EDUTECH

Jurnal Teknologi Pendidikan

Journal homepage https://ejournal.upi.edu/index.php/edutech

Model Gradual Release of Responsibility Untuk Pengembangan Berpikir Algoritmik Siswa SMP: A Systematic Literature Review"

Maryam Nur Azizah Rahmah & Titi Prihatin
Program Studi Pengembangan Kurikulum, Program Pascasarjana
Universitas Negeri Semarang, Indonesia
Correspondence Email :maryam_rahmah@students.unnes.ac.id

ABSTRACT

This study aims to examine the relationship between the Gradual Release of Responsibility (GRR) model, programming algorithm instruction, and the development of middle school students' algorithmic thinking skills. The study employed a Systematic Literature Review approach in accordance with the PRISMA 2020 guidelines. Literature was collected from Scopus, Web of Science, IEEE Xplore, the ACM Digital Library, and Google Scholar, and then screened based on relevant inclusion and exclusion criteria. The final synthesis included 37 articles that were analyzed thematically. The findings indicate that direct evidence of GRR's application in programming algorithm instruction for junior high school students remains limited. However, studies on scaffolding, guided practice, progressive scaffolding, visual programming, and introductory programming instruction provide strong conceptual support for the relevance of GRR. The GRR model is deemed appropriate because it provides learning stages ranging from teacher modeling, guided practice, collaboration, to independent practice. The resulting conceptual framework positions Code.org and Scratch as implementation tools to support the development of algorithmic thinking indicators, namely sequencing, conditionals, looping, tracing, and debugging. This study emphasizes the need for programming algorithm instruction that focuses on the thinking process, rather than merely the program's output.

ABSTRAK

Penelitian ini bertujuan mengkaji keterkaitan antara model Gradual Release of Responsibility (GRR), pembelajaran algoritma pemrograman, dan pengembangan kemampuan berpikir algoritmik siswa SMP. Penelitian menggunakan pendekatan Systematic Literature Review dengan mengacu pada alur PRISMA 2020. Literatur dikumpulkan dari Scopus, Web of Science, IEEE Xplore, ACM Digital Library, dan Google Scholar, kemudian diseleksi berdasarkan kriteria inklusi dan eksklusi yang relevan. Hasil akhir sintesis mencakup 37

ARTICLE INFO

Article History:
Submitted/Received 12 Aprl 2025
First Revised 26 Mei 2026
Accepted 15 June 2026
First Available online 30 June 2026 Publication Date 30 June 2026

Keyword:
Algorithmic thinking; computational thinking; gradual release of responsibility; programming instruction; scaffolding.

artikel yang dianalisis secara tematik. Temuan menunjukkan bahwa bukti langsung penerapan GRR dalam pembelajaran algoritma pemrograman siswa SMP masih terbatas. Namun, studi tentang scaffolding, guided practice, progressive scaffolding, visual programming, dan pembelajaran pemrograman pemula memberikan dukungan konseptual yang kuat terhadap relevansi GRR. Model GRR dinilai sesuai karena menyediakan tahapan pembelajaran dari pemodelan guru, latihan terbimbing, kolaborasi, hingga praktik mandiri. Kerangka konseptual yang dihasilkan menempatkan Code.org dan Scratch sebagai media implementasi untuk mendukung pengembangan indikator berpikir algoritmik, yaitu sequencing, conditional, looping, tracing, dan debugging. Kajian ini menegaskan perlunya pembelajaran algoritma pemrograman yang berfokus pada proses berpikir, bukan sekadar produk program.

A. PENDAHULUAN

Kemampuan berpikir komputasional (*Computational Thinking*/ CT) saat ini menjadi salah satu kompetensi utama yang perlu dimiliki, khususnya digunakan sebagai pendekatan dalam menyelesaikan masalah (Wing, 2026). Dalam konteks ini, salah satu pilar CT, yaitu kemampuan berpikir algoritmik, menjadi fondasi dalam memahami materi algoritma dan pemprograman yang meliputi *sequencing, conditional, looping, tracing, dan debugging* (Brennan & Resnick, 2012; Lister et al., 2004). Di tingkat kebijakan nasional, Indonesia telah merespons tuntutan global ini melalui Kurikulum Merdeka yang mengintegrasikan Informatika sebagai mata pelajaran wajib, dan dalam Permendikdasmen Nomor 13 Tahun 2025 yang menetapkan Koding dan Kecerdasan Artifisial (KKA) sebagai mata pelajaran pilihan pada jenjang SMP. Kebijakan ini secara eksplisit menargetkan pengembangan kemampuan berpikir algoritmik, literasi komputasional, dan pemecahan masalah berbasis teknologi (Kemendikdasmen, 2025).

Namun demikian, kebijakan ini berbenturan dengan realitas yang secara empiris cukup memprihatinkan. Data PISA 2022 menempatkan Indonesia pada peringkat ke-68 dari 79 negara dalam pemahaman literasi matematika, yang erat kaitannya dengan penalaran logis sebagai fondasi berpikir algoritmik dengan skor 366 poin, jauh di bawah rata-rata OECD sebesar 472 poin (OECD, 2023). Lebih spesifik lagi, kajian terhadap Bebras Challenge 2023, menunjukkan bahwa 87% dari 3.648 peserta dari berbagai jenjang memperoleh skor di bawah ambang kompetensi minimum, dengan rata-rata skor peserta yang melampaui ambang hanya 56,25 (Fitriyah et al., 2024). Temuan lain, terdapat pada siswa SMP/MTs di Jawa Timur juga menemukan pola serupa dimana kemampuan CT siswa secara keseluruhan berada pada kategori rendah, dengan kelemahan paling menonjol pada aspek algoritmik dan abstraksi (Wahyudhi et al, 2022).

Penelitian dalam pembelajaran *computer science* mengidentifikasi bahwa kesulitan utama pemula bukan pada penguasaan sintaks, namun pada pemahaman bagaimana program bisa berjalan, bagaimana instruksi berinteraksi, bagaimana alur logika mengalir melalui percabangan dan perulangan, dan bagaimana kesalahan logika terjadi (Robins et al., 2003; Lister et al., 2004). Tanpa desain pembelajaran yang tepat, aktivias koding seringkali tereduksi menjadi *trial and error*

DOI: https://doi.org/10.17509/e.v25i2.99888
p-ISSN 0852-1190 e-ISSN 2502-0781

yang menghambat pengembangan berpikir komputasional, tanpa pemahaman konseptual yang baik (Grover & Pea, 2013).

Dalam menghadapi permasalahan kognitif tersebut, terdapat teori *scaffolding* yang berakar pada *Zone of Proximal Development* (Vygotsky,1978) yang menawarkan kerangka solusi yang relevan. Dimana siswa dapat mencapai tingkat pemahaman yang lebih tinggi melalui bantuan sementara yang secara bertahap dikurangi seiring meningkatnya kompetensi. Operasionalisasi yang sistematis dari prinsip s*caffolding* ini ditemukan dalam *Model Gradual Release of Responsibility* (GRR). Model GRR dioperasionalisasikan melalui empat fase, yaitu *Focused Instruction (I do), Guided Instruction (We do), Collaborative Learning (You do together),* dan *Independent Practice (You do alone)* menawarkan kerangka solusi yang relevan dengan kondisi tersebut (Fisher & Frey, 2008). GRR terbukti efektif dalam mendukung pembelajaran lainnya yaitu literasi membaca (Webb et al., 2019), pembelajaran sains (Fuentes & Casinillo, 2024), matematika (Cerveza & Lapinid, 2024), hingga pembelajaran digital secara asinkron (Stevens et al., 2025).

Namun, penelusuran literatur mengungkapkan kesenjangan yang signifikan dimana penerapan GRR yang spesifik dan teroperasionalisasi dalam pembelajaran algoritma pemrograman SMP, dengan fokus pada peningkatan kemampuan berpikir algoritmik yang terukur, hampir tidak tersedia (Chuang et al., 2022; Loksa et al., 2016).

Systematic Literature Review (SLR) ini bertujuan untuk: (1) memetakan integrasi tentang pembelajaran algoritma-pemrograman, berpikir algoritmik, dan model GRR; (2) menganalisis sejauh mana GRR telah diterapkan dalam pembelajaran pemrograman; (3) mengidentifikasi kesenjangan penelitian kritis; dan (4) merumuskan kerangka konseptual untuk pengembangan perangkat pembelajaran berbasis GRR yang meningkatkan kemampuan berpikir algoritmik siswa SMP. Artikel ini secara khusus menempatkan GRR sebagai kerangka pedagogis untuk memahami bagaimana tanggung jawab belajar dapat dialihkan secara bertahap dari guru kepada siswa dalam proses pembelajaran algoritma dan pemrograman.

Kontribusi utama artikel ini terletak pada tiga aspek. Pertama, kajian ini memperjelas bahwa penerapan GRR dalam konteks pemrograman SMP masih jarang digunakan secara eksplisit. Kedua, kajian ini memetakan hubungan antara fase GRR dengan indikator berpikir algoritmik. Ketiga, kajian ini menghasilkan dasar konseptual bagi pengembangan perangkat pembelajaran informatika SMP berbasis GRR yang lebih terarah, terutama untuk mendukung transisi siswa dari mengikuti contoh guru menuju penyusunan dan perbaikan algoritma secara mandiri.

METODE

A. Desain dan Protokol Penelitian

Penelitian ini menggunakan pendekatan *Systematic Literature Review* (SLR) yang dilaksanakan sesuai panduan PRISMA 2020 (*Preferred Reporting Items for Systematic Reviews and Meta Analyses*) dengan didasarkan pada pertanyaan penelitian, bagaimana efektivitas model GRR dalam meningkatkan kemampuan berpikir algoritmik siswa SMP/menengah?

