



## Optimalisasi Kemampuan *Computational Thinking* Peserta Didik SD Negeri 4 Sragen melalui Pelatihan Koding Berbasis Scratch

Rizka Latifah<sup>1\*</sup>, Dewi Anjarsari<sup>2</sup>, Nur Halimah Widowati<sup>3</sup>

<sup>1</sup>Informatika, Fakultas Informatika dan Kesehatan, <sup>2</sup>Pendidikan Guru Sekolah Dasar, Fakultas Ilmu Keguruan dan Pendidikan,

<sup>3</sup>Hukum, Fakultas Hukum, Universitas Sragen, Indonesia

### ARTICLE INFO

#### Article History

Received : Dec 26, 2025

1<sup>st</sup> Revision : Dec 31, 2025

Accepted : Dec 31, 2025

Available Online : Jan 26, 2026

#### Keywords:

*Computational thinking*;

Scratch;

Sekolah Dasar;

### ABSTRACT

Computational thinking (CT) has become a fundamental prerequisite for adapting to digital transformation. This community service project aims to introduce CT skills to students at SD Negeri 4 Sragen through the use of Scratch. Scratch is used as a learning instrument due to its characteristics as an interactive and beginner-friendly visual programming language. This activity was conducted on November 14, 2025, in the computer laboratory of SD Negeri 4 Sragen, involving 26 fourth-grade students. The methods used included lectures and hands-on training in creating a project, namely a catch game. The results of this community service showed a high success rate, with 90% of students able to complete the game project independently, following the instructions. The remaining 10% finished the task with additional guidance. These findings suggest that the drag-and-drop system in Scratch is effective in facilitating an understanding of algorithm basics in a fun and practical manner. Although the program was overall successful, technical obstacles related to the stability of computer devices at the school are essential notes. Therefore, it is recommended that in future community service activities, devices be checked first to ensure a smooth and optimal learning process.

### ABSTRAK

Penguasaan *computational thinking* (CT) telah menjadi prasyarat fundamental dalam beradaptasi dengan transformasi digital. Pengabdian masyarakat ini bertujuan untuk mengenalkan kemampuan CT kepada peserta didik di SD Negeri 4 Sragen melalui pemanfaatan aplikasi Scratch. Scratch digunakan sebagai instrumen pembelajaran karena karakteristiknya sebagai bahasa pemrograman visual yang interaktif dan ramah bagi pemula. Kegiatan ini dilaksanakan pada 14 November 2025 di laboratorium komputer SD Negeri 4 Sragen dengan melibatkan 26 peserta didik kelas IV. Metode yang digunakan meliputi ceramah dan pelatihan praktik langsung dalam pembuatan proyek yaitu game tangkap. Hasil pengabdian menunjukkan tingkat keberhasilan yang tinggi, di mana 90% peserta didik mampu menyelesaikan proyek permainan secara mandiri sesuai instruksi, sementara 10% sisanya berhasil menyelesaikan tugas dengan bimbingan tambahan. Temuan ini mengindikasikan bahwa sistem *drag-and-drop* pada Scratch efektif dalam memfasilitasi pemahaman dasar-dasar algoritma secara menyenangkan dan aplikatif. Meskipun secara keseluruhan program berjalan sukses, kendala teknis terkait stabilitas performa perangkat komputer di sekolah menjadi catatan penting. Oleh karena itu, disarankan agar pada pengabdian selanjutnya melakukan pengecekan perangkat terlebih dahulu guna menjamin kelancaran proses pembelajaran yang lebih maksimal.

### \*Corresponding Author

Email address:

[rizka.latifah62@unissra.ac.id](mailto:rizka.latifah62@unissra.ac.id)



## 1. LATAR BELAKANG

Transformasi digital menjadi urgensi di berbagai bidang kehidupan, termasuk pendidikan. Salah satu dampak signifikan dari penggunaan teknologi ini adalah munculnya pola pikir komputasional atau *computational thinking* pada peserta didik (Wing, 2008). Sebagaimana dijelaskan oleh Abidin (2023) dan (Angraini *et al.*, 2023), *computational thinking* memungkinkan seseorang untuk mengekspresikan tantangan dan solusi melalui logika komputasi yang efektif. Penguasaan *computational thinking* kini menjadi prasyarat dasar bagi siapa pun yang ingin beradaptasi dan berkembang di era transformasi digital ini.

