


Bridging Theory and Practice: A Review of CT in K-12 Problem-Solving

Dinda Alya Maghfira^{1*}, Suprih Widodo², Ulva Elviani³

^{1,2,3} Program Studi Pendidikan Sistem dan Teknologi Informasi, Universitas Pendidikan Indonesia, Jalan Dr. Setiabudhi No. 229, Bandung, Jawa Barat

E-mail: dindaalya@upi.edu

* Corresponding Author

 <https://doi.org/10.31004/jerkin.v4i3>

ARTICLE INFO

Article history

Received: 23 Nov 2025

Revised: 05 Dec 2025

Accepted: 3 Jan 2025

Kata Kunci:

Berpikir Komputasional,
Pemecahan Masalah,
Pendidikan K-12

Keywords:

Computational Thinking,
Problem Solving, K-12
Education



ABSTRACT

Era digital menuntut siswa menguasai keterampilan abad ke-21, dengan *computational thinking* (CT) sebagai landasan kognitif fundamental untuk *problem solving*. Meskipun kebijakan seperti Kurikulum Merdeka telah mengintegrasikan CT, kesenjangan implementasi di lapangan masih signifikan akibat kurangnya panduan komprehensif. Penelitian ini bertujuan mensintesis bukti empiris tentang penerapan CT dalam meningkatkan kemampuan *problem solving* siswa K-12 melalui *Systematic Literature Review* (SLR). Menggunakan database ERIC, SpringerLink, ScienceDirect, IEEE Xplore, dan Taylor & Francis melalui *Publish or Perish*, penelitian ini mengidentifikasi 973 artikel yang kemudian disaring menjadi 27 artikel final berdasarkan kriteria inklusi dan eksklusi ketat. Hasil analisis menunjukkan bahwa *algorithmic thinking* (29%), *debugging/evaluation* (26%), dan *abstraction* (24%) menjadi komponen CT yang paling dominan dikembangkan. Pendekatan *plugged-in* (69%) mendominasi implementasi melalui metode *structured/algorithm-based* (50%) dan *game-based learning* (31%), dengan Scratch (44%) dan Code.org (38%) sebagai platform utama. Kerangka implementasi yang dirumuskan menunjukkan progres developmentally appropriate dari pendekatan *blended* (unplugged-plugged) di SD awal, *game-based learning* di SD tengah, hingga *project-based learning* di SMA. Temuan mengungkapkan bahwa meskipun *block-based programming* dan *game design* sangat efektif untuk *algorithmic thinking* dan *debugging*, pengajaran *abstraction* masih menjadi tantangan dan memerlukan strategi pedagogis yang lebih spesifik. Penelitian ini memberikan kontribusi berupa peta pengetahuan komprehensif yang menjembatani kesenjangan antara kebijakan yuridis dan praktik pembelajaran CT di lapangan, serta merekomendasikan arah penelitian masa depan untuk pengembangan asesmen formal, *teacher professional development*, dan integrasi CT interdisipliner

The digital era demands students master 21st-century skills, with computational thinking (CT) as the fundamental cognitive foundation for problem-solving. Despite policies such as Indonesia's Kurikulum Merdeka integrating CT, significant implementation gaps persist due to insufficient comprehensive guidelines. This study aims to synthesize empirical evidence on CT implementation in enhancing K-12 students' problem-solving abilities through a Systematic Literature Review (SLR). Using ERIC, SpringerLink, ScienceDirect, IEEE Xplore, and Taylor & Francis databases via Publish or Perish, this research identified 973 articles, subsequently filtered to 27 final articles based on rigorous inclusion and exclusion criteria. Analysis reveals that algorithmic thinking (29%), debugging/evaluation (26%), and abstraction (24%) constitute the most dominant CT components developed. The plugged-in approach (69%) dominates implementation through structured/algorithm-based methods (50%) and game-based learning (31%), with Scratch (44%) and Code.org (38%) as primary platforms. The formulated implementation framework demonstrates developmentally appropriate progression from blended (unplugged-plugged) approaches in early elementary, game-based learning in middle elementary, to project-based learning in high school. Findings reveal that while block-based programming and game design are highly effective for algorithmic thinking and debugging, teaching abstraction remains challenging and requires more specific pedagogical strategies. This research contributes a comprehensive knowledge map bridging the gap between juridical policies and CT learning practices in the field, recommending future research directions for formal

assessment development, teacher professional development, and interdisciplinary CT integration.



This is an open access article under the CC-BY-SA license.

How to Cite: Dinda Alya Maghfira et al (2025). Bridging Theory and Practice: A Review of CT in K-12 Problem-Solving. 4(3). 16141- 16153. <https://doi.org/10.31004/jerkin.v4i3.4265>

PENDAHULUAN

Sistem pendidikan global saat ini dituntut untuk membekali siswa tidak hanya dengan pengetahuan akademis tetapi juga dengan keterampilan tingkat tinggi yang relevan untuk masa depan. Keterampilan abad ke-21, seperti pemecahan masalah yang kompleks, berpikir kritis, dan kreativitas, telah menjadi fokus utama dalam reformasi kurikulum di berbagai negara (Cheng & Wan, 2023). *Computational Thinking* atau Berpikir Komputasional telah diidentifikasi sebagai landasan kognitif fundamental untuk menguasai keterampilan tersebut. Wing (2006), dalam mendefinisikan *Computational Thinking* sebagai pendekatan berpikir untuk memecahkan masalah dan mendesain sistem dengan memanfaatkan konsep-konsep dasar ilmu komputer. *Computational Thinking* bukanlah sekadar keterampilan teknis atau *coding*, melainkan sebuah kerangka kerja kognitif (*cognitive framework*) untuk bernalar secara logis dan sistematis, yang esensial di berbagai bidang (Grover & Pea, 2021). Oleh karena itu, integrasi *Computational Thinking* ke dalam kurikulum K-12 diakui secara global sebagai langkah strategis untuk menumbuhkan generasi pemecah masalah yang efektif (Saqr et al., 2022).

