JIMAT
Jurnal Ilmiah Matematika

e-ISSN: 2774-1729
Volume. 6. Nomor 2. Desember 2025
Halaman: 724 -738

DOI: https://doi.org/10.63976/jimat.v6i2.1140

Profil Berpikir Komputasi Mahasiswa dari Prosedur Matematis ke Algoritma Program: Studi Kasus Metode Bagi Dua

Khadijah [1*]

[1]Program Studi Pendidikan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Makassar, Sulawesi Selatan, Indonesia

Email Corresponding Author : khadijah@unm.ac.id

Info Artikel

Article history:
Kirim: 01 November 2025
Terima: 30 November 2025

Publikasi Online, 6 Desember 2025

Kata-kata kunci:
Berpikir Komputasi;
Metode Bagi Dua;
Penalaran Algoritmik;
Algoritma Program;
Studi Kasus.

ABSTRAK

Penguasaan metode numerik tidak hanya menuntut pemahaman prosedur matematis, tetapi juga kemampuan mengonversinya ke bentuk algoritmik. Namun, banyak mahasiswa masih kesulitan mentransformasi hal tersebut. Kondisi ini menunjukkan pentingnya menelaah bagaimana mahasiswa berpikir secara komputasional saat menyusun sintaks program, khususnya pada algoritma dasar seperti Metode Bagi Dua. Penelitian ini bertujuan untuk mendeskripsikan profil berpikir komputasi (CT) mahasiswa dalam mengonversi prosedur matematika Metode Bagi Dua ke dalam program komputer. Pendekatan deskriptif kualitatif dengan desain studi kasus digunakan, melibatkan dua kelompok mahasiswa yang terdaftar dalam Mata Kuliah Metode Numerik. Data penelitian terdiri dari hasil proyek pemrograman mahasiswa, termasuk kode sumber PHP dan keluaran program. Data dianalisis dengan menggunakan Kerangka Berpikir Komputasi oleh Brennan dan Resnick, (dekomposisi, pengenalan pola, abstraksi, berpikir algoritmik, otomatisasi, dan debugging). Temuan ini mengungkapkan profil CT yang berbeda antara kedua kelompok. Kelompok pertama menunjukkan keterampilan CT dasar yang dicirikan oleh dekomposisi linear dan berpikir algoritmik dasar, dengan abstraksi terbatas dan strategi debugging minimal. Sebaliknya, kelompok kedua menunjukkan karakteristik CT yang lebih maju, termasuk dekomposisi modular, pengenalan pola perilaku konvergensi yang lebih kuat, abstraksi umum model matematika, struktur kontrol algoritmik dinamis, dan mekanisme debugging eksplisit. Struktur sintaksis mereka mencerminkan penalaran komputasi yang lebih mendalam, mengintegrasikan kondisi validasi, pemeriksaan konvergensi otomatis, dan keluaran tabular terstruktur. Secara keseluruhan, hasil menunjukkan bahwa proyek Metode Numerik berbasis pemrograman dapat secara efektif mengembangkan literasi algoritmik dan pemikiran komputasi reflektif siswa. Perbedaan dalam profil komputasi menunjukkan bahwa artefak pemrograman siswa memberikan perspektif yang bermakna

JIMAT
Jurnal Ilmiah Matematika
DOI: https://doi.org/10.63976/jimat.v6i2.1140
e-ISSN: 2774-1729
Volume. 6. Nomor 2. Desember 2025
Halaman: 724 -738

untuk menilai kedalaman pemahaman konseptual dan prosedural mereka ketika menerjemahkan algoritma matematika menjadi kode yang dapat dieksekusi. Penelitian selanjutnya disarankan untuk mengembangkan instrumen asesmen digital yang mampu menilai proses berpikir mahasiswa secara otomatis, serta melakukan kajian longitudinal untuk melihat perkembangan kemampuan CT mahasiswa dari waktu ke waktu.

1. PENDAHULUAN

Perkembangan teknologi digital, otomatisasi, dan kecerdasan buatan (AI) mendorong perubahan besar dalam paradigma pendidikan. Kemampuan berpikir komputasi atau computational thinking (CT) muncul sebagai salah satu *core competencies* abad ke-21 yang menopang literasi digital, pemecahan masalah, penalaran matematis, dan inovasi teknologi dalam berbagai disiplin ilmu. Secara global, CT telah dipandang sebagai salah satu "dominant ways of thinking" di era modern. Analisis bibliometrik menunjukkan bahwa CT berada dalam kelompok lima cara berpikir yang paling sering menjadi fokus kajian ilmiah, sejajar dengan critical thinking, creative thinking, design thinking, dan systems thinking (Crilly, 2025). Hal ini menunjukkan bahwa CT telah menjadi kerangka berpikir lintas disiplin yang semakin mendasar bagi pendidikan. Dalam konteks pendidikan tinggi, mahasiswa tidak hanya dituntut memahami konsep matematis secara teoretis, tetapi juga mampu mengonversi prosedur matematis menjadi bentuk algoritma yang dapat dijalankan komputer (Khadijah & Suradi, 2025). Kemampuan tersebut menjadi bagian penting dari apa yang disebut berpikir komputasi (Computational Thinking / CT). Berpikir komputasi yaitu seperangkat keterampilan kognitif yang melibatkan pemecahan masalah secara sistematis melalui dekomposisi, pengenalan pola, abstraksi, dan perancangan algoritma (Brennan & Resnick, 2012; Shute et al., 2017).

Penguatan CT tidak hanya terjadi pada jenjang menengah dan tinggi, tetapi bahkan sejak pendidikan anak usia dini. Meta-analisis robotika pendidikan menemukan bahwa intervensi robotika secara konsisten meningkatkan CT anak usia 3–8 tahun (Alonso-García et al., 2024). Data Collection and Analysis (DCA) menjadi konteks efektif untuk menumbuhkan CT dan literasi matematika prasekolah (Lewis Presser et al., 2023). Pada konsep Narrative Computational Thinking, menunjukkan bahwa CT muncul ketika anak menyusun, mengevaluasi, dan merevisi alur cerita digital (Schlauch et al., 2025). Interaksi anak dengan agen media berbasis AI dapat meningkatkan transfer konsep CT secara signifikan (Y. Xu et al., 2025). CT dapat diperkenalkan melalui robotika, coding unplugged, dan pendekatan bermain yang sesuai tahap perkembangan (Curi et al., 2025; Su & Yang, 2023; Yang, 2025).