B. Strategi Pencarian Literatur

Pencarian literatur dilakukan secara sistematis pada beberapa basis data, antara lain ; Scopus, Web of Science, IEEE Xplore, ACM Digital Library, dan Google Scholar. Pemilihan database tersebut didasarkan pada cakupan masing-masing. Scopus dan Web of Science digunakan untuk menjangkau jurnal internasional bereputasi lintas disiplin. IEEE Xplore dan ACM Digital Library digunakan karena keduanya sangat relevan dengan publikasi bidang *computer science education, programming education,* dan *computing education.* Google Scholar digunakan sebagai pelengkap

DOI: https://doi.org/10.17509/e.v25i2.99888
p-ISSN 0852-1190 e-ISSN 2502-0781

untuk menjangkau artikel nasional, prosiding, artikel open access, dan publikasi yang mungkin tidak terindeks langsung pada database utama.Strategi pencarian dimulai pada tanggal 22 April 2026, dengan menggunakan kombinasi kata kunci yang dikelompokkan dalam tiga kluster konsep yang dihubungkan oleh operator Boolean.

Tabel 1. Strategi Pencarian dan Kluster Kata Kunci

Kluster	Fokus konsep	Kata kunci utama
A	Model pedagogis	"Gradual Release of Responsibility" OR "GRR model" OR "scaffolded instruction" OR "I do We do You do" OR "gradual release"
B	Domain pembelajaran	"programming education" OR "coding learning" OR "algorithm learning" OR "computational thinking" OR "algorithmic thinking" OR "block-based programming" OR "visual programming"
C	Konteks pendidikan	"junior high school" OR "middle school" OR "secondary education" OR "K-12" OR "novice programmer" OR "beginner programmer"
String utama	Kombinasi pencarian	(Kluster A) AND (Kluster B); pencarian tambahan: (Kluster A) AND (Kluster C)

Sumber: Disusun berdasarkan protokol pencarian SLR dalam naskah penulis.

C. Kriteria Inklusi dan Eksklusi

Kriteria inklusi dan eksklusi ditetapkan sebelum seleksi artikel, bertujuan untuk mempertahankan fokus artikel yang memiliki irisan antara GRR, *Scaffolding*, pembelajaran algoritma pemrogaman, *Computational Thingking* (CT) dan konteks pendidikan SMP sekaligus untuk menjaga kualitas sumber yang disentesis.

Tabel 2. Kriteria Inklusi dan Eksklusi

No	Kriteria	Penjelasan
1	Jenis publikasi	Artikel jurnal atau prosiding konferensi bereputasi; publikasi yang memiliki proses telaah ilmiah.
2	Rentang waktu	Publikasi utama 2015-2026; karya sebelum 2015 digunakan sebagai landasan teoritis.
3	Relevansi substansial	Membahas GRR/*scaffolding*, pembelajaran algoritma-pemrograman, CT, berpikir algoritmik, atau karakteristik pemrogram pemula.
4	Bahasa	Bahasa Indonesia atau Inggris.
5	Ketersediaan teks	Tersedia teks lengkap untuk dievaluasi.
6	Non ilmiah	Blog, opini, laporan teknis tanpa telaah ilmiah, dan sumber non-akademik dikeluarkan.
7	Duplikasi	Artikel duplikat dikeluarkan setelah pemeriksaan manajer referensi dan verifikasi manual.
8	Bahasa di luar cakupan	Artikel selain bahasa Indonesia dan Inggris dikeluarkan.

D. Ekstraksi dan Sintesis Data

Data diekstraksi dari setiap artikel menggunakan formulir terstandar yang memuat: identitas artikel (penulis, tahun, jurnal), negara penelitian, desain/metode penelitian, konteks dan jenjang pendidikan, fokus utama (GRR, CT, pemrograman, atau kombinasinya), instrumen pengukuran yang digunakan, temuan utama, dan keterbatasan yang dilaporkan penulis.

Sintesis dilakukan menggunakan pendekatan tematik (*thematic synthesis*) yang dikembangkan oleh Thomas dan Harden (2008), yang melibatkan tiga tahap: (1) pengodean baris per baris temuan setiap artikel; (2) pembentukan tema deskriptif dari kode yang berulang; (3) dan pengembangan tema analitik yang menjawab pertanyaan penelitian. Dengan demikian sintesis tidak hanya disampaikan secara naratif tetapi menghasilkan struktur konseptual yang lebih komprehensif, dari awal hingga menuju tema akhir.

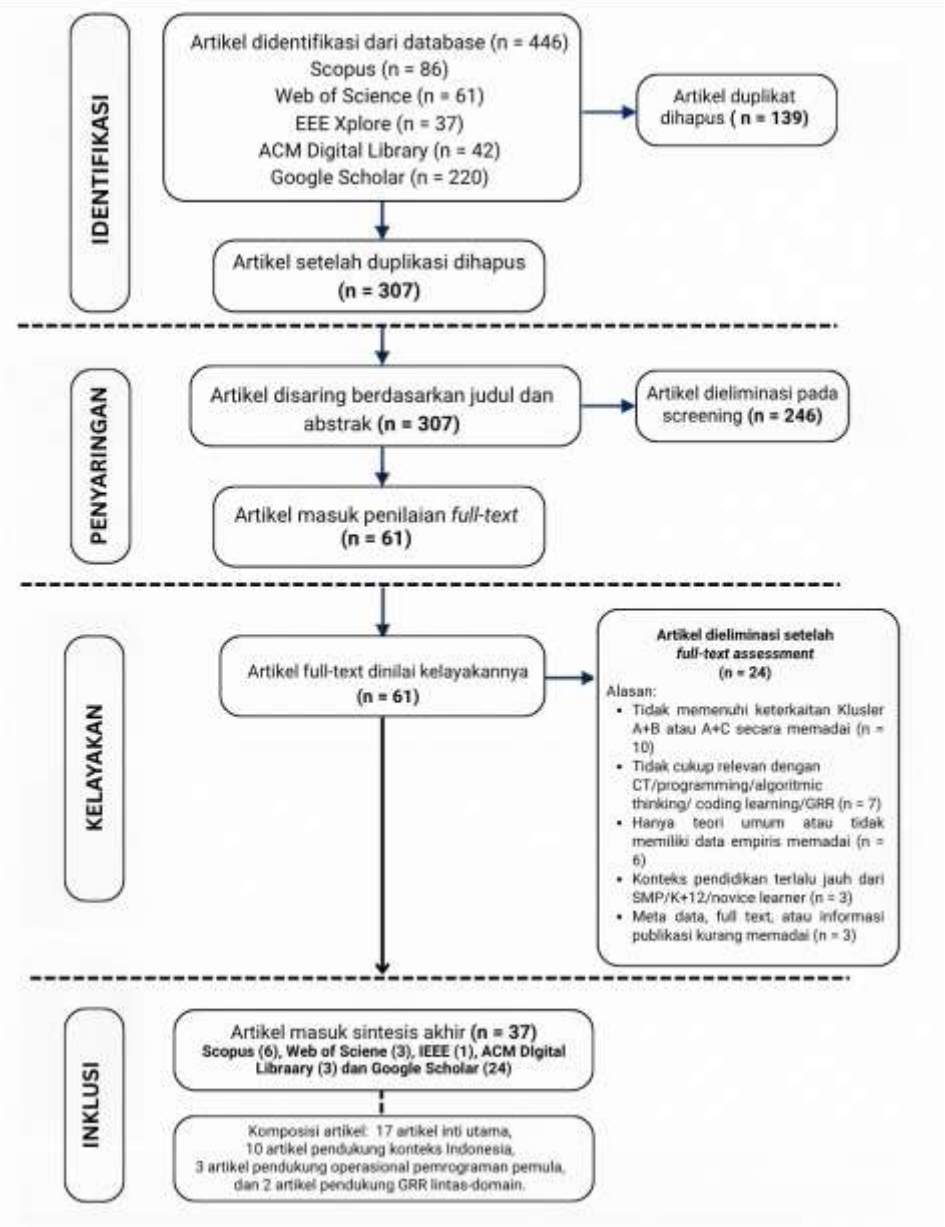
DOI: https://doi.org/10.17509/e.v25i2.99888
p-ISSN 0852-1190 e-ISSN 2502-0781

Tabel 3. Struktur Ekstraksi Data

Aspek Ekstraksi	Informasi Yang Dicatat	Fungsi Dalam Sintesis
Identitas Artikel	Penulis, tahun, judul, jurnal/prosiding	Memastikan keterlacakan sumber.
Konteks Penelitian	Negara, jenjang pendidikan, karakteristik peserta.	Menilai kesesuaian temuan dengan kondisi pendidikan di Indonesia
Desain Penelitian	Quasi-experiment, action research, mixed-method, review, DBR, konseptual.	Menetapkan proses appraisal dan interpretasi data
Fokus Utama	GRR, scaffolding, CT, algorithmic thinking, programming education.	Menghubungkan artikel dengan RQ.
Temuan Utama	Efektivitas, mekanisme, indikator, hambatan, rekomendasi.	Menjadi bahan coding awal.
Keterbatasan	Batas desain, sampel, konteks, instrumen, durasi intervensi.	Mendukung analisis bias dan gap penelitian.

Proses seleksi didokumentasikan menggunakan PRISMA flow diagram untuk memberikan transparansi tentang jumlah artikel yang diidentifikasi, diskrining, dan dimasukkan dalam analisis final (Page et al., 2021).

DOI: https://doi.org/10.17509/e.v25i2.99888
p-ISSN 0852-1190 e-ISSN 2502-0781



Gambar 1. Prosedur PRISMA

HASIL

1. Gambaran Umum Artikel yang Dianalisis

Berdasarkan proses seleksi artikel menggunakan alur PRISMA 2020, menghasilkan 446 artikel yang berasal dari lima database utama, Scopus (86 artikel), Web of Science (61 artikel), IEEE Xplore (37 artikel), ACM Digital Library (42 artikel), dan Google Scholar (220 artikel). Setelah dilakukan proses penghapusan duplikasi, terdapat sebanyak 139 artikel dihapus dan tersisa 307 artikel untuk tahap penyaringan. Pada tahap penyaringan, artikel disaring berdasarkan judul, abstrak, dan kata kunci. Sebanyak 246 artikel dieliminasi karena tidak relevan dengan fokus GRR/*scaffolding*, *computational thinking*, *programming education, algorithmic thinking*, atau konteks SMP.

