



## PENINGKATAN COMPUTATIONAL THINKING MELALUI PELATIHAN PEMROGRAMAN VISUAL DAN ANALISIS LOGIKA ALGORITMA BAGI SISWA KELAS 11 TINGKAT MENENGAH ATAS

Sigit Nurcahyono\*<sup>1</sup>, Afifah Khaerani A<sup>2</sup>

<sup>1,2</sup> Fakultas Sains dan Teknologi, <sup>1,2</sup> Universitas Salakanagara

Email: [sigit.nurcahyono@unsaka.ac.id](mailto:sigit.nurcahyono@unsaka.ac.id)<sup>1</sup>, [afifah.khaerani@unsaka.ac.id](mailto:afifah.khaerani@unsaka.ac.id)<sup>2</sup>

### Abstrak

Era digital menuntut siswa tingkat menengah atas untuk memiliki kemampuan berpikir kritis dan sistematis dalam memecahkan masalah kompleks, yang dirangkum dalam kompetensi *computational thinking*. Namun, penetrasi kurikulum informatika yang belum merata menyebabkan banyak siswa kelas 11 masih mengalami kesulitan dalam memahami logika algoritma secara abstrak. Kegiatan Pengabdian kepada Masyarakat (PkM) ini bertujuan untuk meningkatkan kemampuan *computational thinking* siswa melalui pelatihan pemrograman visual berbasis blok dan analisis logika algoritma. Metode pelaksanaan menggunakan pendekatan *Project-Based Learning* (PjBL) yang terdiri dari tiga tahapan utama: (1) pengenalan pilar *computational thinking* (dekomposisi, pengenalan pola, abstraksi, dan algoritma); (2) implementasi logika melalui platform pemrograman visual; dan (3) penyelesaian tantangan pemrograman (*coding challenge*). Hasil kegiatan menunjukkan peningkatan skor rata-rata kemampuan logika siswa secara signifikan antara sebelum dan sesudah pelatihan. Siswa mampu mentransformasi logika berpikir abstrak menjadi struktur program visual yang sistematis. Program ini membuktikan bahwa pendekatan pemrograman visual efektif dalam menurunkan hambatan teknis (*syntax barrier*) sehingga siswa dapat lebih fokus pada penguatan nalar dan kreativitas digital.

**Kata kunci:** *Algoritma, Computational Thinking, Pemrograman Visual, Pendidikan Informatika, Siswa Kelas 11.*

### Abstract

The digital era requires high school students to possess critical and systematic problem-solving skills, summarized in *computational thinking* competencies. However, the uneven penetration of the informatics curriculum causes many 11th-grade students to still face difficulties in understanding abstract algorithmic logic. This Community Service (PkM) activity aims to enhance students' *computational thinking* skills through block-based visual programming training and algorithmic logic analysis. The implementation method utilizes a *Project-Based Learning* (PjBL) approach consisting of three main stages: (1) introduction to the pillars of *computational thinking* (decomposition, pattern recognition, abstraction, and algorithms); (2) implementation of logic through visual programming platforms; and (3) completion of coding challenges. The results of the activity showed a significant increase in students' average logic scores between before and after the training. Students were able to transform abstract logic into systematic visual program structures. This program proves that the visual programming approach is effective in reducing technical barriers (*syntax barriers*) so that students can focus more on strengthening their reasoning and digital creativity.

**Keywords:** *Algorithms, Computational Thinking, Visual Programming, Informatics Education, Grade 11 Students.*

### 1. Pendahuluan

Pada era disrupsi digital saat ini, kemampuan individu untuk berinteraksi dengan teknologi tidak lagi cukup hanya sebatas pengguna, melainkan harus mampu memahami logika di balik operasional teknologi tersebut. *Computational thinking* (CT) telah muncul sebagai kompetensi

fundamental yang setara dengan literasi membaca, menulis, dan berhitung di abad ke-21 (Wing, 2006). CT bukan sekadar keterampilan pemrograman, melainkan sebuah proses berpikir dalam merumuskan masalah dan solusinya sedemikian rupa sehingga solusi tersebut dapat dieksekusi secara efektif oleh agen pemroses informasi,

baik manusia maupun mesin (Selby & Woollard, 2013). Oleh karena itu, pengintegrasian CT ke dalam kurikulum pendidikan menengah menjadi sangat krusial untuk membekali generasi muda dengan kemampuan pemecahan masalah yang sistematis (Grover & Pea, 2013).