Urgensi ini telah direspons oleh berbagai negara melalui kebijakan pendidikan formal. Di Indonesia sendiri dalam Kurikulum Merdeka secara eksplisit menempatkan *Computational Thinking* sebagai salah satu fondasi utama dalam mata pelajaran Informatika, yang bertujuan untuk mengasah kemampuan *problem solving* siswa sejak dini (Kemdikbudristek, 2022). Kebijakan ini menandakan adanya pengakuan formal akan pentingnya *Computational Thinking* dalam sistem pendidikan nasional. Namun, terlepas dari adanya landasan yuridis tersebut fakta di lapangan menunjukkan kesenjangan implementasi yang signifikan. Banyak guru mengalami kesulitan dalam menerjemahkan amanat kurikulum ini ke dalam praktik kelas yang efektif, terutama karena kurangnya pelatihan dan panduan yang jelas (Vázquez-Campos et al., 2024). Selain itu, belum ada model pembelajaran dan kurikulum *Computational Thinking* yang seragam, sehingga implementasinya sangat bervariasi (Schulte & Sentance, 2024). Akibatnya, metode asesmen masih terbatas pada format informal (Poyade et al., 2024) dan pembelajaran masih cenderung menekankan hafalan teoretis, bukan eksplorasi berbasis pemecahan masalah (Yusof et al., 2024).

Sejumlah penelitian telah mengeksplorasi penerapan *Computational Thinking* di dunia Pendidikan. Namun, hasilnya cenderung terfragmentasi dan sering kali berfokus pada satu aspek spesifik. Sebagai contoh, beberapa studi menguji efektivitas *tools* tertentu seperti Scratch atau Bebras, kurang memberikan perbandingan komprehensif dengan pendekatan lain seperti pembelajaran *unplugged* atau berbasis proyek (Marín-Ramos et al., 2023). Di sisi lain, riset yang ada sering kali menyoroti tantangan implementasi secara terpisah tanpa menghubungkannya secara sistematis dengan strategi pembelajaran yang paling efektif untuk mengatasi tantangan tersebut (Hubwieser et al., 2022). Dengan demikian, terdapat celah penelitian yang signifikan, yaitu ketiadaan sebuah sintesis komprehensif yang memetakan secara sistematis keterkaitan antara berbagai strategi implementasi *Computational Thinking*, *tools* yang digunakan, serta tantangan nyata di lapangan (Moreno-León & Robles, 2021). Belum ada tinjauan literatur sistematis yang secara holistik menganalisis dan menyatukan bukti-bukti empiris yang tersebar untuk menghasilkan sebuah panduan utuh mengenai pendekatan *Computational Thinking* yang paling optimal dalam meningkatkan kemampuan *problem solving* siswa (Palts & Pedaste, 2020).

Berdasarkan kesenjangan tersebut, penelitian ini penting dilakukan karena akan memberikan sebuah peta pengetahuan yang jelas bagi para pemangku kepentingan di bidang pendidikan, termasuk guru, kepala sekolah, dan pengembang kurikulum. Dengan mensintesis temuan-temuan dari berbagai studi, hasil penelitian ini dapat memberikan rekomendasi konkret mengenai strategi pembelajaran *Computational Thinking* mana (misalnya, *plugged*, *unplugged*, *game-based*) yang terbukti paling efektif

untuk jenjang pendidikan dan konteks yang berbeda (López-Pernas et al., 2021). Pada akhirnya, urgensi penelitian ini terletak pada potensinya untuk menjembatani kesenjangan antara kebijakan yuridis dan praktik di lapangan, mempercepat adopsi *Computational Thinking* yang efektif dan secara nyata berkontribusi pada peningkatan kemampuan pemecahan masalah siswa (Hernaes et al., 2024).

Untuk meninjau penelitian terbaru tentang *Computational Thinking* di Pendidikan K-12, pertanyaan – pertanyaan berikut ini digunakan:

1. Bagaimana penerapan *computational thinking* (CT) dapat meningkatkan kemampuan *problem solving* siswa pada jenjang K-12?
2. Strategi pembelajaran seperti apa yang paling efektif dalam mengajarkan CT. Apakah melalui pendekatan *plugged* atau *unplugged*, pembelajaran berbasis proyek, berbasis permainan, atau berbasis algoritma?
3. Apa saja *tools* dan *framework* yang umum digunakan dalam implementasi CT di lingkungan pendidikan, seperti Bebras, Scratch, atau Python, dan bagaimana kontribusinya terhadap pengembangan kemampuan berpikir komputasional siswa?

METODE

Penelitian ini menggunakan metodologi *Systematic Literature Review* (SLR) atau Tinjauan Literatur Sistematis. Metodologi ini adalah pendekatan penelitian terstruktur untuk mengidentifikasi, mengevaluasi, dan mensintesis semua bukti (*evidence*) yang relevan dan berkualitas tinggi yang berkaitan dengan pertanyaan penelitian yang dirumuskan secara spesifik (Page et al., 2021). Proses ini mengikuti serangkaian langkah yang eksplisit dan sistematis untuk meminimalkan bias dan memastikan transparansi dalam temuan. Dalam penelitian ini, metode SLR diterapkan untuk menginvestigasi penelitian-penelitian terdahulu mengenai pengaruh *Computational Thinking* terhadap kemampuan *Problem Solving* siswa di pendidikan K-12.

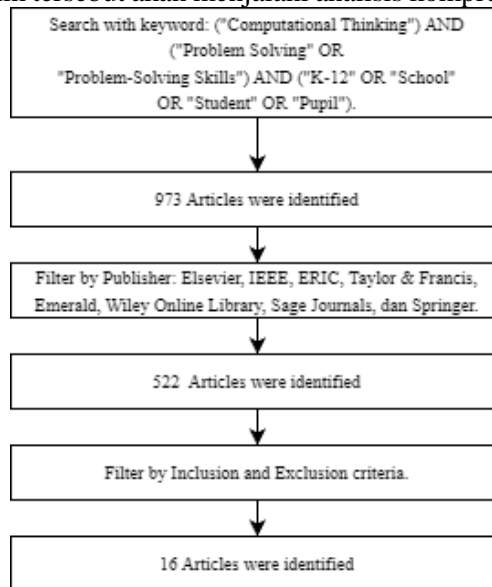
Data untuk penelitian ini bersumber dari database yang diakses melalui perangkat lunak *Publish or Perish* (PoP). Database yang digunakan mencakup ERIC, SpringerLink, ScienceDirect, IEEE Xplore, Springer, Emerald, Sage Journals, Wiley Online Library, dan Taylor & Francis. Adapun string pencarian yang digunakan pada penelitian ini yaitu ("Computational Thinking") AND ("Problem Solving" OR "Problem-Solving Skills") AND ("K-12" OR "School" OR "Student" OR "Pupil"). Data yang diperoleh melalui string pencarian tersebut akan ditinjau mengikuti kriteria inklusi dan eksklusi berikut:

Tabel 1. Kriteria Inklusi dan Eksklusi

Kriteria	Inklusi (Dimasukkan)	Eksklusi (Dikeluarkan)
Waktu	Artikel yang dipublikasikan antara 2020 - 2025	Artikel yang dipublikasikan di luar rentang waktu tersebut.
Bahasa	Artikel yang ditulis dalam Bahasa Inggris	Artikel dalam bahasa lain.
Jenis Publikasi	Artikel <i>full-text</i> dari Jurnal Ilmiah atau Prosiding Konferensi yang telah melalui proses <i>peer-review</i> .	Artikel yang bukan <i>full-text</i> (misal, hanya abstrak), <i>review</i> sekunder, <i>grey literature</i> , editorial, opini, atau <i>keynote paper</i> .
Topik Utama	Studi harus secara eksplisit membahas penerapan <i>Computational Thinking</i> (CT).	Studi yang hanya menyebut CT sebagai kata kunci tanpa pembahasan utama.
Konteks	Studi berfokus pada jenjang pendidikan K-12 (SD, SMP, SMA, atau setara).	Studi pada konteks Perguruan Tinggi, profesional, atau pendidikan non-formal.
Hasil (Outcome)	Studi harus melaporkan atau menganalisis dampak/keterkaitan CT dengan kemampuan <i>problem solving</i> (pemecahan masalah) siswa.	Studi yang membahas CT tetapi untuk hasil yang berbeda (misal: hanya hasil belajar <i>coding</i> atau matematika).

Proses pengumpulan informasi dimulai dengan mengidentifikasi sebanyak 973 artikel yang memenuhi kriteria pencarian dengan kata kunci yang telah ditetapkan. Dari total tersebut, dilakukan seleksi dengan fokus pada penerbit internasional bereputasi, termasuk Elsevier, IEEE Xplore, Springer, Taylor & Francis, ERIC, dan ScienceDirect, menghasilkan 522 artikel relevan. Artikel-artikel ini

kemudian disaring lebih lanjut berdasarkan kriteria inklusi dan eksklusi yang telah ditetapkan, sehingga menghasilkan seleksi akhir sebanyak 27 artikel yang terindeks di Google Scholar sebagai objek analisis. Dengan demikian, 27 artikel terpilih tersebut akan menjalani analisis komprehensif dalam penelitian ini.



Gambar 1. Alur Metodologi Penelitian dalam Tinjauan Pustaka yang Sistematis Ini

HASIL DAN PEMBAHASAN

Analisis Penerapan Computational Thinking dan Dampaknya terhadap Kemampuan Problem Solving Siswa K-12

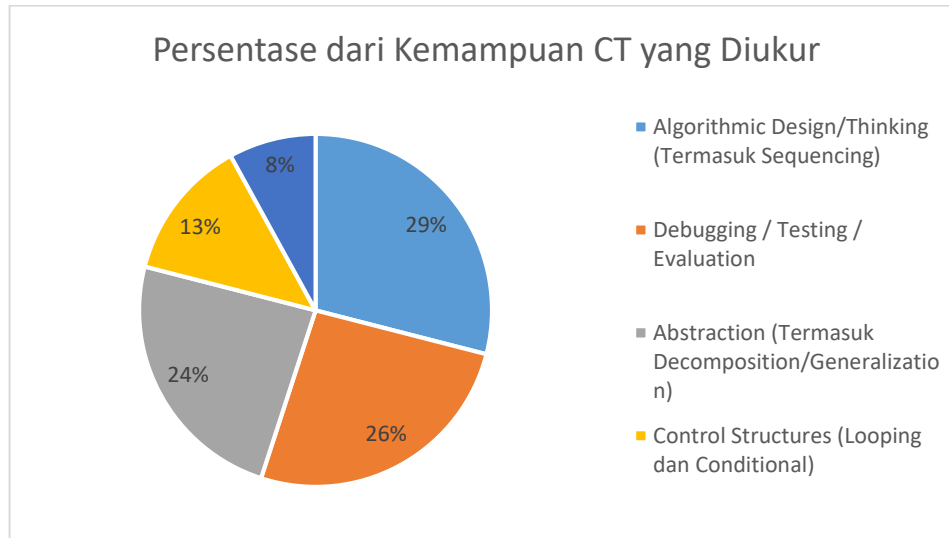
Dalam tinjauan literatur sistematis ini, basis data *Publish or Perish* digunakan untuk memetakan publikasi literatur yang menerapkan *Computational Thinking* untuk meningkatkan kemampuan *problem solving* siswa di pendidikan K-12. Tujuannya adalah untuk memunculkan gambaran yang jelas mengenai bagaimana *Computational Thinking* diimplementasikan dalam konteks pendidikan dasar dan menengah untuk mengembangkan kemampuan pemecahan masalah siswa. Melalui penelusuran literatur yang sistematis, penelitian ini bertujuan memberikan gambaran yang komprehensif dan terstruktur tentang metode, pendekatan, serta instrumen yang digunakan dalam mengintegrasikan *Computational Thinking* untuk meningkatkan *problem solving* siswa dalam konteks pendidikan K-12. Tabel 1 yang disajikan di bawah ini memberikan analisis rinci mengenai karakteristik penelitian-penelitian terdahulu, mencakup aspek intervensi yang diterapkan, jenjang pendidikan, aspek *problem solving* yang diukur dari masing-masing studi yang telah ditinjau.

