

Model Hybrid Project-Based Learning dan AI-Assisted Coding untuk Kompetensi Pemrograman Mahasiswa

Nurita Pratiwi^{1*}, Sulyanah Masithoh², Nuryani Cahyani³

^{1,2}Pendidikan Bahasa Indonesia, STKIP Jaya Pura, Jaya Pura, Indonesia

³Teknologi Pendidikan, Universitas Dharma Kencana, Jambi, Indonesia

Email: ^{1*}nuritapратиwi@stikipjp.ac.id, ²masithoh-yanah@stikipjp.ac.id, ³cahyaniyani@udk.ac.id

Email Penulis Korespondensi: ^{1*}nuritapратиwi@stikipjp.ac.id

Abstrak—Kompetensi pemrograman mahasiswa mencakup pemahaman konsep, perancangan algoritme, implementasi kode, pengujian, debugging, dan kemampuan menjelaskan solusi. Kehadiran Generative AI dan AI-Assisted Coding memberikan peluang dukungan adaptif dalam proses pemrograman, tetapi penggunaan yang tidak terstruktur dapat mendorong penerimaan kode secara pasif tanpa pemahaman yang memadai. Penelitian ini bertujuan menguji efektivitas model hybrid Project-Based Learning (PjBL) dan AI-Assisted Coding dalam meningkatkan kompetensi pemrograman mahasiswa. Penelitian menggunakan pendekatan kuantitatif dengan desain quasi-experimental pretest-posttest control group. Sebanyak 132 mahasiswa terlibat, terdiri atas kelompok eksperimen ($n = 66$) dan kelompok kontrol ($n = 66$). Model hybrid terdiri atas lima tahap, yaitu analisis masalah, desain proyek, AI-assisted coding, pengujian dan debugging, serta refleksi. Data dikumpulkan melalui tes kompetensi pemrograman, tugas kinerja, dan lembar validasi model, kemudian dianalisis menggunakan statistik deskriptif, uji t berpasangan, uji t independen, N -gain, dan Cohen's d . Hasil menunjukkan bahwa rata-rata kelompok eksperimen meningkat dari 60,89 menjadi 82,39, sedangkan kelompok kontrol meningkat dari 60,77 menjadi 70,20. Perbedaan posttest antarkelompok signifikan ($t = 7,229$; $p < 0,001$) dengan ukuran efek besar ($d = 1,258$). Temuan menunjukkan bahwa penggunaan AI sebagai scaffolding untuk menjelaskan, mengevaluasi, menguji, dan memperbaiki kode dapat mendukung peningkatan kompetensi pemrograman ketika mahasiswa tetap bertanggung jawab atas analisis masalah, keputusan teknis, dan refleksi.

Kata Kunci: AI-assisted coding, Kompetensi Pemrograman, Mahasiswa, *Project-Based Learning*, Pendidikan Pemrograman

Abstract— Programming competencies includes conceptual understanding, algorithm design, code implementation, testing, debugging, and the ability to explain solutions. The emergence of generative AI and AI-assisted coding creates opportunities for adaptive support during programming, but unstructured use may encourage students to accept generated code without sufficient understanding. This study aims to examine the effectiveness of a hybrid Project-Based Learning (PjBL) and AI-assisted coding model in improving university students' programming competence. A quantitative quasi-experimental pretest-posttest control group design was used. A total of 132 students participated, comprising an experimental group ($n = 66$) and a control group ($n = 66$). The hybrid model consisted of five stages: problem analysis, project design, AI-assisted coding, testing and debugging, and reflection. Data were collected through a programming competence test, performance task, and model validation sheet, and were analyzed using descriptive statistics, paired-samples t -test, independent-samples t -test, normalized gain, and Cohen's d . The experimental group increased from 60.89 to 82.39, whereas the control group increased from 60.77 to 70.20. The posttest difference was significant ($t = 7.229$; $p < 0.001$), with a large effect size ($d = 1.258$). The findings indicate that positioning AI as scaffolding for explaining, evaluating, testing, and improving code can support programming competence when students remain responsible for problem analysis, technical decisions, and reflection.

Keywords: AI-assisted coding, programming competence, Project-Based Learning, programming education, university students

1. PENDAHULUAN

Pemrograman merupakan kompetensi penting dalam pendidikan tinggi bidang komputasi karena pembelajarannya menuntut mahasiswa menghubungkan pemahaman konsep dengan pemecahan masalah, perancangan algoritme, implementasi, pengujian, dan perbaikan program. Pembelajaran pemrograman yang berorientasi pada praktik karena itu perlu menyediakan aktivitas yang memungkinkan mahasiswa membangun solusi secara bertahap, memeriksa hasilnya, dan menjelaskan alasan di balik keputusan teknis. Pendekatan pembelajaran berbasis proyek relevan dengan kebutuhan tersebut

karena mahasiswa belajar melalui penyelesaian masalah dan pengembangan produk yang menuntut perencanaan, implementasi, pengujian, serta refleksi [1], [2], [3].