Hubungan antara CT dan matematika semakin kuat terlihat pada jenjang sekolah dasar dan menengah. CT berhubungan kuat dengan kelancaran aritmetika, kemampuan penalaran, dan kreativitas (W. Xu et al., 2022). Selain itu, CT merupakan prediktor signifikan mathematical creative thinking (Kim & Park, 2020). Sementara literasi matematika memengaruhi kreativitas melalui mekanisme CT seperti abstraksi dan representasi (Y. Xu et

JIMAT
Jurnal Ilmiah Matematika
DOI: https://doi.org/10.63976/jimat.v6i2.1140
e-ISSN: 2774-1729
Volume. 6. Nomor 2. Desember 2025
Halaman: 724 -738

al., 2025). Relational thinking siswa dalam operasi "difference" sangat berkaitan dengan komponen CT seperti pola, generalisasi, dan strategi algoritmik (Wilkie & Hopkins, 2024). Terdapat hubungan CT dengan mathematical habits of mind, dan kreativitas matematis dapat diperkaya melalui konteks budaya (Pérez-aracil et al., 2023; Suherman & Vidákovich, 2025). Pada jenjang yang lebih tinggi, CT banyak berkembang melalui game-based learning (GBL), constructionism, dan project-based learning (PBL). Pembuatan gim Scratch meningkatkan kemampuan algoritmik, debugging, dan logika kondisional siswa SMP (Troiano et al., 2025). CT-PBL meningkatkan minat jangka panjang siswa berbakat dalam software engineering (Jeong, 2025). Suatu meta-analisis menegaskan bahwa GBL meningkatkan HOTS matematika, yang merupakan komponen esensial CT (Anggoro et al., 2025). Selain itu, terdapat pentingnya peran CT dalam pembuatan artefak digital dan teknologi XR (MacCallum, 2025).

Dalam mata kuliah Metode Numerik, mahasiswa belajar berbagai algoritma penyelesaian masalah matematis yang tidak dapat diselesaikan secara analitik, seperti pencarian akar, sistem persamaan linear, interpolasi, dan integrasi numerik. Proses penyelesaian dalam mata kuliah ini menuntut kemampuan mahasiswa untuk memahami logika matematis dan struktur prosedural metode, serta menerjemahkan logika tersebut ke dalam bentuk algoritma dan sintaks program menggunakan bahasa pemrograman (misalnya Python, MATLAB, atau PHP). Tahap kedua inilah yang menjadi cerminan kemampuan berpikir komputasi, karena mahasiswa harus mampu mengidentifikasi komponen inti dari masalah matematis dan menyusunnya kembali menjadi urutan langkah logis yang dapat dijalankan komputer. Salah satu algoritma yang umum digunakan sebagai entry point bagi mahasiswa dalam memahami CT adalah Metode Bagi Dua (*Bisection Method*). Metode ini secara matematis sederhana, tetapi menuntut penerapan berpikir logis, sistematis, dan berulang (iteratif), yang merupakan ciri khas dari berpikir komputasi. Metode Bagi Dua digunakan untuk mencari akar dari suatu fungsi kontinu $f(x) = 0$ pada selang $[a, b]$ yang memenuhi $f(a) \cdot f(b) < 0$. Prinsip dasarnya adalah membagi selang menjadi dua bagian, mengevaluasi tanda fungsi pada titik tengah, dan mempersempit selang yang mengandung akar hingga mencapai toleransi tertentu.

Secara konseptual, langkah-langkah metode bagi dua ini bersifat matematis deterministik. Namun ketika dikonversi ke dalam bentuk algoritma komputer, mahasiswa harus menentukan variabel dan parameter awal, menyusun perulangan (*looping*) untuk menghitung nilai baru, membuat kondisi if–else untuk memilih subinterval, mengatur kriteria penghentian (*stopping criterion*), menyajikan hasil dalam bentuk output yang terstruktur. Proses ini melibatkan elemen-elemen utama berpikir komputasi yaitu *Decomposition* (memecah persoalan pencarian akar menjadi bagian-bagian kecil seperti input, proses, dan output), *Pattern Recognition* (mengenali pola perubahan tanda fungsi dan pola iterasi yang berulang), *Abstraction* (mengubah rumus matematis menjadi representasi simbolik yang dapat dijalankan komputer), *Algorithmic Thinking* (menyusun urutan logika yang efisien dan terstruktur hingga konvergen). Dengan demikian, implementasi Metode Bagi Dua bukan hanya latihan pemrograman, tetapi juga menjadi alat asesmen alami untuk melihat sejauh mana mahasiswa mampu berpikir komputasi dalam konteks matematis.

JIMAT
Jurnal Ilmiah Matematika
DOI: https://doi.org/10.63976/jimat.v6i2.1140
e-ISSN: 2774-1729
Volume. 6. Nomor 2. Desember 2025
Halaman: 724 -738

Meskipun integrasi antara matematika dan komputasi telah menjadi kebutuhan dalam kurikulum pendidikan matematika modern, dari hasil observasi, masih ditemukan beberapa kesenjangan di lapangan, antara lain mahasiswa cenderung fokus pada hasil perhitungan, bukan pada struktur algoritma yang mereka buat, pemahaman konseptual belum diiringi kemampuan berpikir prosedural dan algoritmik, sehingga kesalahan logika iterasi sering muncul dalam program. Dosen sering menilai dari sisi output (*running program*), bukan dari kualitas berpikir komputasi mahasiswa. Kesenjangan ini menunjukkan bahwa diperlukan analisis yang lebih mendalam tentang bagaimana mahasiswa berpikir saat mengonversi langkah-langkah matematis ke bentuk sintaks program, khususnya pada algoritma yang sederhana tetapi konseptual seperti Metode Bagi Dua.

Berpikir komputasi bukan hanya keterampilan teknis, melainkan bentuk baru dari problem-solving literacy yang mengombinasikan berpikir matematis, logika, dan pemodelan algoritmik (Wing, 2011). Sementara itu, dalam konteks pendidikan matematika dan sains, CT merupakan jembatan antara mathematical reasoning dan computational representation (Weintrop, Beheshti, et al., 2016). Hal ini didukung oleh studi yang menunjukkan bahwa CT melibatkan aspek metakognisi, monitoring strategi, dan debugging (Ocak et al., 2025; Yadav et al., 2017). Dalam konteks ini, mahasiswa pendidikan matematika perlu mengembangkan kemampuan konseptual dalam memahami model matematis, kemampuan prosedural dalam membangun struktur algoritma, kemampuan reflektif dalam mengevaluasi hasil program berdasarkan logika matematis. Ketiga kemampuan tersebut sejalan dengan kerangka *Computational Thinking Framework*.

Penelitian ini menjadi penting untuk mengidentifikasi profil berpikir komputasi mahasiswa melalui analisis sintaks suatu program, mendeskripsikan perbedaan strategi algoritmik dan abstraksi antar mahasiswa, sehingga dapat menjadi dasar pengembangan asesmen CT di bidang pendidikan matematika. Berdasarkan uraian latar belakang tentang pentingnya berpikir komputasi dalam mengonversi prosedur matematis ke bentuk algoritmik, maka fokus utama penelitian ini diarahkan untuk mengidentifikasi dan memetakan profil berpikir komputasi mahasiswa dalam konteks penyusunan sintaks program Metode Bagi Dua.