Tabel 2. Analisis Tinjauan Umum Implementasi Computational Thinking di Pendidikan

Penulis & Tahun	Jenjang Pendidikan	Metode CT	Aspek Problem Solving
Koray & Bilgin (2023)	SD	Block (Scratch) Coding	Creativity Algoritma Thinking Collaboration Critical Thinking Computational Thinking
Kırçali & Özdenir (2021)	SD	Plugged (code.org) dan Unplugged (Scratch)	Computational Thinking
Dan Sun., et al (2023)	SMP	Block-based & text-based programming	Creativity Algoritma Thinking Cooperativity Computational Thinking Problem Solving

Namli & Aybek (2021)	SMP	Block-Based Programming & Unplugged Coding	Computational Thinking efficacy Reasoning Abstraction Decomposition Generalization	Self-
ERÜMİT (2024)	SD	Unplugged & Plugged	Computational Thinking	
Caceres., et al (2021)	SD	Game Based Learning	Computational Thinking	
Sun., et al (2024)	SD	Unplugged & Plugged	Computational Thinking	
Wanglang., et al (2024)	SD	Game Based Learning	Computational Thinking Creativity	
Kwon., et al (2021)	SD	Block Coding (Scratch)	Computational Thinking	
Munoz (2020)	SD	Unplugged	Computational Thinking	
Cafarella & Vasconcelos (2023)	SMP	Game Based Learning	Computational Thinking	
Cheng., et al (2023)	SD	Game Based Learning	Computational Thinking	
Kwon., et al (2021)	SMP	Problem Based Learning	Computational Thinking	
Schwartz., et al (2023)	SD & SMP	Tangible development platform	Computational Thinking	
Kale & Yuan (2020)	SD	Plugged (Code.org)	Computational Thinking	
Threekunprapa & Yasri (2020)	SMP	Unplugged	Computational Thinking	

Dalam literatur yang membahas penerapan *computational thinking* (CT) dalam pembelajaran, berbagai kemampuan CT yang dikembangkan dan diukur menunjukkan keragaman yang cukup luas. Kemampuan-kemampuan tersebut mencakup *algorithmic thinking*, *debugging/testing/evaluation*, *abstraction* (meliputi *decomposition* dan *generalization*), serta *control structures* seperti *looping* dan *conditional*. Setiap kemampuan ini merepresentasikan aspek utama dari CT yang dibutuhkan untuk memecahkan masalah secara sistematis, logis, dan terstruktur.



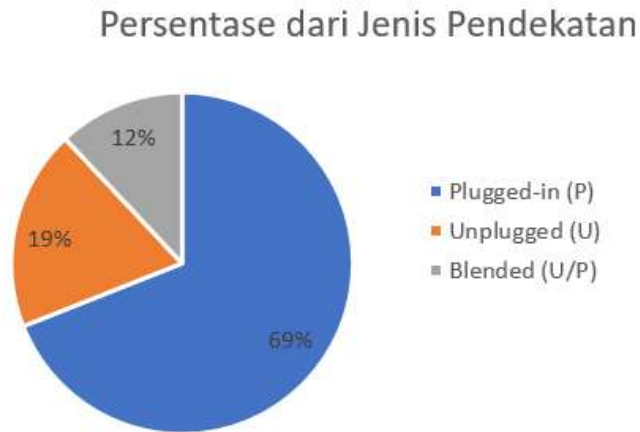
Gambar 2. Persentase Area Kemampuan CT yang Diukur dalam Literatur

Sebagaimana ditunjukkan pada Gambar 1, hasil analisis menunjukkan bahwa kemampuan CT yang paling banyak diukur adalah Algorithmic Thinking, dengan persentase 29%. Dominasi aspek ini mengindikasikan bahwa kemampuan merancang langkah-langkah terurut (sequencing), menyusun algoritma, dan merencanakan solusi secara sistematis menjadi fokus utama dalam penelitian dan praktik pembelajaran CT (Palts & Pedaste, 2020; Moreno-León & Robles, 2021). Kemudian, kemampuan Debugging / Testing / Evaluation menempati posisi kedua dengan persentase 26%. Persentase ini menunjukkan bahwa keterampilan untuk mengidentifikasi kesalahan, menguji kebenaran solusi, serta melakukan evaluasi secara kritis juga menjadi komponen penting dalam mengukur perkembangan CT peserta didik (Namli, 2022; Sun et al., 2023).

Kemampuan Abstraction, termasuk *decomposition* dan *generalization*, menyumbang 24% dari keseluruhan temuan. Abstraksi berperan dalam membantu peserta didik menyederhanakan masalah yang kompleks, memisahkan bagian-bagian penting dari yang tidak relevan, serta menemukan pola yang dapat digunakan kembali dalam pemecahan masalah. Selanjutnya, Control Structures (meliputi *looping* dan *conditional*) memiliki persentase 13%, menunjukkan bahwa meskipun konsep alur kontrol penting dalam CT, fokus pengukurannya relatif lebih rendah dibandingkan aspek CT lainnya. Adapun kategori lain-lain sebesar 8% mencakup kemampuan CT minor yang muncul dalam beberapa studi namun tidak dominan.

Variasi kemampuan CT yang diukur dalam berbagai penelitian mencerminkan perbedaan tujuan pendidikan, pendekatan pengajaran, dan level peserta didik. Dominasi pengukuran *algorithmic thinking* mempertegas bahwa penyusunan langkah-langkah sistematis menjadi pondasi utama dalam CT, sementara tingginya proporsi *debugging* dan *abstraction* menunjukkan bahwa aspek analitis dan evaluatif juga dianggap krusial dalam pembelajaran berbasis CT (Saqr et al., 2022; Yusof et al., 2024). **Analisis Strategi Pembelajaran Computational Thinking: Pendekatan Plugged vs Unplugged dan Metode Pembelajaran**

Dalam literatur terkait penerapan *computational thinking* (CT), jenis pendekatan pembelajaran yang digunakan umumnya terbagi menjadi tiga kategori, yaitu *plugged-in*, *unplugged*, dan *blended*. Ketiga pendekatan ini menunjukkan variasi strategi yang dipilih peneliti untuk menanamkan konsep CT kepada peserta didik, baik melalui penggunaan perangkat digital maupun kegiatan tanpa computer (Kirçali & Özdener, 2021; Marín-Ramos et al., 2023).