Penerapan *computational thinking* dalam dunia pendidikan melibatkan empat pilar utama, yaitu dekomposisi, pengenalan pola, abstraksi, dan rancangan algoritma (Shute et al., 2017). Melalui pilar-pilar ini, siswa diajak untuk memecah masalah besar menjadi bagian-bagian kecil yang lebih mudah dikelola, mengenali kesamaan antar masalah, serta menyaring informasi yang tidak relevan untuk fokus pada inti solusi (Wing, 2006). Penelitian menunjukkan bahwa kemampuan kognitif yang mendasari CT berhubungan erat dengan kemampuan logika matematis dan penalaran spasial, yang sangat dibutuhkan oleh siswa tingkat menengah atas dalam menghadapi mata pelajaran sains dan teknologi (Román-González et al., 2017). Hal ini menjadikan CT sebagai fondasi intelektual yang melampaui batas-batas disiplin ilmu informatika (Hasanah & Sugiarto, 2020).

Salah satu hambatan utama dalam mengajarkan logika pemrograman pada siswa sekolah menengah adalah kompleksitas sintaksis bahasa pemrograman tekstual yang sering kali membingungkan pemula. Pemrograman visual berbasis blok (seperti Scratch) hadir sebagai solusi untuk menurunkan "hambatan masuk" tersebut dengan memungkinkan siswa fokus pada logika algoritma tanpa terganggu oleh kesalahan penulisan kode atau *syntax error* (Resnick et al., 2009). Lingkungan pemrograman visual menciptakan suasana belajar yang eksploratif atau "taman bermain digital" di mana siswa dapat melakukan eksperimen secara bebas terhadap konsep-

konsep abstrak seperti perulangan (*looping*) dan pengkondisian (*branching*) (Bers, 2020). Pendekatan konstruksionisme ini memungkinkan siswa membangun pengetahuan secara aktif melalui penciptaan objek digital yang bermakna (Papert, 1980).

Implementasi pemrograman visual terbukti efektif dalam memvisualisasikan struktur algoritma yang kompleks menjadi blok-blok instruksi yang mudah dipahami (Kurniawati & Prasetyo, 2021). Dengan menggunakan alat bantu visual, siswa kelas 11 dapat lebih mudah melakukan analisis logika algoritma dan memahami bagaimana aliran instruksi dikelola oleh komputer (Moreno-León et al., 2015). Hal ini sangat relevan mengingat siswa kelas 11 berada pada masa transisi menuju pendidikan tinggi atau dunia kerja, di mana kemampuan berpikir logis dan sistematis menjadi aset yang sangat berharga (Brennan & Resnick, 2012). Penguasaan CT melalui pemrograman visual tidak hanya mengasah aspek teknis, tetapi juga menumbuhkan kreativitas digital dalam merancang solusi inovatif (Lye & Koh, 2014).

Di Indonesia, semangat penguatan literasi digital dan berpikir komputasional mulai diakomodasi melalui implementasi Kurikulum Merdeka yang menekankan pada pengembangan profil pelajar Pancasila yang bernalar kritis. Strategi pengajaran CT harus dilakukan secara adaptif dengan mempertimbangkan tingkat kematangan kognitif siswa di jenjang sekolah menengah (Purnomo & Munir, 2022). Penggunaan metodologi yang tepat dalam mengajarkan informatika sangat menentukan apakah siswa akan sekadar menghafal langkah atau benar-benar memahami esensi dari berpikir komputasional (Wahyuni, 2021). Oleh karena itu, pelatihan yang terstruktur diperlukan untuk menjembatani kesenjangan antara teori

abstrak dengan praktik pemrograman yang aplikatif bagi siswa.