DOI: https://doi.org/10.17509/e.v25i2.99888
p-ISSN 0852-1190 e-ISSN 2502-0781

Selanjutnya, terdapat 61 artikel masuk tahap *full-text assessment*. Pada tahap ini, artikel dinilai berdasarkan kriteria inklusi, yaitu diterbitkan pada rentang 2015-2026, memiliki data empiris atau kontribusi konseptual yang kuat, relevan dengan prinsip GRR/scaffolding atau pendekatan pembelajaran bertahap, serta berkaitan dengan *computational thinking, programming, algorithmic thinking,* atau pembelajaran *problem solving*. Dari 61 artikel tersebut, 24 artikel dieliminasi karena tidak memiliki keterkaitan dnegan kluster A, B, C dan string utama, tidak cukup relevan dengan *CT/programming/algorithmic thinking*, hanya berupa teori umum yang konteks yang kurang memadai.

Dengan demikian, pada tahap sintesa akhir diperoleh 37 artikel, yang terdiri dari 17 artikel inti yang secara kuat menghubungkan GRR/*scaffolding* dengan CT atau pemrograman, 10 artikel pendukung dalam konteks Indonesia, 3 artikel pendukung operasional pemrograman pemula, dan 2 artikel pendukung GRR lintas pelajaran (domain). Pembagian klasifikasi ini dilakukan agar sintesis tidak memperlakukan seluruh artikel secara homogen, tetapi membedakan kontribusi artikel berdasarkan kedekatannya dengan fokus penelitian. Pendekatan ini memperkuat validitas sintesis karena artikel inti digunakan untuk menjawab fokus utama SLR, sedangkan artikel pendukung digunakan untuk memperkaya konteks, justifikasi, dan dasar pengembangan perangkat pembelajaran berbasis GRR untuk meningkatkan berpikir algoritmik siswa SMP.

Setelah diperoleh struktur ekstraksi data, maka dilakukan penyusunan matriks ekstraksi data artikel yang bertujuan sebagai dasar dalam proses sintesis dan analisis tematik. Penyusunan matriks ini menjadi penting karena *systematic literature review* melibatkan berbagai artikel dengan karakteristik penelitian yang beragam, sehingga diperlukan format yang terstruktur untuk mengorganisasi informasi secara konsisten. Melalui matriks ekstraksi data, setiap artikel dapat dipetakan berdasarkan identitas penelitian, metode yang digunakan, konteks pendidikan, fokus kajian, media pembelajaran, temuan utama, serta relevansinya terhadap penelitian ini.

Tabel 4. Matriks Ekstraksi Data Artikel Representatif yang Dianalisis dalam SLR

No	Identitas Artikel	Konteks Penelitian	Desain Penelitian	Fokus Utama	Temuan Utama	Keterbatasan
1	Tikva & Tambouris (2023). *The effect of scaffolding programming games and attitudes towards programming on the development of Computational Thinking*. *Education and Information Technologies*.	Yunani; middle school; siswa pemula dalam programming games	Kuasi-eksperimen	Efektivitas scaffolding dalam programming games terhadap CT	Scaffolding mendukung perkembangan CT siswa dan membantu mereka memahami aktivitas pemrograman secara lebih terstruktur.	Perlu perluasan sampel, durasi intervensi lebih panjang, dan pengujian pada konteks sekolah berbeda.
2	Xi et al. (2023). *The effects of a progressive scaffolding approach on middle school students' computational thinking skills and self-efficacy*. ACM Conference Proceedings.	Tiongkok; middle school; siswa dengan pengalaman awal CT/pemrograman	Eksperimen / kuasi-eksperimen	Progressive scaffolding, CT, self-efficacy	Progressive scaffolding meningkatkan CT dan self-efficacy siswa.	Perlu uji longitudinal untuk melihat keberlanjutan efek scaffold fading.
3	Liu et al. (2025). *Teacher Decisions and Perspectives in Scratch TIPP&SEE Implementation*. SIGCSE TS.	AS; upper elementary-middle school; guru dan siswa pengguna Scratch	Mixed-method	Mekanisme keputusan guru dalam implementasi scaffold Scratch	Guru menyesuaikan scaffold berdasarkan usia, kesiapan siswa,	Perlu pengukuran langsung terhadap peningkatan CT siswa, bukan hanya perspektif guru.

DOI: https://doi.org/10.17509/e.v25i2.99888
p-ISSN 0852-1190 e-ISSN 2502-0781

No	Identitas Artikel	Konteks Penelitian	Desain Penelitian	Fokus Utama	Temuan Utama	Keterbatasan
					dan kebutuhan kelas.	
4	Jocius et al. (2021). *Leveraging virtual professional development to build computational thinking literacies in English language arts classrooms*. *Contemporary Issues in Technology and Teacher Education*.	AS; middle-high school; guru ELA	Kualitatif / professional development	GRR minilesson, I do-We do-You do, CT literacies	Guru menggunakan pola I do-We do-You do untuk mengintegrasikan CT dalam literasi.	Fokus pada guru; perlu bukti dampak langsung terhadap kemampuan siswa.
5	Jocius et al. (2024). *Computational thinking infusion as transformative teaching*. *Computer Science Education*.	AS; middle/secondary teachers; guru lintas mata pelajaran	Kualitatif	Mekanisme guided practice dalam infusi CT	Guided practice membantu guru mengintegrasikan CT lintas disiplin.	Perlu uji kuantitatif atau mixed-method untuk mengukur capaian CT siswa.
6	Vasconcelos & Kim (2020). *Coding in scientific modeling lessons (CS-ModeL)*. *Educational Technology Research and Development*.	AS; K-12 teacher education; guru/siswa dalam scientific modeling	Design and development study	Desain scaffolding coding berbasis Scratch	CS-ModeL menawarkan model scaffold untuk coding simulasi sains.	Masih dominan desain model; perlu uji efektivitas lebih luas di kelas nyata.
7	Gao et al. (2025). *Effect of mind mapping-based scaffolding on elementary students' computational thinking in block-based programming*. *Journal of Educational Computing Research*.	Tiongkok; elementary/K-12; siswa pemula block-based programming	Eksperimental / kuasi-eksperimental	Efektivitas mind mapping scaffolding terhadap CT	Mind mapping membantu decomposition, abstraction, dan algorithm design.	Konteks elementary; perlu adaptasi dan validasi pada SMP.
8	Zhang & Rutherford (2024). *Scaffolding expertise: Evaluating scaffolds for block-based coding among experts and novices*. ACM Proceedings.	AS; novice dan expert learners	Eksperimen	Mekanisme scaffold pada pemula dan ahli	Efektivitas scaffold berbeda antara novice dan expert.	Perlu model scaffold adaptif berdasarkan tingkat kesiapan siswa.
9	Sarmento & Reis (2026). *Code Comments as a Pedagogical Scaffold in Digitally Supported Introductory Programming*. *Digital*.	Portugal; vocational secondary; siswa programming awal	Exploratory mixed-method	Code comments sebagai pedagogical scaffold	Komentar kode membantu pemahaman struktur program dan self-regulation.	Perlu sampel lebih besar dan instrumen CT yang lebih tervalidasi.
10	Angeli (2022). *The effects of scaffolded programming scripts on pre-service teachers' computational thinking*. *International Journal of Child-Computer Interaction*.	Siprus; preservice teachers; pemula programming robotics	Eksperimen / intervensi	Algorithmic thinking, debugging, scaffolded scripts	Scaffolded scripts mendukung algorithmic thinking melalui robot programming.	Peserta bukan siswa SMP; perlu replikasi pada siswa K-12/SMP.
11	Greenwald & Krakowski (2021). *Integrating computer science in science classrooms*. NARST/NSF-PAR.	AS; middle school science; siswa dan guru sains	Laporan empiris konferensi	GRR, CT, integrasi CS dalam sains	Task-embedded support dan gradual release membantu siswa menuju praktik CT mandiri.	Publikasi konferensi; perlu artikel jurnal lengkap dengan detail instrumen.
12	Chuang et al. (2022). *Implementation of the gradual release of responsibility informed curriculum and pedagogy for teaching programming*. *Journal of*	Taiwan; perguruan tinggi; mahasiswa non-CS/beginner programmer	Action research; pre-post	GRR eksplisit dalam programming education	GRR meningkatkan self-efficacy dan engagement programming.	Konteks perguruan tinggi, bukan SMP; perlu adaptasi usia dan materi.

