

Dominasi Computational Thinking atas Motivasi dalam Kemampuan Pemrograman C Siswa SMA

Alfajar Yusuffa, Riskayeni Marta, Vera Irma Delianti, Resmidarni

Program Studi Pendidikan Teknik Informatika, Departemen Teknik Elektronika, Fakultas Teknik,
Universitas Negeri Padang, Padang, Indonesia

*Corresponding Author e-mail: maaruffarhan@gmail.com

Received: June 2026; Revised: July 2026; Published: July 2026

Abstrak

Kemampuan pemrograman tidak hanya ditentukan oleh penguasaan sintaks, tetapi juga oleh cara siswa memformulasikan masalah dan mempertahankan usaha ketika menghadapi kesalahan program. Penelitian ini bertujuan menguji kontribusi computational thinking skill dan motivasi belajar terhadap kemampuan pemrograman Bahasa C siswa SMA, sekaligus mengidentifikasi prediktor yang benar-benar dominan. Penelitian menggunakan desain kuantitatif korelasional ex post facto pada 138 siswa kelas XI SMA Negeri 1 Kubung yang dipilih secara proportionate stratified random sampling dari populasi 203 siswa. Computational thinking diukur dengan 13 soal objektif, motivasi dengan 22 pernyataan skala Likert, dan kemampuan pemrograman dengan 18 soal objektif; koefisien reliabilitas masing-masing sebesar 0,884, 0,852, dan 0,897. Hasil menunjukkan computational thinking berkorelasi sangat kuat dengan kemampuan pemrograman ($r = 0,910$; $p < 0,001$; $R^2 = 0,829$), sedangkan motivasi belajar tidak berkorelasi signifikan ($r = -0,107$; $p = 0,210$). Model simultan signifikan, $F(2,135) = 331,81$; $p < 0,001$, dengan $R^2 = 0,831$, tetapi penambahan motivasi hanya meningkatkan daya jelas model sebesar 0,2% ($\Delta R^2 = 0,002$) dan koefisiennya tetap tidak signifikan ($\beta = -0,044$; $p = 0,212$). Kebaruan penelitian terletak pada bukti bahwa signifikansi model gabungan hampir sepenuhnya digerakkan oleh computational thinking, bukan motivasi. Temuan ini menegaskan perlunya pembelajaran Bahasa C yang memprioritaskan dekomposisi, pengenalan pola, abstraksi, perancangan algoritma, tracing, dan debugging, tanpa menafsirkan motivasi sebagai penyebab langsung capaian pemrograman.

Kata kunci: Computational Thinking; Motivasi Belajar; Kemampuan Pemrograman; Bahasa C; Siswa Sma

Computational Thinking Dominates Motivation in High School Students' C Programming Ability

Abstract

Programming ability depends not only on syntax mastery but also on how students formulate problems and sustain effort when code fails. This study examined the contributions of computational thinking skills and learning motivation to high school students' C programming ability and identified the genuinely dominant predictor. A quantitative ex post facto correlational design involved 138 eleventh-grade students from SMA Negeri 1 Kubung, proportionally stratified and randomly selected from a population of 203. Computational thinking was measured using 13 objective items, motivation using 22 Likert statements, and programming ability using 18 objective items; reliability coefficients were 0.884, 0.852, and 0.897, respectively. Computational thinking showed a very strong correlation with programming ability ($r = 0.910$, $p < 0.001$, $R^2 = 0.829$), whereas learning motivation was not significantly related ($r = -0.107$, $p = 0.210$). The simultaneous model was significant, $F(2,135) = 331.81$, $p < 0.001$, with $R^2 = 0.831$; however, adding motivation increased explained variance by only 0.2% ($\Delta R^2 = 0.002$), and its coefficient remained nonsignificant ($\beta = -0.044$, $p = 0.212$). The study's novelty is evidence that the significant joint model was driven almost entirely by computational thinking rather than motivation. C programming instruction should therefore prioritize decomposition, pattern recognition, abstraction, algorithm design, tracing, and debugging while avoiding causal interpretations from correlational data.

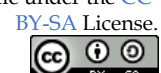
Keywords: computational thinking; learning motivation; programming ability; C language; high school students

How to Cite: Yusuffa, Y., Marta, R., Delianti, V. I., & Resmidarni, R. (2026). Dominasi Computational Thinking atas Motivasi dalam Kemampuan Pemrograman C Siswa SMA. *Journal of Authentic Research*, 5(3), 5401-5415. <https://journal-center.litpam.com/index.php/jar/article/view/6173>



<https://journal-center.litpam.com/index.php/jar/article/view/6173>

Copyright© 2026, Yusuffa et al.
This is an open-access article under the CC-



PENDAHULUAN

Pemrograman merupakan kompetensi inti dalam Informatika karena mengharuskan siswa menerjemahkan persoalan menjadi representasi data, urutan prosedur, struktur kontrol, dan kode yang dapat dieksekusi. Pada Bahasa C, tuntutan tersebut semakin nyata karena siswa harus mengelola deklarasi variabel, tipe data, operator, percabangan, perulangan, larik, pustaka standar, serta aturan sintaks yang ketat. Dengan demikian, kemampuan pemrograman tidak dapat direduksi menjadi kemampuan menyalin baris kode; kemampuan ini mencakup analisis kebutuhan, perancangan algoritma, penelusuran alur program, pengujian, dan debugging (Kadir, 2021; Muslim & Rasim, 2024). Kurikulum Informatika Fase F juga menempatkan strategi algoritmik, pengembangan program terstruktur, pemeliharaan kode, struktur data, dan penggunaan library sebagai capaian yang harus dikuasai siswa (Kementerian Pendidikan Kebudayaan Riset dan Teknologi, 2025). Kerangka tersebut beririsan langsung dengan computational thinking, yaitu proses memodelkan masalah dan menetapkan prosedur yang dapat dijalankan secara efektif oleh agen pemroses informasi (Wing, 2006; Denning & Tedre, 2021). Dalam pendidikan K-12, computational thinking dipandang sebagai kompetensi pemecahan masalah yang mengintegrasikan dekomposisi, pengenalan pola, abstraksi, dan algoritma, bukan sekadar penggunaan perangkat digital (Grover & Pea, 2013; Braun & Huwer, 2022). Karena itu, rendahnya kemampuan pemrograman perlu dibaca sebagai persoalan penalaran komputasional dan praktik pemecahan masalah, bukan hanya kekurangan hafalan sintaks.

Table 1. Dokumentasi awal capaian Informatika siswa kelas XI

Kelas	N	Mencapai KKTP	Tidak mencapai KKTP	Rata-rata
XI 3	35	17 (48,57%)	18 (51,43%)	72,40
XI 4	32	19 (59,38%)	13 (40,63%)	71,95
XI 5	35	18 (51,43%)	17 (48,57%)	73,40
XI 6	34	21 (61,76%)	13 (38,24%)	73,45
XI 7	34	19 (55,88%)	15 (44,12%)	72,60
XI 8	33	16 (48,48%)	17 (51,52%)	71,80
Total	203	110 (54,19%)	93 (45,81%)	72,60

Sumber: dokumentasi sekolah SMA Negeri 1 Kubung, tahun ajaran 2025/2026.

