif dalam memverifikasi output AI [8]. Kegagalan mahasiswa pemula dalam memanfaatkan LLM untuk tugas pembuatan kode bukan disebabkan oleh kurangnya kosakata teknis, melainkan karena mereka tidak memahami sejauh mana informasi kontekstual yang dibutuhkan AI untuk menghasilkan kode yang tepat substansi prompt terbukti jauh lebih menentukan keberhasilan dibandingkan gaya penulisannya [9]. Penelitian terkini tentang interaksi mahasiswa dengan platform *vibe coding* menemukan bahwa mahasiswa tingkat lanjut jauh lebih mampu menyertakan konteks fitur aplikasi dan basis kode yang relevan dalam prompt mereka dibandingkan mahasiswa tingkat pemula [10]. Keseluruhan temuan ini menegaskan bahwa efektivitas *vibe coding* sangat bergantung pada tingkat pengalaman pemrograman penggunanya, sebuah faktor yang belum banyak dikaji secara komparatif dalam konteks pendidikan tinggi.

Permasalahan di atas menjadi sangat relevan untuk dikaji dalam Program Studi S1 Rekayasa Perangkat Lunak di Telkom University Purwokerto. Program ini melibatkan sejumlah mata kuliah pemrograman yang memiliki pendekatan pedagogis berbeda mulai dari Pemrograman Perangkat Bergerak, Struktur Data, Dasar Pemrograman Berorientasi Objek (DPBO), dan Pemrograman 1. Dalam hal ini, mahasiswa tingkat awal yang baru menempuh mata kuliah pemrograman dasar berada pada tahap pembentukan fondasi algoritmik dan konsep dasar pemrograman secara mandiri, sementara mahasiswa tingkat akhir yang telah menempuh seluruh rangkaian mata kuliah pemrograman memiliki skema pemrograman yang lebih matang dan pengalaman yang lebih beragam. Perbedaan jenjang ini secara langsung mempertanyakan apakah *vibe coding* mempengaruhi kemampuan pemrograman kedua kelompok tersebut dengan cara yang sama atau justru berbeda. Pertanyaan ini belum terjawab secara memadai dalam literatur yang ada. F.Geng et al. dalam penelitian terkini tentang interaksi mahasiswa dengan platform *vibe coding* menemukan bahwa mahasiswa tingkat lanjut jauh lebih mampu menyertakan konteks fitur aplikasi dan basis kode yang relevan dalam prompt mereka dibandingkan mahasiswa tingkat pemula [10]. Namun penelitian tersebut dilakukan di luar konteks Indonesia menggunakan metode observasi kualitatif tanpa instrumen kuesioner persepsi, sehingga belum dapat menjelaskan bagaimana persepsi kemampuan mandiri mahasiswa berhubungan dengan kualitas output yang dihasilkan. Sejauh ini belum ada penelitian yang mengkaji perbedaan tersebut dalam satu program studi di perguruan tinggi Indonesia menggunakan pendekatan *mixed methods* yang menggabungkan kuesioner persepsi dan validasi studi kasus oleh dosen.

Maka dari itu, penelitian ini bertujuan untuk menjawab pertanyaan: Apakah pengaruh *vibe coding* terhadap kemampuan pemrograman mahasiswa berbeda antara mahasiswa tingkat awal dan tingkat akhir di Program Studi S1 Rekayasa Perangkat Lunak Telkom University Purwokerto? Untuk menjawab pertanyaan tersebut, penelitian ini secara spesifik akan: (1) mengidentifikasi pola dan intensitas penggunaan *vibe coding* pada mahasiswa tingkat awal dan tingkat akhir; (2) menganalisis hubungan antara intensitas penggunaan *vibe coding* dengan persepsi kemampuan pemrograman pada kedua kelompok; (3) membandingkan perbedaan pengaruh *vibe coding* antara mahasiswa tingkat awal dan tingkat akhir melalui kuesioner persepsi dan penilaian kualitas output prompt oleh dosen pengampu.

Secara praktis, hasil penelitian ini diharapkan dapat memberikan manfaat yang pertama bagi mahasiswa, penelitian ini menyediakan pemahaman berbasis bukti tentang bagaimana *vibe coding* berelevansi dengan proses belajar pemrograman sesuai jenjang studi mereka, sehingga mereka dapat menggunakannya secara lebih reflektif mengetahui kapan teknik ini mendukung pembelajaran dan kapan perlu diminimalkan agar tidak menghalangi perkembangan kemampuan berpikir mandiri dan keterampilan pemrograman fundamental. Kedua, bagi dosen Rekayasa Perangkat Lunak, temuan penelitian ini dapat menjadi acuan dalam merancang intervensi pedagogis yang lebih tepat sasaran, misalnya dengan menyesuaikan panduan penggunaan AI dalam tugas pemrograman berdasarkan tingkat pengalaman mahasiswa dan karakteristik mata kuliah yang diampu.

Secara teoritis, penelitian ini memberikan kontribusi pada setidaknya tiga aspek keilmuan. Pertama, penelitian ini memperkaya literatur tentang dampak AI generatif dalam pendidikan pemrograman dengan menyediakan perspektif komparatif berbasis jenjang studi yang selama ini masih langka, khususnya di konteks perguruan tinggi Indonesia. Sebagian besar studi yang ada dilakukan di luar konteks Indonesia dan tidak membandingkan kelompok mahasiswa secara eksplisit berdasarkan jenjang pengalaman dalam satu program studi yang sama [8], [10]. Kedua, penelitian ini mengembangkan kerangka analisis *mixed methods* yang menggabungkan kuesioner persepsi mahasiswa dengan validasi kualitas output oleh dosen, sehingga memberikan triangulasi data yang lebih kuat dibandingkan pendekatan tunggal berbasis survei atau observasi. Kerangka ini berpotensi diadaptasi oleh peneliti lain dalam konteks institusi yang berbeda. Ketiga, penelitian ini berkontribusi pada pengembangan teori tentang peran metakognisi dan kualitas prompt dalam menentukan efektivitas *vibe coding*, yang dapat memperluas konstruk tentang substansi prompt sebagai faktor penentu keberhasilan penggunaan LLM dalam konteks pembelajaran pemrograman formal [9].

2. METHODS

Penelitian ini mengadopsi pendekatan *mixed methods* dengan desain *explanatory sequential*, yakni pengumpulan dan analisis data kuantitatif dilakukan terlebih dahulu, kemudian diikuti oleh fase kualitatif untuk menjelaskan dan memperdalam hasil statistik yang diperoleh [11]. Pendekatan ini dipilih karena tujuan penelitian mencakup dua dimensi sekaligus: mengukur hubungan statistik antara intensitas penggunaan *vibe coding* dan persepsi

kemampuan pemrograman mahasiswa, sekaligus mengidentifikasi kualitas aktual output yang dihasilkan melalui validasi rubrik oleh dosen pengampu mata kuliah yang bersangkutan.