Project-Based Learning (PjBL) memberikan struktur belajar yang menghubungkan konsep dengan tugas autentik. Dalam konteks pendidikan pemrograman, proyek dapat membantu mahasiswa mempertahankan pengetahuan pemrograman dan mentransfer keterampilan ke penyelesaian masalah yang lebih kontekstual [1]. Meta-analisis [2] juga menunjukkan bahwa PjBL secara umum memberikan dampak positif terhadap hasil belajar, prestasi akademik, sikap, dan keterampilan berpikir. Meskipun demikian, pembelajaran berbasis proyek dapat menghadirkan kompleksitas tugas, perbedaan kemampuan awal, dan kebutuhan umpan balik selama proses pengembangan. Tantangan tersebut membuka ruang bagi pemanfaatan teknologi yang dapat memberikan dukungan tanpa mengambil alih tanggung jawab belajar mahasiswa.

Perkembangan Generative AI menghadirkan bentuk dukungan baru dalam pembelajaran pemrograman. Sistem berbasis AI dapat membantu memberikan penjelasan, contoh, alternatif solusi, dan umpan balik terhadap kode. Kajian tentang Generative AI dalam pendidikan menunjukkan adanya peluang untuk personalisasi, dukungan belajar, dan pemberian umpan balik, tetapi sekaligus terdapat risiko terkait akurasi keluaran, ketergantungan, integritas akademik, dan berkurangnya keterlibatan kognitif apabila teknologi digunakan tanpa desain pedagogis yang jelas [4], [5], [6], [7]. Karena itu, pemanfaatan AI dalam pembelajaran pemrograman perlu diarahkan agar mahasiswa tetap memahami, mengevaluasi, dan mempertanggungjawabkan kode yang digunakan.

Dalam konteks pemrograman, bukti empiris menunjukkan bahwa manfaat AI tidak bersifat otomatis dan sangat dipengaruhi oleh cara teknologi tersebut diintegrasikan ke dalam aktivitas belajar. [8] menemukan bahwa penggunaan alat Generative AI dalam pembelajaran pemrograman dapat mendukung computational thinking, self-efficacy pemrograman, dan motivasi. [9] melalui studi kuasi-eksperimental menunjukkan bahwa pembelajaran pemrograman berbantuan ChatGPT memengaruhi perilaku dan pengalaman belajar mahasiswa, meskipun hasil performa perlu dipahami bersama pola penggunaan AI. [10] juga menemukan bahwa mahasiswa menggunakan chatbot AI untuk pemeriksaan kesalahan, debugging, penjelasan kode, dan pengembangan solusi, tetapi penggunaan tersebut dapat disertai kekhawatiran terhadap kualitas kode dan proses belajar. Temuan [11] pada pembelajaran pemrograman bagi pemula juga menegaskan bahwa penggunaan ChatGPT perlu dipahami sebagai bagian dari perubahan praktik pembelajaran, bukan sekadar sebagai alat pembangkit kode. Temuan tersebut menunjukkan bahwa AI lebih tepat diposisikan sebagai alat pendukung dalam suatu desain pembelajaran yang mengharuskan mahasiswa melakukan verifikasi dan penalaran teknis.

Model pembelajaran yang mengintegrasikan PjBL dan AI-Assisted Coding secara terstruktur berpotensi menjawab kebutuhan tersebut. PjBL dapat menyediakan konteks proyek, tahapan penyelesaian masalah, dan produk yang harus dipertanggungjawabkan, sedangkan AI dapat berfungsi sebagai scaffolding ketika mahasiswa memerlukan bantuan dalam memahami alternatif implementasi, menelusuri kesalahan, atau memperoleh umpan balik. Penelitian [12] menunjukkan bahwa sistem tutor berbasis ChatGPT dapat meningkatkan hasil belajar pemrograman, terutama pada mahasiswa dengan fondasi awal yang lebih lemah, tetapi juga mengingatkan adanya risiko ketergantungan. Temuan tersebut memperkuat pentingnya desain yang menempatkan AI sebagai pendukung, bukan sebagai pengganti proses berpikir dan pemrograman mahasiswa.

Berdasarkan state of the art tersebut, research gap penelitian ini terletak pada keterbatasan model pembelajaran yang secara eksplisit mengintegrasikan sintaks PjBL dengan siklus penggunaan AI-Assisted Coding yang mempertahankan tanggung jawab teknis mahasiswa pada setiap tahap pembelajaran. Penelitian terdahulu telah mengkaji PjBL dalam pendidikan pemrograman dan telah menguji penggunaan ChatGPT atau alat AI dalam pembelajaran pemrograman, tetapi integrasi keduanya dalam satu alur pembelajaran yang mencakup analisis masalah, desain proyek, penggunaan AI untuk mendukung coding, pengujian dan debugging, serta refleksi masih memerlukan pengujian empiris lebih lanjut. Dengan demikian, penelitian ini mengembangkan fokus pada hubungan antara struktur proyek, penggunaan AI sebagai scaffolding, dan kompetensi pemrograman yang diukur melalui tes serta tugas kinerja.