DOI: https://doi.org/10.17509/e.v25i2.99888
p-ISSN 0852-1190 e-ISSN 2502-0781

No	Identitas Artikel	Konteks Penelitian	Desain Penelitian	Fokus Utama	Temuan Utama	Keterbatasan
	Hospitality, Leisure, Sport & Tourism Education.					
13	Noordin (2025). *Computational thinking through scaffolded game development activities*. *European Journal of Educational Research*.	Malaysia; usia 10-15 tahun; siswa K-12/secondary	Kuasi-eksperimen / mixed evidence	Scaffolded game development, graphical programming	Scaffolded game development meningkatkan CT, engagement, dan menurunkan programming anxiety.	Perlu verifikasi metadata dan analisis lebih spesifik per jenjang.
14	Marchelin, Hamidah, & Resti (2022). *Efektifitas Metode Scaffolding dalam Meningkatkan Computational Thinking Siswa SMP pada Materi Perbandingan*. *Jurnal Pengembangan Pembelajaran Matematika*.	Indonesia; SMP; siswa pada materi perbandingan	Eksperimen / kuasi-eksperimen	Efektivitas scaffolding terhadap CT siswa SMP	Scaffolding meningkatkan CT siswa SMP.	Konteks matematika, bukan Informatika; perlu replikasi pada programming/algoritma.
15	Supiarmo, Mardhiyatirrahmah, & Turmudi (2021). *Pemberian scaffolding untuk memperbaiki proses berpikir komputasional siswa*. *Jurnal Cendekia*.	Indonesia; sekolah menengah; siswa pemecahan masalah matematika	Deskriptif kualitatif	Mekanisme scaffolding dalam proses CT	Scaffolding membantu siswa mencapai abstraksi dan berpikir algoritmik.	Sampel terbatas; perlu desain eksperimen dan instrumen CT terstandar.
16	Ningrum & Supeno (2024). *Pengaruh Model Pembelajaran Inkuiri Terbimbing terhadap Keterampilan Berpikir Komputasional pada Pembelajaran IPA Siswa SMP*. *SAP*.	Indonesia; SMP; siswa pembelajaran IPA	Kuasi-eksperimen	Guided inquiry, CT, IPA	Inkuiri terbimbing meningkatkan keterampilan berpikir komputasional siswa SMP.	Tidak eksplisit GRR; perlu pemetaan fase guided inquiry ke fase GRR.
17	Purnani, Mulhamah, & Afifurrahman (2024). *Scaffolding kemampuan berfikir komputasional siswa dalam pemecahan pada materi geometri*. *Journal of Math Tadris*.	Indonesia; sekolah menengah; siswa geometri	Kualitatif / deskriptif	Scaffolding, CT, problem solving geometri	Scaffolding membantu siswa dalam pemecahan masalah dan pengembangan CT.	Perlu validasi instrumen dan perluasan konteks ke Informatika.
18	Wibawa & Agustini (2026). *Pengaruh PBL Berbantuan Aplikasi Bebras terhadap Computational Thinking Informatika Siswa Kelas VII SMP*. *KARMAPATI*.	Indonesia; SMP kelas VII; siswa Informatika	Eksperimen / kuasi-eksperimen	CT, Bebras, Informatika SMP	PBL berbantuan Bebras berpengaruh terhadap CT siswa.	Tidak memuat GRR/scaffolding eksplisit; perlu integrasi dengan fase GRR.
19	Dewi, Juliyanto, & Rahayu (2021). *Pengaruh Pembelajaran IPA dengan Pendekatan CT Berbantuan Scratch terhadap Kemampuan Pemecahan Masalah*. *Indonesian Journal of Natural Science Education*.	Indonesia; SMP; siswa IPA	Eksperimen / kuasi-eksperimen	Scratch, CT, problem solving	Scratch membantu integrasi CT dan pemecahan masalah.	Lebih fokus problem solving; indikator algorithmic thinking perlu dipisahkan.
20	Rosita & Putra (2026). *Needs Analysis: Developing Scratch Media with Sumedang Local Wisdom to Enhance Students' Computational Thinking*. *Mosharafa*.	Indonesia; SMP; siswa kelas VIII	R&D tahap analisis / needs analysis	Scratch, CT, media lokal	Ditemukan kebutuhan pengembangan media Scratch kontekstual untuk CT.	Belum menguji efektivitas; perlu tahap desain, validasi, dan implementasi.

DOI: https://doi.org/10.17509/e.v25i2.99888
p-ISSN 0852-1190 e-ISSN 2502-0781

No	Identitas Artikel	Konteks Penelitian	Desain Penelitian	Fokus Utama	Temuan Utama	Keterbatasan
21	Monalisa (2023). *Analisis berpikir komputasional siswa SMP pada Kurikulum Merdeka mata pelajaran Informatika*. *DIAJAR*.	Indonesia; SMP; siswa Informatika Kurikulum Merdeka	Deskriptif	Indikator CT dalam Informatika SMP	CT siswa SMP masih perlu diperkuat dalam konteks Informatika.	Tidak intervensional; perlu desain pembelajaran untuk meningkatkan CT.
22	Rahma & Lidinillah (2025). *Petualangan Soal Tipe Bebras Unplugged untuk Mengasah Berpikir Komputasional Siswa*. *Jurnal Ilmiah Profesi Pendidikan*.	Indonesia; sekolah dasar/menengah awal; siswa matematika	R&D / pengembangan bahan ajar	Bebras unplugged, CT	Bebras unplugged dapat melatih CT sebelum coding.	Belum menguji transisi dari unplugged ke programming visual.
23	Budiyanto et al. (2025). *Computational thinking development through Code.org activities*. Jurnal/penerbit perlu verifikasi	Indonesia; SD-SMP; siswa pengguna Code.org	Observational study	Code.org, CT, visual coding	Code.org mendukung pengembangan CT bertahap.	Metadata perlu verifikasi; desain observasional membatasi klaim kausal.
24	Wahyudhi & Sa'ida (2022). *Analysis of junior high school students' computational thinking through Bebras Challenge*.	Indonesia; SMP; siswa peserta Bebras	Deskriptif	CT, Bebras, junior high school	CT siswa SMP masih rendah, terutama logika algoritmik.	Tidak menguji intervensi; perlu model pembelajaran lanjutan.
25	Fitriyah et al. (2024). *Computational thinking performance based on Bebras Challenge*. Jurnal/penerbit perlu verifikasi	Indonesia; SD, SMP, SMA; peserta Bebras	Deskriptif kuantitatif	CT performance, Bebras	Banyak siswa belum mencapai kompetensi CT kuat, terutama aspek algoritmik.	Perlu analisis mendalam per indikator dan jenjang SMP.
26	Hoffmeester & Ratumanan (2025). *Enhancing Computational Thinking through STEM Learning*. *Plusminus*	Indonesia; SMP; siswa belajar Pythagorean	R&D / intervensi pembelajaran	CT, STEM, budaya lokal	CT dapat dikembangkan melalui STEM berbasis budaya.	Tidak langsung programming; perlu dikaitkan dengan algorithm design.
27	Yusuf & Noor (2024). *Algorithmic thinking growth using block-based and text-based programming*.	Malaysia; SD-SMP / beginner learners	Eksperimen	Algorithmic thinking, block/text-based programming	Block-based programming mendukung pertumbuhan logika algoritmik.	Metadata perlu verifikasi; perlu detail instrumen algorithmic thinking.
28	Kim et al. (2018). *Debugging during block-based programming*. *Instructional Science*.	AS; preservice teachers / novice programmers	Studi empiris kualitatif	Debugging, block-based programming	Pemula kesulitan pada logic, syntax, control, dan debugging errors.	Peserta bukan siswa SMP; perlu adaptasi ke level K-12.
29	Moon, Cheon, & Kwon (2022). *Difficult Concepts and Practices of Computational Thinking Using Block-Based Programming*. *International Journal of Computer Science Education in Schools*.	AS/Korea-linked; novice learners	Kuantitatif + coding journals	Hambatan konsep CT, block-based programming	Abstraction, debugging, conditionals, data, dan operators sulit bagi pemula.	Konteks undergraduate; perlu pengujian pada SMP.
30	Xie et al. (2019). *A theory of instruction for introductory programming skills*. *Computer Science Education*.	AS; novice programmers	Exploratory mixed-method	Instruksi programming pemula	Instruksi eksplisit dan bertahap meningkatkan penyelesaian latihan dan menurunkan error.	Perlu adaptasi teori instruksi ke pembelajaran Informatika SMP.

DOI: https://doi.org/10.17509/e.v25i2.99888
p-ISSN 0852-1190 e-ISSN 2502-0781

No	Identitas Artikel	Konteks Penelitian	Desain Penelitian	Fokus Utama	Temuan Utama	Keterbatasan
31	Fuentes & Casinillo (2024). *Assessing the effect of the Gradual Release of Responsibility (GRR) model in teaching science*. *Asian Journal of Assessment in Teaching and Learning*.	Filipina; secondary/science class; siswa pada materi Newton's Laws of Motion	Kuasi-eksperimen	Efektivitas GRR lintas-domain	Kelompok GRR mencapai posttest jauh lebih tinggi ($p<0,05$); GRR efektif untuk materi kausalitas bertahap	Bukti empiris GRR pada materi abstrak; analog dengan algoritma-pemrograman
32	Cerveza & Lapinid (2024). *The effects of gradual release of assistance instruction on students' heuristics, confidence, and attitude toward independent problem-solving*.	Filipina; preservice teachers; pembelajaran problem solving matematika	Embedded multiple-case design + time-series	GRA, heuristik, confidence, problem solving	GRA meningkatkan penggunaan heuristik, kepercayaan diri, dan sikap problem solving.	SSRN/preprint; perlu verifikasi peer-review. Tidak langsung CT/programming.
33	Jacob et al. (2020). *Teaching computational thinking to multilingual students through inquiry-based learning*. IEEE RESPECT.	AS; K-12 / upper elementary; multilingual learners	Design-based research; mixed-method	CT, inquiry-based learning, equity	Inquiry terstruktur membantu pengembangan artefak komputasional dan identitas CS.	Tidak eksplisit GRR; perlu pemetaan inquiry terstruktur ke fase We Do/You Do.
34	Suters & Suters (2020). *Coding for the core: Computational thinking and middle grades mathematics*. *Contemporary Issues in Technology and Teacher Education*.	AS; middle grades; guru kelas 6-8	Evaluasi program / professional development	Coding, CT, matematika middle grades	PDmeningkatkan pemahaman guru tentang CT dan integrasi coding.	Fokus guru; dampak langsung terhadap siswa belum kuat.
35	Kaya et al. (2026). *Evaluation of the GRR Framework's Effectiveness on Oral Summarization Skills in Turkish Middle School Classrooms*.	Turki; middle school; siswa literasi bahasa	Mixed-method; quasi-experiment + observasi	GRR lintas-domain, kemandirian belajar	GRR meningkatkan keterampilan oral summarization dan kemandirian siswa.	Domain literasi, bukan CT/programming; metadata perlu verifikasi.
36	Webb et al. (2019). *Gradual Release of Responsibility: A 35-year review*.	AS / lintas-konteks; K-12	Review	Kerangka GRR, teori dan perkembangan model	GRR diposisikan sebagai model pembelajaran bertahap dari modeling menuju kemandirian.	Review umum; bukan empiris CT/programming. Digunakan sebagai landasan teoritis.
37	Wong et al. (2024). *Algorithmic thinking and debugging in visual programming*. Jurnal/penerbit perlu verifikasi	Hong Kong; K-12/elementary; siswa visual programming	Mixed-method / multi-kasus	Algorithmic thinking, debugging, visual programming	Debugging berkaitan dengan sequencing dan conditional reasoning.	Jenjang lebih rendah dari SMP; metadata dan instrumen perlu diverifikasi.

Sumber: Diolah dari hasil pencarian dan seleksi SLR (2026)

2. Kategori Kekuatan Bukti dalam Sintesis

Hasil sintesis diatas menunjukkan bahwa bukti efektivitas GRR dalam pembelajaran algoritma pemrograman siswa SMP dapat dibagi menjadi beberapa tingkat kekuatan bukti.