Secara operasional, desain *explanatory sequential* dalam penelitian ini dijalankan dalam dua fase yang berurutan. Fase pertama (kuantitatif) terdiri atas pengumpulan data kuesioner skala Likert Bagian B dan Bagian C, yang kemudian dianalisis menggunakan uji Mann-Whitney dan korelasi Spearman sebagaimana disajikan pada Tabel 7 dan Tabel 8. Hasil fase ini menunjukkan bahwa mahasiswa tingkat akhir memiliki intensitas penggunaan *vibe coding* yang signifikan lebih tinggi ($U = 57,5$; $p = 0,023$), sementara mahasiswa tingkat awal melaporkan persepsi kemampuan pemrograman yang signifikan lebih tinggi ($U = 39,0$; $p = 0,002$). Pola yang berlawanan ini memunculkan pertanyaan yang tidak dapat dijawab oleh data kuantitatif saja: bagaimana mungkin intensitas penggunaan AI yang lebih tinggi berkaitan dengan persepsi kemampuan mandiri yang lebih rendah? Pertanyaan inilah yang mendasari fase kedua (kualitatif), yang dimulai setelah hasil Mann-Whitney pada Total B dan Total C diperoleh. Data kualitatif berupa prompt dan output aktual mahasiswa pada studi kasus Queue ADT dinilai oleh dosen pengampu menggunakan rubrik tiga dimensi (relevansi prompt, kelengkapan output, dan akurasi teknis), kemudian dianalisis secara tematik untuk mengidentifikasi pola kesalahan yang berulang pada kedua kelompok. Hasil tematik ini digunakan untuk menjelaskan temuan kuantitatif pada Mann-Whitney 3, yang menunjukkan tidak terdapat perbedaan signifikan pada kualitas output prompt ($U = 92,0$; $p = 0,400$): pola kesalahan yang serupa pada prompt kedua kelompok mengindikasikan bahwa persepsi kemampuan yang lebih tinggi pada mahasiswa tingkat awal maupun intensitas penggunaan AI yang lebih tinggi pada mahasiswa tingkat akhir sama-sama belum diiringi oleh penguasaan prompting yang memadai. Dengan demikian, fase kualitatif tidak sekadar melengkapi, tetapi secara langsung menjelaskan mengapa perbedaan signifikan pada persepsi dan intensitas penggunaan tidak tercermin pada perbedaan kualitas output.

Penelitian ini menggunakan desain komparatif dengan melibatkan dua kelompok mahasiswa Program Studi S1 Rekayasa Perangkat Lunak Telkom University Purwokerto dengan mahasiswa tingkat akhir yang telah menempuh sebagian besar mata kuliah pemrograman, serta mahasiswa tingkat awal yang baru menyelesaikan mata kuliah pemrograman dasar. Desain ini sejalan dengan pendekatan yang membandingkan interaksi mahasiswa tingkat pemula dan tingkat lanjut dengan platform *vibe coding*, yang menunjukkan bahwa mahasiswa tingkat lanjut menghasilkan prompt dengan kualitas dan konteks teknis yang lebih baik [10]. Kerangka teori yang mendasari penelitian ini adalah *Cognitive Load Theory* (CLT) yang membagi beban kognitif menjadi tiga komponen: intrinsik, ekstrinsik, dan germane. Alam et al. menegaskan bahwa dari perspektif CLT, hubungan antara penggunaan AI tools dan hasil belajar mahasiswa dapat dijelaskan melalui mekanisme pengurangan beban kognitif ekstrinsik: ketika AI tools menyederhanakan tugas kompleks dan mengotomatisasi proses rutin, mahasiswa berpotensi mengalokasikan lebih banyak sumber daya kognitif untuk pemrosesan germane seperti konstruksi konseptual, namun tanpa panduan instruksional yang tepat, AI tools juga dapat membebani pelajar dan mengurangi efektivitas belajar [12]. Kerangka ini digunakan untuk menjelaskan bagaimana penggunaan *vibe coding* dapat mengubah distribusi beban kognitif secara berbeda antara mahasiswa tingkat awal yang masih membangun fondasi algoritma dan mahasiswa tingkat akhir yang sudah memiliki skema pemrograman yang lebih matang.

2.1 Responden Penelitian

Responden penelitian terdiri dari 30 mahasiswa Program Studi S1 Rekayasa Perangkat Lunak Telkom University Purwokerto, yang dipilih menggunakan teknik purposive sampling berdasarkan perbedaan paparan terhadap mata kuliah pemrograman. Kelompok pertama terdiri dari 15 mahasiswa tingkat awal yang baru menempuh mata kuliah pemrograman dasar, dan kelompok kedua terdiri dari 15 mahasiswa tingkat akhir yang telah menempuh sebagian besar mata kuliah pemrograman dalam kurikulum. Pengambilan data dilakukan pada Mei 2025 menggunakan kuesioner daring melalui Google Form yang didistribusikan kepada responden di masing-masing kelompok.

2.2 Instrumen Penelitian

Instrumen kuesioner pada penelitian ini dikembangkan secara mandiri oleh peneliti (*researcher-developed*) karena belum terdapat instrumen baku yang secara spesifik mengukur intensitas *vibe coding* dan persepsinya terhadap kemampuan pemrograman mahasiswa. Pengembangan butir Bagian B mengacu pada karakteristik perilaku *vibe*

coding sebagaimana diidentifikasi oleh Sarkar dan Drosos [1], yang mendefinisikan *vibe coding* sebagai paradigma pemrograman yang sedang berkembang, di mana pengembang terutama menulis kode melalui interaksi percakapan dengan *large language model* (LLM) penghasil kode, alih-alih menulis kode secara langsung. Butir Bagian C disusun mengacu pada isu kompetensi pemrograman di era AI generatif yang diuraikan oleh Denny et al. [4], yang mencakup antara lain kemampuan mahasiswa memahami kode yang dihasilkan AI, kemampuan menulis kode secara mandiri tanpa bantuan AI, serta kemampuan mengevaluasi dan memperbaiki (*debugging*) kode yang dihasilkan.

Sebelum digunakan dalam pengambilan data utama, instrumen kuesioner Bagian B dan Bagian C terlebih dahulu diuji validitas dan reliabilitasnya menggunakan seluruh data responden ($N = 30$). Uji validitas dilakukan dengan menghitung korelasi item-total terkoreksi (*corrected item-total correlation*), yaitu korelasi Pearson antara skor setiap butir dengan skor total seluruh butir lain dalam bagian yang sama. Butir dinyatakan valid apabila nilai r hitung lebih besar dari 0,30 [18]. Uji reliabilitas dilakukan menggunakan koefisien Cronbach's Alpha, dengan instrumen dinyatakan reliabel apabila nilai α lebih besar dari 0,70 [19]. Kedua pengujian dilakukan secara terpisah untuk Bagian B dan Bagian C. Hasil lengkap uji validitas dan reliabilitas disajikan pada bagian Result (Tabel 1 dan Tabel 2).

Instrumen kuesioner dengan skala Likert 1–5 dipilih sebagai alat pengumpulan data utama karena tujuan penelitian ini adalah mengukur persepsi mahasiswa terhadap kemampuan pemrograman mereka, bukan mengamati proses interaksi secara real-time. Pendekatan ini berbeda dengan eksperimen langsung berbasis *think-aloud* dalam sesi terkontrol yang mengamati perilaku mahasiswa secara observasional saat berinteraksi dengan platform *vibe coding* [10]. Pendekatan tersebut efektif untuk menjawab pertanyaan tentang bagaimana mahasiswa berperilaku saat *vibe coding*, namun karena fokus penelitian ini berbeda yakni bagaimana mahasiswa menilai diri sendiri terhadap kemampuan pemrograman mereka dalam konteks penggunaan *vibe coding* pertanyaan tersebut lebih tepat dijawab melalui instrumen laporan diri (*self-report*) daripada observasi langsung [15]. Pendekatan survei juga memungkinkan perbandingan sistematis antara dua kelompok mahasiswa dengan jumlah yang memadai untuk analisis statistik non-parametrik [11]. Kuesioner terdiri dari tiga bagian: (a) data diri responden yang mencakup nama, NIM, semester aktif, mata kuliah yang pernah ditempuh, dan nilai akhir mata kuliah Struktur Data sebagai variabel kontrol; (b) pengukuran intensitas penggunaan *vibe coding* secara umum dalam kegiatan pemrograman perkuliahan; dan (c) studi kasus pembuatan prompt berbasis topik Queue Circular Single Linked List dari mata kuliah Struktur Data.