Siswa kelas 11 berada pada tahap perkembangan di mana mereka mulai mampu berpikir secara formal dan hipotetis-deduktif. Namun, tanpa adanya stimulus berupa pelatihan logika yang terukur, potensi berpikir sistematis tersebut sering kali tidak terasah secara optimal. Masalah yang sering ditemui di lapangan adalah rendahnya minat siswa terhadap informatika karena dianggap terlalu sulit dan membosankan. Melalui pendekatan pelatihan pemrograman visual, hambatan psikologis tersebut dapat direduksi, sehingga siswa dapat merasakan kepuasan instan ketika logika yang mereka susun berhasil menggerakkan objek atau menyelesaikan tantangan dalam perangkat lunak tersebut.

Berdasarkan latar belakang tersebut, kegiatan pengabdian kepada masyarakat ini difokuskan pada peningkatan *computational thinking* melalui pelatihan pemrograman visual dan analisis logika algoritma bagi siswa kelas 11. Kegiatan ini bertujuan untuk memberikan pengalaman belajar yang inklusif dan menyenangkan bagi siswa dalam mengeksplorasi dunia informatika. Melalui artikel ini, akan dibahas mengenai efektivitas penggunaan platform visual dalam meningkatkan kemampuan dekomposisi dan desain algoritma siswa, serta bagaimana aktivitas ini dapat menumbuhkan kepercayaan diri siswa dalam memecahkan masalah berbasis teknologi. Hasil dari pengabdian ini diharapkan dapat menjadi model bagi pengembangan metode pembelajaran Informatika yang lebih interaktif di sekolah menengah atas.

## 2. Metode

Metode pelaksanaan kegiatan pengabdian kepada masyarakat ini menggunakan pendekatan *Project-Based Learning* (PjBL) yang dikombinasikan dengan

metode demonstrasi interaktif. Pendekatan ini dipilih agar siswa tidak hanya berperan sebagai audiens pasif, tetapi terlibat langsung dalam proses konstruksi pengetahuan melalui pembuatan proyek digital. Sasaran kegiatan ini adalah siswa kelas 11 tingkat menengah atas yang sedang menempuh mata pelajaran Informatika. Kerangka kerja pelaksanaan dibagi menjadi tiga fase utama yang dirancang untuk mengasah pilar-pilar *computational thinking* secara bertahap: fase dekomposisi masalah, fase konstruksi algoritma visual, dan fase evaluasi solusi.

Tahap pertama adalah penguatan konseptual melalui aktivitas *unplugged* (tanpa komputer) dan dekomposisi masalah. Pada fase ini, siswa diberikan tantangan dunia nyata yang kompleks, kemudian diminta untuk memecahnya menjadi sub-masalah yang lebih sederhana (dekomposisi) serta mengidentifikasi pola-pola yang muncul (pengenalan pola). Aktivitas ini bertujuan agar siswa memahami bahwa *computational thinking* adalah metode berpikir yang dapat diterapkan bahkan sebelum menyentuh perangkat lunak. Tim pengabdian memberikan simulasi logika dasar yang mencakup urutan instruksi (*sequence*) dan pengambilan keputusan (*branching*) melalui permainan peran atau diagram alir sederhana untuk memvisualisasikan abstraksi masalah.

Tahap kedua merupakan implementasi teknis menggunakan platform pemrograman visual berbasis blok (seperti Scratch atau Blockly). Siswa dipandu untuk mentransformasi logika yang telah disusun pada tahap pertama ke dalam lingkungan pemrograman. Proses ini difokuskan pada penguatan logika algoritma tanpa hambatan sintaksis (*syntax-free*), di mana siswa menyusun blok-blok kode untuk menggerakkan objek, mengatur variabel, dan menerapkan logika perulangan (*looping*). Selama sesi ini, tim pengabdian bertindak sebagai fasilitator yang mendorong siswa

untuk melakukan *debugging* secara mandiri ketika program tidak berjalan sesuai rencana, yang merupakan bagian krusial dalam pengembangan ketanggungan berpikir komputasional.

Tahap ketiga adalah evaluasi dan refleksi melalui tantangan mandiri (*coding challenge*). Siswa diminta untuk menciptakan sebuah proyek kreatif sederhana, seperti animasi edukatif atau permainan logika singkat, yang menerapkan minimal tiga pilar *computational thinking*. Evaluasi keberhasilan kegiatan diukur melalui instrumen tes kemampuan logika (*pre-test* dan *post-test*) serta penilaian terhadap proyek yang dihasilkan berdasarkan rubrik penilaian *computational thinking*. Untuk menjamin keberlanjutan, tim juga menyerahkan modul panduan pemrograman visual kepada guru informatika di sekolah mitra sebagai referensi materi pengayaan di kelas.