Wing (2014) mendefinisikan *computational thinking* sebagai suatu proses merumuskan solusi dalam bentuk langkah-langkah komputasi atau algoritma yang dilakukan oleh komputer. *Computational thinking* juga merupakan salah satu strategi pemecahan masalah yang bisa digunakan secara luas, bukan hanya pada bidang computer science namun juga pada masalah sehari-hari (Rahayu *et al.*, 2022). Urgensi dari CT adalah mampu melatih peserta didik berpikir kreatif dan inovatif dalam memecahkan masalah dengan bantuan komputer (Huda & Rohaeti, 2024).

Koding merupakan salah satu cara mengenalkan *computational thinking* kepada peserta didik. Koding adalah kegiatan menyusun rangkaian kode yang dapat dikenali dan dieksekusi oleh sistem komputer. Lubis *et al.* (2023) mengemukakan bahwa pengenalan koding sejak dini memberikan kesempatan bagi peserta didik untuk memadukan kemampuan komunikasi, pola pikir, dan strategi dalam pemecahan masalah secara simultan. Koding sejak dini bisa dilakukan sejak peserta didik masih di sekolah dasar.

Target utama dari kegiatan pengabdian ini adalah mengenalkan *computational thinking* (CT) melalui aktivitas pengodean menggunakan Scratch. Aplikasi Scratch adalah bahasa pemrograman visual yang dirancang untuk memudahkan anak-anak atau pemula belajar pemrograman dengan cara menyenangkan dan interaktif. Peserta didik mampu belajar pemrograman tanpa harus mengenal skrip bahasa pemrograman. Aplikasi Scratch mampu membantu anak-anak memahami konsep logika, kreativitas, dan memecahkan masalah (Yunus *et al.*, 2023). Sehingga aplikasi ini media ideal untuk menstimulasi pengenalan kemampuan *computational thinking* kepada peserta didik sekolah dasar.

Berdasarkan urgensi tersebut, program pengabdian masyarakat ini bertujuan mengenalkan kemampuan *computational thinking* peserta didik SD Negeri 4 Sragen. Hasil yang diharapkan adalah peserta didik mampu membuat game tangkap. Melalui proses pembuatan game tersebut, peserta didik akan belajar cara menyusun algoritma pemrograman, mengkombinasikan berbagai macam code, gambar, dan suara, serta memecahkan masalah yang diberikan secara kreatif. Lebih jauh lagi, kemampuan memodifikasi algoritma ini diharapkan dapat memperkuat fondasi berpikir komputasional mereka secara berkelanjutan.

## 2. TINJAUAN PUSTAKA

Di era abad ke-21, *computational thinking* telah menjadi kompetensi yang terintegrasi secara fundamental dalam sistem pendidikan. Implementasi *computational thinking* memungkinkan siswa untuk memecahkan berbagai masalah kompleks dan terbuka melalui pola pikir yang logis serta terstruktur (Masfingatini & Maharani, 2019). Lebih jauh lagi, integrasi *computational thinking* dalam pendidikan berfungsi untuk membekali siswa dengan kemampuan berpikir inovatif, sarana ekspresi diri melalui berbagai media digital, serta kecakapan dalam menganalisis permasalahan dunia nyata dari berbagai perspektif (Agstringtyas *et al.*, 2024). Penguasaan *computational thinking* sejak dini bukan lagi sekadar pilihan, melainkan fondasi esensial bagi siswa untuk bertransformasi dari sekadar pengguna teknologi menjadi kreator yang solutif di masa depan. Oleh karena itu, siswa sekolah dasar diharuskan untuk menguasai kemampuan *computational thinking*.