Dokumentasi awal di SMA Negeri 1 Kubung menunjukkan kesenjangan yang nyata antara tuntutan kurikulum dan capaian siswa. Dari 203 siswa kelas XI yang mengikuti peminatan Informatika, 93 siswa atau 45,81% belum mencapai Kriteria Ketercapaian Tujuan Pembelajaran pada evaluasi semester genap, sedangkan rata-rata keseluruhan hanya 72,60. Ketidaklulusan tertinggi muncul pada kelas XI 8 sebesar 51,52%, diikuti XI 3 sebesar 51,43%. Wawancara pendahuluan dengan guru menguatkan data tersebut: siswa umumnya mampu mengetik ulang contoh, tetapi berhenti ketika variasi masalah berubah karena logika dasar belum terstruktur. Siswa juga mengakui cenderung langsung menulis kode tanpa menyusun algoritma, kemudian kesulitan menelusuri sumber kesalahan ketika program gagal. Pola ini konsisten dengan temuan bahwa kelemahan dekomposisi, abstraksi, dan penyusunan

langkah sistematis membuat siswa terjebak pada prosedur permukaan dan tidak mampu mentransfer solusi ke kasus baru (Sari et al., 2023; Farid & Hidayah, 2024; Rahma et al., 2024). Monalisa (2023) juga menunjukkan bahwa tingkat penguasaan indikator computational thinking membedakan kualitas pemecahan masalah siswa. Oleh sebab itu, data ketuntasan, wawancara guru, dan pengalaman siswa secara bersama-sama mengarah pada satu persoalan substantif: pembelajaran pemrograman belum sepenuhnya membentuk cara berpikir yang memungkinkan siswa merancang, memeriksa, dan memperbaiki solusi secara mandiri.

Sejumlah penelitian mendukung pentingnya computational thinking, tetapi hasil-hasil yang tersedia belum sepenuhnya menjawab persoalan di atas. Risnandar dan Al Maki (2022) melaporkan keberhasilan pendekatan computational thinking dalam penguasaan pemrograman Python, sedangkan Pajow et al. (2024) menemukan hubungan positif antara computational thinking dan pemahaman konsep matematis. Kajian sistematis menegaskan bahwa asesmen computational thinking lazim mengukur kemampuan algoritmik, abstraksi, dekomposisi, dan evaluasi solusi, meskipun definisi serta instrumennya masih beragam (Tang et al., 2020; Hasanah & Yahfizham, 2024). Pada sisi lain, motivasi belajar secara teoretis mengarahkan intensitas, pilihan tindakan, dan persistensi siswa (Uno, 2021). Dalam pembelajaran Informatika, strategi guru, media interaktif, dan pengalaman proyek dapat meningkatkan keterlibatan siswa (Wahyuni et al., 2023; Mawaddah et al., 2024; Ahkam et al., 2025). Namun, peningkatan motivasi tidak selalu identik dengan peningkatan kompetensi teknis. Motivasi yang dilaporkan sendiri dapat tinggi, sementara kemampuan melakukan tracing, memilih struktur kontrol, atau memperbaiki logical error tetap rendah apabila latihan kognitif dan umpan balik teknis tidak memadai. Studi-studi sebelumnya juga banyak berfokus pada intervensi, pembelajaran berbasis blok, matematika, atau kreativitas, bukan pada pemrograman teks Bahasa C yang menuntut ketepatan sintaks dan logika sekaligus (Muslim & Rasim, 2024; Tampubolon et al., 2025).

Kesenjangan penelitian terletak pada belum jelasnya bobot relatif faktor kognitif dan afektif ketika keduanya diuji secara bersamaan dalam konteks pemrograman Bahasa C pada siswa SMA. Penelitian terdahulu cenderung menguji computational thinking atau motivasi secara terpisah, sehingga model gabungan yang signifikan mudah disalahartikan seolah-olah setiap prediktor memberi sumbangan yang bermakna. Padahal, signifikansi simultan dapat digerakkan hampir seluruhnya oleh satu variabel. Penelitian ini menawarkan tiga kebaruan. Pertama, konteksnya adalah pemrograman teks Bahasa C pada Fase F, bukan pemrograman visual atau sekadar pemecahan masalah matematika. Kedua, kemampuan computational thinking dan kemampuan pemrograman diukur dengan tes objektif yang telah melalui uji validitas, reliabilitas, kesukaran, dan daya pembeda, sedangkan motivasi diukur melalui skala afektif. Ketiga, analisis tidak berhenti pada korelasi dan R^2 model gabungan, tetapi memeriksa koefisien parsial, interval kepercayaan, dan perubahan R^2 ketika motivasi ditambahkan setelah computational thinking. Pendekatan ini penting untuk membedakan “model bersama-sama signifikan” dari “kedua prediktor sama-sama signifikan”. Secara kritis, penelitian juga mengakui bahwa tumpang tindih isi antara tes computational thinking dan tes pemrograman dapat memperbesar hubungan yang diamati. Dengan demikian, novelty penelitian bukan hanya angka kontribusi,

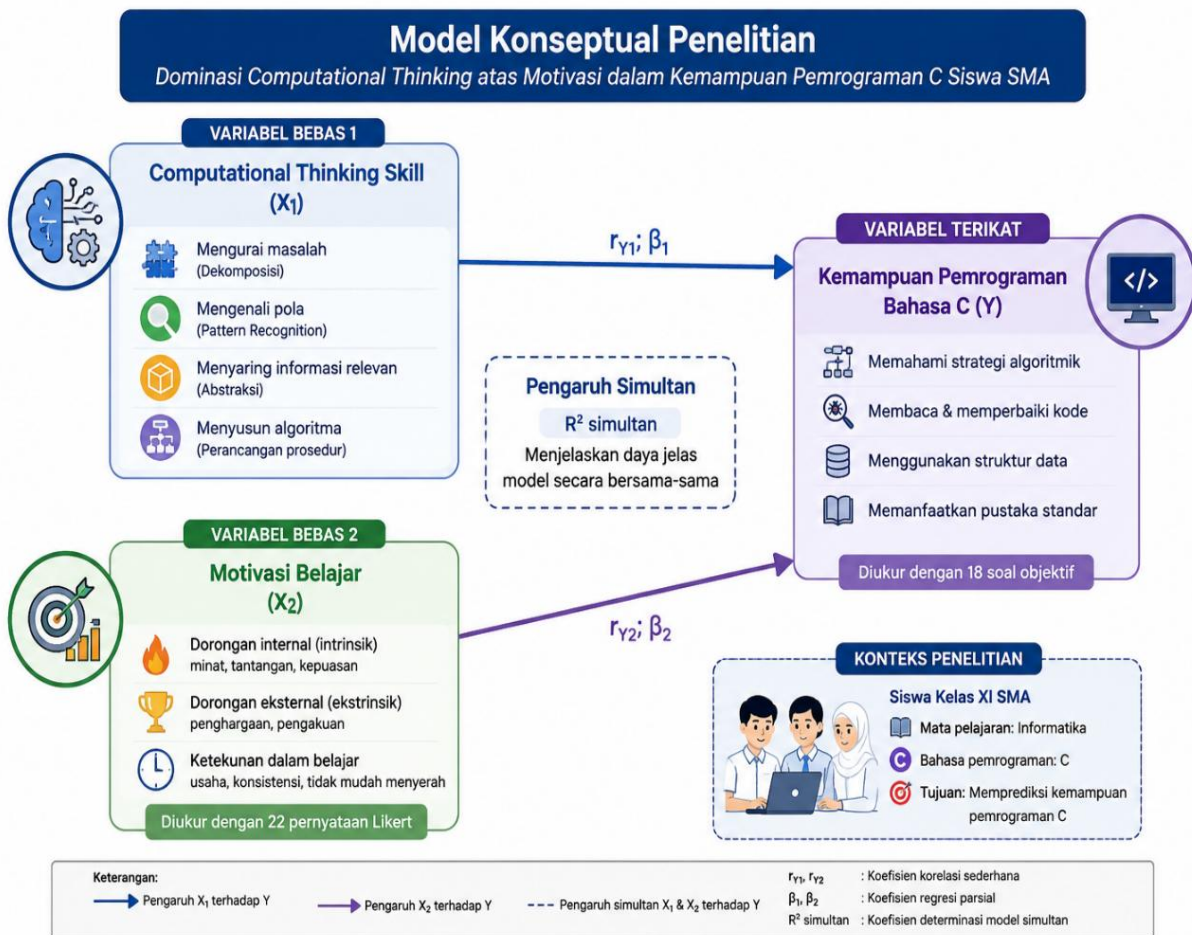
melainkan pembacaan yang lebih tepat tentang sumber daya jelas model dan batas validitas inferensinya.

