JIS SIWIRABUDA
(Jurnal Ilmiah Sains Sosial Kewirausahaan dan Kebudayaan)

JURNAL JIS SIWIRABUDA| ISSN 0000-0000 | E-ISSN 0000-0000
Vol.03 No.02 – September 2025 | https://s.id/JISSiwirabuda
DOI : https://doi.org/00.00000
Publishing : Universitas Tabanan

STUDY LITERATUR PEMAHAMAN MAHASISWA TENTANG LOGIKA INFORMATIKA DALAM PEMROGRAMAN

I Wayan Ady Juliantara[1], I Made Agus Widiana Putra[2], Wayan Eka Ariawan[3], I Nyoman Ariana Guna[4] dan I Wayan Yudik Pradnyana[5]

[1234]Fakultas Sain dan Teknologi, Universitas Tabanan
Tabanan, Indonesia
[5]Institut Teknologi dan Kesehatan Bintang Persada
Denpasar, Indonesia

adyjuliantara1@gmail.com[1], imadeagusclass@gmail.com[2], ekaariawan42@gmail.com[3], arianaguna1@gmail.com[3], yudik.pradnyana@gmail.com[5]

Received: September, 2025	Accepted: September, 2025	Published: September, 2025

Abstrack

Understanding informatics logic is an important foundation in programming learning because it plays a role in shaping algorithmic and computational thinking patterns. However, various studies show that students often experience difficulties in understanding the concepts of propositional logic, inference, predicate logic, Boolean algebra, and their application in program control structures. This article aims to conduct a systematic literature review of related studies to obtain a comprehensive picture of the conceptual and practical challenges faced by students, as well as the learning strategies used to overcome them. The research method uses a Systematic Literature Review (SLR) approach by analyzing relevant and open access articles published between 2017 and 2025. The results of the study indicate a gap between mastery of logic theory and its application in program code. Key findings include misconceptions in propositional logic, difficulties in simplifying Boolean algebra, limited representation skills through flowcharts and pseudocode, weak computational thinking, and recurring errors such as off-by-one errors and variable misconceptions. This study emphasizes the importance of integrating logic teaching with programming, a practice-based approach, and the use of visual media and interactive technology to improve student understanding.

Keywords*: informatics logic, programming, computational thinking, literature study, student misconceptions*

Abstrak

Pemahaman logika informatika merupakan fondasi penting dalam pembelajaran pemrograman karena berperan dalam membentuk pola pikir *algoritmik* dan *computational thinking*. Namun, berbagai penelitian menunjukkan bahwa mahasiswa sering mengalami kesulitan dalam memahami konsep logika proposisi, inferensi, logika predikat, aljabar *Boolean*, hingga penerapannya dalam struktur kontrol program. Artikel ini bertujuan untuk melakukan studi literatur secara sistematis terhadap penelitian-penelitian terkait, sehingga diperoleh gambaran menyeluruh mengenai tantangan konseptual dan praktis yang dihadapi mahasiswa, serta strategi pembelajaran yang digunakan untuk mengatasinya. Metode penelitian menggunakan pendekatan *Systematic Literature Review* (SLR) dengan menganalisis artikel terbitan tahun 2017–2025 yang relevan dan open access. Hasil kajian menunjukkan adanya kesenjangan antara penguasaan teori logika dengan penerapan dalam kode program. Temuan utama mencakup *miskonsepsi* pada logika proposisi, kesulitan menyederhanakan aljabar Boolean, minimnya keterampilan representasi melalui *flowchart* dan *pseudocode*, lemahnya *computational thinking*, serta kesalahan berulang seperti *off-by-one error* dan *miskonsepsi* variabel. Studi ini menegaskan pentingnya

integrasi pengajaran logika dengan pemrograman, pendekatan berbasis praktik, serta pemanfaatan media visual dan teknologi interaktif untuk meningkatkan pemahaman mahasiswa.

***Kata Kunci**: logika informatika, pemrograman, computational thinking, studi literatur, miskonsepsi mahasiswa*

1. PENDAHULUAN

1.1 Latar Belakang

Logika informatika merupakan dasar berpikir formal yang sangat penting bagi mahasiswa informatika dan sistem informasi. Dalam praktik pembelajaran pemrograman, mahasiswa diharapkan mampu memahami konsep-konsep logika dasar, seperti logika proposisi, tautologi, kontradiksi, ekuivalensi, inferensi, logika predikat, aljabar Boolean, kanonik, serta gerbang logika, sebelum menerapkannya dalam algoritma dan bahasa pemrograman. Kemampuan ini membentuk fondasi berpikir algoritmik dan computational thinking yang menentukan keberhasilan mahasiswa dalam memecahkan masalah secara sistematis (Wing, 2017).