Wing (2006) (dalam Selby & Woollard, 2013; Syafaat, 2022) menjelaskan bahwa *computational thinking* mencakup proses berpikir yang terdiri dari abstraksi, dekomposisi, desain algoritmik, evaluasi, dan generalisasi. Secara lebih mendalam, abstraksi merupakan kemampuan menyaring informasi dengan memfokuskan perhatian pada elemen penting dan mengabaikan detail yang tidak relevan. Dekomposisi dilakukan dengan memecah masalah kompleks menjadi bagian-bagian kecil yang lebih sederhana agar lebih mudah dikelola. Selanjutnya, desain algoritmik melibatkan penyusunan langkah-langkah instruksi yang logis dan sistematis untuk mencapai solusi. Proses ini didukung oleh evaluasi untuk menguji ketepatan serta efisiensi solusi yang dihasilkan, serta generalisasi yang memungkinkan penerapan pola solusi tersebut pada berbagai konteks permasalahan serupa lainnya.

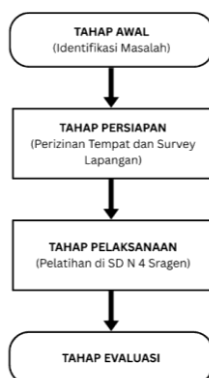
Meskipun pilar-pilar *computational thinking* tersebut sangat penting, mengajarkannya kepada siswa sekolah dasar membutuhkan pendekatan yang lebih nyata dan tidak sekadar teori (Hsu *et al.*, 2018). Konsep seperti abstraksi dan desain algoritma sering kali terasa sulit bagi anak-anak jika hanya dijelaskan secara lisan tanpa adanya praktik langsung (Angeli & Giannakos, 2020). Oleh sebab itu, dibutuhkan alat bantu belajar yang mampu mengubah ide-ide abstrak tersebut menjadi aktivitas yang bisa dilihat dan digerakkan secara visual (Bakala *et al.*, 2021). Melalui pengerjaan proyek kreatif, siswa diberikan kesempatan untuk bereksperimen, mencoba berbagai kemungkinan, dan belajar dari kesalahan secara langsung (Lye & Koh, 2014). Proses interaktif inilah yang akhirnya membuat kemampuan *computational thinking* mereka dapat berkembang secara lebih alami dan intuitif.

*Computational thinking* dapat diperoleh dari belajar pemrograman, di mana koding mendorong siswa untuk berpikir komputasi dengan memecahkan masalah dalam bentuk kode. Sejalan dengan temuan Wong and Jiang (2018), koding terbukti mampu meningkatkan kemampuan *computational thinking* siswa, khususnya dalam aspek analisis masalah. Namun, karena koding sering kali menjadi hambatan bagi siswa sekolah dasar (Kite *et al.*, 2021), aplikasi Scratch hadir sebagai garis awal untuk belajar *computational thinking* (Anwar *et al.*, 2025). Scratch adalah aplikasi pemrograman untuk anak-anak yang memiliki antarmuka interaktif dan menggunakan sistem drag-and-drop yang memudahkan transisi dari konsep abstrak ke praktik nyata. Scratch dirancang secara efektif untuk mengembangkan *computational thinking* melalui koding (Curi *et al.*, 2025). Dengan menggunakan Scratch siswa mampu menerapkan konsep dasar koding dengan cara yang menyenangkan (Alimin *et al.*, 2025). Dengan aplikasi ini, siswa sekolah dasar mampu mengenal dan menguasai dasar-dasar *computational thinking* dengan lebih mudah (Ariyan *et al.*, 2021; Yang *et al.*, 2024).