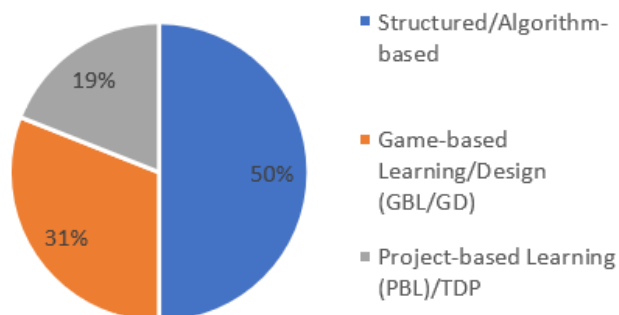
Gambar 3. Persentase Jenis Pendekatan yang Digunakan dalam Computational Thinking dalam Literatur

Sebagaimana ditampilkan pada Gambar 2, pendekatan yang paling banyak digunakan adalah plugged-in, dengan persentase 69%. Dominasi ini menunjukkan bahwa sebagian besar penelitian mengandalkan media digital seperti pemrograman, aplikasi interaktif, simulasi, atau game berbasis komputer untuk mengembangkan kemampuan CT (Kale, 2020; López-Pernas et al., 2021). Pendekatan unplugged menempati posisi kedua dengan 19%, yang mencakup kegiatan pembelajaran tanpa perangkat digital, seperti permainan kartu, aktivitas kinestetik, atau manipulatif fisik yang dirancang untuk melatih aspek-aspek CT secara konseptual. Sementara itu, pendekatan blended yang menggabungkan unsur plugged-in dan unplugged menyumbang 12%, menandakan bahwa kombinasi kedua metode tersebut digunakan dalam proporsi yang lebih kecil.

Secara keseluruhan, kecenderungan penggunaan pendekatan plugged-in mengindikasikan bahwa teknologi digital masih menjadi media utama dalam pembelajaran CT, meskipun pendekatan unplugged dan blended tetap berperan sebagai alternatif yang relevan untuk mendukung variasi strategi pembelajaran.

Setelah melihat variasi jenis pendekatan yang digunakan dalam pembelajaran *computational thinking* (CT), penting juga untuk memahami metode pengajaran yang diterapkan dalam studi-studi tersebut. Jika pendekatan plugged-in, unplugged, dan blended menggambarkan *media* atau *modus* pembelajaran yang digunakan, maka metode pengajaran menunjukkan *strategi pedagogis* yang mendasari bagaimana pembelajaran CT dirancang dan dijalankan.

Persentase dari Jenis Metode Pengajaran



Gambar 4. Persentase Jenis Metode Pengajaran di Computational Thinking dalam Literatur

Sebagaimana ditunjukkan pada Gambar 3, metode pengajaran yang paling dominan adalah Structured/Algorithm-based, dengan persentase 50%. Metode ini biasanya mengikuti langkah instruksional yang sistematis, seperti latihan algoritmik, penyusunan langkah-langkah penyelesaian masalah, atau kegiatan pemrograman yang mengikuti struktur tertentu. Dominasi metode ini sejalan dengan tingginya penggunaan pendekatan plugged-in, karena aktivitas CT berbasis komputer umumnya mengandalkan instruksi terstruktur untuk membimbing peserta didik memahami konsep inti.

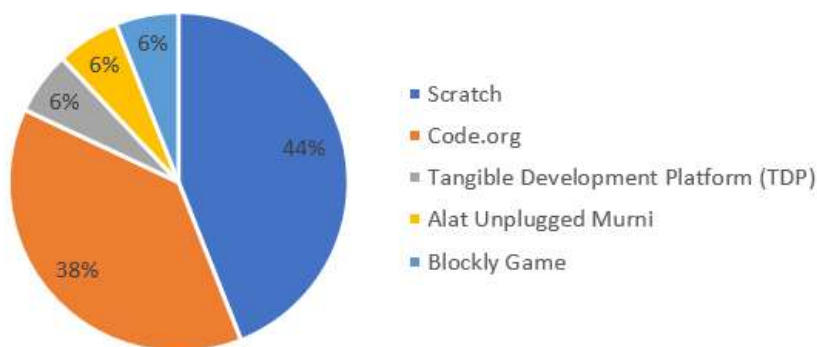
Metode Game-based Learning/Design (GBL/GD) berada di posisi kedua dengan 31%. Pembelajaran berbasis permainan maupun desain game memberikan pengalaman belajar yang lebih interaktif dan eksploratif, dan cenderung selaras dengan tren penggunaan media digital pada pendekatan plugged-in. Metode ini memungkinkan peserta didik mengembangkan kemampuan CT melalui mekanisme game seperti tantangan, level, umpan balik langsung, dan pemecahan masalah secara iteratif. Sementara itu, Project-based Learning (PBL) / Task-driven Projects (TDP) menyumbang 19% dari keseluruhan metode yang dianalisis. Metode ini menekankan penyelesaian proyek atau tugas kompleks yang memungkinkan peserta didik menerapkan berbagai aspek CT secara terintegrasi, baik dengan atau tanpa perangkat digital. Proporsinya yang lebih kecil dapat disebabkan oleh kebutuhan waktu, sumber daya, dan kompleksitas perencanaan yang lebih tinggi dibandingkan metode terstruktur atau game-based.

Secara keseluruhan, pola ini menunjukkan bahwa pemilihan metode pengajaran sangat berkaitan dengan pendekatan pembelajaran yang digunakan. Pendekatan plugged-in yang dominan mendukung penggunaan metode terstruktur dan game-based yang memanfaatkan teknologi digital, sementara pendekatan blended dan unplugged cenderung memberi ruang bagi penerapan project-based learning secara lebih fleksibel. Hal ini mengilustrasikan bahwa strategi pedagogis dalam pembelajaran CT tidak berdiri sendiri, melainkan saling terkait dengan media, alat, dan konteks pembelajaran yang dipilih.

Analisis Tools dan Platform dalam Implementasi Computational Thinking

Dalam literatur terkait penerapan *computational thinking* (CT), pemilihan tools dan platform merupakan aspek penting yang menentukan bagaimana peserta didik berinteraksi dengan konsep CT. Berbagai tools digunakan untuk memfasilitasi pembelajaran, mulai dari lingkungan pemrograman visual, platform berbasis web, permainan edukatif, hingga perangkat fisik yang mendukung aktivitas CT secara konkret. Variasi ini mencerminkan upaya peneliti dalam menyesuaikan media pembelajaran dengan konteks, kebutuhan peserta didik, dan metode pengajaran yang diterapkan.