Kebaruan penelitian ini terletak pada perancangan alur hybrid yang menempatkan AI-Assisted Coding di dalam lima tahap pembelajaran, yaitu analisis masalah, desain proyek, AI-assisted coding, pengujian dan debugging, serta refleksi. Pada alur tersebut, mahasiswa tetap menjadi pengambil keputusan utama: mereka harus merumuskan masalah dan rancangan sebelum menggunakan AI, mengevaluasi keluaran AI sebelum digunakan, memverifikasi program melalui pengujian, dan menjelaskan keputusan teknis pada tahap refleksi. Dengan desain tersebut, penggunaan AI tidak diperlakukan sebagai intervensi yang berdiri sendiri, melainkan sebagai bagian dari strategi pembelajaran berbasis proyek yang diarahkan pada penguasaan kompetensi pemrograman.

Penelitian ini bertujuan menguji efektivitas model hybrid Project-Based Learning dan AI-Assisted Coding dalam meningkatkan kompetensi pemrograman mahasiswa. Hipotesis penelitian adalah: H1, terdapat perbedaan kompetensi pemrograman yang signifikan antara mahasiswa yang mengikuti model hybrid dan kelompok pembanding; dan H2, peningkatan kompetensi pemrograman kelompok yang mengikuti model hybrid lebih tinggi daripada kelompok pembanding.

Penelitian menggunakan pendekatan kuantitatif dengan desain quasi-experimental pretest-posttest control group. Kelompok eksperimen mengikuti model hybrid Project-Based Learning dan AI-Assisted Coding, sedangkan kelompok kontrol mengikuti pembelajaran pembandingan dengan materi, target capaian, dan alokasi waktu yang setara. Pengukuran dilakukan sebelum dan sesudah pembelajaran untuk membandingkan perubahan kompetensi pemrograman pada kedua kelompok.

2. METODOLOGI PENELITIAN

2.1 Desain Penelitian

Penelitian menggunakan pendekatan kuantitatif dengan desain quasi-experimental pretest-posttest control group. Kelompok eksperimen mengikuti model *hybrid Project-Based Learning* dan AI-Assisted Coding, sedangkan kelompok kontrol mengikuti pembelajaran pembandingan dengan materi, target capaian, dan alokasi waktu yang setara.

2.2 Subjek Penelitian

Subjek penelitian terdiri atas 132 mahasiswa aktif, dengan 66 mahasiswa pada kelompok eksperimen dan 66 mahasiswa pada kelompok kontrol. Karakteristik responden berdasarkan kelompok, jenis kelamin, semester, dan program studi disajikan pada Tabel 1. Naskah sumber belum menjelaskan secara rinci lokasi penelitian, mata kuliah, teknik penentuan sampel, serta prosedur penempatan mahasiswa ke dalam kelompok; informasi tersebut diperlukan untuk memastikan keterlacakan desain dan keterbatasan generalisasi hasil.

Tabel 1. Karakteristik Responden

Karakteristik	Kategori	Frekuensi	Persentase (%)
Kelompok	Eksperimen	66	50,0
	Kontrol	66	50,0
Jenis kelamin	Laki-laki	61	46,2
	Perempuan	71	53,8
Semester	2	42	31,8
	4	57	43,2
	6	33	25,0
Program studi	Informatika/Komputer	74	56,1
	Pendidikan terkait TIK	32	24,2
	Lainnya	26	19,7
Total		132	100,0