Beberapa studi empiris terbaru menunjukkan bahwa penggunaan *programming* berbasis block seperti Scratch tidak hanya meningkatkan *computational thinking* tetapi juga memberikan pengalaman belajar yang lebih *engaging* dan kontekstual bagi peserta didik sekolah dasar. Penelitian oleh Molina-Ayuso *et al.* (2024) menemukan bahwa penerapan Scratch dalam pembelajaran geometri di kelas lima SD memberikan proses belajar yang lebih positif serta meningkatkan keterlibatan dan motivasi peserta didik dalam mengembangkan keterampilan *computational thinking* melalui aktivitas pemrograman visual. Hasil lain juga menunjukkan bahwa pendekatan *project-based programming* dengan Scratch dapat meningkatkan keterampilan pemecahan masalah siswa, yang merupakan bagian integral dari *computational thinking* (Putra *et al.*, 2025). Selain itu, evidensi meta-analitik mengindikasikan bahwa Scratch merupakan alat yang efektif dalam menumbuhkan berbagai dimensi *computational thinking* seperti konsep dan praktik komputasi pada siswa sekolah dasar (Irawan *et al.*, 2024). Penelitian lain menggambarkan bahwa desain tugas Scratch yang terstruktur dan sesuai usia dapat memperkuat pemahaman siswa terhadap konsep *algorithmic thinking*, termasuk penggunaan loop dan kondisi dalam pemrograman visual (Ouahouda *et al.*, 2025). Temuan-temuan ini memperkuat fakta bahwa Scratch serta pendekatan pembelajaran berbasis aktivitas koding merupakan strategi pembelajaran yang relevan untuk mengembangkan *computational thinking* secara sistematis di pendidikan dasar (Hermita *et al.*, 2024).

### 3. METODE PELAKSANAAN

Pengabdian ini dilakukan dengan metode pelatihan dan ceramah (Latifah *et al.*, 2025) dengan tujuan agar Peserta didik mampu menggunakan aplikasi Scratch untuk mengenal *computational thinking*. Kegiatan dilaksanakan pada Jum'at, 14 November 2025, bertempat di laboratorium komputer SD Negeri 4 Sragen. Sebanyak 26 peserta didik kelas empat berpartisipasi dalam pelatihan ini. Aplikasi Scratch bisa diunduh secara gratis di website resminya (<https://scratch.mit.edu/>) atau dapat diakses secara web-based. Dalam pengabdian ini, aplikasi Scratch berbasis website yang digunakan dengan tujuan untuk mempersingkat waktu dan mengurangi beban kerja pada komputer laboratorium.



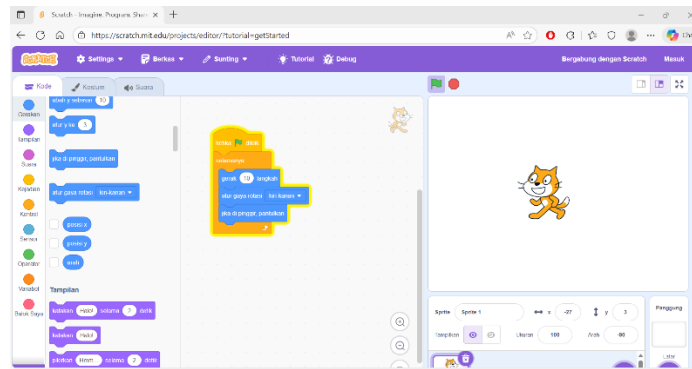
**Gambar 1.** Diagram Alir Metode Pelaksanaan

Pelaksanaan kegiatan ini dibagi menjadi tiga tahapan yaitu persiapan, pelatihan, dan evaluasi. Diagram alir pelaksanaan kegiatan bisa dilihat pada Gambar 1. Tahapan persiapan dilakukan sebelum pelatihan dimulai yaitu dari analisis masalah, menuliskan judul, dan menentukan lokasi. Tahapan pelatihan dibagi menjadi dua tahap yaitu tahap pengenalan kode dan elemen serta tahap pembuatan algoritma dan game. Pada tahap pengenalan kode dan elemen, peserta didik dikenalkan dengan berbagai macam kode dan elemen di aplikasi Scratch. Untuk kode sebagai contohnya adalah kode “*move (10) steps*” yang artinya peserta didik mampu menggerakkan objek 10 langkah. Untuk elemen pada Scratch terdapat gambar, background, dan suara. Pada pengabdian ini, instruktur mengajarkan semua elemen agar peserta didik semakin kreatif dalam membuat game. Pada tahapan pembuatan algoritma dan game, pada tahap ini dilakukan penyusunan algoritma bersama-sama. Peserta didik memperhatikan instruktur menerangkan pada layar untuk kemudian melakukan penyusunan algoritma. Pada tahapan terakhir, yaitu evaluasi. Pada tahapan evaluasi, peserta didik menunjukkan hasil kinerja mereka pada para instruktur. Peserta didik menceritakan game yang mereka buat. Selain itu, peserta didik juga diberikan cara untuk memodifikasi elemen sehingga peserta didik mampu mengubah interface game sesuai keinginan mereka.