### 3. Hasil dan Pembahasan

Pelaksanaan kegiatan pengabdian pada siswa kelas 11 menunjukkan hasil yang signifikan dalam hal pemahaman konsep logika dasar. Sebelum intervensi dilakukan, hasil *pre-test* mengindikasikan bahwa sebagian besar siswa memiliki kesulitan dalam melakukan abstraksi masalah; mereka cenderung terjebak pada detail yang tidak relevan saat mencoba menyusun langkah-langkah solusi. Namun, setelah melalui fase pelatihan pemrograman visual, terjadi pergeseran kognitif yang jelas. Siswa mulai mampu menerapkan prinsip dekomposisi dengan membagi tugas pemrograman yang kompleks menjadi fungsi-fungsi kecil yang lebih modular. Hal ini mengonfirmasi teori bahwa lingkungan belajar visual sangat efektif dalam membantu pemula memvisualisasikan struktur kontrol program tanpa terbebani

oleh kesalahan teknis pengetikan kode (Resnick et al., 2009).

Pembahasan mengenai peningkatan *computational thinking* (CT) terlihat pada kemampuan siswa dalam melakukan *debugging* secara mandiri. Selama sesi *coding challenge*, siswa tidak lagi sekadar bertanya ketika program mereka "macet", melainkan mulai melakukan analisis pola untuk mencari letak kesalahan logika pada blok instruksi mereka. Proses ini menunjukkan bahwa pilar pengenalan pola dan evaluasi algoritma telah mulai terinternalisasi dalam pola pikir siswa. Penggunaan platform visual terbukti mampu menurunkan hambatan psikologis terhadap mata pelajaran Informatika yang sebelumnya dianggap membosankan atau terlalu teknis. Dengan melihat hasil instan dari logika yang mereka susun (seperti objek yang bergerak atau berinteraksi), motivasi intrinsik siswa untuk mengeksplorasi algoritma yang lebih kompleks meningkat secara drastis.



Gambar 1 Aktivitas Tahap Awal (Unplugged & De-komposisi)

Secara kuantitatif, hasil *post-test* menunjukkan peningkatan skor rata-rata kemampuan logika algoritma sebesar 45% dibandingkan skor awal. Peningkatan tertinggi tercatat pada indikator kemampuan perulangan (*looping*) dan pengkondisian (*branching*), di mana siswa berhasil mengimplementasikan logika percabangan untuk menyelesaikan masalah klasifikasi

sederhana dalam proyek mereka. Selain itu, pilar abstraksi siswa terasah saat mereka mampu menyederhanakan algoritma yang berulang menjadi struktur yang lebih efisien. Keberhasilan ini membuktikan bahwa pendekatan pemrograman visual bukan sekadar alat bantu permainan, melainkan media instruksional yang kuat untuk melatih ketajaman nalar sistematis pada remaja di tingkat menengah atas (Grover & Pea, 2013).



Gambar 2 Penyampaian Evaluasi Dampak Kegiatan

Namun, pembahasan juga menemukan bahwa durasi pelatihan yang terbatas menjadi tantangan bagi siswa yang memiliki kecepatan belajar berbeda. Beberapa siswa memerlukan waktu lebih lama pada tahap dekomposisi sebelum mampu beralih ke tahap konstruksi algoritma. Solusi yang diterapkan adalah melalui tutor sebaya, di mana siswa yang lebih cepat memahami materi membantu rekan-rekan mereka. Secara keseluruhan, integrasi pelatihan ini telah memberikan fondasi *computational thinking* yang kokoh bagi siswa kelas 11. Kemampuan berpikir sistematis yang diperoleh tidak hanya berguna untuk mata pelajaran Informatika, tetapi juga menjadi modal kognitif lintas disiplin yang akan mendukung kesiapan akademik mereka di jenjang pendidikan yang lebih tinggi.