Berdasarkan kesenjangan tersebut, penelitian ini bertujuan: (1) mendeskripsikan profil computational thinking, motivasi belajar, dan kemampuan pemrograman Bahasa C; (2) menguji hubungan computational thinking dengan kemampuan pemrograman; (3) menguji hubungan motivasi belajar dengan kemampuan pemrograman; dan (4) menilai kontribusi simultan serta kontribusi unik masing-masing prediktor. Computational thinking dibatasi pada dekomposisi, pengenalan pola, abstraksi, dan algoritma (Wing, 2006; Sari et al., 2023). Motivasi belajar mencakup hasrat berhasil, kebutuhan belajar, harapan masa depan, penghargaan, kegiatan menarik, dan lingkungan kondusif (Uno, 2021). Kemampuan pemrograman mengacu pada pemahaman strategi algoritmik, penerjemahan notasi ke program terstruktur, pemeliharaan dan debugging kode, penggunaan struktur data, serta pemanfaatan library sesuai capaian Fase F. Hipotesis penelitian menyatakan bahwa computational thinking berhubungan positif dengan kemampuan pemrograman, motivasi belajar berhubungan dengan kemampuan pemrograman, dan kedua prediktor secara simultan menjelaskan variasi kemampuan pemrograman. Karena desainnya *ex post facto* dan korelasional, istilah “kontribusi” dalam artikel ini digunakan dalam arti proporsi variasi statistik yang dijelaskan, bukan bukti sebab-akibat. Pembatasan ini diperlukan agar implikasi pedagogis tetap tajam tanpa melampaui kekuatan data.

METODE

Desain dan konteks penelitian

Penelitian menggunakan pendekatan kuantitatif dengan desain korelasional *ex post facto*. Desain ini dipilih karena variabel computational thinking, motivasi belajar, dan kemampuan pemrograman telah melekat pada siswa sebelum pengukuran, sehingga peneliti tidak memberikan perlakuan atau memanipulasi proses pembelajaran (Sugiyono, 2023; Hasbi et al., 2023). Penelitian dilaksanakan di SMA Negeri 1 Kubung, Kabupaten Solok, Sumatera Barat, pada semester genap tahun ajaran 2025/2026. Konteks pembelajaran adalah mata pelajaran Informatika Fase F, khususnya elemen Algoritma dan Pemrograman menggunakan Bahasa C. Unit analisis adalah siswa individual. Variabel bebas pertama, computational thinking skill (X1), didefinisikan sebagai kemampuan menguraikan masalah, mengenali pola, menyaring informasi relevan, dan menyusun algoritma. Variabel bebas kedua, motivasi belajar (X2), didefinisikan sebagai dorongan internal dan eksternal yang mengarahkan ketekunan belajar Informatika. Variabel terikat, kemampuan pemrograman Bahasa C (Y), didefinisikan sebagai kemampuan memahami strategi algoritmik, membaca dan memperbaiki kode, menggunakan struktur data, serta memanfaatkan pustaka standar. Model konseptual menguji dua hubungan sederhana dan satu model simultan. Karena penelitian bersifat observasional, pengujian diarahkan pada kekuatan asosiasi dan proporsi variasi yang dijelaskan, bukan pada estimasi efek kausal.



Gambar 1. Model konseptual penelitian

Populasi dan sampel

Populasi penelitian terdiri atas 203 siswa kelas XI yang mengambil peminatan Informatika pada enam rombongan belajar, yaitu XI 3 sampai XI 8. Kelas XI 1 dan XI 2 tidak dimasukkan karena tidak memperoleh mata pelajaran peminatan Informatika. Ukuran sampel minimum dihitung dengan rumus Slovin pada tingkat kesalahan 5%, menghasilkan 134,66 atau sekurang-kurangnya 135 siswa. Untuk menjaga keterwakilan tiap kelas, alokasi dilakukan secara proporsional dan setiap hasil desimal dibulatkan ke atas. Prosedur tersebut menghasilkan 138 siswa: 24 siswa dari XI 3, 22 dari XI 4, 24 dari XI 5, 23 dari XI 6, 23 dari XI 7, dan 22 dari XI 8. Dengan demikian, teknik yang secara operasional digunakan adalah proportionate stratified random sampling, bukan simple random sampling murni. Siswa dipilih secara acak di dalam tiap kelas setelah daftar populasi diperoleh dari guru Informatika. Sebanyak 30 siswa lain yang memiliki karakteristik setara digunakan untuk uji coba instrumen dan tidak diikuti ke dalam sampel utama. Pemisahan ini mencegah kebocoran butir dan mengurangi bias karena responden telah mengenal soal pada saat pengumpulan data utama.