### 4. HASIL DAN PEMBAHASAN

Kegiatan pengabdian masyarakat ini dilaksanakan pada tanggal 14 November 2025 selama 1 hari di SD Negeri 4 Sragen. Tahapan persiapan pengabdian ini dimulai dari melakukan analisis kebutuhan dan diskusi bersama Wakil Kepala Sekolah bidang Kurikulum SD Negeri 4 Sragen. Kemampuan yang dibutuhkan oleh siswa sekolah dasar pada era transformasi digital adalah kemampuan *computational thinking*. Kami kemudian mengidentifikasi bagaimana bidang informatika dapat memfasilitasi kebutuhan tersebut. Setelah melakukan proses pertimbangan, kami memutuskan melakukan pelatihan peningkatan *computational thinking* menggunakan aplikasi Scratch. Aplikasi Scratch mampu

meningkatkan *computational thinking*. Aplikasi ini gratis dan bisa diunduh di website resmi Scratch. Aplikasi ini juga ringan dan memiliki aplikasi berbasis website sehingga tidak perlu melakukan instalasi untuk menggunakan aplikasi ini. Tampilan aplikasi bisa dilihat pada Gambar 2.



**Gambar 2.** Tampilan Aplikasi Scratch Berbasis Website

Pada tahapan pelatihan, kegiatan meliputi pengenalan kode dan elemen dasar serta tahap penyusunan algoritma dalam pembuatan game yang dilaksanakan di laboratorium komputer. Instruktur memulai dengan menjelaskan fungsi berbagai kode dan elemen dalam aplikasi Scratch, sementara peserta didik diminta untuk menirukan instruksi melalui layar proyektor. Hal ini dilakukan, agar peserta didik memahami makna setiap kode dan elemen dalam aplikasi. Proses ini kemudian dilanjutkan dengan pembuatan algoritma dan game "menangkap benda" yang dirancang secara interaktif. Pembuatan algoritma tidak dilakukan melalui penulisan teks yang rumit, melainkan dengan memberikan petunjuk tujuan game sehingga peserta didik harus menentukan sendiri kombinasi kode dan elemen yang tepat. Proses membuat algoritma dilakukan bersama-sama dan dalam bimbingan instruktur. Meskipun demikian, peserta didik diberikan kebebasan penuh dalam memilih elemen visual game mereka, yang bertujuan untuk memicu kreativitas sekaligus memberikan fondasi bahwa pemrograman adalah tentang pemecahan masalah secara logis. Kegiatan pengabdian bisa dilihat pada Gambar 3.



**Gambar 3.** Kegiatan Pengabdian di SD Negeri 4 Sragen

Pada tahapan evaluasi, instruktur melakukan pengecekan secara mendalam terhadap hasil aplikasi yang telah dikembangkan oleh setiap peserta didik guna memastikan prinsip koding telah terpenuhi dengan benar. Peserta didik tidak hanya diminta untuk menunjukkan hasil teknisnya, tetapi juga didorong untuk bercerita mengenai pengalaman serta proses logika yang mereka lalui dalam menciptakan game menggunakan aplikasi Scratch. Aktivitas ini memperkuat teori bahwa *computational thinking* dapat diperoleh secara efektif dari belajar pemrograman, di mana koding membiasakan siswa untuk berpikir komputasi dengan memecahkan masalah dalam bentuk kode. Hal ini selaras dengan temuan Wong and Jiang (2018) bahwa pemrograman mampu meningkatkan kemampuan CT siswa, khususnya pada aspek kemampuan analisis masalah. Semua peserta didik pada akhirnya mampu membuat game “tangkap benda” dengan aplikasi Scratch, yang membuktikan bahwa antarmuka interaktif dan sistem *drag-and-drop* pada aplikasi ini sangat efektif dalam membantu siswa mengonstruksi alur logika secara konkret.