Instrumen kedua adalah studi kasus pembuatan prompt di mana seluruh responden diminta membuat prompt untuk menghasilkan algoritma enqueue dan dequeue sesuai ADT yang diberikan, kemudian menjelaskan pemahaman mereka terhadap kode yang dihasilkan dan mengevaluasi kesesuaiannya dengan ADT. Studi kasus prompt dipilih sebagai instrumen kualitatif karena kemampuan mahasiswa dalam menyusun prompt yang efektif untuk menghasilkan kode merupakan kompetensi inti yang dapat diukur secara langsung melalui artefak yang dihasilkan. Pendekatan ini telah terbukti mampu mengukur kemampuan mahasiswa dalam memahami masalah, merumuskan instruksi yang tepat, dan mengevaluasi kode yang dihasilkan AI dimensi kompetensi yang tidak dapat ditangkap melalui observasi proses *real-time* semata [16]. Dengan memberikan ADT yang sama kepada seluruh responden, studi kasus ini memastikan komparabilitas hasil antarkelompok sekaligus menjaga relevansi terhadap capaian pembelajaran mata kuliah Struktur Data.

Instrumen ketiga adalah rubrik penilaian prompt yang dikembangkan bersama dosen pengampu mata kuliah Struktur Data. Validasi berbasis rubrik dosen ahli dipilih dibandingkan tes objektif atau eksperimen terkontrol karena rubrik yang dikembangkan secara spesifik oleh instruktur terbukti lebih mampu menangkap nuansa penalaran logis dan akurasi konseptual dibandingkan kriteria penilaian generik, khususnya pada mata kuliah Data Structures and Algorithms di mana pemahaman terhadap struktur dan perilaku program lebih penting daripada sekadar kebenaran sintaksis [17]. Tes objektif seperti unit test otomatis hanya mengukur kebenaran output kode, sementara rubrik dosen memungkinkan penilaian terhadap kualitas prompt dari sisi relevansi pedagogis, kelengkapan output, dan kesesuaian teknis terhadap ADT tiga dimensi yang tidak dapat ditangkap oleh evaluasi otomatis semata. Rubrik mencakup tiga kriteria yaitu relevansi prompt terhadap tujuan pembelajaran, kelengkapan output yang dihasilkan, dan akurasi teknis kode terhadap ADT yang diberikan.

2.3 Teknik Analisis Data

Data yang dikumpulkan dianalisis melalui dua jalur yang kemudian ditriangulasi. Pada jalur kuantitatif, analisis statistik deskriptif digunakan untuk mengukur distribusi skor persepsi kemampuan pemrograman dan

intensitas penggunaan *vibe coding* pada kedua kelompok. Uji korelasi Spearman (Spearman's rho) digunakan untuk menganalisis hubungan antara intensitas penggunaan *vibe coding* dan persepsi kemampuan pemrograman. Pemilihan uji Spearman didasarkan pada sifat data ordinal yang diperoleh dari skala Likert, di mana uji non-parametrik ini tidak mensyaratkan distribusi normal dan tepat digunakan untuk mengukur hubungan monoton antar variabel ordinal [15]. Uji beda Mann-Whitney dipilih untuk membandingkan perbedaan skor antara kedua kelompok mahasiswa karena uji ini tidak mengasumsikan distribusi normal dan sesuai untuk membandingkan dua kelompok independen dengan data ordinal [15].

Pada jalur kualitatif, hasil validasi dosen terhadap tugas prompt dianalisis secara tematik untuk menemukan pola kualitas output *vibe coding* berdasarkan perbedaan jenjang mahasiswa. Triangulasi antara kedua jalur dilakukan untuk memverifikasi konsistensi temuan, yakni apakah kelompok yang menunjukkan persepsi kemampuan tertentu pada kuesioner juga menghasilkan kualitas prompt yang konsisten menurut penilaian dosen

$$r_s = 1 - \frac{6 \sum d_i^2}{n(n^2-1)} \quad (1)$$

$$t = r_s \sqrt{\frac{n-2}{1-r_s^2}} \quad (2)$$

$$U_1 = n_1 n_2 + \frac{n_1(n_1+1)}{2} - \sum R_1 \quad (3)$$

3. RESULT

Dalam penelitian ini melibatkan 30 mahasiswa Program Studi S1 Rekayasa Perangkat Lunak di Telkom University Purwokerto, yang terbagi menjadi 2 kelompok yaitu mahasiswa tingkat awal (n=15) dan mahasiswa tingkat akhir (n=15). Data kuesioner dikumpulkan menggunakan Google Form dengan mencakup tiga bagian: identitas responden, intensitas penggunaan *vibe coding* (Bagian B), dan persepsi kemampuan pemrograman (Bagian C). Sebagaimana dijelaskan pada bagian 2.2, instrumen kuesioner Bagian B dan Bagian C terlebih dahulu diuji validitas dan reliabilitasnya menggunakan seluruh data responden (N = 30) sebelum digunakan dalam analisis utama. Hasil uji validitas (*corrected item-total correlation*) dan reliabilitas (Cronbach's Alpha) Bagian B disajikan pada Tabel 1, sedangkan hasil pengujian Bagian C disajikan pada Tabel 2.

Table 1 Hasil Uji Validitas dan Reliabilitas Bagian B – Intensitas Vibe Coding (N = 30).

Butir	Pernyataan	r hitung	p-value	Keterangan
B1	Saya menggunakan AI untuk menghasilkan kode secara langsung dari deskripsi yang saya tulis	0,8095	0,000	Valid
B2	Saya menyerahkan pembuatan kode sepenuhnya kepada AI tanpa menulis kode sendiri terlebih dahulu	0,8143	0,000	Valid
B3	Saya menggunakan AI sebagai langkah pertama sebelum mencoba memahami atau mengerjakan soal sendiri	0,4541	0,012	Valid
B4	Saya mengumpulkan tugas dari kode yang dihasilkan AI tanpa memodifikasinya secara signifikan	0,7073	0,000	Valid

B5	Saya merasa lebih mudah menyelesaikan tugas di mata kuliah yang sudah saya tempuh dengan bantuan AI	0,7110	0,000	Valid
Cronbach's Alpha		0,8678	–	Reliabel

Berdasarkan Tabel 1, seluruh butir pada Bagian B memiliki nilai *r hitung* berkisar antara 0,454 hingga 0,814, yang semuanya berada di atas nilai kritis 0,30 sehingga dinyatakan valid. Nilai Cronbach's Alpha sebesar 0,868 menunjukkan bahwa instrumen Bagian B memiliki konsistensi internal yang tinggi dan reliabel digunakan sebagai alat ukur intensitas penggunaan *vibe coding* pada penelitian ini.