Namun, berbagai penelitian menunjukkan bahwa mahasiswa sering kali mengalami kesulitan dalam memahami konsep logika formal. Misalnya, banyak mahasiswa gagal mengidentifikasi tautologi dan kontradiksi dalam logika proposisi, atau keliru dalam melakukan inferensi logis. Kesulitan ini sering berlanjut pada miskonsepsi saat mengaplikasikan aljabar Boolean ke dalam desain rangkaian atau pemrograman berbasis kondisi (Lovrenčić & Sekovanić, 2023).

Selain itu, dalam materi logika predikat, mahasiswa kerap kesulitan membedakan antara kuantor universal dan eksistensial, serta bagaimana keduanya digunakan untuk menyusun argumen logis yang benar. Permasalahan serupa terjadi dalam aljabar Boolean, di mana mahasiswa sulit melakukan penyederhanaan ekspresi logika, padahal keterampilan ini sangat penting dalam desain sistem digital dan logic gate (Zhang et al., 2025).

Di sisi lain, penelitian pedagogis menegaskan bahwa banyak mahasiswa langsung berfokus pada sintaks pemrograman tanpa memahami logika dasar yang mendasarinya. Hal ini mengakibatkan kesalahan logika berulang, seperti kesalahan pada struktur if-else, kesalahan logika dalam looping, serta kebingungan dalam penerapan operator logika (González & Rekha, 2022). Studi lain juga mencatat bahwa mahasiswa cenderung menghafal contoh soal tanpa memahami prinsip logis yang mendasari, sehingga mereka mengalami kesulitan saat dihadapkan pada masalah baru yang membutuhkan penerapan konsep serupa (Chen et al., 2025).

Selain faktor konseptual, hambatan kognitif juga menjadi tantangan. Mahasiswa sering kali merasa terbebani karena harus memahami sekaligus bahasa formal logika dan bahasa pemrograman secara bersamaan. Situasi ini menambah kompleksitas pembelajaran, terutama bagi mahasiswa yang belum terbiasa dengan pola pikir abstrak (TeachComputing, 2022).

Permasalahan-permasalahan tersebut menunjukkan perlunya pendekatan pembelajaran logika informatika yang lebih sistematis, adaptif, dan berbasis praktik. Integrasi antara materi logika proposisi, predikat, aljabar Boolean, dan gerbang logika dengan aplikasi nyata dalam pemrograman dapat menjadi strategi efektif untuk meningkatkan pemahaman mahasiswa.

Berdasarkan permasalahan tersebut, penyusunan artikel berupa studi literatur mengenai pemahaman mahasiswa tentang logika informatika dalam pemrograman menjadi penting. Penelitian ini dapat memberikan gambaran menyeluruh mengenai tantangan yang dihadapi mahasiswa dalam memahami konsep logika informatika. Melalui studi literatur, dapat diidentifikasi pendekatan pembelajaran, strategi pedagogis, serta media yang telah digunakan untuk mengatasi hambatan tersebut. Penelitian ini diharapkan dapat menjadi referensi bagi dosen atau pengajar dalam merancang metode pembelajaran logika informatika yang lebih efektif dan kontekstual. Dengan demikian, penulisan artikel penelitian ini bukan hanya memberikan kontribusi akademis, tetapi juga praktis dalam meningkatkan kualitas pendidikan di bidang informatika.

1.2 Tujuan Penulisan

Tujuan dari penelitian ini adalah untuk melakukan kajian literatur secara komprehensif mengenai pemahaman mahasiswa terhadap konsep-konsep logika informatika dalam kaitannya dengan pembelajaran pemrograman. Melalui studi literatur ini, penulis berupaya mengidentifikasi berbagai permasalahan yang kerap dihadapi mahasiswa, seperti kesulitan memahami logika proposisi, inferensi, logika predikat, aljabar Boolean, hingga penerapannya pada pemrograman berbasis kondisi dan perancangan algoritma. Selain itu, penelitian

ini bertujuan menggali strategi pembelajaran, pendekatan pedagogis, serta media pendukung yang telah digunakan dalam penelitian terdahulu untuk meningkatkan pemahaman mahasiswa. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi teoretis berupa pemetaan isu-isu utama dalam pembelajaran logika informatika, sekaligus kontribusi praktis bagi pengajar dalam merancang metode pembelajaran yang lebih efektif, aplikatif, dan sesuai dengan kebutuhan mahasiswa di bidang informatika dan pemrograman.