Table 2. Distribusi populasi dan sampel penelitian

Kelas	Populasi	Sampel	Proporsi sampel (%)
XI 3	35	24	68,57
XI 4	32	22	68,75
XI 5	35	24	68,57
XI 6	34	23	67,65
XI 7	34	23	67,65
XI 8	33	22	66,67
Total	203	138	67,98

Instrumen dan prosedur

Tiga instrumen digunakan. Tes computational thinking awalnya terdiri atas 20 soal pilihan ganda yang memetakan dekomposisi, pengenalan pola, abstraksi, dan algoritma. Setelah uji korelasi butir-total pada 30 siswa dengan r tabel 0,361, 13 butir dinyatakan valid. Tes kemampuan pemrograman awalnya terdiri atas 20 soal yang memuat strategi algoritmik, program terstruktur, sintaks dan debugging, larik, serta library; 18 butir dipertahankan. Angket motivasi awalnya berisi 30 pernyataan skala Likert empat tingkat; 22 butir valid digunakan dalam penelitian. Reliabilitas instrumen berturut-turut sebesar 0,884 untuk computational thinking, 0,852 untuk motivasi, dan 0,897 untuk kemampuan pemrograman, seluruhnya melampaui batas 0,70. Soal kognitif juga diuji tingkat kesukaran dan daya pembeda; butir yang digunakan berada pada kategori sukar sampai sedang/mudah dan memiliki daya pembeda minimal cukup. Pengujian validitas dan reliabilitas mengikuti prinsip evaluasi instrumen pendidikan (Arikunto, 2021; Fadli et al., 2023). Pengumpulan data dilakukan secara langsung: siswa mengerjakan tes computational thinking, mengisi angket motivasi, dan mengerjakan tes kemampuan pemrograman. Dokumentasi nilai sekolah dan wawancara awal digunakan untuk membangun konteks masalah, bukan sebagai variabel dalam pengujian hipotesis.

Table 3. Karakteristik dan kualitas instrumen

Variabel	Bentuk	Butir awal	Butir digunakan	Reliabilitas	Rentang kualitas butir
Computational thinking	Tes pilihan ganda	20	13	0,884	Kesukaran sukar-sedang; daya beda cukup-sangat baik
Motivasi belajar	Likert 4 tingkat	30	22	0,852	Butir valid berdasarkan korelasi item-total
Kemampuan pemrograman C	Tes pilihan ganda	20	18	0,897	Kesukaran sukar-mudah; daya beda cukup-sangat baik

Analisis data

Analisis dilakukan pada matriks skor 138 responden. Statistik deskriptif meliputi mean, simpangan baku, median, modus, rentang, serta persentase terhadap skor maksimum. Capaian per indikator dihitung dengan membagi rerata skor

indikator dengan skor maksimumnya. Hubungan bivariat diuji menggunakan korelasi Pearson dan regresi linear sederhana; interval kepercayaan 95% untuk r dihitung melalui transformasi Fisher. Model simultan diestimasi dengan regresi linear berganda $Y = a + b_1X_1 + b_2X_2$. Pelaporan mencakup R , R^2 , adjusted R^2 , F , koefisien tidak terstandar, beta terstandar, t , p , interval kepercayaan, serta perubahan R^2 ketika motivasi ditambahkan setelah computational thinking. Asumsi diperiksa melalui normalitas distribusi dan residual, linearitas, homoskedastisitas, independensi residual, dan multikolinearitas. Uji Kolmogorov-Smirnov mengacu pada prinsip goodness-of-fit (Massey, 1951), sedangkan keputusan statistik menggunakan $\alpha = 0,05$ (Syahroni, 2023). VIF di bawah 10 dan tolerance di atas 0,10 dipakai sebagai indikator tidak adanya multikolinearitas. Hasil utama dihitung melalui SPSS versi 27 dan diverifikasi ulang dari data mentah untuk memastikan tanda koefisien, nilai p , dan kontribusi unik setiap prediktor konsisten.

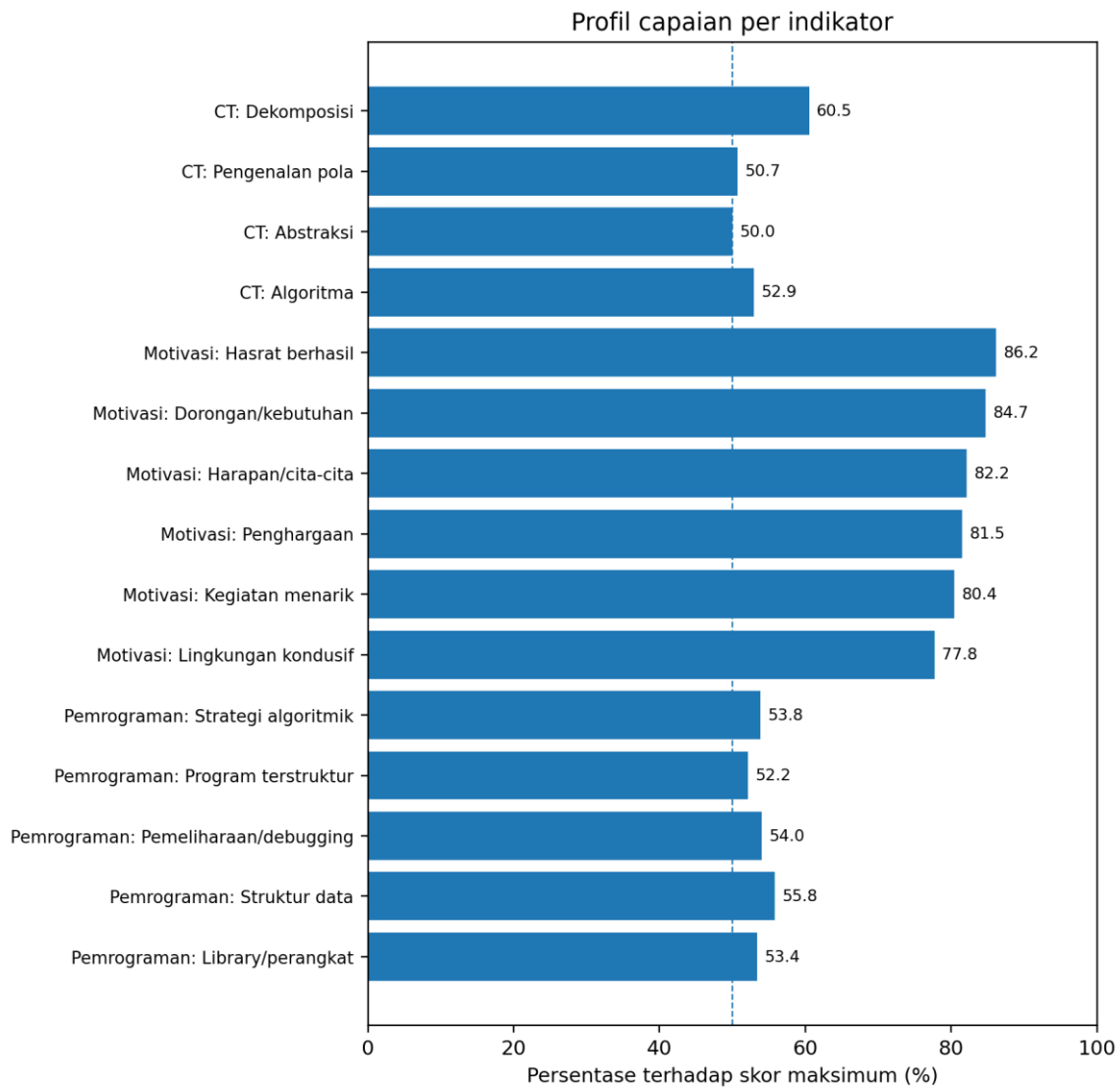
HASIL DAN PEMBAHASAN

Profil data dan kesenjangan kognitif-afektif

Table 4. Statistik deskriptif variabel penelitian

Variabel	N	Mean	SD	Median	Min-Maks	% skor maksimum
Computational thinking	138	7,05	3,34	7,00	1-13	54,24
Motivasi belajar	138	71,96	8,75	72,00	40-87	81,77
Kemampuan pemrograman C	138	9,67	4,76	9,50	1-18	53,74