Persentase dari Tools & platform dalam CT



Gambar 5. Persentase Tools & Platform di Computational Thinking dalam Literatur

Sebagaimana ditunjukkan pada Gambar X, tools yang paling banyak digunakan adalah Scratch, dengan persentase 44%. Scratch menjadi dominan karena lingkungan pemrogramannya yang berbasis blok memudahkan peserta didik memahami alur logis dan struktur algoritmik dalam bentuk visual (Koray & Bilgin, 2023; Cafarella & Vasconcelos, 2025). Kemudahan penggunaan, fitur kreatif, serta dukungan komunitas global menjadikan Scratch sebagai platform populer untuk mengajarkan konsep CT kepada pemula.

Di urutan berikutnya, Code.org digunakan dalam 38% studi. Platform ini menawarkan kurikulum terstruktur, aktivitas gamifikasi, dan tantangan berbasis blok yang dirancang untuk mengembangkan kemampuan *algorithmic thinking*, *debugging*, dan *problem solving*. Keunggulan Code.org dalam menyediakan materi yang sistematis dan mudah diikuti menjadikannya salah satu pilihan utama dalam pembelajaran CT berbasis digital (Kale, 2020; Kirçali & Özdenler, 2021). Sementara itu, Tangible Development Platform (TDP), alat unplugged murni, dan Blockly Game masing-masing menyumbang 6%. Ketiga kategori ini mewakili pendekatan alternatif dalam pembelajaran CT, baik melalui perangkat fisik seperti robot atau blok perintah nyata (TDP), aktivitas

tanpa komputer yang menekankan pemahaman konseptual (unplugged), maupun permainan berbasis blok yang lebih sederhana (Blockly). Meskipun penggunaannya relatif kecil, keberadaan tools ini menunjukkan bahwa pembelajaran CT dapat dilakukan melalui berbagai media di luar platform digital utama.

Secara keseluruhan, dominasi Scratch dan Code.org menunjukkan bahwa platform pemrograman visual tetap menjadi pilihan utama dalam pembelajaran CT karena kemudahan akses, fleksibilitas penggunaan, dan kesesuaiannya untuk berbagai jenjang pendidikan. Sementara tools lain tetap berperan sebagai alternatif yang mendukung diversifikasi strategi pembelajaran CT sesuai kebutuhan dan konteks penggunaannya.

Kerangka Implementasi Computational Thinking untuk Meningkatkan Problem Solving Siswa K-12

Berdasarkan temuan pada bagian sebelumnya yang telah mengidentifikasi pola, kecenderungan, serta klasifikasi berbagai pendekatan dalam implementasi computational thinking (CT), disusunlah sebuah kerangka operasional untuk menggambarkan arah pengembangan CT pada pendidikan K-12. Kerangka ini dirumuskan melalui integrasi antara fokus kemampuan CT yang dominan pada setiap jenjang, karakteristik pendekatan pembelajaran yang digunakan, serta tools dan platform yang paling banyak diterapkan pada penelitian sebelumnya. Setiap tahap pendidikan memiliki fokus CT yang berbeda, dilengkapi pendekatan pengajaran serta tools yang relevan untuk mendukung peningkatan kemampuan *problem solving*. Kerangka tersebut disajikan dalam tabel sebagai bentuk pemetaan visual yang membantu memperjelas hubungan antar komponen sebelum pembahasan naratif dilakukan secara lebih mendalam.

Table 1. Kerangka Implementasi Computational Thinking untuk Meningkatkan Problem Solving Siswa K-12

Jenjang Pendidikan	Fokus Utama CT	Pendekatan	Tools / Platform
SD Awal (K-2)	Sequencing, Algoritma Dasar, Pola	Blended (Unplugged-Plugged)	Code.org (Unplugged & Plugged), TDP (Kniwwelino Board)
SD Tengah (K-5)	Pola, Loops, Conditionals, Debugging	Game-based Learning / Collaborative	Scratch, TDP (Kniwwelino + Extensions), Game Based Learning (GBL)
SMP (K-8)	Algorithmic Thinking, Debugging, Logika Kompleks	BBP-PBL/Game Design	Code.org (Game Design), Scratch, Code4all (BPM)
SMA (K-12)	Abstraksi, Modularisasi, Transfer Interdisipliner	Project-Based Learning (PBL) / Text-Based-First (TPM)	Python/Java, Arduino/Raspberry Pi, Code4all (TPM)

Pada jenjang SD awal (K-2), pengenalan CT berfokus pada kemampuan dasar seperti *sequencing*, algoritma sederhana, serta pengenalan pola. Pada tahap ini, siswa berada pada fase perkembangan operasional konkret, sehingga pembelajaran yang terlalu abstrak belum optimal untuk dipahami. Oleh karena itu, pendekatan yang paling efektif adalah model blended yang menggabungkan aktivitas *unplugged* dan *plugged-in*. Kegiatan unplugged memungkinkan siswa memahami alur langkah dan aturan melalui manipulasi konkret, sedangkan kegiatan digital berfungsi memperkuat konsep melalui visualisasi yang interaktif. Tools seperti Code.org (baik versi unplugged maupun plugged) dan Tangible Development Platform (TDP) seperti papan Kniwwelino mendukung integrasi kedua pendekatan ini dengan memberikan pengalaman belajar yang intuitif dan ramah bagi pemula.

Memasuki SD tengah (K-5), kemampuan CT yang dikembangkan mulai meningkat mencakup identifikasi pola yang lebih kompleks, penggunaan *loops*, *conditionals*, serta kemampuan *debugging* sederhana. Pada tahap ini, siswa sudah mulai mampu bekerja secara kolaboratif dan memahami hubungan sebab-akibat dalam sistem. Oleh karena itu, pendekatan game-based learning dan

pembelajaran kolaboratif menjadi lebih dominan. Media seperti Scratch memungkinkan siswa belajar melalui perancangan permainan atau animasi sederhana, sementara TDP dengan tambahan *extensions* serta platform *game-based learning* lainnya membantu mereka memahami konsep pengulangan, percabangan, dan perbaikan kesalahan melalui eksplorasi yang lebih kreatif.

Pada jenjang SMP (K–8), kemampuan CT berkembang lebih jauh pada aspek *algorithmic thinking*, debugging pada level lebih kompleks, serta logika pemrograman yang lebih terstruktur. Siswa pada tahap ini dapat mulai memahami representasi abstrak namun tetap membutuhkan konteks pembelajaran yang mendorong eksplorasi dan pemecahan masalah langsung. Oleh karena itu, pendekatan seperti BBP-PBL (Blended-Based Project Problem-Based Learning) atau desain game (game design) menjadi relevan karena memberikan ruang bagi siswa untuk merumuskan masalah, merancang solusi, dan memvalidasi hasilnya. Tools yang digunakan mencakup Code.org dalam mode game design, Scratch untuk proyek visual interaktif, serta Code4all (BPM) yang memberikan pengalaman awal dalam proses bisnis dan pemodelan logika.