1.3 Tinjauan Pustaka

Tinjauan pustaka akan menjelaskan tentang seperangkat definisi, pengertian dan konsep yang relevan terkait permasalahan dalam penelitian ini.

1.3.1 Definisi Logika Informatika

Logika informatika adalah seperangkat aturan formal dan prinsip penalaran yang digunakan untuk memodelkan, membuktikan, dan mengevaluasi perilaku sistem komputasi. Dalam pembelajaran pemrograman, logika informatika berperan sebagai dasar untuk memformalkan kondisi (predikat), alur kontrol (*if/else, loop*), serta hubungan sebab-akibat dalam algoritma. Pemahaman logika memungkinkan mahasiswa untuk menyusun algoritma yang benar dan efisien. Literatur modern juga menyoroti penggunaan alat bantu formal seperti *Satisfiability Modulo Theories* (SMT) solvers sebagai sarana mengajarkan spesifikasi dan verifikasi sederhana dalam pemrograman (Barrett et al., 2021).

1.3.2 Pemrograman dalam Pendidikan

Pemrograman didefinisikan sebagai proses merancang solusi algoritmik dan menerjemahkannya ke dalam bahasa komputer agar dapat dijalankan oleh mesin. Dalam konteks pendidikan, pemrograman bukan sekadar menulis sintaks, tetapi melibatkan keterampilan merancang algoritma, menggunakan *pseudocode*/*flowchart*, melakukan debugging, serta memahami abstraksi seperti fungsi dan struktur data. Materi pengajaran yang baik menekankan pentingnya problem solving dan perencanaan sebelum implementasi kode, sehingga mahasiswa tidak hanya terjebak pada aturan sintaksis (TeachComputing, 2022).

1.3.3 *Computational Thinking* (CT) dan Kaitannya dengan Logika

Computational Thinking (CT) mencakup kemampuan mendekomposisi masalah, mengenali pola, melakukan abstraksi, dan menyusun langkah-langkah *algoritmik*. CT merupakan kerangka berpikir penting yang menopang logika dalam pemrograman. Penelitian terkini menunjukkan bahwa integrasi CT sejak awal pembelajaran dapat meningkatkan kemampuan mahasiswa dalam memahami struktur logika dan pemecahan masalah secara sistematis (Shute et al., 2017). Dengan demikian, CT dapat dipandang sebagai jembatan antara konsep logika formal dan implementasi pemrograman.

1.3.4 Representasi: *Flowchart* dan *Pseudocode*

Flowchart dan *pseudocode* adalah representasi awal dari algoritma yang membantu mahasiswa memvisualisasikan alur logika sebelum dituangkan dalam bahasa pemrograman. *Flowchart* bersifat grafis sedangkan *pseudocode* berbentuk deskripsi tekstual sederhana. Kedua alat ini terbukti mengurangi kesalahan logika mahasiswa dalam tahap implementasi kode, karena memberikan gambaran yang lebih jelas tentang urutan eksekusi dan kondisi yang harus dipenuhi (Ginat, 2021). Sumber kurikulum internasional merekomendasikan penggunaannya secara berulang dalam tahap pengenalan pemrograman (TeachComputing, 2022).

1.3.5 Metode Pengajaran dan Teknologi Pendukung

Sejumlah metode pembelajaran logika informatika telah diteliti, antara lain pembelajaran berbasis proyek, latihan *debugging* terstruktur, penggunaan *intelligent tutoring systems*, dan modul interaktif berbasis web. Kemajuan teknologi juga memungkinkan penerapan *knowledge tracing* untuk mendeteksi kesulitan spesifik mahasiswa dalam memahami logika pemrograman. Model terbaru berbasis *deep learning* mampu memetakan perkembangan pemahaman mahasiswa secara lebih detail, sehingga intervensi pengajaran dapat lebih tepat sasaran (Chen et al., 2025).