2. METODE PENELITIAN

Jenis dan Pendekatan Penelitian

Penelitian ini menggunakan pendekatan kualitatif deskriptif dengan bentuk studi kasus (*case study*). Pendekatan ini dipilih karena penelitian tidak berfokus pada hasil numerik atau uji hipotesis, melainkan bertujuan untuk mendalami proses berpikir mahasiswa dalam konteks tertentu, yaitu bagaimana mereka mengonversi prosedur matematis Metode Bagi Dua ke dalam bentuk algoritma program komputer. Pendekatan kualitatif memungkinkan peneliti untuk menggali makna, strategi, dan struktur berpikir komputasi (Computational Thinking/CT). Sedangkan bentuk studi kasus digunakan karena data diperoleh kemudian menjadi unit analisis khusus dengan karakteristik implementasi algoritma yang berbeda, namun berada dalam satu

JIMAT
Jurnal Ilmiah Matematika
DOI: https://doi.org/10.63976/jimat.v6i2.1140
e-ISSN: 2774-1729
Volume. 6. Nomor 2. Desember 2025
Halaman: 724 -738

mata kuliah, Metode Numerik pada Program Studi Pendidikan Matematika Universitas Negeri Makassar.

Lokasi dan Waktu Penelitian

Penelitian ini dilaksanakan di Program Studi Pendidikan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA), Universitas Negeri Makassar (UNM). Lokasi ini dipilih karena program studi tersebut telah menerapkan mata kuliah Metode Numerik yang mengintegrasikan konsep matematika dan pemrograman komputer, serta mendorong mahasiswa untuk menghasilkan proyek sintaks algoritmik. Waktu pelaksanaan penelitian dimulai pada semester genap tahun akademik 2023/2024.

Subjek Penelitian

Subjek penelitian terdiri atas dua kelompok mahasiswa (Kelompok 1 dan Kelompok 2) dari Program Studi Pendidikan Matematika yang mengikuti mata kuliah Metode Numerik. Kedua kelompok dipilih secara purposif berdasarkan kriteria, antara lain menyusun laporan proyek akhir Metode Numerik yang memuat algoritma dan sintaks lengkap Metode Bagi Dua (Bisection Method). Data penelitian dalam bentuk produk pemrograman mahasiswa berupa sintaks program (kode PHP) dari Metode Bagi Dua, deskripsi algoritma dan hasil running program, dan struktur logika perulangan, kondisi, dan pengendalian kesalahan yang digunakan mahasiswa. Melalui analisis objek tersebut, peneliti menelusuri indikator berpikir komputasi yang muncul dalam setiap tahapan penulisan kode dan penyusunan algoritma.

Teknik Pengumpulan Data

Pengumpulan data dilakukan melalui tiga tahap utama yaitu dokumentasi, analisis produk hasil proyek (artefak digital), dan analisis perbandingan kasus. Dokumentasi dilakukan dengan mengumpulkan dua laporan akhir proyek mahasiswa yang memuat sintaks dan hasil implementasi algoritma metode bagi dua, dan menyeleksi bagian kode yang relevan untuk analisis CT, seperti perulangan, kondisi logika, dan perhitungan fungsi. Analisis Artefak Digital dilakukan dengan melakukan code reading dan code interpretation terhadap setiap sintaks program mahasiswa, dan mengidentifikasi struktur kode yang menggambarkan komponen berpikir komputasi, seperti decomposition, pattern recognition, abstraction, dan algorithmic thinking. Analisis Perbandingan Kasus (Cross-case Analysis) dengan membandingkan hasil analisis antar dua kelompok untuk mengidentifikasi perbedaan strategi berpikir dan pendekatan algoritmik.

Teknik Analisis Data

Analisis data menggunakan pendekatan Computational Thinking Framework dengan Analisis dilakukan melalui tiga tahapan utama yaitu Reduksi Data, Koding dan Kategorisasi, Interpretasi dan Sintesis (Brennan & Resnick, 2012; Shute et al., 2017). Pada Reduksi Data,

JIMAT
Jurnal Ilmiah Matematika
DOI: https://doi.org/10.63976/jimat.v6i2.1140
e-ISSN: 2774-1729
Volume. 6. Nomor 2. Desember 2025
Halaman: 724 -738

Peneliti menyeleksi bagian-bagian sintaks yang merepresentasikan proses berpikir mahasiswa. Fokus reduksi diarahkan pada penentuan input dan parameter awal (indikator decomposition), struktur perulangan dan kondisi logika (indikator pattern recognition), translasi rumus ke operasi komputasi (indikator abstraction), urutan kontrol program (indikator algorithmic thinking), dan pengendalian galat dan hasil (automation dan debugging). Untuk Koding dan Kategorisasi, setiap baris atau blok sintaks dikodekan berdasarkan komponen CT yang muncul. Hasil koding disajikan dalam tabel analisis yang mengaitkan bagian kode dengan indikator CT yang relevan. Selanjutnya, Interpretasi dan Sintesis, bertujuan untuk menafsirkan profil berpikir komputasi mahasiswa berdasarkan hasil koding dan kategori yang diperoleh. Analisis dilakukan dengan menginterpretasi keterkaitan antara bagian kode dan proses berpikir, menyusun narasi deskriptif untuk setiap komponen CT, membandingkan pola berpikir antar kelompok, menyimpulkan kecenderungan CT yang dominan pada masing-masing kelompok.

Triangulasi Data

Untuk menjaga keabsahan data, dilakukan Triangulasi sumber, yaitu membandingkan hasil analisis dengan dokumentasi laporan dan pedoman proyek, dan catatan analisis agar prosesnya dapat ditelusuri ulang. Agar proses analisis lebih sistematis, indikator yang digunakan untuk mengidentifikasi komponen CT ditetapkan pada Tabel 1.