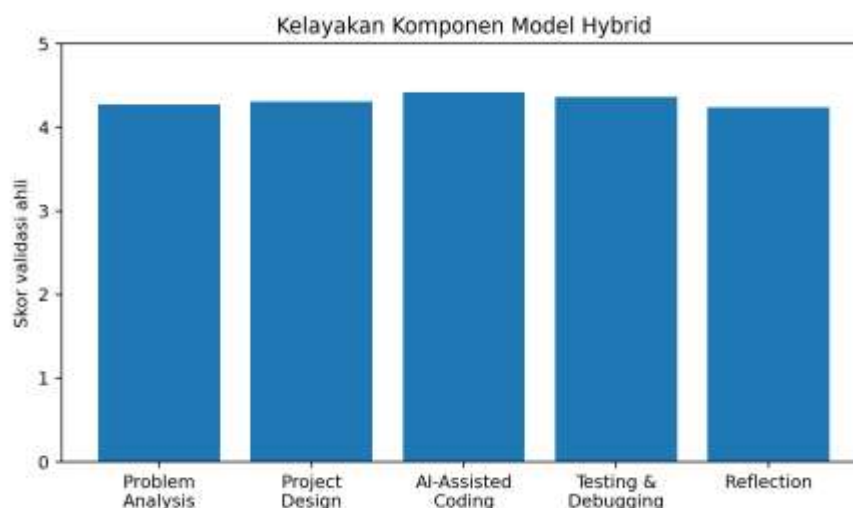
2.3 Sintaks Model Hybrid

Model hybrid menggabungkan prinsip pembelajaran berbasis proyek dengan penggunaan AI-Assisted Coding secara terstruktur. AI ditempatkan sebagai scaffolding yang digunakan mahasiswa untuk mengeksplorasi alternatif, memperoleh penjelasan, membandingkan rancangan, menelusuri kemungkinan kesalahan, dan memperoleh umpan balik. Mahasiswa tetap bertanggung jawab terhadap analisis masalah, perancangan algoritme, keputusan implementasi, pengujian, debugging, dan refleksi. Lima tahap model disajikan pada Tabel 2.

Tabel 2. Tahapan Model Hybrid Project-Based Learning dan AI-Assisted Coding

Tahap	Aktivitas Mahasiswa	Peran AI-Assisted Coding	Kompetensi yang Dikembangkan
1. Analisis masalah	Menganalisis kebutuhan, batasan, dan spesifikasi proyek.	Membantu mengeksplorasi alternatif pendekatan.	Analisis masalah dan dekomposisi.
2. Desain proyek	Merancang algoritme, struktur program, dan pembagian modul.	Membantu memberikan contoh atau membandingkan rancangan.	Perancangan algoritme dan arsitektur.
3. AI-Assisted Coding	Mengimplementasikan kode dan meminta penjelasan atas alternatif solusi.	Memberikan contoh, penjelasan, dan umpan balik kode.	Implementasi dan pemahaman sintaks.

Tahap	Aktivitas Mahasiswa	Peran AI-Assisted Coding	Kompetensi yang Dikembangkan
4. Pengujian dan debugging	Membuat test case, menemukan kesalahan, dan memperbaiki program.	Membantu menelusuri kemungkinan penyebab kesalahan.	Testing, debugging, dan evaluasi.
5. Refleksi	Menjelaskan keputusan teknis dan mengevaluasi proses penggunaan AI.	Memicu pertanyaan reflektif.	Metakognisi dan tanggung jawab teknis.



Gambar 1. Hasil Validasi Kelayakan Komponen Model Hybrid

2.4 Instrumen Penelitian

Kompetensi pemrograman diukur menggunakan tes dan tugas kinerja yang mencakup pemahaman konsep, perancangan algoritme, implementasi kode, ketepatan fungsi program, debugging, dan kemampuan menjelaskan solusi. Instrumen tes terdiri atas 25 butir dan dilengkapi tugas kinerja. Hasil pengujian menunjukkan korelasi butir sebesar 0,468–0,817 dan Cronbach's alpha sebesar 0,914. Validasi isi dilaporkan memperoleh skor ahli 4,34 dari skala 5. Naskah sumber belum menjelaskan jumlah validator, karakteristik validator, prosedur validasi, jenis korelasi yang digunakan untuk validitas butir, serta rubrik dan rentang skor tugas kinerja; rincian tersebut perlu dilengkapi untuk menjamin reproduktibilitas instrumen.

Tabel 3. Kualitas Instrumen Kompetensi Pemrograman

Aspek	Hasil	Kriteria	Keputusan
Validitas butir	$r = 0,468-0,817$	$r > 0,30$	Valid
Reliabilitas	Cronbach's alpha = 0,914	$\alpha > 0,70$	Reliabel
Validasi isi	Skor ahli = 4,34/5	$\geq 4,00$	Sangat baik

2.5 Teknik Analisis Data

Data dianalisis menggunakan statistik deskriptif, uji normalitas, uji homogenitas, paired-samples t-test, independent-samples t-test, perhitungan normalized gain (N-gain), dan Cohen's d dengan tingkat signifikansi $\alpha = 0,05$. Uji pretest antarkelompok digunakan untuk memeriksa kesetaraan kemampuan awal, sedangkan uji posttest antarkelompok digunakan untuk menguji perbedaan hasil akhir. N-gain digunakan untuk menggambarkan peningkatan relatif skor dari pretest ke posttest, sedangkan Cohen's d digunakan untuk menginterpretasikan besar efek perbedaan posttest. Naskah sumber belum menampilkan hasil uji normalitas dan homogenitas sehingga hasil pengujian asumsi perlu ditambahkan sebelum naskah dinyatakan final.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Validasi Model

Tabel 4. Hasil Validasi Model Hybrid

Komponen	Skor Rata-rata	Kategori
Analisis masalah	4,27	Sangat baik
Desain proyek	4,31	Sangat baik
AI-Assisted Coding	4,42	Sangat baik
Pengujian dan debugging	4,36	Sangat baik
Refleksi	4,24	Sangat baik
Rata-rata	4,32	Sangat baik