Seluruh data dari 138 responden dapat dianalisis. Rerata computational thinking sebesar 7,05 dari skor maksimum 13 atau 54,24%, sedangkan kemampuan pemrograman sebesar 9,67 dari skor maksimum 18 atau 53,74%. Kedua capaian tersebut menunjukkan penguasaan yang masih berada di sekitar separuh skor maksimum. Sebaliknya, rerata motivasi mencapai 71,96 dari maksimum 88 atau 81,77%. Pola indikator menguatkan perbedaan itu. Pada computational thinking, capaian tertinggi terdapat pada dekomposisi (60,51%), sedangkan abstraksi menjadi yang terendah (50,00%). Pada kemampuan pemrograman, seluruh indikator terkonsentrasi pada rentang 52,17–55,80%; struktur data relatif tertinggi, sementara penerjemahan program terstruktur terendah. Semua indikator motivasi berada pada rentang 77,76–86,17%. Dengan demikian, siswa melaporkan dorongan belajar yang tinggi, tetapi kinerja objektif computational thinking dan pemrograman tetap moderat-rendah. Ketidaksejajaran ini penting karena menunjukkan bahwa persepsi afektif positif belum otomatis berubah menjadi keterampilan teknis yang dapat didemonstrasikan pada soal algoritma, tracing, dan debugging.



Gambar 2. Persentase capaian tiap indikator terhadap skor maksimum

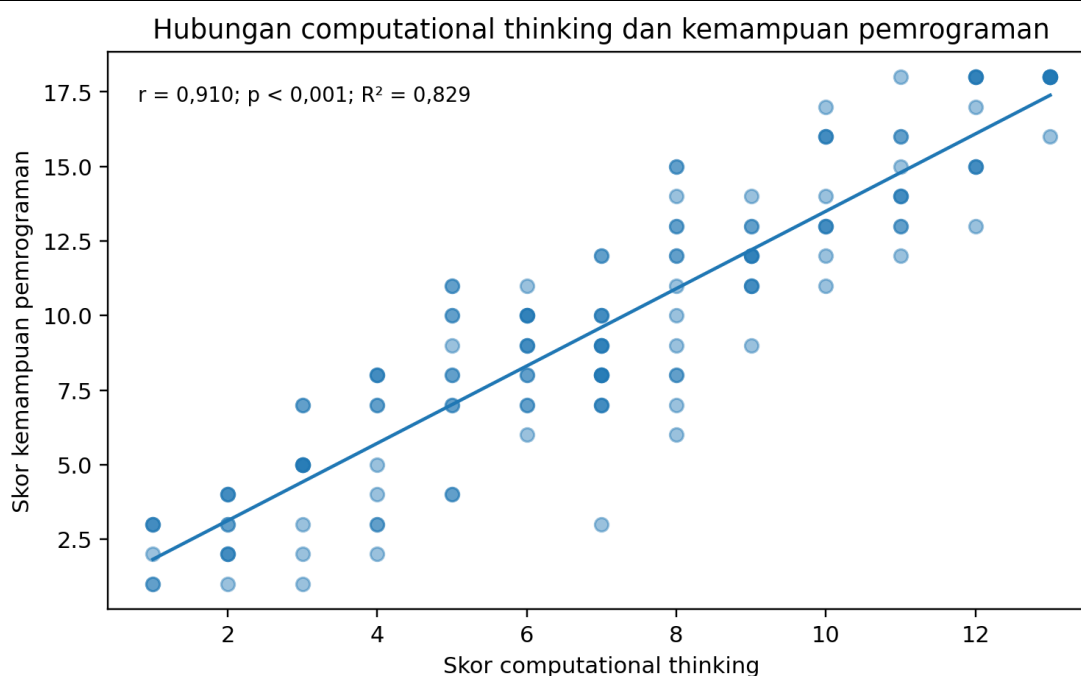
Secara rasional, kesenjangan tersebut dapat muncul ketika siswa memiliki keinginan berhasil, harapan masa depan, atau menyukai kegiatan praktikum, tetapi belum memiliki strategi kognitif yang cukup untuk memecah masalah dan memeriksa kode. Wawancara awal mendukung penjelasan ini: siswa menyatakan ingin mampu memprogram, namun sering langsung mengetik kode tanpa rencana dan menyerah ketika error berulang. Guru juga menilai pembelajaran sebelumnya lebih banyak menggunakan demonstrasi sehingga siswa mudah meniru contoh tetapi sulit melakukan transfer. Temuan tersebut sejalan dengan Sari et al. (2023) dan Farid dan Hidayah (2024), yang menempatkan dekomposisi dan abstraksi sebagai fondasi pengelolaan kompleksitas. Rahma et al. (2024) menegaskan bahwa computational thinking diperlukan agar pemecahan masalah berlangsung sistematis. Sementara itu, penelitian tentang motivasi Informatika menunjukkan bahwa antusiasme dapat dipengaruhi strategi guru dan media pembelajaran (Mawaddah et al., 2024; Wahyuni et al., 2023), tetapi peningkatan afeksi memerlukan latihan teknis agar menghasilkan kompetensi. Oleh karena itu, motivasi yang tinggi dalam data ini lebih tepat dibaca sebagai kesiapan psikologis, bukan bukti penguasaan pemrograman.

Computational thinking sebagai prediktor dominan

Korelasi computational thinking dengan kemampuan pemrograman sangat kuat dan positif, $r = 0,910$, $p < 0,001$, dengan interval kepercayaan 95% antara 0,877 dan 0,935. Regresi sederhana menghasilkan persamaan $\hat{Y} = 0,532 + 1,297X_1$ dan $R^2 = 0,829$. Artinya, setiap tambahan satu poin computational thinking terkait dengan kenaikan sekitar 1,30 poin kemampuan pemrograman, dan 82,9% variasi skor pemrograman dapat dijelaskan secara statistik oleh skor computational thinking. Analisis per indikator menunjukkan seluruh dimensi computational thinking berhubungan signifikan dengan kemampuan pemrograman: dekomposisi $r = 0,681$, pengenalan pola $r = 0,648$, abstraksi $r = 0,697$, dan algoritma $r = 0,743$; seluruh $p < 0,001$. Algoritma menjadi indikator dengan hubungan terkuat, sesuai karakter Bahasa C yang mengharuskan siswa merancang urutan instruksi, memprediksi keluaran, dan mengevaluasi efisiensi sebelum memperbaiki sintaks. Pola titik pada grafik juga memperlihatkan kenaikan kemampuan pemrograman yang konsisten seiring meningkatnya skor computational thinking.