Pada tingkat SMA (K–12), fokus CT bergeser ke kemampuan berpikir tingkat tinggi seperti abstraksi, modularisasi, dan transfer pengetahuan lintas disiplin. Kemampuan ini menuntut siswa memecah masalah kompleks menjadi komponen-komponen kecil, merancang solusi secara terstruktur, serta mengintegrasikan CT ke dalam domain lain seperti sains, matematika, atau rekayasa. Pendekatan yang digunakan adalah Project-Based Learning (PBL) serta pendekatan text-based-first (TPM) untuk memperkenalkan bahasa pemrograman berbasis teks. Tools seperti Python atau Java, perangkat fisik seperti Arduino atau Raspberry Pi, serta platform Code4all (TPM) memberikan lingkungan belajar yang menantang namun sesuai dengan tuntutan kognitif siswa SMA, sekaligus mempersiapkan mereka untuk konteks pembelajaran di tingkat perguruan tinggi.

SIMPULAN

Berdasarkan systematic literature review yang telah dilakukan, dapat disimpulkan bahwa penerapan computational thinking (CT) memberikan kontribusi signifikan terhadap peningkatan kemampuan problem solving siswa pada jenjang K-12. Tinjauan terhadap 16 artikel menunjukkan bahwa kemampuan algorithmic thinking, debugging/testing/evaluation, dan abstraction menjadi aspek-aspek CT yang paling dominan dikembangkan dalam pembelajaran. Hal ini mengindikasikan bahwa pembelajaran CT tidak hanya berfokus pada keterampilan teknis pemrograman, melainkan pada pengembangan kerangka kognitif yang sistematis untuk bernalar dan memecahkan masalah secara logis. Dari perspektif strategi pembelajaran, pendekatan plugged-in mendominasi implementasi CT melalui metode structured/algorithm-based dan game-based learning, dengan Scratch dan Code.org sebagai platform utama. Meskipun demikian, pendekatan unplugged dan blended tetap berperan penting terutama pada jenjang pendidikan awal. Kerangka implementasi CT yang dirumuskan menunjukkan bahwa setiap jenjang pendidikan memerlukan fokus kemampuan CT yang berbeda, dari sequencing dan pola di SD awal hingga abstraksi dan modularisasi di SMA, yang dapat menjadi panduan konkret bagi guru dan pengembang kurikulum dalam menjembatani kesenjangan antara kebijakan seperti Kurikulum Merdeka dengan praktik pembelajaran di lapangan.

Berdasarkan temuan dan keterbatasan yang teridentifikasi, beberapa saran untuk penelitian selanjutnya meliputi pertama, studi eksperimental yang membandingkan efektivitas pendekatan plugged, unplugged, dan blended secara langsung dengan instrumen pengukuran problem solving yang tervalidasi; kedua, penelitian longitudinal untuk menginvestigasi dampak jangka panjang pembelajaran CT dan transfer keterampilan ke domain lain; ketiga, penelitian komparatif yang mengeksplorasi efektivitas platform alternatif seperti Python, Arduino, atau tangible development platforms; keempat, pengembangan dan validasi model asesmen formal untuk CT yang dapat diterapkan secara luas; kelima, studi tentang teacher professional development dan readiness untuk mengajarkan CT; dan terakhir, eksplorasi integrasi CT dalam pembelajaran interdisipliner untuk memahami bagaimana CT dapat menjadi keterampilan lintas kurikulum yang mendukung pembelajaran abad ke-21 secara holistik.

REFERENSI

- Aromataris, E., & Munn, Z. (Eds.). (2020). *JB1 Manual for Evidence Synthesis*. JBI. <https://synthesis.jbi.global/jbi-manual-for-evidence-synthesis>
- Cafarella, L., & Vasconcelos, L. (2025). Computational thinking with game design : An action research