Tabel 1. Indikator Komponen CT

Komponen CT	Indikator Analisis
Decomposition	Mahasiswa memecah permasalahan ke dalam bagian input, proses, dan output
Pattern Recognition	Mahasiswa mengenali pola perubahan tanda fungsi dan pola iterasi
Abstraction	Mahasiswa menerjemahkan model matematis ke bentuk komputasi
Algorithmic Thinking	Mahasiswa menyusun urutan logika *if–else–loop* yang efisien
Automation	Mahasiswa mengotomatisasi perhitungan hingga mencapai konvergensi
Debugging / Refinement	Mahasiswa memverifikasi hasil, memperbaiki kesalahan, dan menampilkan output akhir

3. HASIL DAN PEMBAHASAN

Hasil Penelitian

Hasil analisis profil berpikir komputasi mahasiswa berdasarkan dua hasil proyek pemrograman kelompok (K1 dan K2) pada materi Metode Bagi Dua (Bisection Method) dalam mata kuliah Metode Numerik. Analisis difokuskan pada kemampuan mahasiswa dalam mengonversi prosedur matematis ke dalam sintaks program komputer menggunakan bahasa PHP. Proses analisis mengacu pada kerangka Computational Thinking (CT) menurut Brennan dan Resnick dan Shute, dkk, yang terdiri dari enam komponen utama: decomposition, pattern recognition, abstraction, algorithmic thinking, automation, dan debugging (Brennan & Resnick,

JIMAT
Jurnal Ilmiah Matematika
DOI: https://doi.org/10.63976/jimat.v6i2.1140
e-ISSN: 2774-1729
Volume. 6. Nomor 2. Desember 2025
Halaman: 724 -738

2012; Shute et al., 2017). Data penelitian diperoleh dari dua kelompok mahasiswa yang masing-masing mengembangkan sintaks program Metode Bagi Dua pada Tabel 2 berikut.

Tabel 2. Data Penelitian

Kelompok	Bahasa Pemrograman	Karakteristik Sintaks	Format Output
K1	PHP dasar	Struktur linear, iterasi manual	Teks konsol sederhana
K2	PHP + HTML	Struktur modular, iterasi otomatis, menampilkan tabel hasil	Tabel HTML dengan hasil konvergen

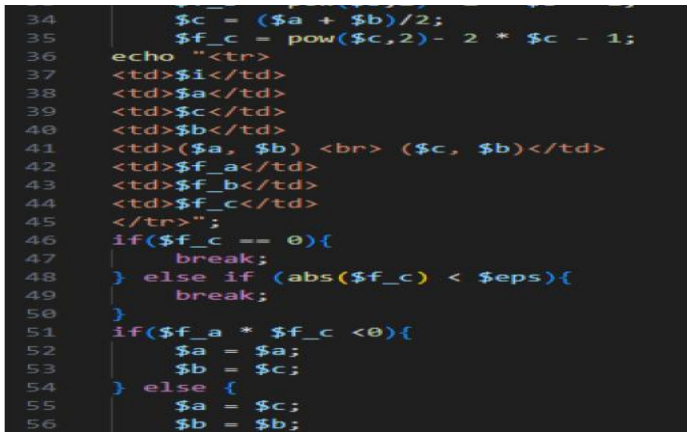
Kedua kelompok menggunakan fungsi yang sama, yaitu: $f(x) = x^2 - 5x + 2$, dan sama-sama menetapkan batas awal a=1, b=5, serta toleransi $\epsilon = 0.0001$.

Analisis Berpikir Komputasi Kelompok 1

Kelompok 1 menyusun algoritma secara berurutan berdasarkan langkah matematis manual. Menentukan nilai batas awal, menghitung titik tengah, mengevaluasi tanda fungsi, dan memperbarui batas interval. Struktur kode bersifat linear tanpa modularisasi fungsi. Pada tahap Decomposition, Mahasiswa K1 menunjukkan kemampuan dasar dalam memecah masalah ke dalam bagian-bagian sederhana, yaitu:

a. Input: variabel a, b, dan ϵ;
b. Proses: iterasi perhitungan nilai tengah dan evaluasi tanda fungsi;
c. Output: nilai akar hampiran c.

Blok program disusun sesuai urutan langkah algoritma matematis. Hal ini menunjukkan bahwa mahasiswa mampu mengidentifikasi elemen-elemen dasar dari prosedur matematis dan mengimplementasikannya secara langsung dalam program. Kutipan sintaks pada Gambar 1.



Gambar 1. Kutipan Sintaks Kelompok 1

Pada Gambar 1 ini menunjukkan segmentasi logis input–proses–output.

Pada tahap Pattern Recognition, pengenalan pola terlihat pada penggunaan kondisi if ($f_a * $f_c < 0) yang menjadi acuan dalam menentukan subinterval baru. Mahasiswa memahami bahwa perubahan tanda fungsi menandakan adanya akar di antara dua titik, dan pola

JIMAT
Jurnal Ilmiah Matematika
DOI: https://doi.org/10.63976/jimat.v6i2.1140
e-ISSN: 2774-1729
Volume. 6. Nomor 2. Desember 2025
Halaman: 724 -738

tersebut diulang dalam setiap iterasi. Namun, belum ada eksplorasi terhadap pola konvergensi (misalnya menghitung selisih $|c_n - c_{n-1}|$), sehingga pola iteratif hanya diimplementasikan secara simbolik. Pada tahap Abstraction, Abstraksi dilakukan dengan mengubah rumus matematis $c = (a + b)/2$ menjadi operasi dalam bahasa pemrograman. Mahasiswa juga menyederhanakan konsep *kriteria berhenti* menjadi "batas jumlah iterasi" tanpa menerapkan perhitungan galat. Hal ini menunjukkan bahwa mahasiswa telah melakukan abstraksi parsial, mampu merepresentasikan model matematis, namun belum mengeneralisasi kondisi konvergensi secara penuh.

Pada tahap Algorithmic Thinking, Mahasiswa K1 telah mampu menyusun urutan logika perhitungan dengan kontrol *if–else* dan *looping*. Meskipun sederhana, struktur algoritmiknya menunjukkan pemahaman tentang bagaimana komputer menjalankan prosedur matematis secara berulang untuk mencapai solusi mendekati akar. Logika kontrol yang dilakukan pada Gambar 2. Iterasi dibatasi hingga 500 kali, menunjukkan pendekatan prosedural tanpa kriteria dinamis.

```
31  for ($i = 1; $i<=500; $i++) {
32      $f_a = pow($a,2)- 2 * $a - 1;
33      $f_b = pow($b,2)- 2 * $b - 1;
34      $c = ($a + $b)/2;
35      $f_c = pow($c,2)- 2 * $c - 1;
```

Gambar 2. Logika Kontrol K1

Untuk tahap Automation, program berjalan otomatis dalam menghitung titik tengah dan memperbarui nilai batas, tetapi masih bergantung pada jumlah iterasi tetap. Ini mencerminkan *automation* dasar, mahasiswa memahami prinsip pengulangan otomatis, namun belum menerapkan mekanisme penghentian berbasis galat konvergensi. Debugging dan Refinement, dari hasil *running program*, output yang dihasilkan konvergen mendekati akar. Hal ini menunjukkan mahasiswa mampu mengevaluasi hasil dan memperbaiki kesalahan sintaks dasar. Namun, tidak ditemukan indikator eksplisit dari proses *debugging* yang reflektif (misalnya menambahkan pesan kesalahan atau kondisi validasi tambahan). Berdasarkan analisis tersebut, maka ditemukan bahwa mahasiswa telah menguasai CT tingkat dasar, khususnya *decomposition* dan *algorithmic thinking*, tetapi belum sepenuhnya menginternalisasi *abstraction* dan *debugging*.