Table 2 Hasil Uji Validitas dan Reliabilitas Bagian C – Persepsi Kemampuan Pemrograman (N = 30)

Butir	Pernyataan	r hitung	p-value	Keterangan
C1	Saya memahami kode yang dihasilkan AI di mata kuliah yang sudah saya tempuh, bukan hanya menyalinnya	0,6424	0,000	Valid
C2	Setelah menggunakan AI di mata kuliah yang sudah saya tempuh, saya tetap mampu menulis kode serupa secara mandiri	0,8536	0,000	Valid
C3	Penggunaan AI di mata kuliah ini membantu saya memahami konsep pemrograman lebih dalam	0,4359	0,016	Valid
C4	Saya mampu menemukan dan memperbaiki kesalahan (bug) pada kode secara mandiri	0,6981	0,000	Valid
C5	Tanpa bantuan AI, saya merasa mampu menyelesaikan tugas dasar di mata kuliah yang saya pilih ini	0,7857	0,000	Valid
C6	Saya merasa kemampuan pemrograman saya berkembang meskipun menggunakan AI di mata kuliah yang saya pilih ini	0,7932	0,000	Valid
Cronbach's Alpha		0,8806	–	Reliabel

Berdasarkan Tabel 2, seluruh butir pada Bagian C memiliki nilai *r hitung* berkisar antara 0,436 hingga 0,854, yang semuanya berada di atas nilai kritis 0,30 sehingga dinyatakan valid. Nilai Cronbach's Alpha sebesar 0,881 menunjukkan konsistensi internal yang tinggi. Meskipun Cronbach's Alpha Bagian C akan meningkat menjadi 0,897 apabila butir C3 dihapus, butir tersebut tetap dipertahankan karena nilai *r hitung* = 0,436 masih memenuhi kriteria validitas dan butir tersebut merupakan komponen penting dalam mengukur dimensi pemahaman konseptual pemrograman. Dengan demikian, kedua instrumen (Bagian B dan Bagian C) dinyatakan valid dan reliabel sehingga layak digunakan untuk analisis statistik lebih lanjut.

Untuk memberikan gambaran mengenai latar belakang akademik responden, Tabel 3 menyajikan distribusi mata kuliah yang telah ditempuh oleh masing-masing kelompok.

Table 3 Distribusi mata kuliah yang telah ditempuh oleh masing-masing kelompok.

Mata Kuliah	Tingkat Awal	Tingkat Akhir
Pemrograman 1	15 (100%)	15 (100%)
DPBO	15 (100%)	15 (100%)
Struktur Data	15 (100%)	15 (100%)
Pemrograman Perangkat Bergerak	0 (0%)	15 (100%)

Tabel 4 menyajikan rata-rata jawaban responden pada setiap butir Bagian B yang mengukur intensitas penggunaan *vibe coding* berdasarkan kelompok.

Table 4 Rata-rata jawaban responden pada setiap butir Bagian B.

Butir	Pertanyaan	Tingkat Awal	Tingkat Akhir
B1	Menggunakan AI untuk menghasilkan kode secara langsung	3.33	4.27
B2	Menyerahkan pembuatan kode sepenuhnya kepada AI	3.13	4.00
B3	Menggunakan AI sebagai langkah pertama sebelum mengerjakan sendiri	3.80	4.07
B4	Mengumpulkan tugas dari kode AI tanpa modifikasi signifikan	2.27	3.07
B5	Merasa lebih mudah menyelesaikan tugas dengan bantuan AI	4.07	4.40
Total Mean		16.60	19.80

Tabel 5 menyajikan rata-rata jawaban responden pada setiap butir Bagian C yang mengukur persepsi kemampuan pemrograman setelah menggunakan *vibe coding*.

Table 5 Rata-rata jawaban responden pada setiap butir Bagian C.

Butir	Pertanyaan	Tingkat Awal	Tingkat Akhir
C1	Memahami kode yang dihasilkan AI, bukan hanya menyalin	4.20	3.47
C2	Tetap mampu menulis kode serupa secara mandiri	3.73	2.60
C3	AI membantu memahami konsep pemrograman lebih dalam	4.40	4.00
C4	Mampu menemukan dan memperbaiki <i>bug</i> secara mandiri	3.80	2.80
C5	Mampu menyelesaikan tugas dasar tanpa bantuan AI	3.20	2.33
C6	Kemampuan pemrograman saya berkembang meskipun menggunakan AI di mata kuliah yang saya pilih ini	3.87	3.13
Total Mean		23.20	18.33

Tabel 6 menunjukkan skor deskriptif intensitas penggunaan *vibe coding* (Total B, rentang 5–25) dan persepsi kemampuan pemrograman (Total C, rentang 6–30) per kelompok sebagai gambaran keseluruhan distribusi data.

Table 6 Skor deskriptif intensitas penggunaan *vibe coding*.

Statistik	Total B Tingkat Awal	Total B Tingkat Akhir	Total C Tingkat Awal	Total C Tingkat Akhir
Mean	16.60	19.80	23.20	18.33
Median	16	19	22	17
SD	4.08	2.78	4.09	3.83
Minimum	11	15	18	12
Maksimum	25	24	30	26
n	15	15	15	15

Tabel 6 menunjukkan bahwa mahasiswa tingkat akhir menggunakan *vibe coding* dengan intensitas yang lebih tinggi dibandingkan mahasiswa tingkat awal. Ini ditunjukkan oleh rata-rata total B yang lebih besar (Mean=19,80) Di sisi lain, mahasiswa tingkat awal menunjukkan persepsi kemampuan pemrograman yang lebih tinggi melalui rata-rata Total C sebesar 23,20 dibandingkan kelompok tingkat akhir dengan rata-rata 18,33.

Pengujian hipotesis dilakukan menggunakan uji korelasi Rank Spearman dengan taraf signifikansi $\alpha = 0,05$ dan uji dua sisi (*two-tailed*). Derajat kebebasan yang digunakan adalah $df = n - 2 = 13$ untuk masing-masing kelompok. Koefisien korelasi (rs) dihitung menggunakan rata-rata peringkat untuk menangani nilai yang sama (ties). Hipotesis yang diuji adalah $H_0: \rho_s = 0$ (tidak terdapat korelasi yang signifikan) dan $H_1: \rho_s \neq 0$ (terdapat korelasi yang signifikan). Hasil uji korelasi Spearman secara lengkap disajikan pada Tabel 6.

Table 7 Hasil uji korelasi Spearman

Kelompok	rs	p-value	t _{hitung}	t _{tabel}	df	Keterangan	Keputusan
Tingkat Awal	-0.1455	0,551	-0.5304	±2.160	13	Sangat lemah	H ₀ Diterima
Tingkat Akhir	-0.5009	0,057s	-2.0866	±2.160	13	Sedang	H ₀ Diterima

Berdasarkan Tabel 7, pada kelompok mahasiswa tingkat awal diperoleh nilai $rs = -0,1455$ dengan nilai $t_{hitung} = 0,5304 < t_{tabel} = 2,160$ ($df = 13$, $\alpha = 0,05$), sehingga H_0 diterima. Hal ini menunjukkan bahwa tidak terdapat korelasi yang signifikan secara statistik antara intensitas penggunaan *vibe coding* dan persepsi kemampuan pemrograman pada kelompok mahasiswa tingkat awal. Nilai $|rs| = 0,1455$ termasuk ke dalam kategori korelasi sangat lemah [3], dengan arah korelasi negatif.