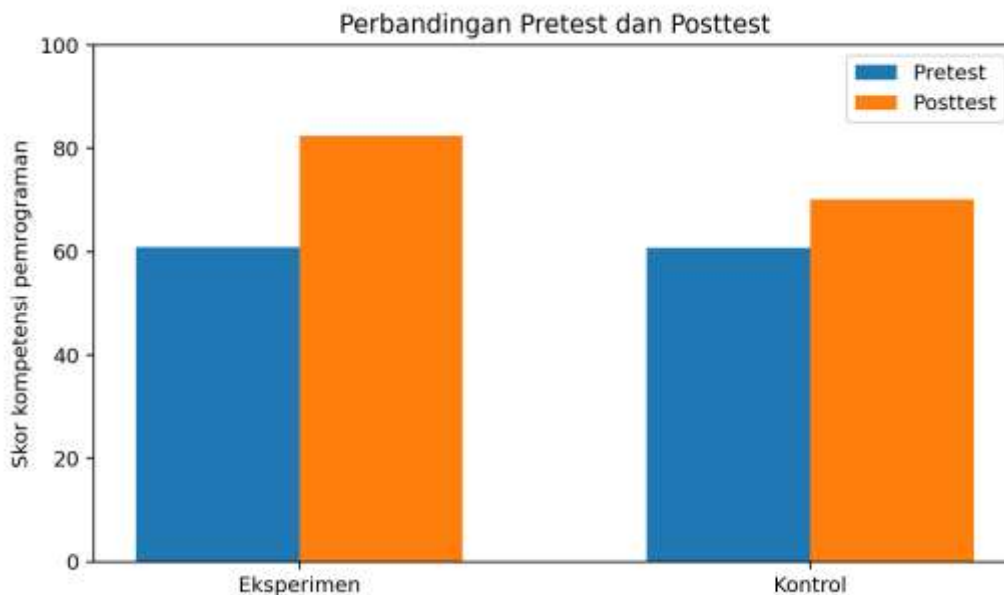
Hasil validasi menunjukkan bahwa seluruh komponen model memperoleh kategori sangat baik, dengan skor rata-rata keseluruhan 4,32. Komponen AI-Assisted Coding memperoleh skor tertinggi (4,42), diikuti pengujian dan debugging (4,36), desain proyek (4,31), analisis masalah (4,27), dan refleksi (4,24). Temuan ini menunjukkan bahwa struktur model dinilai layak oleh validator berdasarkan data validasi yang dilaporkan. Namun, interpretasi hasil validasi perlu dilengkapi dengan informasi jumlah dan karakteristik validator serta prosedur penilaian.

3.2 Perbandingan Kompetensi Pemrograman

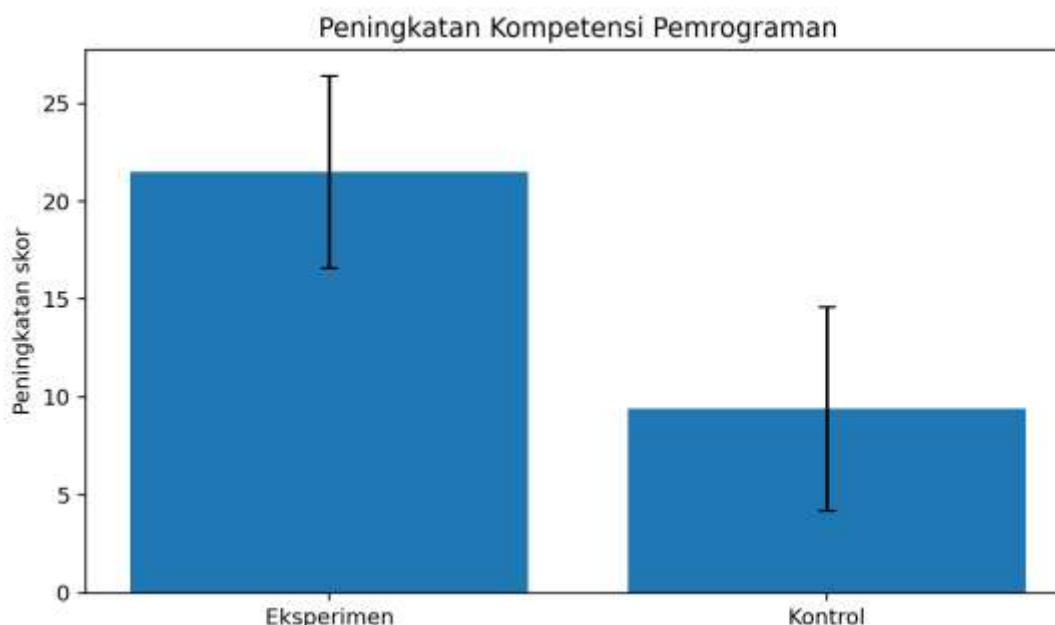
Tabel 5. Statistik Deskriptif Pretest dan Posttest