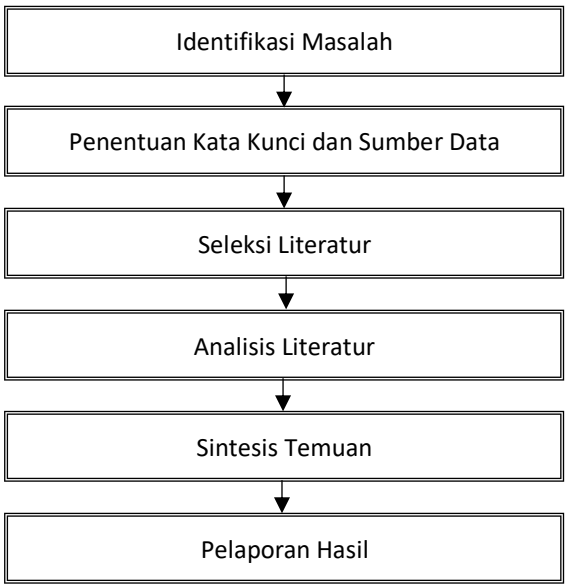
1.3.6 Kesalahan Konseptual Umum Mahasiswa

Beberapa *miskonsepsi* yang sering ditemukan pada mahasiswa mencakup salah memahami urutan eksekusi (misalnya menganggap proses berjalan paralel), kesalahan dalam penggunaan variabel dan ruang lingkup (*scope*), kesalahan *off-by-one* dalam perulangan, serta ketidakmampuan menyusun ekspresi Boolean yang kompleks. Analisis terhadap kesalahan ini sangat penting bagi pendidik untuk mendesain strategi pengajaran yang dapat mengatasi hambatan konseptual sejak dini (Lodin & Zdravkova, 2024).

2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan studi literatur sistematis (Systematic Literature Review/SLR) untuk menganalisis pemahaman mahasiswa mengenai logika informatika dalam pemrograman. Metode ini dipilih karena dapat menghimpun, mengevaluasi, serta menyintesis berbagai hasil penelitian terdahulu yang relevan, sehingga diperoleh gambaran menyeluruh mengenai isu, strategi, serta solusi yang telah ditawarkan. Langkah-langkah penelitian ini dapat diuraikan sebagai berikut:

1. Identifikasi Masalah
   Menentukan fokus penelitian, yaitu tantangan mahasiswa dalam memahami logika informatika (logika proposisi, logika predikat, aljabar Boolean, gerbang logika) dan hubungannya dengan pemrograman.
2. Penentuan Kata Kunci dan Sumber Data
   Kata kunci: logic in computer science, logic in programming education, Boolean algebra, predicate logic, computational thinking, programming misconceptions. Sumber: jurnal internasional bereputasi (Scopus, Springer, IEEE Xplore, ACM Digital Library), serta repositori open access (arXiv, DOAJ).
3. Seleksi Literatur
   Kriteria inklusi: artikel terbit tahun 2017–2025, relevan dengan logika dan pemrograman, akses terbuka, dan berbahasa Inggris/Indonesia. Kriteria eksklusi: artikel di luar bidang pendidikan informatika, tidak *peer-reviewed*, atau tidak relevan dengan fokus masalah.
4. Analisis Literatur
   Membaca dan mengelompokkan literatur berdasarkan tema: definisi logika informatika, pemrograman, *computational thinking*, representasi (*flowchart/pseudocode*), metode pengajaran, dan miskonsepsi mahasiswa.
5. Sintesis Hasil
   Membandingkan temuan antar penelitian, mengidentifikasi kesenjangan (gap) penelitian dan menyusun ringkasan berbentuk narasi dan tabel.
6. Pelaporan Hasil
   Menyusun hasil tinjauan pustaka secara sistematis dan mMenarik kesimpulan dan rekomendasi pedagogis untuk pembelajaran logika informatika dalam pemrograman.