DOI: https://doi.org/10.17509/e.v25i2.99888
p-ISSN 0852-1190 e-ISSN 2502-0781

Tabel 5. Kategori Bukti dalam Sintesis

Kategori Bukti	Jumlah / Posisi	Fungsi dalam Sintesis	Artikel
Bukti langsung GRR dalam *programming education*	Terbatas	Menunjukkan bahwa GRR dapat diterapkan dalam pembelajaran pemrograman, tetapi belum spesifik pada siswa SMP dan lima indikator berpikir algoritmik.	Chuang et al. (2022)
Bukti *scaffolding*/GRR-konseptual dalam CT dan programming	Dominan dalam artikel inti	Menunjukkan bahwa pembelajaran bertahap, scaffolded programming, progressive scaffolding, dan guided practice mendukung CT, programming, dan algorithmic thinking.	Tikva & Tambouris; Xi et al.; Noordin; Gao et al.; Angeli; Zhang & Rutherford
Bukti konteks Indonesia	10 artikel	Menunjukkan kebutuhan lokal terhadap model pembelajaran yang mampu meningkatkan CT dan berpikir algoritmik siswa SMP.	Marchelin et al.; Supiarmo et al.; Ningrum & Supeno; Wibawa & Agustini; Fitriyah et al.; Wahyudhi & Sa'ida
Bukti operasional pemrograman pemula	3 artikel utama	Menjelaskan kesulitan novice programmer dalam memahami tracing, debugging, struktur kontrol, dan alur program.	Kim et al.; Moon et al.; Xie et al.
Bukti GRR lintas-domain	4 artikel	Menguatkan bahwa GRR efektif untuk pembelajaran abstrak, bertahap, dan problem solving.	Fuentes & Casinillo; Cerveza & Lapinid; Kaya et al.; Webb et al.
Bukti tambahan CT/inquiry/visual programming	3 artikel	Memperkaya konteks implementasi CT dan visual programming pada K–12/middle grades.	Jacob et al.; Suters & Suters; Wong et al.

Dari pemetaan tersebut tampak bahwa bukti langsung GRR pada pembelajaran algoritma pemprograman untuk siswa SMP masih terbatas terlebih yang mendukung peningkatan kemampuan algoritmik, tetapi bukti tidak langsung dari studi *scaffolding, progressive scaffolding, guided instruction, dan scaffolded programming* cukup kuat untuk mendukung pengembangan GRR dalam pembelajaran algoritma pemrograman.

PEMBAHASAN

1. Efektivitas GRR Berdasarkan Bukti Sintesis

Dukungan penggunaan GRR dalam pembelajaran algoritma pemrogramman dari hasil sintesis yang telah dilakukan menunjukkan bahwa tidak muncul sebagai bukti tunggal, tetapi sebagai bukti dari bebeapa kelompok penelitian. Penelitian Chuang et al. (2022) menunjukkan bahwa kurikulum dan pedagogi berbasis GRR dapat diterapkan dalam pembelajaran pemrograman dan berdampak positif terhadap *self-efficacy* pada objek mahasiswa *non-Computer Science*. Namun, bukti penelitian tersebut belum sepenuhnya dapat menjawab konteks pada objek siswa SMP dan belum spesifik mengukur kemampuan berpikir algoritmik melalui indikator *sequencing*, *conditional*, *looping*, *tracing, dan debugging*. Meskipun demikian, hasil sintesis memperlihatkan bahwa prinsip utama GRR banyak muncul dalam studi *scaffolding, progressive scaffolding, guided practice, dan scaffolded programming*. Fisher dan Frey (2008, 2014) menjelaskan GRR sebagai kerangka pembelajaran yang memindahkan tanggung jawab belajar secara bertahap dari guru kepada siswa, sedangkan Webb et al. (2019) menegaskan bahwa GRR merupakan bentuk *scaffolding* yang mendukung pembelajaran yang kompleks.

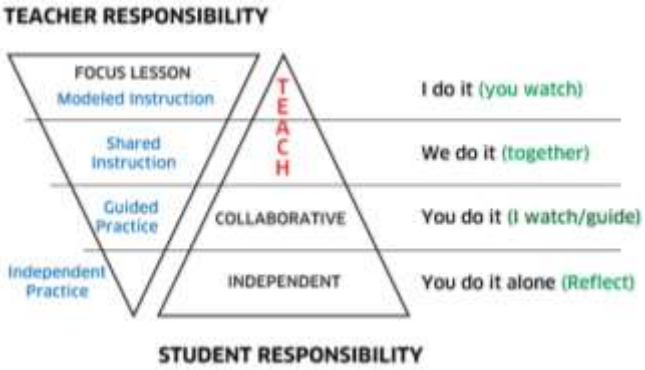
DOI: https://doi.org/10.17509/e.v25i2.99888
p-ISSN 0852-1190 e-ISSN 2502-0781

Dalam konteks *computational thinking* dan pemrgraman, Tikva dan Tambouris (2023), Angeli (2022), dan Noordin (2025) menunjukkan bahwa *scaffolding* dapat membantu siswa atau pemula dalam mengembangkan *computational thinking*, *algorithmic thinking*, dan keterampilan pemrograman. Bukti langsung GRR dalam pembelajaran algoritma pemrograman masih terbatas, tetapi bukti dukungan dari studi *scaffolding* dan bukti konseptual dari teori GRR memberikan dukungan yang cukup kuat. Oleh karena itu, GRR memiliki potensi kuat dan dasar teoretis yang memadai untuk dikembangkan serta diuji lebih lanjut dalam pembelajaran algoritma dan pemrograman.

2. Kesesuaian Karakteristik GRR dengan Pembelajaran Algoritma Pemrograman

Secara teoretis, GRR memiliki kesesuaian yang kuat dengan karakteristik pembelajaran algoritma dan pemrograman. Pembelajaran ini tidak hanya menuntut siswa memahami sintaks atau blok program, tetapi juga memahami alur eksekusi, hubungan antar instruksi, struktur percabangan, perulangan, prediksi output, serta strategi menemukan dan memperbaiki kesalahan. Para pemogram pemula sering mengalami kesulitan dalam membangun pemahaman yang konseptual secara stabil (Robins et al, 2003), selain itu kemampuan membaca dan menelusuri program juga merupakan keterampilan fundamental yang belum tentu dikuasai oleh para pemula ini (Lister et al, 2004). Hal-hal tersebut dijelaskan melalui perspektif *Cognitive Load Theory*, yang menyatakan bahwa pembelajaran dapat terhambat ketika beban kognitif melebihi kapasitas kerja memori siswa (Sweller, 1988).

Dalam pembelajaran algoritma pemrograman, beban tersebut muncul karena siswa harus memproses beberapa elemen secara bersamaan, seperti logika masalah, struktur kontrol, alur eksekusi, dan kemungkinan kesalahan program. Oleh karena itu, pembelajaran algoritma pemrograman membutuhkan model yang mampu memberi dukungan kognitif secara bertahap, dimana pembelajaran tidak langsung menempatkan siswa pada praktik mandiri penuh, tetapi memberi dukungan bertahap sampai siswa siap berpikir secara mandiri. Dalam konteks ini, GRR dapat dipahami sebagai bentuk operasional dari *scaffolding*. Prinsip pelepasan tanggung jawab secara bertahap, melalui fase I Do, We Do, You Do Together, dan You Do Alone**.**



Gambar 2. Model *The Gradual Release of Responsibility (*Fisher, D., & Frey, N, 2008)

3. Mekanisme Efektivitas GRR terhadap Berpikir Algoritmik

Berpikir algoritmik dioperasionalisasikan melalui lima indikator, yaitu *sequencing, conditional, looping, tracing, dan debugging (*Brennan et.al, 2012; Lister et al, 2004).

dan debugging. Guru tidak lagi mendominasi, tetapi berperan sebagai pemantau, pemberi umpan balik, dan fasilitator.

Pada fase You Do Alone, siswa menunjukkan kemandirian dalam menyusun, menelusuri, dan memperbaiki algoritma. Siswa dituntut menyusun instruksi, menentukan

DOI: https://doi.org/10.17509/e.v25i2.99888
p-ISSN 0852-1190 e-ISSN 2502-0781

conditional dan looping, tracing, serta memperbaiki kesalahan tanpa ketergantungan penuh pada guru. Dengan demikian, GRR tidak hanya berfungsi sebagai model pembelajaran umum, tetapi dapat dioperasionalisasikan sebagai strategi pembelajaran yang secara langsung mengarah pada pengembangan *sequencing, conditional, looping, tracing, dan debugging*.