Pada kelompok mahasiswa tingkat akhir diperoleh nilai $rs = -0,5009$ dengan nilai $t_{hitung} = 2,0866 < t_{tabel} = 2,160$ ($df = 13$, $\alpha = 0,05$), sehingga H_0 diterima. Tidak terdapat korelasi yang signifikan secara statistik antara intensitas penggunaan *vibe coding* dan persepsi kemampuan pemrograman pada kelompok mahasiswa tingkat akhir. Meskipun demikian, nilai $rs = -0,5009$ berada pada kategori korelasi sedang, dengan nilai $t_{hitung} = 2,0866$ yang sangat mendekati $t_{tabel} = 2,160$. Arah korelasi bersifat negatif pada kedua kelompok.

Pengujian hipotesis dilakukan menggunakan uji Mann-Whitney untuk membandingkan skor kedua variabel antara kelompok mahasiswa tingkat awal dan tingkat akhir. Pengujian dilakukan dua kali: Mann-Whitney 1 untuk Total B dan Mann-Whitney 2 untuk Total C. Hipotesis yang diuji pada setiap uji adalah H_0 : tidak terdapat perbedaan yang signifikan antara kedua kelompok, dan H_1 : terdapat perbedaan yang signifikan antara kedua kelompok. Hasil uji disajikan pada Tabel 8.

Table 8 Hasil uji Mann-Whitney perbandingan dua kelompok.

Variabel (Uji)	n_1	n_2	U	p-value	α	Keputusan	Keterangan
Total B	15	15	57.5	0.023	0.05	Ditolak	Signifikan ($p < 0,05$)
Total C	15	15	39.0	0.002	0.05	Ditolak	Sangat Signifikan ($p < 0,01$)

Berdasarkan Tabel 8, pada Mann-Whitney 1 (Total B) diperoleh nilai $U = 57,5$ dengan $p\text{-value} = 0,023$. Karena $p < 0,05$ maka H_0 ditolak. Terdapat perbedaan yang signifikan pada intensitas penggunaan *vibe coding* antara kelompok mahasiswa tingkat awal dan tingkat akhir. Berdasarkan nilai median, kelompok tingkat akhir (Median = 19) memiliki intensitas penggunaan *vibe coding* yang lebih tinggi dibandingkan kelompok tingkat awal (Median = 16). Pada Mann-Whitney 2 (Total C) diperoleh nilai $U = 39,0$ dengan $p\text{-value} = 0,002$. Karena $p < 0,01$ maka H_0 ditolak. Terdapat perbedaan yang sangat signifikan pada persepsi kemampuan pemrograman antara kedua kelompok. Berdasarkan nilai median, kelompok tingkat awal (Median = 22) memiliki persepsi kemampuan pemrograman yang lebih tinggi dibandingkan kelompok tingkat akhir (Median = 17).

Statistik deskriptif skor rubrik validasi dosen pada masing-masing kelompok disajikan pada Tabel 9, sedangkan distribusi kategori skor disajikan pada Tabel 10.

Table 9 Statistik deskriptif skor rubrik validasi dosen per kelompok.

Kelompok	n	Min	Max	Mean	Median	SD	Modus Kategori
Tingkat awal	15	3	8	5.53	5.0	1.88	Cukup
Tingkat akhir	15	3	9	6.13	6.0	1.73	Cukup

Table 10 Distribusi kategori skor rubrik validasi dosen per kelompok.

Kelompok	Baik (7-9)	Cukup(4-6)	Sangat Buruk (3)	Total
Tingkat awal	6	6	3	15
Tingkat akhir	4	10	1	15

Berdasarkan Tabel 9, rata-rata skor rubrik kelompok tingkat awal (Mean = 5,53; Median = 5,0) sedikit lebih rendah dibandingkan kelompok tingkat akhir (Mean = 6,13; Median = 6,0). Berdasarkan Tabel 8, pada kelompok tingkat awal terdapat 6 mahasiswa (40%) dengan kategori Baik dan 3 mahasiswa (20%) dengan kategori Sangat Buruk. Pada kelompok tingkat akhir, mayoritas mahasiswa (10 orang; 66,7%) berada pada kategori Cukup, dengan 4 mahasiswa (26,7%) berkategori Baik dan 1 mahasiswa (6,7%) berkategori Sangat Buruk.

Uji Mann-Whitney dilakukan untuk mengetahui apakah ada perbedaan signifikan padapada kualitas output prompt antara kedua kelompok. Hipotesis yang diujikan adalah H_0 : tidak terdapat perbedaan yang signifikan pada skor rubrik antara kelompok mahasiswa tingkat awal dan kelompok mahasiswa tingkat akhir, dan H_1 : terdapat perbedaan yang signifikan pada skor rubrik antara kedua kelompok. Hasil uji disajikan pada Tabel 11.

Table 11 Hasil uji Mann-Whitney 3 pada skor rubrik validasi dosen.

Variabel (Uji)	n_1	n_2	$\sum R_1$	$\sum R_2$	U	p-value	Keputusan
Skor rubrik	15	15	212.0	253.0	92.0	0.400	Diterima

Berdasarkan Tabel 11, uji Mann-Whitney 3 dilakukan terhadap skor rubrik validasi dosen dengan masing-masing kelompok berjumlah $n_1 = 15$ (tingkat awal) dan $n_2 = 15$ (tingkat akhir). Hasil uji menunjukkan nilai $U = 92,0$ dengan $p\text{-value} = 0,400$. Karena $p = 0,400 > \alpha = 0,05$, maka H_0 diterima, yang berarti tidak terdapat perbedaan yang signifikan secara statistik pada kualitas output prompt antara kelompok mahasiswa tingkat awal dan tingkat akhir. Kesimpulan ini diperkuat oleh besarnya efek ($r = 0,182$), yang menunjukkan bahwa perbedaan antar kelompok tidak hanya tidak signifikan secara statistik, tetapi juga tidak bermakna secara substantif. Hasil ini konsisten dengan gambaran deskriptif pada Tabel 9 dan 10, di mana kedua kelompok berada pada modus kategori "Cukup" dengan rentang skor yang tumpang tindih, sehingga secara keseluruhan kualitas output prompt kedua kelompok dapat dianggap setara.

Triangulasi dilakukan dengan membandingkan skor rubrik validasi dosen dan skor Total C (persepsi kemampuan pemrograman) per responden untuk menguji konsistensi antara persepsi kemampuan yang dilaporkan sendiri oleh mahasiswa dengan kualitas output aktual yang dihasilkan. Data triangulasi disajikan pada Tabel 12.

Table 12 Triangulasi skor rubrik validasi dosen dan skor Total C

Kelompok	rs (Rubrik vs Total C)	p-value	Kesimpulan
Tingkat Awal	-0,01	0.972	Tidak signifikan
Tingkat Akhir	-0,30	0.286	Tidak signifikan

Hasil triangulasi menunjukkan tidak terdapat konsistensi antara skor rubrik dan skor Total C pada kedua kelompok. Pada kelompok tingkat awal, beberapa mahasiswa dengan skor rubrik sangat buruk (skor 3) justru memiliki Total C yang tergolong sedang hingga tinggi. Sebaliknya, pada kelompok tingkat akhir, beberapa mahasiswa dengan skor rubrik baik memiliki Total C yang rendah. Ketidakkonsistenan ini tampak pada beberapa kasus di kelompok tingkat akhir, di mana mahasiswa dengan skor rubrik tertinggi (9) justru memiliki Total C yang tergolong rendah (≤ 15), dan sebaliknya.