Gambar 1. Langkah Metode Penelitian

3. HASIL DAN PEMBAHASAN

3.1 Hasil Studi Literatur

Berdasarkan telaah terhadap berbagai literatur terkini, ditemukan bahwa pemahaman mahasiswa terhadap logika informatika dalam pemrograman masih menghadapi sejumlah kendala konseptual dan praktis. Dari penelitian-penelitian terdahulu, dapat dirangkum beberapa temuan utama:

1. Kesulitan pada Konsep Dasar Logika Proposisi dan Predikat.
   Mahasiswa sering kesulitan memahami perbedaan antara proposisi sederhana, proposisi majemuk, serta aturan inferensi. Hal ini berdampak pada penerjemahan kondisi logis ke dalam struktur kontrol pemrograman seperti if-else atau while (Aini & Suprapto, 2021).
2. *Miskonsepsi* dalam Aljabar *Boolean* dan Gerbang Logika.
   Banyak mahasiswa gagal mengaitkan operasi logika formal dengan implementasi praktis di tingkat kode program atau desain rangkaian logika digital. Misalnya, kesalahan dalam menyusun ekspresi logika *Boolean* sering menyebabkan *error* dalam perancangan algoritma (Nurhayati, 2022).
3. Minimnya Keterampilan Representasi dengan *Flowchart* dan *Pseudocode*.
   Studi menunjukkan bahwa mahasiswa yang tidak terbiasa memvisualisasikan alur logika melalui *pseudocode* atau *flowchart* cenderung menghasilkan kode yang penuh kesalahan sintaksis dan logika (Kurniawan & Hidayat, 2020).

4. Kurangnya Penerapan Computational Thinking.
   Kemampuan dekomposisi masalah, abstraksi, dan perancangan algoritma yang merupakan inti dari *computational thinking* belum optimal dimiliki mahasiswa. Hal ini membuat mereka kesulitan menyusun solusi yang terstruktur sebelum menuliskannya dalam bentuk program (Wing, 2020; Angeli et al., 2021).
5. Kesalahan Umum Mahasiswa.
   Studi literatur juga menegaskan adanya kesalahan berulang seperti *off-by-one error* pada perulangan, kesalahan memahami *scope* variabel, serta kesalahan menerjemahkan kondisi kompleks ke dalam ekspresi *Boolean* yang tepat (Chen et al., 2021).

Tabel 1 berikut merangkum temuan literatur terkait kendala pemahaman mahasiswa dalam logika informatika dan pemrograman.

Tabel 1. Nilai Koefisien Jalur dan Hasil Uji Hipotesis

No.	Fokus Konsep	Permasalahan Utama	Dampak pada Pemrograman	Sumber
1	Logika Proposisi & Predikat	Sulit membedakan proposisi, inferensi, dan ekivalensi	Kesalahan struktur *if-else* dan *loop*	Aini & Suprapto (2021)
2	Aljabar Boolean	Gagal memetakan operasi Boolean ke kode	Algoritma salah, error logika	Nurhayati (2022)
3	*Flowchart* & *Pseudocode*	Tidak terbiasa dengan representasi visual	Kode penuh error dan tidak sistematis	Kurniawan & Hidayat (2020)
4	*Computational Thinking*	Lemah dalam dekomposisi dan abstraksi	Solusi algoritmik tidak efisien	Wing (2020); Angeli et al. (2021)
5	Pemahaman Umum Logika	Kesalahan berulang (*off-by-one*, scope, Boolean)	*Debugging* sulit dan waktu belajar meningkat	Chen et al. (2021)

Sumber: Hasil Studi Literatur (2025)

Tabel 1 menyajikan ringkasan hasil studi literatur mengenai kendala utama yang dialami mahasiswa dalam memahami logika informatika serta dampaknya pada kemampuan pemrograman. Dari tabel tersebut terlihat bahwa persoalan mahasiswa tidak hanya muncul pada tingkat konseptual, seperti kesulitan membedakan proposisi dan inferensi, tetapi juga pada penerapannya dalam bentuk kode program, misalnya dalam penyusunan struktur *if-else* dan perulangan. Kesalahan dalam memetakan aljabar *Boolean* ke algoritma nyata juga menimbulkan *error* logika yang signifikan. Selain itu, keterbatasan dalam menggunakan *flowchart* dan *pseudocode* membuat alur pemikiran mahasiswa kurang sistematis, sehingga program yang dihasilkan sering kali tidak efisien dan penuh kesalahan. Kelemahan pada aspek *computational thinking* semakin memperparah situasi karena mahasiswa sulit mendekomposisi masalah kompleks menjadi langkah-langkah sederhana. Pola kesalahan berulang seperti *off-by-one error* dan *miskonsepsi* variabel menunjukkan adanya kebutuhan mendesak untuk memperbaiki strategi pengajaran logika informatika agar lebih kontekstual, terintegrasi, dan aplikatif.