Tabel 6. Penyesuaian Fase GRR terhadap Pengembangan Berpikir Algoritmik

Komponen GRR	Fokus Tanggung Jawab Belajar	Operasionalisasi dalam Pembelajaran Algoritma-Pemrograman	Indikator Berpikir Algoritmik
Focused Instruction I Do	Tanggung jawab belajar dominan pada guru. Guru memodelkan proses berpikir secara eksplisit.	Guru memberikan contoh penyusunan program dan melakukan *think-aloud* untuk menjelaskan alasan di balik urutan instruksi, hubungan antarblok, serta alur eksekusi program. Proses berpikir yang abstrak dibuat terlihat agar siswa memahami bagaimana algoritma dibangun dan dijalankan.	Sequencing dan tracing awal
Guided Instruction We Do	Tanggung jawab mulai dibagi antara guru dan siswa melalui bimbingan, pertanyaan penuntun, dan umpan balik.	Guru dan siswa menyelesaikan masalah bersama. Guru mengarahkan siswa memahami kapan suatu kondisi dijalankan, bagaimana percabangan bekerja, berapa kali perulangan terjadi, dan kapan perulangan berhenti. Siswa mulai memahami hubungan sebab-akibat dalam struktur algoritma, bukan sekadar mencoba blok program.	Conditional, looping, dan tracing
Collaborative Learning You Do Together	Tanggung jawab semakin berpindah kepada siswa melalui kerja kolaboratif, sementara guru berperan sebagai fasilitator.	Siswa bekerja dalam kelompok untuk menguji solusi, membaca alur program, membandingkan prediksi output, menemukan kesalahan, dan memperbaiki program. Guru tidak lagi mendominasi, tetapi memantau, memberi umpan balik, dan membantu jika diperlukan.	Tracing dan debugging
Independent Learning You Do Alone	Tanggung jawab belajar berada pada siswa secara mandiri.	Siswa menyelesaikan masalah algoritmik secara mandiri dengan menyusun instruksi, menentukan percabangan dan perulangan, menelusuri alur eksekusi, serta memperbaiki kesalahan program tanpa ketergantungan penuh pada guru.	Integrasi seluruh indikator: sequencing, conditional, looping, tracing, dan debugging

4. Peran Visual Programming sebagai Media Implementasi GRR.

Integrasi GRR dan *visual programming* dalam penelitian ini diposisikan sebagai kerangka pembelajaran bertahap untuk mengembangkan berpikir algoritmik siswa SMP. GRR menyediakan struktur pelepasan tanggung jawab belajar dari guru kepada siswa melalui fase I Do, We Do, You Do Together, dan You Do Alone (Fisher & Frey, 2008), sedangkan Code.org dan Scratch berperan sebagai media yang mendukung pelaksanaan setiap fase. Code.org lebih sesuai untuk fase awal yang masih terstruktur dan terbimbing, sedangkan Scratch lebih sesuai untuk fase kolaboratif dan mandiri karena memberi ruang eksplorasi, revisi, dan integrasi konsep komputasional dalam proyek yang lebih terbuka (Brennan & Resnick, 2012; Kale & Yuan, 2021; Zhang & Nouri, 2019).

Tabel. 7 Mapping Penggunaan aplikasi Code.org dan Scratch dalam Fase GRR

Fase GRR	Aplikasi	Kesesuaian Aplikasi dan Pembelajaran	Indikator
Focused Instruction (I Do)	Code.org	Code.org digunakan sebagai media demonstrasi karena aktivitasnya terstruktur, bertahap, dan cocok untuk pemodelan guru. Guru menunjukkan cara menyusun instruksi, membaca alur eksekusi, dan menjelaskan logika program secara eksplisit (Fisher & Frey, 2008; Kale & Yuan, 2021).	Sequencing, tracing awal

DOI: https://doi.org/10.17509/e.v25i2.99888
p-ISSN 0852-1190 e-ISSN 2502-0781

Guided Instruction (We Do)	Code.org	Code.org mendukung latihan terbimbing melalui tantangan/puzzle dan umpan balik langsung. Guru dan siswa menyelesaikan masalah bersama untuk memahami percabangan, perulangan, dan prediksi output (Fisher & Frey, 2008; Kale & Yuan, 2021; Sorva et al., 2013).	Conditional, looping, tracing
Collaborative Learning You Do Together	Scratch	Scratch digunakan untuk kerja kelompok karena memungkinkan siswa mengembangkan, menguji, memodifikasi, dan memperbaiki program secara kolaboratif. Aktivitas ini mendukung diskusi alur program dan debugging bersama (Brennan & Resnick, 2012; Zhang & Nouri, 2019).	Tracing, debugging
Independent Learning You Do Alone	Scratch	Scratch digunakan untuk proyek mandiri karena memberi ruang eksplorasi yang lebih terbuka. Siswa menyusun program, menggunakan percabangan dan perulangan, menelusuri alur eksekusi, serta memperbaiki kesalahan secara mandiri (Brennan & Resnick, 2012; Zhang & Nouri, 2019; Lister et al., 2004).	Sequencing, conditional, looping, tracing, debugging

Dengan pemetaan tersebut, Code.org berfungsi sebagai media untuk fase pembelajaran yang masih sangat terbimbing, sedangkan Scratch digunakan ketika tanggung jawab belajar mulai berpindah kepada siswa. Pola ini memperkuat prinsip GRR karena penggunaan aplikasi disesuaikan dengan tingkat kemandirian siswa dalam memahami, menelusuri, dan memperbaiki algoritma.

5. Kelebihan dan Keterbatasan Studi-studi Sebelumnya

Artikel-artikel yang disintesis menunjukkan bahwa penelitian sebelumnya telah memberikan dasar yang penting bagi pengembangan pembelajaran algoritma pemrograman berbasis bantuan bertahap. Berdasarkan kata kunci dan hasil akhir sintesis, kekuatan utama studi-studi tersebut terletak pada konsistensi temuan bahwa *scaffolding, guided practice, progressive scaffolding, visual programming,* dan aktivitas pemrograman bertahap dapat membantu siswa atau pemula memahami konsep komputasional yang abstrak. Studi Chuang et al. (2022), Tikva dan Tambouris (2023), Xi et al. (2023), Angeli (2022), Gao et al. (2025), dan Noordin (2025) memperlihatkan bahwa dukungan yang diberikan secara eksplisit dan bertahap berperan dalam meningkatkan *computational thinking*, keterampilan pemrograman, *self efficacy*, maupun keterlibatan belajar. Selain itu, studi terkait *visual programming* seperti Budiyanto et al. (2025), Yusuf dan Noor (2024), dan Wong et al. (2024) memperkuat bahwa lingkungan pemrograman visual dapat membantu siswa membangun pemahaman terhadap alur program, struktur logika, dan proses debugging.

Meskipun demikian, keterbatasan utama dari studi-studi tersebut adalah masih terbatasnya penelitian yang secara langsung menguji GRR dalam pembelajaran algoritma-pemrograman siswa SMP. Chuang et al. (2022) memang secara eksplisit menerapkan GRR dalam pembelajaran programming, tetapi konteksnya belum spesifik pada siswa SMP dan pengembangan berpikir algoritmik. Sebagian besar artikel lain lebih banyak menggunakan istilah *scaffolding, guided instruction, progressive scaffolding, atau scaffolded programming*, namun belum memetakan mekanisme pembelajaran secara eksplisit ke dalam fase I Do, We Do, You Do Together, dan You Do Alone yang terdapat pada GRR. Dengan demikian, dukungan terhadap GRR dalam sintesis ini lebih banyak bersifat proksimal, yaitu berasal dari prinsip-prinsip pembelajaran bertahap yang sejalan dengan GRR.

Keterbatasan lain terlihat pada aspek pengukuran. Beberapa studi mengukur *computational thinking* secara umum, *self-efficacy, engagement, problem solving*, atau hasil belajar, tetapi

DOI: https://doi.org/10.17509/e.v25i2.99888
p-ISSN 0852-1190 e-ISSN 2502-0781

belum secara terpadu menilai indikator berpikir algoritmik yang digunakan dalam penelitian ini, yaitu *sequencing, conditional, looping, tracing, dan debugging.* Selain itu, penggunaan aplikasi seperti Code.org dan Scratch belum selalu dikaitkan dengan desain instruksional yang menunjukkan perpindahan tanggung jawab belajar secara bertahap. Oleh karena itu, studi-studi sebelumnya memberikan dasar yang relevan, tetapi belum sepenuhnya menjawab kebutuhan pengembangan model GRR yang spesifik untuk pembelajaran Informatika SMP. Keterbatasan inilah yang menjadi ruang kontribusi penelitian ini, yaitu merumuskan GRR sebagai kerangka pembelajaran efektifyang lebih operasional untuk mengembangkan berpikir algoritmik siswa melalui fase pembelajaran yang terstruktur, bertahap, dan terukur.

6. *Research Gaps* dan Agenda Penelitian Masa Depan

Dalam konteks pendidikan Indonesia, penerapan GRR dalam pembelajaran algoritmapemrograman relevan dengan posisi mata pelajaran Informatika dalam Kurikulum Merdeka, khususnya pada pengembangan berpikir komputasional dan algoritma pemrograman di jenjang SMP. Kemampuan berpikir komputasional siswa SMP pada mata pelajaran Informatika masih perlu dianalisis dan dikembangkan secara sistematis (Monalisa, 2023). Disamping itu implementasi Informatika di SMP dalam Kurikulum Merdeka membutuhkan kesiapan sekolah, guru, dan perangkat pembelajaran (Nabilah et al., 2022). Temuan ini memperkuat argumen bahwa pendekatan GRR dapat menjadi kerangka pedagogis yang sesuai karena menyediakan pembelajaran bertahap dari pemodelan guru menuju kemandirian siswa.

Selain itu, beberapa studi lokal menunjukkan bahwa tantangan utama pembelajaran Informatika tidak hanya terletak pada materi, tetapi juga pada kesiapan guru dalam mengajarkan berpikir komputasional dan algoritma pemrograman. Rachmawati dan Yulianto (2024), Ciptasari dan Meliana (2024), serta Hafidhoh dan Atmaja (2025) menekankan pentingnya pendampingan dan penguatan kompetensi guru Informatika SMP dalam memahami *computational thinking* dan algoritma pemrograman. Dalam konteks ini, GRR dapat direkomendasikan sebagai pendekatan yang membantu guru menyusun pembelajaran secara lebih operasional melalui fase I Do, We Do, You Do Together, dan You Do Alone.