Table 5. Korelasi dan regresi sederhana terhadap kemampuan pemrograman

Prediktor	r	95% CI r	p	R ²	B	Persamaan
Computational thinking	0,910	0,877-0,935	<0,001	0,829	1,297	$\hat{Y} = 0,532 + 1,297X_1$
Motivasi belajar	-0,107	-0,270-0,061	0,210	0,012	-0,058	$\hat{Y} = 13,877 - 0,058X_2$



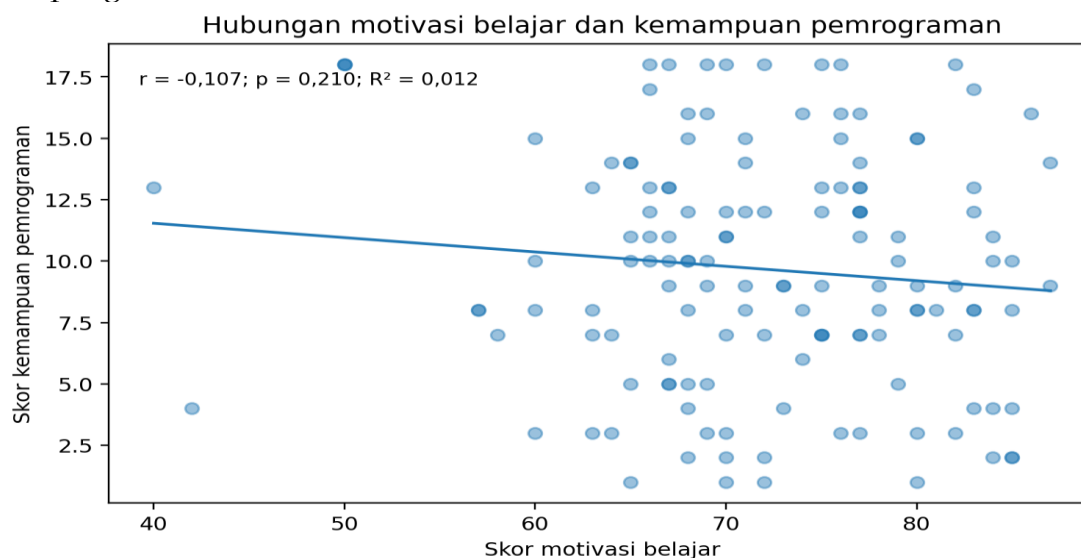
Gambar 3. Hubungan computational thinking dengan kemampuan pemrograman

Temuan ini mendukung gagasan bahwa programming merupakan manifestasi teknis dari proses berpikir komputasional. Wing (2006) dan Denning dan Tedre (2021) menempatkan formulasi masalah dan spesifikasi prosedur sebagai inti computational thinking. Dalam konteks siswa, hubungan positif juga dilaporkan oleh Pajow et al. (2024) pada pemahaman konsep dan oleh Risnandar dan Al Maki (2022) pada

pembelajaran pemrograman. Monalisa (2023) menunjukkan bahwa penguasaan indikator computational thinking membedakan tingkat pemecahan masalah, sedangkan Tampubolon et al. (2025) menekankan peran pola dan algoritma dalam penyelesaian persoalan terstruktur. Namun, besarnya $r = 0,910$ harus ditafsirkan secara kritis. Tes computational thinking dan tes pemrograman sama-sama memuat unsur strategi algoritmik, struktur kontrol, tracing, serta evaluasi keluaran. Kedekatan konten ini dapat mencerminkan hubungan substantif sekaligus shared measurement domain. Dengan kata lain, hasil kuat tersebut mendukung dominasi penalaran komputasional, tetapi belum membuktikan bahwa computational thinking sebagai konstruk yang sepenuhnya terpisah menyebabkan kemampuan pemrograman. Penelitian lanjutan perlu memakai tugas kinerja autentik dan analisis validitas diskriminan agar batas kedua konstruk lebih tegas.

Motivasi tinggi tidak menjelaskan capaian pemrograman

Motivasi belajar menunjukkan korelasi negatif yang lemah dan tidak signifikan dengan kemampuan pemrograman, $r = -0,107$, $p = 0,210$, dengan interval kepercayaan 95% $-0,270$ sampai $0,061$. Regresi sederhana menghasilkan $R^2 = 0,012$, sehingga hanya 1,2% variasi kemampuan pemrograman yang terkait dengan motivasi dan arah koefisiennya tidak stabil secara statistik. Hasil ini memperbaiki pembacaan yang hanya melihat besaran R tanpa tanda: berdasarkan matriks data mentah, arah hubungan adalah negatif, bukan positif. Meskipun demikian, interval kepercayaan melintasi nol, sehingga data tidak cukup untuk menyimpulkan adanya hubungan negatif populasi. Secara empiris, siswa yang mengaku sangat termotivasi dapat memperoleh skor pemrograman rendah maupun tinggi; pola titik tersebar tanpa kecenderungan linear yang jelas. Temuan ini juga tidak berarti motivasi “tidak penting” secara universal, melainkan bahwa skala motivasi umum yang digunakan tidak mampu memprediksi skor tes pemrograman secara bermakna pada sampel dan waktu pengukuran ini.



Gambar 4. Hubungan motivasi belajar dengan kemampuan pemrograman

Ada beberapa penjelasan yang masuk akal. Pertama, skor motivasi relatif tinggi pada hampir semua indikator, sehingga variasi afektif lebih kecil daripada variasi kemampuan kognitif. Kedua, angket mengukur persepsi dan niat, sedangkan tes

pemrograman menuntut performa objektif; kesenjangan niat-perilaku dan bias keinginan sosial dapat melemahkan hubungan. Ketiga, motivasi mungkin bekerja secara tidak langsung melalui intensitas latihan, ketekunan debugging, efikasi diri, atau kualitas strategi belajar—variabel mediator yang tidak diukur. Uno (2021) menempatkan motivasi sebagai pendorong dan pengarah tindakan, tetapi pendorong tersebut tetap membutuhkan kemampuan dan kondisi belajar yang memadai. Mawaddah et al. (2024) juga menunjukkan bahwa motivasi dalam Informatika berinteraksi dengan disiplin, kesiapan kognitif, dan strategi guru. Sementara Wahyuni et al. (2023) menemukan peningkatan hasil melalui media interaktif, efek tersebut terjadi dalam konteks intervensi yang mengubah pengalaman belajar, bukan sekadar perbedaan skor motivasi. Karena itu, hasil penelitian ini lebih mendukung model bahwa motivasi perlu dioperasionalkan sebagai perilaku belajar spesifik—misalnya frekuensi latihan, durasi debugging, atau penyelesaian proyek—agar hubungan dengan kemampuan pemrograman dapat diuji secara lebih sensitif.

Model simultan: signifikansi digerakkan oleh computational thinking

Model regresi berganda yang memasukkan computational thinking dan motivasi signifikan, $F(2,135) = 331,81$, $p < 0,001$, dengan $R = 0,912$, $R^2 = 0,831$, adjusted $R^2 = 0,828$, dan standard error of estimate 1,973. Persamaan model adalah $\hat{Y} = 2,305 + 1,292X_1 - 0,024X_2$. Computational thinking tetap menjadi prediktor sangat kuat ($\beta = 0,907$; $t = 25,582$; $p < 0,001$), sedangkan motivasi tidak signifikan setelah computational thinking dikendalikan ($\beta = -0,044$; $t = -1,254$; $p = 0,212$). Penambahan motivasi pada model yang sebelumnya hanya memuat computational thinking menaikkan R^2 dari 0,829 menjadi 0,831, atau $\Delta R^2 = 0,002$. Dengan demikian, tambahan daya jelas motivasi hanya sekitar 0,2 poin persentase. VIF sebesar 1,005 menunjukkan bahwa hasil tersebut bukan akibat multikolinearitas antara kedua prediktor. Interpretasi yang tepat bukan “computational thinking dan motivasi sama-sama berkontribusi”, melainkan “model gabungan signifikan karena dominasi computational thinking, sementara motivasi tidak memberikan kontribusi unik yang terdeteksi”.