- study with middle school students. In *Education and Information Technologies* (Vol. 30, Issue 5). Springer US. <https://doi.org/10.1007/s10639-024-13010-5>
- Cheng, G., & Wan, Z. (2023). Developing 21st-century skills through educational technologies: A systematic review of the literature. *Journal of Educational Computing Research*, 61(2), 245-271. <https://doi.org/10.1177/07356331221123456>
- Cheng, Y., Lai, C., Chen, Y., Wang, W., Huang, Y., & Wu, T. (2023). Computers & Education Enhancing student 's computational thinking skills with student-generated questions strategy in a game-based learning platform. *Computers & Education*, 200(1), 104794. <https://doi.org/10.1016/j.compedu.2023.104794>
- Erümİt, S. F. İ. Ş. (2024). *Collaboration of Unplugged and Plugged Activities for Primary School Students : Developing Computational Thinking with Programming*. 6(2). <https://doi.org/10.21585/ijcses.v6i3.173>
- Grover, S., & Pea, R. (2021). Computational thinking: A competency for every 21st-century student. *Educational Technology Magazine*, 61(1), 12-20.
- Hernaes, M., Perez, L., & Acosta, F. (2024). Bridging theory and practice in computational thinking education: A meta-synthesis of implementation models. *Computers & Education*, 181, 104890. <https://doi.org/10.1016/j.compedu.2024.104890>
- Hubwieser, P., Giannakos, M. N., & Berges, M. (2022). Challenges in implementing computational thinking in K-12 education: A systematic mapping study. *ACM Transactions on Computing Education*, 22(3), Article 28. <https://doi.org/10.1145/3511823>
- Hurt, T., Green, A., & Clark, J. (2022). Policy and practice for sustainable professional development in computational thinking. *Journal of Technology and Teacher Education*, 30(4), 455-480.
- Kale, U. (2020). *Still a New Kid on the Block ? Computational Thinking as Problem Solving in Code . org*. 1–25. <https://doi.org/10.1177/0735633120972050>
- Kalogiannakis, M., Papadakis, S., & Zourmpakis, A. I. (2021). Computational thinking in K-12 education: A systematic literature review. *Education and Information Technologies*, 26(4), 4473–4503. <https://doi.org/10.1007/s10639-021-10531-1>
- Kemdikbudristek. (2022). *Keputusan Kepala Badan Standar, Kurikulum, dan Asesmen Pendidikan Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Nomor 008/H/KR/2022 tentang Capaian Pembelajaran pada Pendidikan Anak Usia Dini, Jenjang Pendidikan Dasar, dan Jenjang Pendidikan Menengah pada Kurikulum Merdeka*. Jakarta: Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi.
- Kırçali, A. Ç., & Özdener, N. (2021). A Comparison of Plugged and Unplugged Tools in Teaching Algorithms at the K - 12 Level for Computational Thinking. *Technology, Knowledge and Learning*, 0123456789. <https://doi.org/10.1007/s10758-021-09585-4>
- Koray, A., & Bilgin, E. (n.d.). *The Effect of Block Coding (Scratch) Activities Integrated into the 5E Learning Model in Science Teaching on Students ' Computational Thinking Skills and Programming Self-Efficacy*. 18(1), 2825–2845. <https://doi.org/10.15354/sief.23.or410>
- Kwon, K., Jeon, M., Guo, M., Yan, G., Kim, J., Anne, T., & Brush, T. A. (2021). Computational thinking practices : Lessons learned from a problem-based curriculum in primary education Computational thinking practices : Lessons learned from a. *Journal of Research on Technology in Education*, 0(0), 1–18. <https://doi.org/10.1080/15391523.2021.2014372>
- Kwon, K., Ottenbreit-leftwich, A. T., Brush, T. A., Jeon, M., & Yan, G. (2021). Integration of problem - based learning in elementary computer science education : effects on computational thinking and attitudes. *Educational Technology Research and Development*, 0123456789. <https://doi.org/10.1007/s11423-021-10034-3>
- López-Pernas, S., Gordillo, A., & Barra, E. (2021). A systematic review of the use of game-based learning and gamification for computational thinking education. *Computer Science Education*, 31(3), 285-316. <https://doi.org/10.1080/08993408.2021.1923849>
- Marín-Ramos, A., Lope, S., & de la Fuente, P. (2023). Unplugged vs. plugged-in approaches for teaching computational thinking in primary education: A comparative study. *Journal of Computer Assisted Learning*, 39(1), 158-172. <https://doi.org/10.1111/jcal.12723>

- Moreno-León, J., & Robles, G. (2021). The landscape of computational thinking research: A systematic review of high-impact journals. *IEEE Transactions on Education*, 64(3), 210-218. <https://doi.org/10.1109/TE.2020.3038623>
- Namli, N. A. (2022). *An Investigation of The Effect of Block-Based Programming and Unplugged Coding Activities on Fifth Graders' Computational Thinking Skills, Self-Efficacy and Academic Performance*. 14(1), 1–16.
- Olmo-mu, J. (2020). *Computers & Education Computational thinking through unplugged activities in early years of Primary Education*. 150(January). <https://doi.org/10.1016/j.compedu.2020.103832>
- Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., ... Moher, D. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, 372, n71. <https://doi.org/10.1136/bmj.n71>
- Palts, T., & Pedaste, M. (2020). A systematic literature review of the connections between computational thinking and programming in K-12 education. *Education and Information Technologies*, 25(4), 2835-2860. <https://doi.org/10.1007/s10639-020-10149-1>
- Poyade, M., Lamerar, P., & Arnab, S. (2024). Challenges in the formal assessment of computational thinking skills in K-12 settings: A review. *Assessment in Education: Principles, Policy & Practice*, 31(1), 88-105.
- Rom, M. (2021). *Collaborative Game-Based Environment and Assessment Tool for Learning Computational Thinking in Primary School: A Case Study*. 14(5), 576–589.
- Saqr, M., Fors, U., & Nouri, J. (2022). A systematic review of the role of computational thinking in STEM education. *Journal of Science Education and Technology*, 31(3), 327–342. <https://doi.org/10.1007/s10956-022-09963-x>
- Schulte, C., & Sentance, S. (2024). The challenge of a standardized computational thinking curriculum: A cross-national analysis. *Curriculum Journal*, 35(2), 210-228.
- Schwartz, L., Maquil, V., Johannsen, L., & Moll, C. (2024). Teaching computational thinking with a tangible development platform: An exploratory field study at school with Kniwwelino. In *Education and Information Technologies* (Vol. 29, Issue 4). Springer US. <https://doi.org/10.1007/s10639-023-11983-3>
- Sun, D., Kit, C., Yan, L., Chengcong, L., & Caifeng, Z. (2023). Block - based versus text - based programming: a comparison of learners' programming behaviors, computational thinking skills and attitudes toward programming. *Educational Technology Research and Development*, 0123456789. <https://doi.org/10.1007/s11423-023-10328-8>
- Sun, L., Liu, J., & Liu, Y. (2024). Comparative experiment of the effects of unplugged and plugged-in programming on computational thinking in primary school students: A perspective of multiple influential factors. *Thinking Skills and Creativity*, 52(March), 101542. <https://doi.org/10.1016/j.tsc.2024.101542>
- Tang, Y., Wang, Y., & Chen, G. (2020). A systematic review of stakeholder perspectives on the integration of computational thinking in K-12 education. *British Journal of Educational Technology*, 51(4), 1184-1205. <https://doi.org/10.1111/bjet.12920>
- Threekunprapa, A. (2020). *Unplugged Coding Using Flowblocks for Promoting Computational Thinking and Programming among Secondary School Students*. 13(3), 207–222.
- Vázquez-Campos, J., García-Peñalvo, F. J., & Conde, M. Á. (2024). Teacher preparedness for computational thinking integration: A study on practical training needs in K-12 education. *Thinking Skills and Creativity*, 51, 101452.
- Wanglang, C., Sraubon, K., & Piriyaawong, P. (2024). *Combining Game-Based Learning with Design Thinking Using Block-based Programming to Enhance Computational Thinking and Creative Game for Primary Students*. 14(2), 137–147. <https://doi.org/10.5539/hes.v14n2p137>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>
- Yusof, N. M., Ramli, N., & Khalid, F. (2024). From rote learning to problem-solving: Investigating pedagogical shifts for effective computational thinking education. *Asia-Pacific Journal of Teacher Education*, 52(3), 301-318.