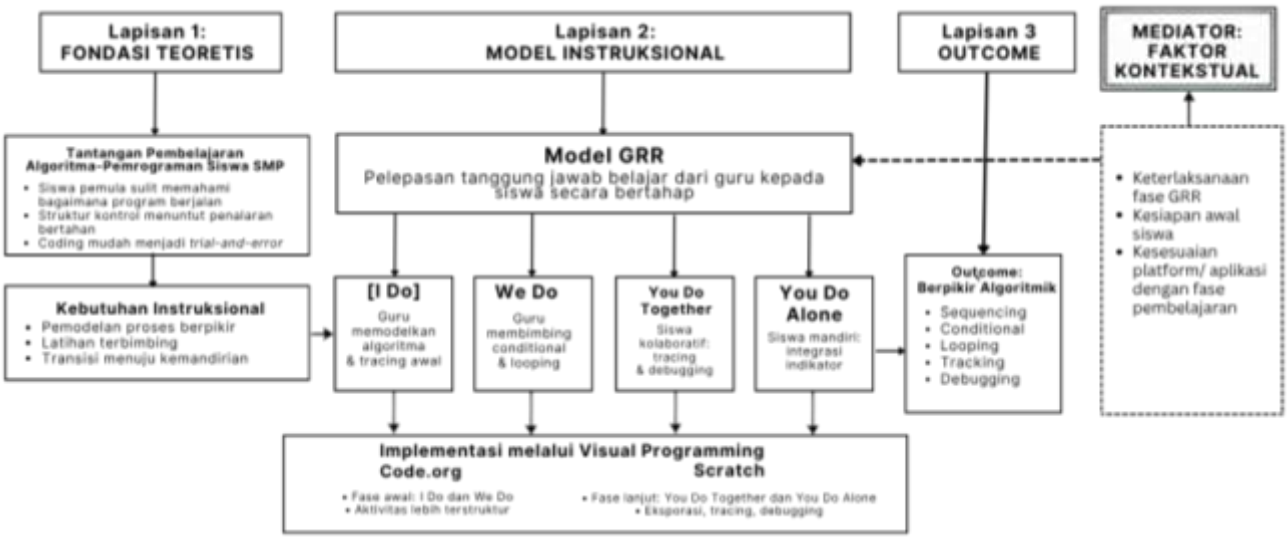
Keterkaitan ini juga semakin relevan dengan munculnya perhatian terhadap Koding dan Kecerdasan Artifisial (KKA). Studi lokal seperti Imawati et al. (2025), Irfandi et al. (2025), dan Fattah et al. (2025) menunjukkan bahwa pembelajaran koding dan kecerdasan artifisial membutuhkan kesiapan guru, penguatan *computational thinking*, serta desain pembelajaran yang tidak hanya berorientasi pada penggunaan teknologi. Oleh karena itu, rekomendasi penerapan GRR dalam konteks Indonesia adalah mengembangkan sintaks pembelajaran yang selaras dengan mata pelajaran Informatika dan KKA, menyusun rubrik berpikir algoritmik, serta menyiapkan panduan guru untuk memodelkan *sequencing, conditional, looping, tracing, dan debugging* secara eksplisit.

KERANGKA KONSEPTUAL

Berdasarkan hasil sintesis, kerangka konseptual penelitian ini disusun untuk menjelaskan hubungan antara karakteristik pembelajaran algoritma pemrograman, kebutuhan pembelajaran bertahap, model GRR, media *visual programming*, dan *outcome* berpikir algoritmik. Pembelajaran algoritma pemrograman pada siswa SMP dipandang sebagai proses kognitif yang kompleks karena siswa tidak hanya perlu menyusun instruksi, tetapi juga memahami alur eksekusi, percabangan, perulangan, tracing, dan debugging. Oleh karena itu, pembelajaran membutuhkan dukungan instruksional yang eksplisit, bertahap, dan terstruktur.

DOI: https://doi.org/10.17509/e.v25i2.99888
p-ISSN 0852-1190 e-ISSN 2502-0781

Dalam kerangka ini, GRR diposisikan sebagai model utama yang mengatur perpindahan tanggung jawab belajar dari guru kepada siswa melalui fase I Do, We Do, You Do Together, dan You Do Alone. Sedangkan *Visual programming*, yaitu Code.org dan Scratch, tidak diposisikan sebagai fondasi teori utama, tetapi sebagai media implementasi yang mendukung keterlaksanaan fase GRR. Code.org digunakan untuk fase awal yang lebih terbimbing, sedangkan Scratch digunakan untuk fase kolaboratif dan mandiri dengan *Outcome* yang dituju adalah kemampuan berpikir algoritmik siswa SMP, yang mencakup *sequencing, conditional, looping, tracing, dan debugging.*



Gambar 3. Kerangka Konseptual

KESIMPULAN

Systematic Literature Review (SLR) ini menunjukkan bahwa model *Gradual Release of Responsibility* (GRR) memiliki potensi baik secara konseptual maupun pedagogis untuk mendukung pembelajaran algoritma pemrograman di jenjang SMP. Temuan sintesis memperlihatkan bahwa pembelajaran algoritma pemrograman tidak cukup hanya bertumpu pada penggunaan aplikasi atau aktivitas *coding* semata, tetapi membutuhkan desain instruksional yang mampu membimbing siswa memahami alur eksekusi, dan struktur kontrol secara bertahap.

Hasil kajian juga menegaskan bahwa kemampuan berpikir algoritmik dapat dipahami sebagai konstruk yang terukur melalui indikator *sequencing, conditional, looping, tracing, dan debugging*. Kelima indikator tersebut memerlukan pembelajaran yang terstruktur, dan memberi ruang bagi siswa untuk bergerak dari pemahaman yang dimodelkan guru menuju kemandirian dalam menyelesaikan masalah algoritmik. Dalam konteks ini, GRR relevan digunakan karena menyediakan tahapan I Do, We Do, You Do Together, dan You Do Alone sebagai mekanisme pelepasan tanggung jawab belajar secara bertahap.

Meskipun demikian, bukti langsung mengenai penerapan GRR dalam pembelajaran algoritma pemrograman siswa SMP masih terbatas. Sebagian besar dukungan yang ditemukan berasal dari studi tentang *scaffolding, guided practice, progressive scaffolding, visual programming*, serta penerapan GRR pada domain pembelajaran lain. Oleh karena itu, kontribusi utama artikel ini adalah merumuskan dasar konseptual bahwa GRR dapat dikembangkan sebagai kerangka pembelajaran yang relevan untuk mendukung kemampuan berpikir algoritmik siswa SMP, terutama jika diintegrasikan dengan media *visual programming* seperti Code.org dan Scratch.

DOI: https://doi.org/10.17509/e.v25i2.99888
p-ISSN 0852-1190 e-ISSN 2502-0781

Dengan demikian, penelitian ini memperkuat urgensi pengembangan pembelajaran algoritma-pemrograman yang tidak hanya berorientasi pada produk program, tetapi juga pada proses berpikir siswa. Kajian lanjutan diperlukan untuk menguji penerapan GRR secara empiris dalam konteks pembelajaran Informatika SMP dan menilai dampaknya terhadap indikator berpikir algoritmik secara lebih terukur.

DAFTAR PUSTAKA

Angeli, C. (2022). The effects of scaffolded programming scripts on pre-service teachers' computational thinking: Developing algorithmic thinking through programming robots. *International Journal of Child-Computer Interaction, 31*, Article 100329. https://doi.org/10.1016/j.ijcci.2021.100329

Bakara, A. (2015). Perkembangan kognitif siswa berdasarkan teori Piaget dalam operasi logis di sekolah menengah pertama. *Jurnal Pendidikan dan Pembelajaran Khatulistiwa, 4*(12). https://doi.org/10.26418/jppk.v4i12.12653

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the Annual Meeting of the American Educational Research Association*, 1–25.

Budiyanto, C. W., Fenyvesi, K., Maharani, Y. I., Yuana, R. A., Nashiroh, P. K., & Latifah, R. (2025). Investigating computational thinking in K–12 visual programming activities on Code.org: A Brennan-Resnick framework approach. *Elinvo: Electronics, Informatics, and Vocational Education, 10*(1), 54–62. https://doi.org/10.21831/elinvo.v10i1

Cerveza, R. L., & Lapinid, M. R. C. (2024). The effects of gradual release of assistance instruction on students' heuristics, confidence, and attitude toward independent problem-solving. *Journal on Mathematics Education, 15*(3), 771–792. https://doi.org/10.22342/jme.v15i3.pp771-792

Chuang, K. L., Kee, Y. H., & Chen, H. H. (2022). Implementation of the gradual release of responsibility informed curriculum and pedagogy for teaching programming. *Journal of Hospitality, Leisure, Sport & Tourism Education, 30*, Article 100367. https://doi.org/10.1016/j.jhlste.2021.100367

Dewi, N. R., Juliyanto, E., & Rahayu, R. (2021). Pengaruh pembelajaran IPA dengan pendekatan computational thinking berbantuan Scratch terhadap kemampuan pemecahan masalah. *Indonesian Journal of Natural Science Education*.

Fisher, D., & Frey, N. (2008). *Better learning through structured teaching: A framework for the gradual release of responsibility*. ASCD.

Fisher, D., & Frey, N. (2014). *Better learning through structured teaching: A framework for the gradual release of responsibility* (2nd ed.). ASCD.

Fitriyah, Y., Wahyudin, W., Nurhayati, H., & Febrianti, T. S. (2024). Indonesian students' computational thinking performance based on level and gender. *International Journal of Pedagogy and Teacher Education, 8*(1), 50–67. https://doi.org/10.20961/ijpte.v8i1.89464

Fuentes, A. G. P., & Casinillo, L. F. (2024). Assessing the effect of the Gradual Release of Responsibility (GRR) model in teaching science. *Asian Journal of Assessment in Teaching and Learning, 14*(1), 15–24. https://doi.org/10.37134/ajatel.vol14.1.2.2024

Gao, Y., Li, Y., & Zhang, X. (2025). Effect of mind mapping-based scaffolding on elementary students' computational thinking in block-based programming. *Journal of Educational Computing Research*.

DOI: https://doi.org/10.17509/e.v25i2.99888
p-ISSN 0852-1190 e-ISSN 2502-0781

Greenwald, E., & Krakowski, A. (2021). Integrating computer science in science classrooms. *Proceedings of the National Association for Research in Science Teaching*.

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

Herlambang, A. D., Ramadana, M. R., Wijoyo, S. H., & Phadung, M. (2024). Students' cognitive load on computer programming instructional process. *Elinvo: Electronics, Informatics, and Vocational Education, 9*(2), 309–320. https://doi.org/10.21831/elinvo.v9i2.57882

Hoffmeester, D., & Ratumanan, T. G. (2025). Enhancing computational thinking through STEM learning. *Plusminus*.

Jacob, S. R., Warschauer, M., & Yu, C. (2020). Teaching computational thinking to multilingual students through inquiry-based learning. *Proceedings of the IEEE Research on Equity and Sustained Participation in Engineering, Computing, and Technology*.

Jocius, R., O'Byrne, W. I., Albert, J., Joshi, D., Blanton, M., & Andrews, A. (2021). Leveraging virtual professional development to build computational thinking literacies in English language arts classrooms. *Contemporary Issues in Technology and Teacher Education*.