Table 6. Hasil regresi linear berganda

Prediktor	B	SE	β	t	p	95% CI B
Konstanta	2,305	1,467	-	1,571	0,119	-0,597–5,207
Computational thinking	1,292	0,051	0,907	25,582	<0,001	1,192–1,392
Motivasi belajar	-0,024	0,019	-0,044	-1,254	0,212	-0,062–0,014

Model: $R = 0,912$; $R^2 = 0,831$; adjusted $R^2 = 0,828$; $F(2,135) = 331,81$; $p < 0,001$; $SEE = 1,973$; ΔR^2 motivasi setelah computational thinking = 0,002.

Inilah temuan paling penting sekaligus novelty penelitian. Banyak laporan korelasional berhenti pada R^2 model gabungan dan menyimpulkan semua variabel bebas berpengaruh. Analisis koefisien parsial dan perubahan R^2 menunjukkan bahwa kesimpulan semacam itu tidak didukung data ini. Secara pedagogis, prioritas intervensi perlu diarahkan pada latihan yang secara eksplisit mengembangkan dekomposisi, abstraksi, pengenalan pola, algoritma, tracing, dan debugging. Hasanah dan Yahfizham (2024) menekankan bahwa computational thinking membantu siswa memanipulasi masalah kompleks menjadi langkah yang terkelola, sedangkan Muslim dan Rasim (2024) menunjukkan pentingnya pengalaman proyek untuk mengintegrasikan perancangan dan implementasi. Motivasi tetap perlu dipelihara

sebagai kondisi yang memungkinkan siswa bertahan, tetapi strategi motivasional sebaiknya melekat pada aktivitas kognitif yang autentik: proyek bertahan, umpan balik error yang informatif, pair debugging, dan refleksi algoritma. Dengan cara itu, dorongan belajar tidak berdiri sebagai sikap deklaratif, melainkan menjadi energi untuk melakukan praktik yang memperkuat kemampuan. Temuan juga menegaskan bahwa sekolah tidak cukup hanya membuat pembelajaran “menarik”; pembelajaran harus memberi struktur berpikir dan kesempatan latihan yang dapat diverifikasi melalui produk program.

Kelayakan model, implikasi, dan keterbatasan

Diagnostik menunjukkan model linear layak digunakan. Nilai signifikansi Kolmogorov-Smirnov untuk computational thinking, motivasi, dan kemampuan pemrograman masing-masing 0,070, 0,051, dan 0,200. Deviation from Linearity untuk hubungan X1-Y sebesar 0,404 dan X2-Y sebesar 0,361. Pada model regresi, residual berdistribusi normal (Shapiro-Wilk $p = 0,479$), tidak menunjukkan heteroskedastisitas (Breusch-Pagan $p = 0,870$), dan spesifikasi linear tidak ditolak (RESET $p = 0,644$). Durbin-Watson sebesar 1,828 menunjukkan tidak ada indikasi kuat autokorelasi residual. Karena itu, ketidaksignifikanan motivasi tidak dapat dijelaskan oleh pelanggaran asumsi dasar model. Hasil tersebut memperkuat keyakinan bahwa perbedaan daya prediksi kedua variabel memang terdapat pada data, walaupun generalisasi tetap terbatas pada konteks sekolah dan instrumen penelitian.

Table 7. Ringkasan diagnostik asumsi

Asumsi/diagnostik	Statistik	p/keputusan
Normalitas variabel (K-S)	X1 = 0,070; X2 = 0,051; Y = 0,200	Seluruh $p > 0,05$
Linearitas (deviation from linearity)	X1-Y = 0,404; X2-Y = 0,361	Hubungan linear
Normalitas residual (Shapiro-Wilk)	W = 0,991	$p = 0,479$
Homoskedastisitas (Breusch-Pagan)	$\chi^2 = 0,280$	$p = 0,870$
Spesifikasi linear (RESET)	F = 0,215	$p = 0,644$
Independensi residual	Durbin-Watson = 1,828	Tidak ada indikasi kuat autokorelasi
Multikolinearitas	VIF = 1,005; tolerance = 0,995	Tidak ada multikolinearitas

Tiga keterbatasan harus menjadi batas interpretasi. Pertama, desain ex post facto tidak memungkinkan penetapan urutan sebab-akibat; siswa yang mahir memprogram mungkin mengembangkan computational thinking lebih baik, atau keduanya dipengaruhi pengalaman latihan sebelumnya. Kedua, kemampuan pemrograman diukur melalui tes pilihan ganda, bukan produk program lengkap, sehingga aspek desain, efisiensi, dokumentasi, dan ketahanan debugging belum terobservasi secara utuh. Ketiga, kedekatan indikator computational thinking dan pemrograman berpotensi meningkatkan korelasi. Selain itu, motivasi diukur dengan self-report umum, bukan rekaman perilaku belajar. Oleh sebab itu, penelitian berikutnya perlu mengombinasikan tes konstruk yang lebih diskriminatif, tugas coding autentik, log aktivitas IDE, dan desain longitudinal atau eksperimen. Kajian juga dapat memasukkan efikasi diri pemrograman, kebiasaan latihan, kualitas umpan

balik, fasilitas laboratorium, dan pengalaman coding sebelumnya sebagai mediator atau moderator. Langkah tersebut akan menguji apakah dominasi computational thinking tetap bertahan ketika kemampuan pemrograman dinilai melalui performa nyata dan hubungan temporal dapat diamati.

KESIMPULAN

Penelitian menunjukkan bahwa kemampuan pemrograman Bahasa C siswa terutama berkaitan dengan computational thinking. Computational thinking menjelaskan 82,9% variasi kemampuan pemrograman dan tetap menjadi satu-satunya prediktor signifikan dalam model simultan. Motivasi belajar memiliki skor deskriptif tinggi, tetapi tidak berhubungan signifikan dengan kemampuan pemrograman dan hanya menambah 0,2% daya jelas setelah computational thinking dikendalikan. Dengan demikian, signifikansi model gabungan tidak boleh ditafsirkan sebagai bukti bahwa kedua prediktor sama-sama berkontribusi. Temuan utama penelitian adalah dominasi penalaran komputasional atas motivasi deklaratif dalam konteks pemrograman teks Bahasa C. Kesimpulan ini bersifat asosiasional, bukan kausal, dan perlu dibaca bersama keterbatasan tumpang tindih konstruk serta penggunaan tes pilihan ganda.