Analisis Berpikir Komputasi Kelompok 2

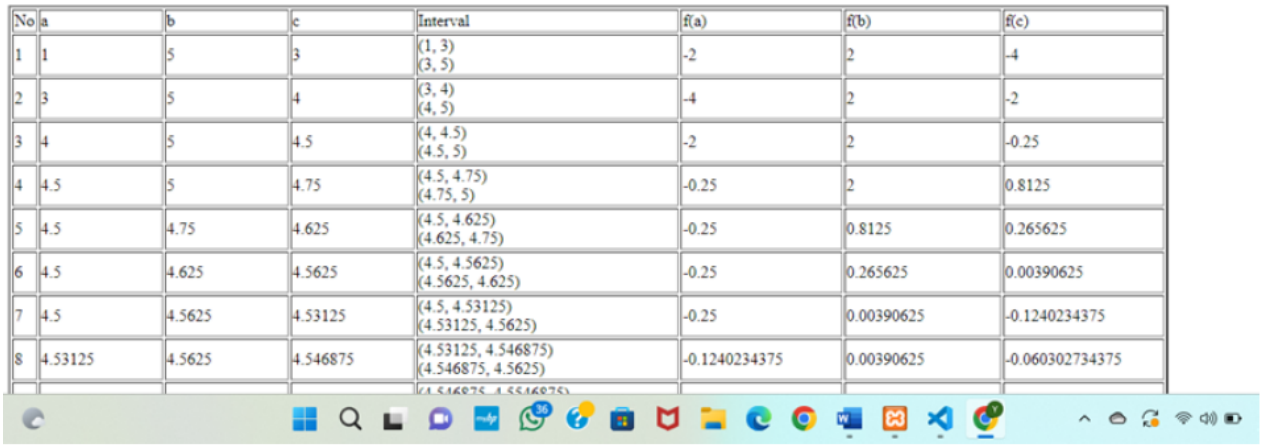
Kelompok 2 menampilkan implementasi yang lebih kompleks dan sistematis. Sintaks disusun modular dengan integrasi tampilan tabel HTML yang menampilkan hasil iterasi. Kelompok ini menunjukkan kesadaran algoritmik yang lebih tinggi dan efisiensi dalam otomatisasi program. Pada tahap Decomposition, Mahasiswa K2 secara eksplisit membagi program menjadi bagian-bagian fungsional, yaitu:

1. Definisi Fungsi: menggunakan ekspresi pow($x,2) - 5*$x + 2;
2. Inisialisasi: variabel batas a, b, dan ϵ;

JIMAT
Jurnal Ilmiah Matematika
DOI: https://doi.org/10.63976/jimat.v6i2.1140
e-ISSN: 2774-1729
Volume. 6. Nomor 2. Desember 2025
Halaman: 724 -738

3. Iterasi: menggunakan struktur for ($i=1; $i<=50; $i++);
4. Keluaran: tabel hasil berisi nilai a, b, c, f(a), f(b), dan f(c).

Struktur ini menunjukkan bahwa mahasiswa tidak hanya menyalin langkah matematis, tetapi juga memetakan ulang proses matematis menjadi sistem informasi terstruktur, sebuah indikator kuat *decomposition* dalam konteks CT. Pada tahap *Pattern Recognition*, pola iteratif dan pola tanda fungsi dikenali secara jelas. Mahasiswa menyadari bahwa pola perubahan tanda menentukan subinterval akar, dan mereka memperkuatnya dengan visualisasi tabel berulang di setiap iterasi pada Gambar 3.



No	a	b	c	Interval	f(a)	f(b)	f(c)
1	1	5	3	(1, 3) (3, 5)	-2	2	-4
2	3	5	4	(3, 4) (4, 5)	-4	2	-2
3	4	5	4.5	(4, 4.5) (4.5, 5)	-2	2	-0.25
4	4.5	5	4.75	(4.5, 4.75) (4.75, 5)	-0.25	2	0.8125
5	4.5	4.75	4.625	(4.5, 4.625) (4.625, 4.75)	-0.25	0.8125	0.265625
6	4.5	4.625	4.5625	(4.5, 4.5625) (4.5625, 4.625)	-0.25	0.265625	0.00390625
7	4.5	4.5625	4.53125	(4.5, 4.53125) (4.53125, 4.5625)	-0.25	0.00390625	-0.1240234375
8	4.53125	4.5625	4.546875	(4.53125, 4.546875) (4.546875, 4.5625)	-0.1240234375	0.00390625	-0.060302734375

Gambar 3. Iterasi Metode Bagi Dua K2

Melalui Gambar 3 ini, mahasiswa membangun kesadaran pola konvergensi numerik, indikator *pattern recognition* yang lebih tinggi dibanding K1. Pada tahap Abstraction, Kelompok 2 mengabstraksi konsep matematis lebih baik dengan menulis perhitungan umum dan parameterisasi fungsi. Mereka tidak hanya memindahkan rumus ke kode, tetapi juga menambahkan mekanisme evaluasi hasil, seperti pada potongan sintaks Gambar 4.

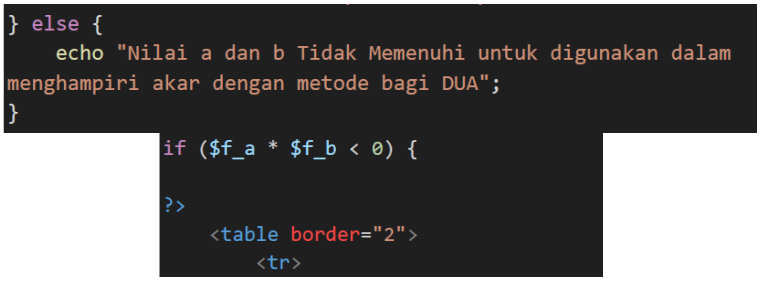
```
if ($f_c == 0) {
    break;
}
if ($f_a * $f_c < 0) {
    $a = $a;
    $b = $c;
} else {
    $a = $c;
    $b = $b;
}
```

Gambar 4. Potongan Sintaks K2

Gambar 4 menunjukkan kemampuan *abstraction and generalization*, mahasiswa menyadari syarat matematis yang harus terpenuhi agar algoritma valid. Pada tahap Algorithmic Thinking, program K2 mencerminkan *algorithmic thinking* yang matang. Mereka menyusun

JIMAT
Jurnal Ilmiah Matematika
DOI: https://doi.org/10.63976/jimat.v6i2.1140
e-ISSN: 2774-1729
Volume. 6. Nomor 2. Desember 2025
Halaman: 724 -738

urutan logika dengan jelas (*input* → *proses* → *kondisi* → *output*), menggunakan struktur kontrol yang efisien (for loop dan if–else), menyisipkan mekanisme penghentian otomatis saat kondisi konvergensi tercapai. Potongan Logika control pada Gambar 5.