Rekapitulasi seluruh hasil uji statistik dan analisis dalam penelitian ini disajikan pada Tabel 13.

Table 13 Rekapitulasi seluruh hasil analisis statistik

Uji/ Analisis	Statistik	Nilai	p-value	Keputusan
Korelasi Spearman (Tingkat Awal)	rs	-0.1455	0.551	Tidak terdapat korelasi signifikan, kekuatan sangat lemah, arah negatif
Korelasi Spearman (Tingkat Akhir)	rs	-0.5009	0.057	Tidak terdapat korelasi signifikan, kekuatan sedang, arah negatif
Mann-Whitney 1 (Total B, Awal vs Akhir)	U	57.5	0.023	Signifikan, tingkat akhir lebih intensif
Mann-Whitney 2 (Total C, Awal vs Akhir)	U	39.0	0.002	Sangat signifikan, tingkat awal lebih tinggi persepsinya

Mann-Whitney 3 (Rubrik, Awal vs Akhir)	U	92.0	0.400	Tidak signifikan, kualitas output setara
Triangulasi (Rubrik vs Total C)	rs	-0,01 / -0,30	0,97 / 0,29	Tidak konsisten, persepsi ≠ kualitas aktual

4. DISCUSSION

Penelitian ini memiliki tujuan untuk mengetahui apakah *vibe coding* memengaruhi kemampuan pemrograman mahasiswa di Program Studi Rekayasa Perangkat Lunak S1 di Telkom University Purwokerto berbeda antara mahasiswa tingkat awal dan mahasiswa tingkat akhir Berdasarkan keseluruhan hasil analisis, ditemukan bahwa kedua kelompok memang menunjukkan perbedaan yang bermakna pada intensitas penggunaan *vibe coding* dan persepsi kemampuan pemrograman. Namun, tidak ada perbedaan yang signifikan dalam kualitas output prompt yang dihasilkan. Hasil ini membentuk gambaran yang kompleks yang layak untuk dikaji lebih mendalam.

Hasil uji Mann-Whitney pada Total B menunjukkan bahwa mahasiswa tingkat akhir menggunakan *vibe coding* dengan intensitas yang secara signifikan lebih tinggi dibandingkan mahasiswa tingkat awal ($U = 57,5$; $p = 0,023$), dengan median 19,0 berbanding 16,0. Pola ini konsisten di seluruh butir Bagian B, dengan perbedaan paling mencolok pada B1 (menggunakan AI untuk menghasilkan kode secara langsung: 4,27 vs 3,33) dan B4 (mengumpulkan tugas tanpa modifikasi signifikan: 3,07 vs 2,27). Temuan ini sejalan dengan penelitian Geng et al. yang menemukan bahwa mahasiswa tingkat lanjut lebih aktif memanfaatkan platform *vibe coding* karena memiliki skema pemrograman yang lebih matang untuk mengarahkan dan mengevaluasi output yang dihasilkan AI. Dalam penelitian ini, mahasiswa tingkat akhir yang telah menempuh seluruh rangkaian mata kuliah pemrograman menunjukkan kenyamanan yang lebih besar dalam mendelegasikan implementasi kode kepada AI. Namun penelitian ini memperluas dan mempersoalkan asumsi Geng et al. jika mahasiswa tingkat lanjut memang lebih mampu berinteraksi secara canggih dengan *vibe coding*, mengapa interaksi yang lebih intensif ini tidak menghasilkan kualitas output yang secara statistik lebih baik, dan justru berkaitan dengan persepsi kemampuan mandiri yang lebih rendah?

Jawaban atas pertanyaan tersebut ditemukan dalam pola penelitian yang paling substantif. Mahasiswa tingkat akhir melaporkan persepsi kemampuan pemrograman yang jauh lebih rendah dibandingkan mahasiswa tingkat awal, meskipun mereka menggunakan *vibe coding* yang lebih intens. ($U = 39,0$; $p = 0,002$; Median Awal = 22 vs Median Akhir = 17). Perbedaan paling tajam terjadi pada butir yang mengukur kemampuan tanpa AI: C2 (menulis kode mandiri: 3,73 vs 2,60), C4 (debugging mandiri: 3,80 vs 2,80), dan C5 (menyelesaikan tugas tanpa AI: 3,20 vs 2,33). Paradoks ini dapat dipahami melalui dua mekanisme yang saling melengkapi. Secara kognitif, ketika AI mengambil alih proses implementasi secara konsisten, mahasiswa kehilangan kesempatan mengalami productive struggle proses berjuang secara mandiri yang diperlukan untuk membentuk skema algoritmik jangka panjang. Dalam kerangka Cognitive Load Theory, pengurangan beban kognitif ekstrinsik yang ditawarkan *vibe coding* tidak secara otomatis dialihkan menjadi germane load yang dibutuhkan untuk membangun kemampuan mandiri, sebagaimana diperingatkan oleh Alam et al. Secara metakognitif, mahasiswa tingkat akhir yang telah menghadapi kompleksitas algoritmik yang lebih tinggi cenderung memiliki kesadaran yang lebih realistis tentang batas kemampuannya, berbeda dengan mahasiswa tingkat awal yang kepercayaan dirinya masih relatif tinggi karena belum terekspos pada kegagalan-kegagalan algoritmik yang kompleks. Temuan ini memperluas argumen Prather et al. tentang kesenjangan yang melebar akibat AI generatif: dalam penelitian ini, kesenjangan tersebut terjadi tidak hanya di antara individu dalam satu kelompok, tetapi juga secara sistematis antara jenjang mahasiswa yang berbeda.

Uji korelasi Spearman memperkuat pola di atas. Meskipun tidak ada korelasi yang signifikan secara statistik pada kedua kelompok, kelompok tingkat akhir menunjukkan tendensi hubungan negatif dengan kekuatan sedang ($rs = -0,5009$; $|t \text{ hitung}| = 2,0866$ yang sangat mendekati $t \text{ tabel} = 2,160$). Arah negatif yang konsisten pada kedua kelompok mendukung argumen bahwa semakin tinggi intensitas penggunaan *vibe coding*, semakin rendah persepsi kemampuan pemrograman mandiri mahasiswa sebuah pola yang belum mencapai ambang signifikansi statistik pada ukuran sampel ini, namun secara substantif bermakna dan layak diuji pada replikasi dengan sampel yang lebih besar.