Kelompok	N	Pretest Mean $\pm$ SD	Posttest Mean $\pm$ SD	Peningkatan
Eksperimen	66	60.89 $\pm$ 8.89	82.39 $\pm$ 9.45	21.50
Kontrol	66	60.77 $\pm$ 7.66	70.20 $\pm$ 9.92	9.43

Kelompok eksperimen mengalami peningkatan rata-rata dari 60,89 pada pretest menjadi 82,39 pada posttest, atau meningkat 21,50 poin. Kelompok kontrol meningkat dari 60,77 menjadi 70,20, atau meningkat 9,43 poin. Dengan demikian, peningkatan rata-rata pada kelompok eksperimen lebih besar 12,07 poin dibandingkan kelompok kontrol.



Gambar 2. Perbandingan Rata-rata Pretest dan Posttest Kompetensi Pemrograman



Gambar 3. Perbandingan Peningkatan Kompetensi Pemrograman

3.3 Pengujian Hipotesis

Tabel 6. Hasil Uji Hipotesis

Pengujian	t	p	Keputusan
Pretest eksperimen vs kontrol	0.079	0.937	Tidak berbeda signifikan
Pretest–posttest eksperimen	35.623	<0,001	Signifikan
Pretest–posttest kontrol	14.721	<0,001	Signifikan
Posttest eksperimen vs kontrol	7.229	<0,001	H1 diterima
Gain eksperimen vs kontrol	13.721	<0,001	H2 diterima

Tabel 7. Efektivitas Model Hybrid

Indikator	Eksperimen	Kontrol
N-gain rata-rata	0.578	0.255
Kategori	Sedang	Rendah
Cohen's d posttest	1.258	–
Interpretasi efek	Besar	–

Uji pretest menunjukkan tidak terdapat perbedaan kemampuan awal yang signifikan antara kelompok eksperimen dan kontrol ($t = 0,079$; $p = 0,937$). Pada kelompok eksperimen, skor pretest dan posttest berbeda secara signifikan ($t = 35,623$; $p < 0,001$), dan perbedaan yang sama juga ditemukan pada kelompok kontrol ($t = 14,721$; $p < 0,001$). Perbandingan posttest antarkelompok menunjukkan perbedaan yang signifikan ($t = 7,229$; $p < 0,001$), sedangkan perbandingan gain juga signifikan ($t = 13,721$; $p < 0,001$). Cohen's d sebesar 1,258 menunjukkan ukuran efek besar. Karena kemampuan awal kedua kelompok tidak berbeda secara signifikan, hasil tersebut memberikan dukungan empiris terhadap hipotesis bahwa kelompok yang mengikuti model hybrid mencapai hasil akhir yang lebih tinggi. Interpretasi tetap perlu mempertimbangkan karakteristik desain quasi-experimental dan rincian penempatan peserta ke kelompok. Nilai N-gain 0,578 pada kelompok eksperimen berada pada kategori sedang, sedangkan 0,255 pada kelompok kontrol berada pada kategori rendah berdasarkan kriteria [13].

3.4 Pembahasan

Temuan penelitian menunjukkan bahwa kelompok yang mengikuti integrasi Project-Based Learning dan AI-Assisted Coding memperoleh peningkatan kompetensi pemrograman yang lebih tinggi daripada kelompok pembanding. Hasil ini

sejalan dengan bukti bahwa PjBL dapat mendukung hasil belajar dan pengembangan keterampilan melalui aktivitas proyek yang autentik [1], [2]. Dalam penelitian ini, kontribusi model tidak hanya terletak pada keberadaan proyek, tetapi pada pengaturan penggunaan AI di dalam tahapan proyek sehingga mahasiswa tetap harus membangun, menguji, dan menjelaskan solusi.

Tahap analisis masalah dan desain proyek menjadi bagian penting karena mahasiswa diminta merumuskan kebutuhan, batasan, algoritme, dan struktur program sebelum memanfaatkan AI untuk membantu implementasi. Urutan tersebut membantu menjaga agar AI tidak menjadi sumber solusi awal yang diterima secara pasif. Pendekatan semacam ini relevan dengan literatur yang menekankan pentingnya kemampuan memahami, mengevaluasi, dan menggunakan AI secara bertanggung jawab, bukan sekadar menghasilkan keluaran dari sistem AI [14], [15], [16].

Pada tahap AI-Assisted Coding, AI digunakan sebagai scaffolding ketika mahasiswa menghadapi kesulitan sintaks atau implementasi. Mahasiswa tidak hanya menerima kode, tetapi diminta memahami alternatif, membandingkannya dengan rancangan, dan melakukan modifikasi apabila diperlukan. Temuan ini memiliki kesesuaian dengan penelitian [12] yang menunjukkan bahwa tutor berbasis ChatGPT dapat mendukung pembelajaran pemrograman, khususnya bagi mahasiswa dengan fondasi awal yang lebih lemah. Namun, penggunaan AI tetap perlu dikendalikan karena dukungan yang terlalu dominan dapat menimbulkan ketergantungan dan mengurangi kesempatan membangun kemampuan pemecahan masalah secara mandiri.