Gambar 5. Potongan Logika Kontrol K2

Logika pada Gambar 5 ini menunjukkan pemahaman mendalam tentang konsep konvergensi dan pemilihan interval baru secara sistematis. Untuk tahap Automation, Program Kelompok 2 menunjukkan otomatisasi penuh, menghitung nilai f(a), f(b), f(c) dalam setiap iterasi tanpa intervensi manual, menghentikan perulangan secara otomatis saat akar ditemukan (break statement), dan menampilkan hasil secara langsung dalam bentuk tabel. Automasi ini mengindikasikan bahwa mahasiswa telah memahami konsep pemrograman sebagai representasi proses matematis berulang, hal ini merupakan inti dari CT dalam konteks numerik.

Pada tahap Debugging dan Refinement, Kelompok 2 menunjukkan refleksi terhadap potensi kesalahan dengan menambahkan kondisi validasi if ($f_a * $f_b < 0) dan menampilkan pesan kesalahan bila interval tidak memenuhi syarat akar. Ini mencerminkan kesadaran *debugging* yang eksplisit. Selain itu, hasil output program menampilkan nilai akar yang mendekati hasil teoritis, menunjukkan proses *refinement* terhadap logika iterasi. Berdasarkan analisis tersebut, maka ditemukan bahwa mahasiswa telah menunjukkan CT yang lebih komprehensif, dengan kemampuan *decomposition*, *pattern recognition*, *abstraction*, *algorithmic thinking*, dan *automation* yang tinggi, serta kesadaran *debugging* yang lebih baik dibanding kelompok 1.

Tabel 3. Perbandingan Profil CT Kelompok 1 dan 2

Komponen CT	Kelompok 1	Kelompok 2	Analisis Perbandingan
Decomposition	Linear, mengikuti urutan manual	Modular dan sistematis	K2 lebih terstruktur dalam memisah fungsi program
Pattern Recognition	Mengenali pola tanda fungsi	Mengenali pola konvergensi dan memvisualisasikannya	K2 memiliki kesadaran pola numerik lebih kuat
Abstraction	Translasi rumus langsung ke sintaks	Generalisasi rumus dan kondisi matematis	K2 menunjukkan tingkat abstraksi lebih tinggi
Algorithmic Thinking	Urutan sederhana berbasis perulangan tetap	Struktur logika kontrol dinamis dan efisien	K2 menerapkan algoritma lebih optimal
Automation	Iterasi otomatis dasar	Otomatisasi penuh hingga kondisi berhenti	K2 lebih matang dalam otomatisasi

JIMAT
Jurnal Ilmiah Matematika
DOI: https://doi.org/10.63976/jimat.v6i2.1140
e-ISSN: 2774-1729
Volume. 6. Nomor 2. Desember 2025
Halaman: 724 -738

Komponen CT	Kelompok 1	Kelompok 2	Analisis Perbandingan
Debugging	Evaluasi sederhana terhadap output	Validasi dan pesan kesalahan eksplisit	K2 memiliki refleksi dan kontrol kesalahan lebih baik

Pembahasan

Analisis menunjukkan bahwa proses menulis sintaks algoritma Metode Bagi Dua tidak hanya bersifat teknis, tetapi mencerminkan proses berpikir tingkat tinggi. Berdasarkan teori (Brennan & Resnick, 2012), berpikir komputasi tidak terlepas dari kemampuan kognitif untuk merepresentasikan model matematis secara sistematis, mengembangkan logika procedural, dan mengevaluasi hasil dengan refleksi terhadap kesalahan (debugging). Berdasarkan hasil penelitian, diperoleh sejumlah temuan yang mencerminkan profil berpikir komputasi mahasiswa pendidikan matematika pada konteks pembelajaran Metode Numerik.

1. Kemampuan *Decomposition* (Pemecahan Masalah), terlihat dari mahasiswa pada kedua kelompok mampu mengidentifikasi dan memecah permasalahan pencarian akar fungsi menjadi beberapa langkah logis: menentukan nilai awal a dan b, menghitung nilai fungsi f(a) dan f(b), mencari titik tengah $c = (a + b)/2$, serta mengevaluasi tanda fungsi untuk mempersempit interval. Namun, Kelompok 2 menunjukkan kemampuan dekomposisi yang lebih terstruktur dengan membagi program ke dalam blok input, proses, dan output secara modular. Ini menunjukkan bahwa mahasiswa tidak hanya meniru prosedur matematis, tetapi juga memahami struktur sistematis dari algoritma komputasi.
2. Kemampuan *Pattern Recognition* (Pengenalan Pola). Kedua kelompok mengenali pola dasar tanda fungsi $f(a) \cdot f(b) < 0$ untuk menentukan keberadaan akar. Tetapi, Kelompok 2 memperluas pengenalan pola ini menjadi pola konvergensi numerik dengan menampilkan hasil iteratif dalam tabel HTML yang memperlihatkan pola penyempitan interval dan nilai akar yang semakin mendekati hasil sebenarnya. Hal ini menunjukkan bahwa mahasiswa pada kelompok tersebut mulai berpikir reflektif terhadap perilaku fungsi dan pola hasil komputasi, bukan sekadar menjalankan perintah berulang.
3. Kemampuan *Abstraction* (Abstraksi Konseptual). Mahasiswa mampu mengubah representasi matematis menjadi bentuk komputasi. Mereka menuliskan rumus $c = (a + b)/2$ dan kondisi $f(a) \cdot f(b) < 0$ sebagai ekspresi logis dalam bahasa pemrograman PHP. Namun, Kelompok 1 masih melakukan abstraksi pada tingkat *translation* (langsung menerjemahkan rumus ke kode), sementara Kelompok 2 sudah mencapai tingkat *generalization*, yaitu dengan memformulasikan fungsi dalam bentuk parameterisasi dan menambahkan validasi terhadap kondisi matematis sebelum iterasi dilakukan. Abstraksi pada tingkat ini menandakan bahwa mahasiswa mulai memahami peran algoritma sebagai model umum yang dapat diterapkan untuk berbagai fungsi kontinu.
4. Kemampuan *Algorithmic Thinking* (Berpikir Algoritmik).Semua mahasiswa menunjukkan kemampuan menyusun urutan langkah algoritmik yang benar. Mereka memahami bahwa

JIMAT
Jurnal Ilmiah Matematika
DOI: https://doi.org/10.63976/jimat.v6i2.1140
e-ISSN: 2774-1729
Volume. 6. Nomor 2. Desember 2025
Halaman: 724 -738

proses iterasi berulang perlu dikontrol melalui *looping* dan *conditional branching* (if–else). Perbedaan mendasar muncul pada kualitas struktur logika, dimana pada Kelompok 1, algoritma masih bersifat linear dengan batas iterasi tetap, dan pada Kelompok 2, algoritma telah disusun secara efisien, dinamis, dan dilengkapi dengan mekanisme penghentian otomatis (break ketika akar ditemukan). Dengan demikian, kelompok 2 menunjukkan *algorithmic thinking* yang lebih matang, ditandai dengan kemampuan mengatur alur logika agar efisien dan dapat beradaptasi terhadap kondisi program.