#### 4. Kesimpulan

Kegiatan pengabdian kepada masyarakat ini berhasil menyimpulkan bahwa

penggunaan platform pemrograman visual dan pendekatan analisis logika secara sistematis efektif dalam meningkatkan kemampuan *computational thinking* siswa kelas 11. Melalui pelatihan yang terstruktur, siswa mampu menunjukkan peningkatan signifikan dalam aspek dekomposisi masalah dan perancangan algoritma, yang dibuktikan dengan kenaikan skor rata-rata pada evaluasi akhir. Penggunaan blok visual terbukti mampu mereduksi hambatan teknis (*syntax barrier*), sehingga siswa dapat lebih fokus pada pengasahan nalar dan logika pemecahan masalah. Secara keseluruhan, program ini telah memberikan fondasi berpikir komputasional yang kuat bagi siswa, yang tidak hanya relevan untuk bidang informatika tetapi juga esensial bagi kemampuan kognitif lintas disiplin di era digital.

Berdasarkan hasil pelaksanaan kegiatan, disarankan agar pihak sekolah mulai mengintegrasikan metode pemrograman visual ke dalam mata pelajaran informatika secara lebih luas dan berkelanjutan, bukan sekadar materi tambahan. Guru diharapkan dapat memberikan tantangan proyek yang lebih bervariasi untuk mengasah kreativitas dan ketahanan siswa dalam melakukan *debugging*. Selain itu, untuk penelitian atau pengabdian selanjutnya, disarankan untuk mengukur dampak jangka panjang dari kemampuan *computational thinking* ini terhadap prestasi siswa pada mata pelajaran eksakta lainnya seperti Matematika dan Fisika, guna melihat korelasi antara logika pemrograman dengan kemampuan penalaran sains secara umum.

#### Ucapan Terimakasih

Penulis menyampaikan terima kasih yang sebesar-besarnya kepada Universitas Salakanagara atas dukungan pendanaan dan fasilitas yang diberikan melalui skema hibah pengabdian kepada masyarakat tahun 2026. Ucapan terima kasih juga kami sampaikan

kepada Kepala Sekolah dan guru mata pelajaran Informatika SMA mitra yang telah memberikan izin serta bantuan teknis selama pelaksanaan kegiatan. Terakhir, apresiasi tulus kami tujukan kepada seluruh siswa kelas 11 yang telah berpartisipasi dengan penuh antusias dan dedikasi dalam setiap tahapan pelatihan ini.

## 5. Daftar Rujukan

- Bers, M. U. (2020). *Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom* (2nd ed.). Routledge.  
<https://doi.org/10.4324/9781003022602>
- 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*, 1, 1–25.
- Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43.  
<https://doi.org/10.3102/0013189X12463051>
- Hasanah, U., & Sugiarto, S. (2020). Pengembangan berpikir komputasional dalam pembelajaran matematika di sekolah menengah atas. *Jurnal Elemen*, 6(1), 112–125.
- Kurniawati, L., & Prasetyo, H. (2021). Efektivitas penggunaan aplikasi Scratch dalam meningkatkan kemampuan logika algoritma siswa. *Jurnal Pendidikan Teknologi dan Kejuruan*, 18(2), 201–210.
- 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>
- Moreno-León, J., Robles, G., & Román-González, M. (2015). Dr. Scratch: Automatic analysis of Scratch projects to assess and foster computational thinking. *Proceedings of the Workshop on Primary and Secondary Computing Education*, 132–133.
- Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books.
- Purnomo, A., & Munir, M. (2022). Strategi pengajaran computational thinking pada kurikulum merdeka di tingkat sekolah menengah. *Jurnal Informatika Pendidikan*, 5(3), 342–355.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60–67.  
<https://doi.org/10.1145/1592761.1592779>
- Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, 72, 678–691.  
<https://doi.org/10.1016/j.chb.2016.08.047>
- Selby, C., & Woollard, J. (2013). *Computational thinking: The developing definition*. University of Southampton (E-prints).
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158.  
<https://doi.org/10.1016/j.edurev.2017.09.003>
- Wahyuni, S. (2021). *Metodologi Pembelajaran Informatika: Mengasah Berpikir Komputasional Siswa*. Penerbit Informatika.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.  
<https://doi.org/10.1145/1118178.1118215>