Tahap pengujian dan debugging memperkuat keterampilan teknis karena mahasiswa harus membuat test case, mengidentifikasi kesalahan, dan memperbaiki program berdasarkan bukti hasil pengujian. AI dapat membantu menghasilkan hipotesis mengenai sumber kesalahan, tetapi keputusan perbaikan tetap harus diverifikasi melalui eksekusi program. Pola penggunaan ini konsisten dengan temuan [10] bahwa mahasiswa menggunakan AI untuk error checking dan debugging, sekaligus menunjukkan adanya kekhawatiran terhadap kualitas kode dan proses belajar ketika penggunaan AI tidak diimbangi dengan tujuan pembelajaran yang jelas. Dengan demikian, nilai pedagogis AI lebih kuat ketika teknologi tersebut ditempatkan di dalam siklus verifikasi, bukan sebagai pengganti proses debugging.

Tahap refleksi memperkuat aspek metakognitif dan tanggung jawab teknis. Mahasiswa diminta mengevaluasi bagian yang dapat diselesaikan secara mandiri, alasan menggunakan bantuan AI, kesesuaian keluaran AI dengan rancangan, dan keputusan teknis yang diambil. Pendekatan ini sejalan dengan prinsip penggunaan AI yang menempatkan manusia sebagai pengambil keputusan dan menekankan penggunaan teknologi secara bertanggung jawab [7]. Refleksi juga penting karena literasi AI mencakup kemampuan memahami, menggunakan, mengevaluasi, dan mempertimbangkan aspek etis dari teknologi AI [16].

Implikasi praktis penelitian adalah bahwa penggunaan AI dalam pembelajaran pemrograman sebaiknya diatur melalui desain aktivitas, bukan hanya melalui kebijakan memperbolehkan atau melarang AI. Pengajar dapat meminta mahasiswa mendokumentasikan rancangan awal, penggunaan AI yang relevan, versi kode, hasil pengujian, proses debugging, dan refleksi. Dokumentasi tersebut memungkinkan asesmen menilai proses kompetensi pemrograman sekaligus kemampuan mahasiswa menggunakan AI secara kritis. Pendekatan ini juga sejalan dengan perhatian terhadap integritas akademik dan kebutuhan penyesuaian asesmen pada era Generative AI [5], [17].

3.5 Keterbatasan Penelitian

Model hybrid Project-Based Learning dan AI-Assisted Coding menunjukkan efektivitas dalam meningkatkan kompetensi pemrograman mahasiswa berdasarkan perbandingan hasil posttest dan peningkatan gain antarkelompok. Kelompok eksperimen meningkat dari rata-rata 60,89 menjadi 82,39, sedangkan kelompok kontrol meningkat dari 60,77 menjadi 70,20. Perbedaan posttest antarkelompok signifikan ($t = 7,229$; $p < 0,001$) dengan Cohen's $d = 1,258$ yang menunjukkan ukuran efek besar. Temuan tersebut mendukung penggunaan AI sebagai scaffolding yang terintegrasi dalam pembelajaran berbasis proyek, dengan mahasiswa tetap bertanggung jawab terhadap analisis masalah, desain algoritme, implementasi, pengujian, debugging, dan refleksi. Hasil penelitian perlu ditafsirkan sesuai keterbatasan desain quasi-experimental dan keterbatasan pelaporan konteks intervensi.

Penelitian selanjutnya disarankan menguji model pada konteks institusi dan bahasa pemrograman yang berbeda, melibatkan durasi intervensi yang lebih panjang, melaporkan prosedur penggunaan AI secara rinci, serta menggunakan pengukuran yang dapat membedakan penguasaan konsep, kemampuan coding, debugging, dan kemampuan mengevaluasi keluaran AI.

4. KESIMPULAN

Model *hybrid Project-Based Learning* dan AI-Assisted Coding terbukti efektif meningkatkan kompetensi pemrograman mahasiswa. Kelompok eksperimen menunjukkan peningkatan yang lebih tinggi daripada kelompok kontrol, dengan perbedaan posttest yang signifikan dan ukuran efek besar ($d = 1.258$). Efektivitas model diperoleh melalui integrasi lima tahap

yang menempatkan AI sebagai scaffolding untuk eksplorasi, penjelasan, evaluasi, pengujian, dan debugging, sementara mahasiswa tetap bertanggung jawab terhadap analisis masalah, desain algoritme, keputusan implementasi, dan refleksi. Penelitian berikutnya dapat menguji model pada bahasa pemrograman yang berbeda, tingkat kemampuan awal yang beragam, serta durasi proyek yang lebih panjang.