Berbeda dengan temuan pada persepsi, hasil uji Mann-Whitney pada skor rubrik tidak menunjukkan perbedaan yang signifikan antara kedua kelompok ($U = 92,0$; $p = 0,400$), dengan keduanya berada pada modus kategori "Cukup." Ketidakberbedaan ini bukan berarti *vibe coding* menghasilkan output yang baik pada kedua kelompok, melainkan bahwa keduanya sama-sama belum menggunakannya secara efektif. Catatan evaluatif dosen validator mengungkap permasalahan yang bersifat fundamental dan lintas kelompok dengan sebagian besar mahasiswa belum mampu menyusun prompt yang kontekstual dan spesifik terhadap ADT yang diberikan; sejumlah output hanya mencakup operasi enqueue tanpa dequeue; beberapa output dihasilkan dalam bentuk kode siap jalan alih-alih representasi algoritma yang diminta; inkonsistensi penggunaan parameter address mengindikasikan pemahaman ADT yang belum utuh dan penamaan fungsi yang berlebihan menunjukkan mahasiswa tidak memahami batasan cakupan yang diminta. Pola ini sejalan dengan temuan Lucchetti et al. yang menegaskan bahwa kegagalan dalam menggunakan LLM untuk tugas pembuatan kode bukan karena kurangnya kosakata teknis, melainkan karena tidak memahami informasi kontekstual apa yang dibutuhkan AI untuk menghasilkan kode yang tepat. Penelitian ini memperluas temuan tersebut dengan menunjukkan bahwa keterbatasan ini tidak hanya berlaku pada mahasiswa pemula, tetapi juga pada mahasiswa tingkat akhir mengisyaratkan bahwa *prompting literacy* dalam konteks tugas algoritmik formal merupakan keterampilan yang perlu diajarkan secara eksplisit, bukan sesuatu yang muncul otomatis dari pengalaman pemrograman umum.

Hasil triangulasi memperkuat dan menyempurnakan temuan-temuan sebelumnya. Tidak ditemukan konsistensi yang signifikan antara persepsi kemampuan mahasiswa (Total C) dan kualitas output aktual mereka (skor rubrik) pada kedua kelompok baik pada kelompok mahasiswa tingkat awal ($rs = -0,01$; $p = 0,972$) maupun mahasiswa tingkat akhir ($rs = -0,30$; $p = 0,286$). Ini berarti mahasiswa yang merasa sangat mampu tidak selalu menghasilkan prompt berkualitas baik, dan sebaliknya. Fenomena ini mengindikasikan adanya dua kondisi yang berlawanan sekaligus yaitu *performance without competence* pada sebagian mahasiswa mampu menghasilkan output berkualitas melalui AI tanpa memiliki kemampuan pemrograman mandiri yang mendasarinya dan *confidence without output quality* pada sebagian yang lain. Kondisi ini merupakan bukti empiris dalam *Cognitive Load Theory* bahwa *vibe coding* yang digunakan tanpa panduan instruksional yang terstruktur menciptakan ilusi kompetensi. AI menghilangkan hambatan nyata yang diperlukan untuk membangun kemampuan jangka panjang, sehingga proses belajar berlangsung tanpa skema yang memadai. Temuan ini memperkuat dan memperluas argumen Prather et al. tentang kesulitan metakognitif mahasiswa pemula dalam memverifikasi output AI. Penelitian ini menunjukkan bahwa kesulitan tersebut tidak terbatas pada verifikasi output, tetapi meluas pada penilaian diri terhadap kemampuan pemrograman secara menyeluruh, dan bahwa mahasiswa tingkat akhir pun tidak sepenuhnya terbebas dari masalah ini.

Secara keseluruhan, pola temuan ini memiliki dua implikasi praktis yang saling berkaitan. Pertama, penggunaan *vibe coding* dalam pembelajaran pemrograman memerlukan panduan instruksional yang terstruktur agar tidak memperlebar kesenjangan antara persepsi dan kompetensi aktual misalnya dengan mewajibkan mahasiswa untuk menjelaskan, memodifikasi, dan membuktikan pemahamannya terhadap kode yang dihasilkan AI sebelum mengumpulkan tugas. Kedua, *prompting literacy* perlu diintegrasikan secara eksplisit ke dalam kurikulum sebagai keterampilan tersendiri, karena pengalaman pemrograman yang lebih luas terbukti tidak secara otomatis menghasilkan kemampuan *prompting* yang lebih efektif dalam konteks tugas algoritmik formal. Penelitian ini memiliki beberapa keterbatasan yang perlu dipertimbangkan dalam menginterpretasikan temuan. Ukuran sampel yang terbatas ($n = 15$ per kelompok) mengurangi kekuatan statistik analisis terutama pada uji Spearman kelompok tingkat akhir yang menghasilkan nilai $|t \text{ hitung}|$ sangat mendekati batas signifikansi, sehingga penambahan sampel berpotensi menghasilkan korelasi yang signifikan dengan kekuatan korelasi yang sama. Studi kasus hanya mencakup satu topik dari satu mata kuliah, validasi rubrik dilakukan oleh satu penilai, dan data kuesioner bersifat *self-report* yang rentan terhadap bias persepsi meskipun bias persepsi itu sendiri menjadi bagian dari temuan substantif penelitian ini. Penelitian selanjutnya disarankan untuk memperluas ukuran sampel, mencakup topik dari beberapa mata kuliah dengan tingkat kesulitan algoritmik yang berbeda, menggunakan dua penilai independen untuk meningkatkan reliabilitas rubrik, serta mempertimbangkan tes kemampuan pemrograman terstandar sebagai pelengkap data persepsi yang lebih objektif.

5. CONCLUSION

Penelitian ini berhasil menjawab ketiga tujuan yang telah ditetapkan. Pertama, mahasiswa tingkat akhir terbukti menggunakan *vibe coding* dengan intensitas yang lebih tinggi secara signifikan dibandingkan mahasiswa tingkat awal. Kedua, tidak ada korelasi statistik yang signifikan antara intensitas penggunaan *vibe coding* dan persepsi kemampuan pemrograman pada kedua kelompok, tetapi kelompok tingkat akhir menunjukkan tendensi negatif terhadap kekuatan. Ketiga, meskipun terdapat perbedaan signifikan pada intensitas penggunaan dan persepsi kemampuan antar kelompok, tidak terdapat perbedaan yang signifikan pada kualitas output prompt yang dihasilkan, dengan kedua kelompok berada pada kategori "Cukup" berdasarkan validasi dosen.

Temuan penelitian ini memberikan kontribusi terhadap perkembangan literatur tentang dampak *vibe coding* dalam pendidikan pemrograman, khususnya di konteks perguruan tinggi Indonesia. Penelitian ini menjadi yang mengkaji perbedaan pengaruh *vibe coding* berdasarkan jenjang studi menggunakan pendekatan *mixed-methods* yang menggabungkan kuesioner persepsi dan validasi studi kasus algoritmik. Hasil triangulasi yang menunjukkan ketidakkonsistenan antara persepsi kemampuan mahasiswa dan kualitas *output* aktual mengisyaratkan bahwa penggunaan *vibe coding* tanpa panduan yang memadai berpotensi membentuk kesenjangan antara kepercayaan diri dan kompetensi pemrograman yang nyata sebuah temuan yang relevan bagi perancangan kurikulum dan strategi pembelajaran berbasis AI.

Secara praktis, penelitian ini menyarankan agar program studi Rekayasa Perangkat Lunak mempertimbangkan pembuatan pedoman penggunaan *vibe coding* yang terstruktur, yang mencakup penjelasan tentang literasi perintah sebagai bagian dari kurikulum pemrograman. Untuk mendapatkan gambaran yang lebih mendalam dan dapat digeneralisasi, penelitian lebih lanjut harus dilakukan dengan ukuran sampel yang lebih besar. Ini akan mencakup studi kasus dari berbagai mata kuliah dengan fitur yang berbeda, serta penggunaan alat pengukuran kemampuan pemrograman yang terstandar sebagai pelengkap data persepsi.