5. Kemampuan *Automation* (Otomatisasi Proses). Kedua kelompok telah memahami bahwa proses komputasi bersifat berulang dan otomatis. Namun, Kelompok 1 masih menggunakan pendekatan *bounded iteration* (jumlah iterasi tetap), sedangkan Kelompok 2 telah mengimplementasikan *conditional automation* (otomatisasi berbasis kondisi konvergensi). Hal ini memperlihatkan adanya peningkatan dari sekadar menjalankan pengulangan ke arah pemahaman tentang konsep otomatisasi yang disertai logika kendali, yang merupakan sebuah elemen kunci dalam berpikir komputasi tingkat tinggi.
6. Kemampuan *Debugging* dan *Refinement* (Evaluasi dan Penyempurnaan). Dalam tahap akhir, mahasiswa menunjukkan kemampuan untuk mengevaluasi hasil perhitungan dan memperbaiki kesalahan logika. Kelompok 2 menambahkan pesan kesalahan (*error message*) ketika syarat $f(a) \cdot f(b) < 0$ tidak terpenuhi, sedangkan Kelompok 1 hanya memverifikasi output akhir tanpa mekanisme kontrol kesalahan eksplisit. Kehadiran kondisi validasi ini menunjukkan munculnya kesadaran *debugging reflective*, yaitu kemampuan untuk tidak hanya memperbaiki hasil, tetapi juga memeriksa kebenaran struktur logika program terhadap prinsip matematisnya.

Hasil penelitian ini memperkuat pandangan Weintrop, dkk (2016) bahwa integrasi CT dalam pembelajaran matematika mendorong terbentuknya computational modeling mindset, yaitu kemampuan mahasiswa untuk memahami dan membangun representasi matematis dalam bentuk algoritmik. Perbedaan hasil antara K1 dan K2 juga menunjukkan adanya gradasi tingkat CT mahasiswa, dari CT dasar (reproduktif) menuju CT menengah (produktif–reflektif). K1 cenderung meniru langkah matematis tanpa modifikasi logika, sedangkan K2 mulai berpikir reflektif dan mengembangkan struktur kontrol yang lebih efisien. Kemampuan analisis berpikir komputasi ini, dapat menjadi bekal keterampilan mahasiswa di masa depan, hal ini sesuai dengan hasil temuan yang menemukan bahwa literatur global menunjukkan bahwa CT berperan penting dalam membangun kemampuan representasi, analisis, dan generalisasi yang diperlukan dalam berbagai domain STEM dan non-STEM (Xiaolei & Teng, 2024).

Berdasarkan hasil penelitian, diperoleh temuan utama yaitu semua mahasiswa menunjukkan kemampuan berpikir komputasi dasar, terutama pada aspek decomposition dan algorithmic thinking. Mahasiswa yang menggunakan pendekatan sintaks modular (K2) menampilkan tingkat berpikir komputasi lebih tinggi, dengan indikator abstraction, automation, dan debugging yang lebih lengkap. Perbedaan struktur sintaks mencerminkan perbedaan

JIMAT
Jurnal Ilmiah Matematika
DOI: https://doi.org/10.63976/jimat.v6i2.1140
e-ISSN: 2774-1729
Volume. 6. Nomor 2. Desember 2025
Halaman: 724 -738

strategi berpikir, dimana K1 berorientasi langkah matematis, dan K2 berorientasi pada sistem logika komputasi. Pembelajaran Metode Numerik berbasis proyek coding dapat menjadi sarana efektif untuk menumbuhkan literasi algoritmik dan refleksi komputasi mahasiswa pendidikan matematika.

4. KESIMPULAN

Hasil analisis terhadap dua kelompok mahasiswa (K1 dan K2) menunjukkan bahwa keduanya mampu menerapkan unsur-unsur berpikir komputasi dalam mengonversi prosedur matematis *Metode Bagi Dua* ke dalam sintaks program komputer. Namun, kualitas penerapannya berbeda. Pertama, pada aspek *decomposition*, kedua kelompok mampu memecah masalah pencarian akar ke dalam langkah-langkah logis, tetapi Kelompok 2 lebih sistematis karena menyusun program secara modular ke dalam input–proses–output. Kedua, pada *pattern recognition*, kedua kelompok mengenali pola tanda fungsi, namun Kelompok 2 mampu mengidentifikasi pola konvergensi melalui tampilan tabel iteratif sehingga menunjukkan refleksi yang lebih dalam terhadap perilaku fungsi numerik. Ketiga, dalam *abstraction*, Kelompok 1 hanya menerjemahkan langsung rumus ke kode, sedangkan Kelompok 2 sudah melakukan generalisasi dan menambahkan validasi matematis, menunjukkan pemahaman algoritmik yang lebih konseptual. Keempat, pada *algorithmic thinking*, seluruh mahasiswa mampu menyusun logika iteratif yang benar, tetapi Kelompok 2 lebih efisien karena mengatur alur program secara dinamis dan adaptif. Kelima, dalam *automation*, Kelompok 1 masih tergantung pada jumlah iterasi tetap, sedangkan Kelompok 2 menerapkan otomatisasi berbasis kondisi konvergensi, yang lebih tepat secara numerik. Keenam, pada aspek *debugging*, Kelompok 2 menunjukkan kemampuan jauh lebih baik dengan menambahkan pesan kesalahan dan validasi interval, sedangkan Kelompok 1 hanya melakukan pemeriksaan hasil akhir. Secara keseluruhan, kedua kelompok menunjukkan kemampuan berpikir komputasi, namun Kelompok 2 menampilkan profil CT yang lebih matang, terstruktur, dan reflektif dibanding Kelompok 1. Berdasarkan hasil penelitian ini, rekomendasi penelitian selanjutnya yaitu sebaiknya melakukan penelitian pengembangan instrumen asesmen digital yang mampu menilai proses berpikir mahasiswa secara otomatis, serta melakukan kajian longitudinal untuk melihat perkembangan kemampuan CT mahasiswa dari waktu ke waktu.