REKOMENDASI

Pembelajaran perlu memprioritaskan latihan berjenjang yang mengintegrasikan dekomposisi masalah, abstraksi data, pengenalan pola, penyusunan algoritma, tracing, dan debugging sebelum siswa diminta menulis program lengkap. Strategi motivasional sebaiknya ditanamkan di dalam praktik tersebut melalui proyek autentik, umpan balik cepat, pair debugging, dan refleksi kesalahan, bukan hanya melalui penghargaan atau penyajian materi yang menarik. Penelitian selanjutnya disarankan menggunakan tugas coding autentik, log aktivitas IDE, desain longitudinal atau eksperimen, serta menguji efikasi diri, kebiasaan latihan, fasilitas laboratorium, dan pengalaman pemrograman sebelumnya. Validitas diskriminan antara computational thinking dan kemampuan pemrograman juga perlu diuji agar besarnya hubungan tidak dipengaruhi oleh kemiripan butir.

ACKNOWLEDGMENT

Penulis mengucapkan terima kasih kepada SMA Negeri 1 Kubung, guru mata pelajaran Informatika, dan seluruh siswa yang berpartisipasi dalam pengumpulan data penelitian.

REFERENSI

- Ahkam, J., Farhan, N., & Hakim, A. (2025). Pengaruh pembelajaran berbasis proyek digital terhadap kreativitas siswa sekolah menengah. *Journal of Modern Social and Humanities*, 1(6), 201–209.
- Apriani, A., Ibrahim, I. D. K., & Muhid, A. (2024). Penerapan Kurikulum Merdeka melalui Kompetisi Bebras Computational Thinking untuk siswa sekolah dasar. *ADMA: Jurnal Pengabdian dan Pemberdayaan Masyarakat*, 4(2), 463–470. <https://doi.org/10.30812/adma.v4i2.3434>
- Arikunto, S. (2021). *Dasar-dasar evaluasi pendidikan* (Edisi ke-3). Bumi Aksara.
- Braun, D., & Huwer, J. (2022). Computational literacy in science education – A systematic review. *Frontiers in Education*, 7, 937048. <https://doi.org/10.3389/feduc.2022.937048>

- Denning, P. J., & Tedre, M. (2021). Computational thinking: A disciplinary perspective. *Informatics in Education*, 20(3), 361–390. <https://doi.org/10.15388/infedu.2021.21>
- Fadli, R., Hidayati, S., Cholifah, M., Siroj, A. R., & Afgani, W. M. (2023). Validitas dan reliabilitas pada penelitian motivasi belajar pendidikan agama Islam menggunakan product moment. *JIIP-Jurnal Ilmiah Ilmu Pendidikan*, 6(3), 1734–1739.
- Farid, S. A., & Hidayah, I. N. (2024). Analisis computational thinking pondasi dekomposisi siswa SMP kelas VII pada pembelajaran model project based learning. *Jurnal Integrasi dan Harmoni Inovatif Ilmu-Ilmu Sosial*, 4(5). <https://doi.org/10.17977/um063.v4.i5.2024.2>
- Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43. <https://doi.org/10.3102/0013189X12463051>
- Hasanah, S. Z. H., & Yahfizham, Y. (2024). Systematic literature review: Analisis kemampuan berpikir komputasional siswa sekolah dasar menggunakan Scratch. *PENDEKAR: Jurnal Pendidikan Berkarakter*, 2, 90–97. <https://doi.org/10.51903/pendekar.v2i3.732>
- Hasbi, A. Z. E., Damayanti, R., Hermina, D., & Mizani, H. (2023). Penelitian korelasional (metodologi penelitian pendidikan). *Al-Furqan: Jurnal Agama, Sosial, dan Budaya*, 2(6), 784–808.
- Kadir, A. (2021). Dasar logika pemrograman komputer. Elex Media Komputindo.
- Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi. (2025). Capaian pembelajaran dan alur tujuan pembelajaran Informatika Fase F kelas XI.
- Massey, F. J. (1951). The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253), 68–78.
- Mawaddah, Darman, & Saputra, H. N. (2024). Strategi guru dalam meningkatkan motivasi belajar siswa pembelajaran Informatika. *Jurnal Ilmiah Ilmu Pendidikan*, 7(9).
- Monalisa, M. (2023). Analisis berpikir komputasional siswa SMP pada Kurikulum Merdeka mata pelajaran Informatika. *DIAJAR: Jurnal Pendidikan dan Pembelajaran*, 2(3), 298–304. <https://doi.org/10.54259/diajar.v2i3.1596>
- Muslim, A. A., & Rasim. (2024). The effectiveness of the project based learning model on web programming material in enhancing students' creative thinking skills. *Jurnal PPS-TP*, 12(1). <https://doi.org/10.23960/jt.1214>
- Pajow, M. A., Regar, V. E., & Maukar, M. G. (2024). Hubungan kemampuan computational thinking dan pemahaman konsep matematika siswa materi pola bilangan. *Kognitif: Jurnal Riset HOTS Pendidikan Matematika*, 4(1), 544–554. <https://doi.org/10.51574/kognitif.v4i1.1661>
- Rahma, F. L. P., Putri, I. A., Tanjung, M. S., & Siregar, R. (2024). Studi literatur: Pentingnya berpikir komputasional dalam meningkatkan kemampuan pemecahan masalah matematis peserta didik. *Bilangan: Jurnal Ilmiah Matematika, Kebumian dan Angkasa*, 2, 17–27. <https://doi.org/10.62383/bilangan.v2i2>
- Risnandar, & Al Maki, W. F. (2022). Metode computational thinking untuk pengabdian masyarakat dalam peningkatan kemampuan bahasa pemrograman

- Python siswa SMK. *Jurnal Pengabdian Masyarakat Teknologi Informasi dan Informatika (DIMASLOKA)*, 1(1), 1-7.
- Sari, E. N. I., Nursanti, E. D. R., Fadholi, M., Aprinastuti, C., & Rismiyati, Y. (2023). Penerapan pendekatan computational thinking terhadap kemampuan pondasi pada materi energi dan perubahannya kelas 3 SD Negeri Bhaktikarya. *EDUKASI: Jurnal Penelitian & Artikel Pendidikan*, 15(1), 1-12.
- Sugiyono. (2023). *Metode penelitian kuantitatif, kualitatif, dan R&D*. Alfabeta.
- Syahroni, M. I. (2023). Analisis data kuantitatif. *E-Jurnal Al Musthafa*, 3(3), 19-27. <https://doi.org/10.62552/ejam.v3i3.64>
- Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, 148, 103798. <https://doi.org/10.1016/j.compedu.2019.103798>
- Tampubolon, A., Hutagalung, C. F., & Kemberen, S. N. B. (2025). Analisis kemampuan berpikir komputasional matematis siswa SMA pada materi fungsi. *Pedagogy: Jurnal Pendidikan Matematika*, 10(4), 2432-2450.
- Uno, H. B. (2021). *Teori motivasi dan pengukurannya: Analisis di bidang pendidikan* (Edisi ke-17). Bumi Aksara.
- Wahyuni, R., Irsyadunas, & Alfiriani, A. (2023). Efektivitas media pembelajaran berbasis TikTok pada mata pelajaran Informatika kelas X TKJ. *Jurnal Informatika*, 11(2).
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>