6. AI TOOLS

Dalam proses penulisan penelitian ini, beberapa artificial intelligence tools digunakan sebagai alat bantu pada tahap tertentu. Pada bagian Discussion Claude membantu dalam menelusuri kerangka teoretis yang relevan pada bagian Diskusi, khususnya dalam mengaitkan hasil penelitian dengan teori tekanan kognitif dan literatur tentang pembelajaran yang dibantu AI, khususnya dalam mengaitkan temuan penelitian dengan *Cognitive Load Theory* dan literatur terkait *AI-assisted learning*, serta membantu merumuskan struktur argumentasi secara logis dan koheren.

7. REFERENCE

- [1] A. Sarkar and I. Drosos, "Vibe coding: Programming through conversation with artificial intelligence," in Proc. 36th Annu. Conf. Psychol. Program. Interest Group (PPIG 2025), Sep. 2025, doi: 10.48550/arXiv.2506.23253.
- [2] H.-F. Chang, M. Shokrolah, S. E. S. Witchger, L. Cao, and S. Koolmanojwong Mobasser, "Coding with AI: From a reflection on industrial practices to future computer science and software engineering education," arXiv preprint arXiv:2512.23982, Dec. 2025.
- [3] M. Kazemitabaar, J. Chow, C. K. T. Ma, B. J. Ericson, D. Weintrop, and T. Grossman, "Studying the effect of AI code generators on supporting novice learners in introductory programming," in Proc. 2023 CHI Conf. Human Factors Comput. Syst. (CHI '23), Hamburg, Germany, Apr. 2023, doi: 10.1145/3544548.3580919.
- [4] P. Denny et al., "Computing education in the era of generative AI," Commun. ACM, vol. 67, no. 2, pp. 56–67, Feb. 2024, doi: 10.1145/3624720.
- [5] E. Kasneci et al., "ChatGPT for good? On opportunities and challenges of large language models for education," Learn. Individ. Differ., vol. 103, p. 102274, Apr. 2023, doi: 10.1016/j.lindif.2023.102274.
- [6] J. Prather et al., "It's weird that it knows what I want: Usability and interactions with Copilot for novice programmers," *ACM Transactions on Computer-Human Interaction*, vol. 31, no. 1, Article 4, Nov. 2023, doi: 10.1145/3617367.

- [7] I. Samsyudin, “Vibe coding and AI-led conversational programming: Emerging trends in software development,” *SSRN Electron. J.*, Aug. 2025, doi: 10.2139/SSRN.5469367.
- [8] J. Prather et al., “The widening gap: The benefits and harms of generative AI for novice programmers,” in *Proc. ACM Conf. Int. Comput. Educ. Res. (ICER '24)*, Melbourne, Australia, Aug. 2024, pp. 469–486, doi: 10.1145/3632620.3671116.
- [9] F. Lucchetti, Z. Wu, A. Guha, M. Q. Feldman, and C. J. Anderson, “Substance beats style: Why beginning students fail to code with LLMs,” in *Proc. 2025 Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol. (NAACL)*, Albuquerque, NM, USA, Apr. 2025, pp. 8541–8610, doi: 10.18653/v1/2025.naacl-long.433.
- [10] F. Geng et al., “Exploring student-AI interactions in vibe coding,” *arXiv preprint arXiv:2507.22614*, Jul. 2025.
- [11] N. V. Ivankova, J. W. Creswell, and S. L. Stick, “Using mixed-methods sequential explanatory design: From theory to practice,” *Field Methods*, vol. 18, no. 1, pp. 3–20, Feb. 2006, doi: 10.1177/1525822X05282260.
- [12] M. N. Alam, M. A. Islam, M. O. A. Babiker, M. S. Siddiqui, M. Bin Amin, and J. Oláh, “AI-assisted learning tools and student learning outcomes: A cognitive load theory perspective,” *Comput. Hum. Behav. Rep.*, vol. 21, p. 100986, Mar. 2026, doi: 10.1016/j.chbr.2026.100986.
- [13] G. J. Bhattacharjee, *Social Science Research: Principles, Methods, and Practices*, 2nd ed. Tampa, FL, USA: University of South Florida, 2012. [Online]. Available: https://scholarcommons.usf.edu/oa_textbooks/3. Accessed: Jun. 2026.
- [14] M. A. Bujang, E. D. Omar, D. H. P. Foo, and Y. K. Hon, “Sample size determination for conducting a pilot study to assess reliability of a questionnaire,” *Restor. Dent. Endod.*, vol. 49, no. 1, p. e3, 2024, doi: 10.5395/rde.2024.49.e3.
- [15] S. S. Sullivan and A. R. Artino Jr., “Analyzing and interpreting data from Likert-type scales,” *J. Grad. Med. Educ.*, vol. 5, no. 4, pp. 541–542, Dec. 2013, doi: 10.4300/JGME-5-4-18.
- [16] P. Denny, J. Leinonen, J. Prather, A. Luxton-Reilly, T. Amarouche, B. A. Becker, and B. N. Reeves, “Prompt problems: A new programming exercise for the generative AI era,” in *Proc. 55th ACM Tech. Symp. Comput. Sci. Educ. (SIGCSE)*, Portland, OR, USA, 2024, pp. 296–302, doi: 10.1145/3626252.3630909.
- [17] A. Pathak et al., “Rubric is all you need: Improving LLM-based code evaluation with question-specific rubrics,” in *Proc. ACM Int. Conf. Comput. Educ. Res. (ICER)*, 2025, doi: 10.1145/3702652.3744220.
- [18] D. F. Polit and C. T. Beck, *Nursing Research: Generating and Assessing Evidence for Nursing Practice*, 10th ed. Philadelphia, PA, USA: Wolters Kluwer, 2017.
- [19] K. S. Taber, “The use of Cronbach’s alpha when developing and reporting research instruments in science education,” *Res. Sci. Educ.*, vol. 48, no. 6, pp. 1273–1296, Dec. 2018, doi: 10.1007/s11165-016-9602-2.

8. AUTHORS

Novita Syahwa Tri Hapsari adalah mahasiswa Program Studi S1 Rekayasa Perangkat Lunak, Fakultas Informatika, Telkom University Purwokerto. Bidang minat penelitiannya meliputi kecerdasan buatan dalam pendidikan, pembelajaran pemrograman, rekayasa perangkat lunak, serta pemanfaatan Generative AI dalam pengembangan perangkat lunak. Email: novitasyahwa@student.telkomuniversity.ac.id.

Rifki Adhitama adalah Dosen Program Studi S1 Rekayasa Perangkat Lunak, Fakultas Informatika, Telkom University Purwokerto. Bidang penelitian dan pengembangan yang menjadi minatnya meliputi Rekayasa Perangkat Lunak, Software Engineering, Topic Modelling, Data Mining, and Intelligent Information Systems. Email: rifkiadhitama@telkomuniversity.ac.id.

Yudha Islami Sulistya adalah Dosen pada Program Studi Rekayasa Perangkat Lunak, Fakultas Informatika, Telkom University Purwokerto. Bidang penelitian dan pengembangan yang diminati meliputi Machine Learning, Deep Learning, metode Ensemble Learning, teknik reduksi dimensi, serta penerapan kecerdasan buatan dalam sistem klasifikasi dan prediksi. Email: yudhaislamisulistya@telkomuniversity.ac.id.