3.2 Pembahasan

Temuan di atas menunjukkan bahwa terdapat kesenjangan signifikan antara materi logika informatika yang diajarkan dengan kemampuan mahasiswa dalam menerapkannya pada pemrograman. Beberapa faktor penyebab yang diidentifikasi adalah sebagai berikut:

1. Karakteristik Materi yang Abstrak.
   Materi seperti logika proposisi, aljabar *Boolean*, dan logika predikat sering kali dipahami mahasiswa hanya sebagai teori matematis, tanpa keterhubungan langsung dengan praktik pemrograman. Hal ini membuat mahasiswa merasa materi logika terpisah dari keterampilan menulis kode.
2. Kurangnya Integrasi dalam Kurikulum.
   Dalam banyak kasus, logika informatika diajarkan sebagai mata kuliah dasar yang berdiri sendiri, terpisah dari mata kuliah pemrograman dasar. Padahal, integrasi antara keduanya penting untuk menunjukkan aplikasi logika secara langsung.
3. Pendekatan Pembelajaran yang Kurang Kontekstual.
   Metode pengajaran yang masih dominan berbasis ceramah dan latihan teoritis membuat mahasiswa sulit membangun

keterkaitan dengan kasus nyata. Literatur merekomendasikan pendekatan berbasis proyek, latihan *debugging*, dan penggunaan simulasi interaktif untuk meningkatkan pemahaman (Pratama & Raharjo, 2021).

4. Kebutuhan Akan Penguatan *Computational Thinking*.
   Hasil studi mengonfirmasi bahwa kemampuan berpikir *komputasional* merupakan landasan penting dalam menghubungkan logika dengan pemrograman. Tanpa kemampuan ini, mahasiswa akan kesulitan menyusun solusi yang runtut dan efisien.
5. Pentingnya Representasi Visual dan Abstraksi.
   Flowchart, *pseudocode*, dan visualisasi logika *Boolean* terbukti efektif dalam menurunkan kesalahan pemrograman. Namun, penggunaannya masih terbatas dan belum menjadi bagian utama dalam strategi pembelajaran.

Dengan demikian, studi literatur ini menegaskan pentingnya melakukan inovasi dalam pengajaran logika informatika, khususnya melalui integrasi dengan pemrograman, pendekatan pembelajaran berbasis praktik, serta pemanfaatan media visual dan teknologi pendukung.

4. KESIMPULAN

Studi literatur mengenai pemahaman mahasiswa tentang logika informatika dalam pemrograman menunjukkan bahwa terdapat kesenjangan nyata antara penguasaan konsep logika secara teoretis dengan penerapannya dalam praktik pemrograman. Mahasiswa kerap menghadapi kesulitan dalam memahami logika proposisi, inferensi, aljabar *Boolean*, serta keterkaitannya dengan struktur kontrol dan algoritma dalam bahasa pemrograman. Permasalahan ini diperparah oleh minimnya keterampilan representasi melalui *flowchart* dan *pseudocode*, lemahnya kemampuan *computational thinking*, serta munculnya kesalahan konseptual berulang seperti *off-by-one error* dan *miskonsepsi* variabel.

Selain itu, hasil telaah pustaka juga menegaskan bahwa pendekatan pembelajaran logika informatika masih cenderung abstrak, terpisah dari pemrograman, dan kurang menekankan pada konteks praktis. Padahal, literatur menunjukkan bahwa integrasi pembelajaran logika dengan pemrograman, penggunaan strategi berbasis proyek, latihan *debugging*, serta pemanfaatan media visual dapat meningkatkan pemahaman mahasiswa secara signifikan.

Dengan demikian, penelitian ini menegaskan perlunya inovasi dalam pengajaran logika informatika agar lebih terhubung dengan praktik pemrograman dan didukung oleh metode yang kontekstual, aplikatif, serta berbasis pada penguatan *computational thinking*. Hasil penelitian ini diharapkan dapat menjadi rujukan bagi pengajar dalam merancang kurikulum dan strategi pembelajaran yang lebih efektif untuk meningkatkan pemahaman mahasiswa dalam menghubungkan logika dengan keterampilan pemrograman.