Berdasarkan rekapitulasi hasil observasi partisipatif pada Tabel 1, ditemukan bahwa peserta didik di SD Negeri 4 Sragen menunjukkan progres positif dalam mengimplementasikan *computational thinking*. Scratch terbukti menjadi media transisi yang efektif bagi siswa sekolah dasar karena mampu meminimalisir hambatan teknis bahasa pemrograman yang kompleks. Keberhasilan ini teridentifikasi dari tingkat penyelesaian tugas, di mana sekitar 90% peserta didik berhasil menyelesaikan pembuatan game secara mandiri dengan mengikuti seluruh instruksi algoritma yang diberikan. Sementara itu, 10% peserta didik lainnya mampu menyelesaikan proyek dengan pendampingan intensif akibat kendala kecepatan adaptasi pada instruksi teknis. Secara keseluruhan, penggunaan Scratch memfasilitasi siswa untuk mengenal dan menerapkan pilar-pilar berpikir komputasional secara aplikatif dan menyenangkan melalui karya digital.

Tabel 1. Rekapitulasi Kemampuan Akhir Siswa SD N 4 Sragen (n=26)

KATEGORI CAPAIAN	PERSENTASI	DESKRIPSI KINERJA PESERTA
Mandiri dan Kreatif	60%	Berhasil menyelesaikan game dan mampu melakukan modifikasi mandiri (mengubah suara, mempercepat gerakan, atau mengganti objek).
Mandiri (sesuai Instruksi)	30%	Berhasil menyelesaikan game tepat waktu sesuai dengan seluruh algoritma yang diajarkan instruktur.
Perlu Pendampingan	10%	Menyelesaikan game dengan bantuan teknis dari tim pengabdian karena kendala kecepatan belajar.
Total	100%	

## 5. KESIMPULAN

Pengabdian ini bertujuan untuk memberikan pemahaman kemampuan *computational thinking*. Kegiatan ini dilaksanakan oleh Universitas Sragen pada 14 November 2025 di SD Negeri 4 Sragen. Metode pengabdian yang digunakan adalah pelatihan dan ceramah yang terbagi menjadi tiga tahapan yaitu persiapan, pelatihan, dan evaluasi. Melalui kegiatan ini, peserta didik mampu memahami dan mengenal kemampuan *computational thinking*. Para peserta didik terbukti mampu menerapkan empat pilar utama *computational thinking*, mulai dari dekomposisi saat menganalisis karakter game, pengenalan pola dalam menerapkan kode, abstraksi saat berfokus pada rancangan solusi, hingga penyusunan algoritma untuk menggerakkan *sprite* sesuai instruksi permainan. Dekomposisi dilakukan pada saat peserta didik menganalisis karakter yang ada di game. Kedua, pengenalan pola. Peserta didik menerapkan pola sesuai kode yang telah diajarkan. Ketiga, abstraksi, dimana peserta didik berfokus pada merancang solusi, dan yang terakhir algoritma, yaitu peserta didik membuat langkah-langkah agar *sprite* (karakter) bergerak sesuai instruksi dalam game yang dikembangkan. Sehingga diperoleh kesimpulan bahwa pengabdian ini berjalan dengan baik dan sesuai dengan tujuan awal. Implikasi

kegiatan ini adalah untuk mengenalkan pada siswa sekolah dasar bahwa pemrograman itu mudah dan menyenangkan. Meskipun demikian, pada pelatihan ini ditemukan beberapa kendala teknis antara lain perangkat yang disediakan oleh sekolah. Laptop yang disediakan di laboratorium komputer adalah laptop. Laptop yang disediakan masih menggunakan Windows 7 dan baterai BIOS nya mati, sehingga harus dilakukan pengaturan waktu terlebih dahulu agar bisa terhubung ke internet. Oleh karena itu, disarankan pada pelatihan selanjutnya untuk melakukan pengecekan pada perangkat yang disediakan oleh sekolah agar pelatihan bisa dilaksanakan dengan lebih maksimal.