REFERENSI

Alonso-García, S., Rodríguez Fuentes, A. V., Ramos Navas-Parejo, M., & Victoria-Maldonado, J. J. (2024). Enhancing computational thinking in early childhood education with educational robotics: A meta-analysis. *Heliyon*, *10*(13). https://doi.org/10.1016/j.heliyon.2024.e33249

Anggoro, B. S., Dewantara, A. H., Suherman, S., Muhammad, R. R., & Saraswati, S. (2025). Effect of game-based learning on students' mathematics high order thinking skills: A meta-analysis. *Revista de Psicodidáctica (English Ed.)*, *30*(1), 500158. https://doi.org/10.1016/j.psicoe.2024.500158

JIMAT
Jurnal Ilmiah Matematika
DOI: https://doi.org/10.63976/jimat.v6i2.1140
e-ISSN: 2774-1729
Volume. 6. Nomor 2. Desember 2025
Halaman: 724 -738

Brennan, K., & Resnick, M. (2012). New Frameworks for Studying and Assessing the Development of Computational Thinking. *Proceedings of the 2012 Annual Meeting of the American Educational Research Association (AERA)*.

Crilly, N. (2025). Critical thinking, creative thinking, systems thinking and many more: a comparative bibliometric analysis of prevalence and distribution. *Thinking Skills and Creativity*, *59*(September 2025), 102014. https://doi.org/10.1016/j.tsc.2025.102014

Curi, M. E., Gerosa, A., Viera, M., & Carboni, A. (2025). A review of computational thinking interventions in upper elementary education. *Computers and Education Open*, *9*(March), 100252. https://doi.org/10.1016/j.caeo.2025.100252

Jeong, H. (2025). Supporting interest development in gifted software education through computational thinking and project-based learning. *Computers and Education Open*, *9*(April), 100282. https://doi.org/10.1016/j.caeo.2025.100282

Khadijah, & Suradi. (2025). Penalaran Pada Pemecahan Akar Non- Linear : Review Integratif Literasi. *Pedagogy : Jurnal Pendidikan Matematika*, *10*(4), 1742–1758.

Kim, H., & Park, N. (2020). STEAM-Based Assessment Framework: Designing Tasks for Reflective and Creative Reasoning. *International Journal of Science Education*, *42*(8), 1243–1261. https://doi.org/10.1080/09500693.2020.1748259

Lewis Presser, A. E., Young, J. M., Rosenfeld, D., Clements, L. J., Kook, J. F., Sherwood, H., & Cerrone, M. (2023). Data collection and analysis for preschoolers: An engaging context for integrating mathematics and computational thinking with digital tools. *Early Childhood Research Quarterly*, *65*(June), 42–56. https://doi.org/10.1016/j.ecresq.2023.05.012

MacCallum, K. (2025). Integrating computational thinking through digital creation: The (TPAC)2K model. *Teaching and Teacher Education*, *161*(October 2024), 105056. https://doi.org/10.1016/j.tate.2025.105056

Ocak, C., Yadav, A., & Rich, K. M. (2025). Exploring young children's metacognition during unplugged computational thinking. *International Journal of Child-Computer Interaction*, *45*(August 2024), 100767. https://doi.org/10.1016/j.ijcci.2025.100767

Pérez-aracil, J., Hernández-díaz, A. M., Madalin, C., & Salcedo-sanz, S. (2023). Improving numerical methods for the steel yield strain calculation in reinforced concrete members with Machine Learning algorithms. *Expert Systems With Applications*, *225*(December 2022), 119987. https://doi.org/10.1016/j.eswa.2023.119987

Schlauch, M., Sylla, C., & Gil, M. (2025). More than words: Conceptualizing narrative computational thinking based on a multicase study. *International Journal of Child-Computer Interaction*, *43*(October 2024). https://doi.org/10.1016/j.ijcci.2024.100704

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying Computational Thinking. *Educational Research Review*, *22*, 74–100.

Su, J., & Yang, W. (2023). A systematic review of integrating computational thinking in early childhood education. *Computers and Education Open*, *4*(December 2022), 100122. https://doi.org/10.1016/j.caeo.2023.100122

Suherman, S., & Vidákovich, T. (2025). Ethnomathematical test for mathematical creative thinking. *Journal of Creativity*, *35*(2). https://doi.org/10.1016/j.yjoc.2025.100099

Troiano, G. M., Abdollahi, A., Cassidy, M., Puttick, G., Machado, T., & Harteveld, C. (2025). Leveling the computational playing field: Inquiring about factors predicting computational thinking in constructionist game-based learning. *Computers and Education*, *237*(May), 105347. https://doi.org/10.1016/j.compedu.2025.105347

JIMAT
Jurnal Ilmiah Matematika
DOI: https://doi.org/10.63976/jimat.v6i2.1140
e-ISSN: 2774-1729
Volume. 6. Nomor 2. Desember 2025
Halaman: 724 -738

Weintrop, D., Beheshti, E., Horn, & Al, M. et. (2016). Defining computational thinking for mathematics and science. *Journal of Science Education and Technology*, *25*, 127–147. https://link.springer.com/article/10.1007/s10956-015-9581-5?utm_source=chatgpt.com#citeas

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147.

Wilkie, K. J., & Hopkins, S. (2024). Primary students' relational thinking and computation strategies with concrete-to-symbolic representations of subtraction as difference. *Journal of Mathematical Behavior*, *73*(December 2023), 101121. https://doi.org/10.1016/j.jmathb.2023.101121

Wing, J. M. (2011). Research Notebook: Computational Thinking—What and Why? *The Link Magazine*.

Xiaolei, S., & Teng, M. F. (2024). Three-wave cross-lagged model on the correlations between critical thinking skills, self-directed learning competency and AI-assisted writing. *Thinking Skills and Creativity*, *52*(October 2023). https://doi.org/10.1016/j.tsc.2024.101524

Xu, W., Geng, F., & Wang, L. (2022). Relations of computational thinking to reasoning ability and creative thinking in young children: Mediating role of arithmetic fluency. *Thinking Skills and Creativity*, *44*(August 2021), 101041. https://doi.org/10.1016/j.tsc.2022.101041

Xu, Y., Pan, E. Z., Levine, J., He, K., Thomas, T., & Warschauer, M. (2025). The effects of interactions with AI-enhanced media characters on learning computational thinking. *Learning and Instruction*, *98*(May), 102149. https://doi.org/10.1016/j.learninstruc.2025.102149

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2017). Computational Thinking in Teacher Education. *ACM Transactions on Computing Education (TOCE)*, *17*(4), 1–16.

Yang, W. (2025). A three-phase professional development approach to improving robotics pedagogical knowledge and computational thinking attitude of early childhood teachers. *Computers and Education*, *231*(December 2024). https://doi.org/10.1016/j.compedu.2025.105282