PERNYATAAN PENGGUNAAN KECERDASAN ARTIFISIAL

Dalam proses penyempurnaan bahasa, struktur, dan keterbacaan naskah ini, penulis menggunakan kecerdasan artifisial generatif sebagai bantuan penyuntingan. AI tidak digunakan untuk menghasilkan atau memanipulasi data penelitian, menetapkan hasil, maupun menggantikan penilaian ilmiah penulis. Seluruh keluaran telah diperiksa, diverifikasi, disunting, dan menjadi tanggung jawab penuh penulis.

REFERENCES

- [1] U. Ubaidah and M. Loeis, "Improving computer programming competency for first semester computer science students through immersive project-based learning," *J. Educ. Res. Pract.*, vol. 12, no. 1, pp. 179–189, 2022, doi: 10.5590/JERAP.2022.12.1.13.
- [2] L. Zhang and Y. Ma, "A study of the impact of project-based learning on student learning effects: A meta-analysis study," *Front. Psychol.*, vol. 14, p. 1202728, 2023, doi: 10.3389/fpsyg.2023.1202728.
- [3] R. A. Jabbar and N. D. A. Halim, "The impact of project-based learning through integrating the use of technology in computer science courses on students' acquisition of programming skills," *Innov. Teach. Learn. J.*, vol. 8, no. 1, pp. 1–14, 2024, doi: 10.11113/itlj.v8.152.
- [4] E. Kasneci *et al.*, "ChatGPT for good? On opportunities and challenges of large language models for education," *Learn. Individ. Differ.*, vol. 103, p. 102274, 2023, doi: 10.1016/j.lindif.2023.102274.
- [5] C. K. Lo, "What is the impact of ChatGPT on education? A rapid review of the literature," *Educ. Sci.*, vol. 13, no. 4, p. 410, 2023, doi: 10.3390/educsci13040410.
- [6] A. Tlili *et al.*, "What if the devil is my guardian angel: ChatGPT as a case study of using chatbots in education," *Smart Learn. Environ.*, vol. 10, p. 15, 2023, doi: 10.1186/s40561-023-00237-x.
- [7] U.N.E.S.C.O., *Guidance for generative AI in education and research*. UNESCO, 2023.
- [8] R. Yilmaz and F. G. K. Yilmaz, "The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation," *Comput. Educ. Artif. Intell.*, vol. 4, p. 100147, 2023, doi: 10.1016/j.caeai.2023.100147.
- [9] D. Sun, A. Boudouaia, C. Zhu, and Y. Li, "Would ChatGPT-facilitated programming mode impact college students' programming behaviors, performances, and perceptions? An empirical study," *Int. J. Educ. Technol. High. Educ.*, vol. 21, p. 14, 2024, doi: 10.1186/s41239-024-00446-5.
- [10] S. Groothuisen, A. A. J. Beemt, J. C. Remmers, and L. W. Meeuwen, "AI chatbots in programming education: Students' use in a scientific computing course and consequences for learning," *Comput. Educ. Artif. Intell.*, vol. 7, p. 100290, 2024, doi: 10.1016/j.caeai.2024.100290.
- [11] T. Kosar, D. Ostojić, Y. D. Liu, and M. Mernik, "Computer science education in ChatGPT era: Experiences from an experiment in a programming course for novice programmers," *Mathematics*, vol. 12, no. 5, p. 629, 2024, doi: 10.3390/math12050629.
- [12] A. C. M. Yang, J.-Y. Lin, C.-Y. Lin, and H. Ogata, "Enhancing Python learning with PyTutor: Efficacy of a ChatGPT-based intelligent tutoring system in programming education," *Comput. Educ. Artif. Intell.*, vol. 7, p. 100309, 2024, doi: 10.1016/j.caeai.2024.100309.
- [13] R. R. Hake, "Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses," *Am. J. Phys.*, vol. 66, no. 1, pp. 64–74, 1998, doi: 10.1119/1.18809.
- [14] C. K. Y. Chan, "A comprehensive AI policy education framework for university teaching and learning," *Int. J. Educ. Technol. High. Educ.*, vol. 20, p. 38, 2023, doi: 10.1186/s41239-023-00408-3.
- [15] M. Hornberger, A. Bewersdorff, and C. Nerdel, "What do university students know about artificial intelligence? Development and validation of an AI literacy test," *Comput. Educ. Artif. Intell.*, vol. 5, p. 100165, 2023, doi: 10.1016/j.caeai.2023.100165.
- [16] D. T. K. Ng, J. K. L. Leung, S. K. W. Chu, and M. S. Qiao, "Conceptualizing AI literacy: An exploratory review," *Comput. Educ. Artif. Intell.*, vol. 2, p. 100041, 2021, doi: 10.1016/j.caeai.2021.100041.

- [17] Y. K. Dwivedi *et al.*, “So what if ChatGPT wrote it?” Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy,” *Int. J. Inf. Manag.*, vol. 71, p. 102642, 2023, doi: 10.1016/j.ijinfomgt.2023.102642.