## 6. UCAPAN TERIMA KASIH

Terima kasih kami ucapkan kepada Sekolah Dasar Negeri 4 Sragen yang telah berkenan menjadi tempat pengabdian masyarakat dan memfasilitasi kegiatan pengabdian kami.

## 7. DAFTAR RUJUKAN

- Agstringtyas, A. S., Sugiarto, V., & Imanda, A. C. (2024). Pelatihan Scratch Application untuk Pengenalan Computational Thinking Skill pada Tenaga Pendidik SD Islam Tompokersan-Lumajang. *J-Dinamika: Jurnal Pengabdian Masyarakat*, 9(2). <https://doi.org/10.25047/j-dinamika.v9i2.4698>
- Alimin, R. A., Misrawita, Hendri, G., & Darul, I. (2025). Evaluasi Pembelajaran Coding Scratch Berbasis Computational Thinking dengan Model Kirkpatrick di Sekolah. *AJMIE: Alhikam Journal of Multidisciplinary Islamic Education*, 6(1), 54–72. <https://doi.org/10.32478/spwy7n37>
- Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior*, 105, 106185. <https://doi.org/10.1016/j.chb.2019.106185>
- Angraini, L. M., Yolanda, F., & Muhammad, I. (2023). Augmented reality: The improvement of computational thinking based on students' initial mathematical ability. *International Journal of Instruction*, 16(3), 1033–1054. <https://doi.org/10.29333/iji.2023.16355a>
- Anwar, K., Kamid, K., Kurniawan, W., Sofnidar, S., & Alrizal, A. (2025). Peningkatan Ketrampilan Berfikir Computational Thinking Dalam Pembelajaran Berbasis Programming-Matematika Menggunakan Scratch. *Jurnal JUPEMA*, 4(1), 50–60. <https://doi.org/10.22437/jupema.v4i1.38084>
- Ariyan, Z., Andy Hidayat, J., Wirarama, W., & Ahmad Zafrullah, M. (2021). Pengenalan Algoritma Pemrograman Menggunakan Aplikasi Scratch Bagi Siswa SD 13 Mataram. *Jurnal Begawe Teknologi Informasi (JBegaTI)*, 2(1). <https://doi.org/10.29303/jbegati.v2i1.423>
- Bakala, E., Gerosa, A., Hourcade, J. P., & Tejera, G. (2021). Preschool children, robots, and computational thinking: A systematic review. *International Journal of Child-Computer Interaction*, 29, 100337. <https://doi.org/10.1016/j.ijcci.2021.100337>
- Curi, M. E., Gerosa, A., Viera, M., & Carboni, A. (2025). A review of computational thinking interventions in upper elementary education. *Computers and Education Open*, 100252. <https://doi.org/10.1016/j.caeo.2025.100252>
- Hermita, N., Alim, J. A., Almais, A. T. W., Vebrianto, R., Thahir, M., Wijaya, T. T., & Mandiro, M. A. (2024). Evaluating primary students' motivation and computational thinking in scratch-based learning: a confusion matrix analysis. *Primary: Jurnal Pendidikan Guru Sekolah Dasar*, 13(6), 264–273. <https://doi.org/10.33578/jpkip-v13i6.p264-273>
- Hsu, T.-C., Chang, S.-C., & Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, 126, 296–310. <https://doi.org/10.1016/j.compedu.2018.07.004>
- Huda, N., & Rohaeti, E. (2024). Computational Thinking Skill Level of Senior High School Students Majoring in Natural Science. *International Journal of Learning, Teaching and Educational Research*, 23(1), 339–359. <https://doi.org/10.26803/ijlter.23.1.17>