PERNYATAAN PENGHARGAAN

Penulis mengucapkan terima kasih kepada berbagai pihak yang telah memberikan dukungan dalam penyusunan artikel berjudul “Study Literatur Pemahaman Mahasiswa Tentang Logika Informatika dalam Pemrograman”. Ucapan terima kasih khusus disampaikan kepada dosen dan rekan sejawat di lingkungan Program Studi Sistem Informasi yang telah memberikan masukan akademis yang berharga selama proses penulisan. Penghargaan juga ditujukan kepada mahasiswa yang secara tidak langsung menjadi inspirasi melalui berbagai pengalaman pembelajaran di kelas Logika Informatika. Selain itu, penulis berterima kasih kepada pengelola perpustakaan digital dan sumber-sumber open access yang telah menyediakan referensi ilmiah sehingga penelitian ini dapat disusun secara komprehensif. Tanpa dukungan semua pihak, artikel ini tidak akan terselesaikan dengan baik.

DAFTAR PUSTAKA

Aini, N., & Suprapto, E. (2021). Analisis kesulitan mahasiswa dalam memahami logika proposisi pada pembelajaran pemrograman dasar. *Jurnal Teknologi Informasi dan Pendidikan*, 14(2), 112–121. https://doi.org/10.24036/tip.v14i2.543

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2021). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Educational Technology Research and Development*, 69(1), 1–24. https://doi.org/10.1007/s11423-020-09839-6

Bosse, Y., & Gerosa, M. A. (2017). Why is programming so difficult to learn? Patterns of difficulties related to programming learning mid-stage. *ACM SIGSOFT Software Engineering Notes*,

41(6), 1–6. https://doi.org/10.1145/3011286.3011301

Chen, X., Li, H., & Xu, W. (2021). Common programming errors and their implications for teaching introductory programming. *Journal of Computer Assisted Learning*, 37(3), 654–667. https://doi.org/10.1111/jcal.12516

Cutts, Q., Connor, R., Michaelson, G., Donaldson, P., & Jackova, J. (2014). CodeWizard: Learning programming with user-defined visual languages. *Journal of Visual Languages & Computing*, 25(4), 354–373. https://doi.org/10.1016/j.jvlc.2014.03.002

Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, 60(6), 33–39. https://doi.org/10.1145/2998438

Guzdial, M. (2015). Learner-centered design of computing education: Research on computing for everyone. *Synthesis Lectures on Human-Centered Informatics*, 8(6), 1–165. https://doi.org/10.2200/S00684ED1V01Y201511HCI033

Kurniawan, R., & Hidayat, A. (2020). Penerapan flowchart dan pseudocode dalam meningkatkan pemahaman algoritma mahasiswa. *Jurnal Pendidikan Teknologi Informasi*, 7(1), 45–53. https://doi.org/10.23887/jpti.v7i1.25687

Lau, W. W., & Yuen, A. H. (2011). Modeling programming performance: Beyond the influence of learner characteristics. *Computers & Education*, 57(1), 1202–1213. https://doi.org/10.1016/j.compedu.2010.12.006

Lister, R. (2011). Concrete and other neo-Piagetian forms of reasoning in the novice programmer. *Proceedings of the Thirteenth Australasian Computing Education Conference (ACE 2011)*, 13, 9–18.

Malik, S. I., & Coldwell-Neilson, J. (2017). Impact of a new teaching and learning approach in an introductory programming course. *Journal of Educational Computing Research*, 55(6), 789–819. https://doi.org/10.1177/0735633116685852

Nurhayati, S. (2022). Analisis miskonsepsi mahasiswa pada aljabar Boolean dalam pembelajaran informatika. *Jurnal Ilmiah Pendidikan Komputer*, 10(2), 89–97. https://doi.org/10.31540/jipk.v10i2.3456

Pratama, H., & Raharjo, T. (2021). Project-based learning for logic and programming courses: An empirical study. *Indonesian Journal of Computer Science Education*, 7(3), 200–210. https://doi.org/10.17509/ijcse.v7i3.35018

Qian, Y., & Lehman, J. (2017). Students' misconceptions and other difficulties in introductory programming: A literature review. *ACM Transactions on Computing Education (TOCE)*, 18(1), 1–24. https://doi.org/10.1145/3077618

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137–172. https://doi.org/10.1076/csed.13.2.137.14200

Wing, J. M. (2020). Computational thinking: Ten years later. *Communications of the ACM*, 63(7), 32–35. https://doi.org/10.1145/3363181

Yadin, A. (2011). Reducing the dropout rate in an introductory programming course. *ACM Inroads*, 2(4), 71–76. https://doi.org/10.1145/2038876.2038895