- Irawan, E., Rosjanuardi, R., & Prabawanto, S. (2024). Promoting Computational Thinking through Programming Trends, Tools, and Educational Approaches: a Systematic Review. *JTAM (Jurnal Teori dan Aplikasi Matematika)*, 8(4), 1327–1348. <https://doi.org/10.31764/jtam.v8i4.26407>
- Kite, V., Park, S., & Wiebe, E. (2021). The code-centric nature of computational thinking education: A review of trends and issues in computational thinking education research. *SAGE Open*, 11(2). <https://doi.org/10.1177/21582440211016418>
- Latifah, R., Setiani, H., & Trisanti, N. (2025). Pelatihan Pengenalan Huruf dan Angka Menggunakan Aplikasi Qreatif. *Masyarakat: Jurnal Pengabdian*, 2(1), 81–87. <https://doi.org/10.58740/m-jp.v2i1.352>
- Lubis, S., Saragih, A., & Sinar, T. S. (2023). The Effectiveness of Scratch Coding Activities in English Language Learning. *World Journal of English Language*, 13(7). <https://doi.org/10.5430/wjel.v13n7p508>
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51–61. <https://doi.org/10.1016/j.chb.2014.09.012>
- Masfingatin, T., & Maharani, S. (2019). Computational thinking: Students on proving geometry theorem. *International Journal of Scientific & Technology Research*, 8(09), 2216–2223.
- Molina-Ayuso, Á., Adamuz-Povedano, N., Bracho-López, R., & Torralbo-Rodríguez, M. (2024). Computational thinking with Scratch: A tool to work on geometry in the fifth grade of primary education. *Sustainability*, 16(1), 110. <https://doi.org/10.3390/su16010110>
- Ouahouda, F., Khadija, A., & Achtaich, N. (2025). Incorporation of Scratch Programming and Algorithmic Resource Design in Primary Education. *Engineering Proceedings*, 107(1), 40. <https://doi.org/10.3390/engproc2025107040>
- Putra, G. M. C., Prasetyaningtyas, F. D., Ansori, I., Kurnianto, B., Wahyuni, N. I., Xinzhe, Z., & Nurmanto, T. N. (2025). Effectiveness of Project-Based Scratch Programming to Improve Problem-Solving Skills of Elementary School Students. *International Journal of Social Learning (IJSL)*, 5(2), 460–478. <https://doi.org/10.47134/ijsl.v5i2.417>
- Rahayu, Y. D., Dewi, I. C., Bahariawan, R. A., Gumilang, A. S., Aini, S. N., & Dinata, N. A. S. (2022). Introduction to Computational Thinking at SD Muhammadiyah 1 Jember Through Game Making Training Among Us. *Mattawang: Jurnal Pengabdian Masyarakat*, 3(4), 485–488. <https://doi.org/10.35877/454RI.mattawang1275>
- Selby, C., & Woollard, J. (2013). *Computational thinking: The developing definition*. University of Southampton.
- Syafaat, M. (2022). Pelatihan Problem Solving Pada HMJ Teknik Informatika UIN Alauddin Makassar Menggunakan Metode Computational Thinking. *Abdi Samulang: Jurnal Pengabdian Kepada Masyarakat*, 1(1), 17–23. <https://doi.org/10.61477/abdisamulang.v1i1.6>
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717–3725. <https://doi.org/10.1098/rsta.2008.0118>
- Wing, J. M. (2014). *Computational thinking benefits society*. 40th anniversary blog of social issues in computing.
- Wong, G. K., & Jiang, S. (2018). Computational thinking education for children: Algorithmic thinking and debugging. *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*. <https://doi.org/10.1109/TALE.2018.8615232>
- Yang, S.-Y., Lin, Y.-C., & Lin, Y.-T. (2024). Improving elementary students' computational thinking skills through an educational robot intervention: A quasi-experimental study. *International Journal of Learning, Teaching and Educational Research*, 23(9), 343–360. <https://doi.org/10.26803/ijlter.23.9.18>
- Yunus, A., Artin Bayu, M., & Sri, M. (2023). Pelatihan Pemrograman Scratch Bagi Guru-Guru SD Islam Al Madina Semarang. *Jurnal Pengabdian Masyarakat Bumi Raflesia*, 6(2), 198–205. <https://doi.org/10.36085/jpmb.v6i2.5292>