Jocius, R., Joshi, D., Albert, J., & O'Byrne, W. I. (2024). Computational thinking infusion as transformative teaching. *Computer Science Education*.

Kale, U., & Yuan, J. (2021). Still a new kid on the block? Computational thinking as problem solving in Code.org. *Journal of Educational Computing Research*. https://doi.org/10.1177/0735633120972050

Kaya, K., Kurudayıoğlu, M., Yıldırım, K., Galeza, A., & Rasinski, T. (2026). Evaluation of the GRR framework's effectiveness on oral summarization skills in Turkish middle school classrooms. *Evaluation and Program Planning, 114*, Article 102720. https://doi.org/10.1016/j.evalprogplan.2025.102720

Kemendikdasmen. (2025). *Permendikdasmen Nomor 13 Tahun 2025 tentang perubahan atas Permendikbudristek Nomor 12 Tahun 2024*. Kementerian Pendidikan Dasar dan Menengah Republik Indonesia.

Kim, B., Yuan, J., Vasconcelos, L., Shin, M., & Hill, R. B. (2018). Debugging during block-based programming. *Instructional Science*.

Kitchenham, B., & Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering* (EBSE-2007-01). Keele University.

Lister, R., Adams, E. S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., McCartney, R., Moström, J. E., Sanders, K., Seppälä, O., Simon, B., & Thomas, L. (2004). A multi-national study of reading and tracing skills in novice programmers. *ACM SIGCSE Bulletin, 36*(4), 119–150. https://doi.org/10.1145/1041624.1041673

Lister, R., Fidge, C., & Teague, D. (2009). Further evidence of a relationship between explaining, tracing and writing skills in introductory programming. *ACM SIGCSE Bulletin, 41*(3), 161–165. https://doi.org/10.1145/1595496.1562930

Liu, J., Franklin, D., & Weintrop, D. (2025). Teacher decisions and perspectives in Scratch TIPP&SEE implementation. *Proceedings of the ACM Technical Symposium on Computer Science Education*.

Loksa, D., Ko, A. J., Jernigan, W., Oleson, A., Mendez, C. J., & Burnett, M. M. (2016). The role of programming environment design in cultivating computational thinking. *ACM Transactions on Computing Education, 16*(4), 1–25. https://doi.org/10.1145/2958905

DOI: https://doi.org/10.17509/e.v25i2.99888
p-ISSN 0852-1190 e-ISSN 2502-0781

Marchelin, Hamidah, & Resti. (2022). Efektifitas metode scaffolding dalam meningkatkan computational thinking siswa SMP pada materi perbandingan. *Jurnal Pengembangan Pembelajaran Matematika*.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review, 63*(2), 81–97. https://doi.org/10.1037/h0043158

Monalisa, M. (2023). Analisis berpikir komputasional siswa SMP pada Kurikulum Merdeka mata pelajaran Informatika. *DIAJAR: Jurnal Pendidikan dan Pembelajaran*.

Moon, J., Cheon, J., & Kwon, K. (2022). Difficult concepts and practices of computational thinking using block-based programming. *International Journal of Computer Science Education in Schools*.

Nabilah, B., Zakir, S., & Murtiyastuti, E. (2022). Analisis penerapan mata pelajaran Informatika dalam implementasi Kurikulum Merdeka tingkat SMP. *PIJAR: Jurnal Pendidikan dan Pengajaran*.

Ningrum, A. P., & Supeno. (2024). Pengaruh model pembelajaran inkuiri terbimbing terhadap keterampilan berpikir komputasional pada pembelajaran IPA siswa SMP. *SAP: Susunan Artikel Pendidikan*.

Noordin, N. H. (2025). Computational thinking through scaffolded game development activities: A study with graphical programming. *European Journal of Educational Research, 14*(4), 1137–1149. https://doi.org/10.12973/eu-jer.14.4.1137

O'Donnell, C. L. (2008). Defining, conceptualizing, and measuring fidelity of implementation and its relationship to outcomes in K–12 curriculum intervention research. *Review of Educational Research, 78*(1), 33–84. https://doi.org/10.3102/0034654307313793

OECD. (2023). *PISA 2022 results: The state of learning and equity in education* (Vol. I). OECD Publishing. https://doi.org/10.1787/53f23881-en

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*, Article n71. https://doi.org/10.1136/bmj.n71

Pearson, P. D., & Gallagher, M. C. (1983). The instruction of reading comprehension. *Contemporary Educational Psychology, 8*(3), 317–344. https://doi.org/10.1016/0361-476X(83)90019-X

Petticrew, M., & Roberts, H. (2006). *Systematic reviews in the social sciences: A practical guide*. Blackwell.

Purnani, Mulhamah, & Afifurrahman. (2024). Scaffolding kemampuan berpikir komputasional siswa dalam pemecahan pada materi geometri. *Journal of Math Tadris*.

Rahma, A., & Lidinillah, D. A. M. (2025). Petualangan soal tipe Bebras unplugged untuk mengasah berpikir komputasional siswa. *Jurnal Ilmiah Profesi Pendidikan*.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education, 13*(2), 137–172. https://doi.org/10.1076/csed.13.2.137.14200

Rosita, R., & Putra, Z. H. (2026). Needs analysis: Developing Scratch media with Sumedang local wisdom to enhance students' computational thinking. *Mosharafa: Jurnal Pendidikan Matematika*.

DOI: https://doi.org/10.17509/e.v25i2.99888
p-ISSN 0852-1190 e-ISSN 2502-0781

Sarmento, J., & Reis, A. (2026). Code comments as a pedagogical scaffold in digitally supported introductory programming. *Digital*.

Sentance, S., & Waite, J. (2017). PRIMM: Exploring pedagogical approaches for teaching text-based programming in school. *Proceedings of the 12th Workshop on Primary and Secondary Computing Education*, 81–88. https://doi.org/10.1145/3137069.3137088

Sorva, J., Karavirta, V., & Malmi, L. (2013). A review of generic program visualization systems for introductory programming education. *ACM Transactions on Computing Education, 13*(4), 1–64. https://doi.org/10.1145/2490822

Stevens, E. Y., Wilson, N. S., Baumann, J., Adams, B., Dussling, T. M., Smetana, L., & Bean-Folkes, J. (2025). Exploring asynchronous implementation of the GRR framework to support graduate students' metacognition. *Education Sciences, 15*(8), Article 1007. https://doi.org/10.3390/educsci15081007

Sukirman, Pramudita, D. A., Afiyanto, A., & Utaminingsih. (2022). Block-based visual programming as a tool for learning programming concepts for novices. *International Journal of Information and Education Technology, 12*(5), 365–371. https://doi.org/10.18178/ijiet.2022.12.5.1628

Supiarmo, Mardhiyatirrahmah, & Turmudi. (2021). Pemberian scaffolding untuk memperbaiki proses berpikir komputasional siswa. *Jurnal Cendekia*.

Suters, L., & Suters, H. (2020). Coding for the core: Computational thinking and middle grades mathematics. *Contemporary Issues in Technology and Teacher Education*.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science, 12*(2), 257–285. https://doi.org/10.1207/s15516709cog1202_4

Sweller, J., Ayres, P., & Kalyuga, S. (2011). *Cognitive load theory*. Springer. https://doi.org/10.1007/978-1-4419-8126-4

Thomas, J., & Harden, A. (2008). Methods for the thematic synthesis of qualitative research in systematic reviews. *BMC Medical Research Methodology, 8*, Article 45. https://doi.org/10.1186/1471-2288-8-45

Tikva, C., & Tambouris, E. (2023). The effect of scaffolding programming games and attitudes towards programming on the development of computational thinking. *Education and Information Technologies*.

Vasconcelos, L., & Kim, B. (2020). Coding in scientific modeling lessons: CS-ModeL. *Educational Technology Research and Development*.

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Harvard University Press.

Wahyudhi, S., & Sa'ida, I. A. (2022). Level of computational thinking skill among junior high school students in East Java using Bebras Challenge. *Media Bina Ilmiah, 17*(1), 131–140.

Webb, S., Massey, D., Goggans, M., & Flajole, K. (2019). Thirty-five years of the gradual release of responsibility: Scaffolding toward complex and responsive teaching. *The Reading Teacher, 73*(1), 75–83. https://doi.org/10.1002/trtr.1799

Wibawa, I. M. A., & Agustini, K. (2026). Pengaruh PBL berbantuan aplikasi Bebras terhadap computational thinking Informatika siswa kelas VII SMP. *KARMAPATI: Kumpulan Artikel Mahasiswa Pendidikan Teknik Informatika*.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35. https://doi.org/10.1145/1118178.1118215

Wong, G. K. W., Jian, S., & Cheung, H. Y. (2024). Engaging children in developing algorithmic thinking and debugging skills in primary schools: A mixed-methods multiple case study.

DOI: https://doi.org/10.17509/e.v25i2.99888
p-ISSN 0852-1190 e-ISSN 2502-0781

Education and Information Technologies, 29(13), 16205–16254. https://doi.org/10.1007/s10639-024-12448-x

Wood, D., Bruner, J. S., & Ross, G. (1976). The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry, 17*(2), 89–100. https://doi.org/10.1111/j.1469-7610.1976.tb00381.x

Xi, T., Zhang, M., & Li, Y. (2023). The effects of a progressive scaffolding approach on middle school students' computational thinking skills and self-efficacy. *ACM Conference Proceedings*.

Xie, B., Loksa, D., Nelson, G. L., Davidson, M. J., Dong, D., Kwik, H., Tan, A. H., Hwa, L., Li, M., & Ko, A. J. (2019). A theory of instruction for introductory programming skills. *Computer Science Education*.

Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM, 60*(4), 55–62. https://doi.org/10.1145/2994591

Yusuf, A., & Noor, N. M. (2024). Modeling students' algorithmic thinking growth trajectories in different programming environments. *Smart Learning Environments, 11*(1). https://doi.org/10.1186/s40561-024-00324-7

Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K–9. *Computers & Education, 141*, Article 103607. https://doi.org/10.1016/j.compedu.2019.103607

Zhang, Y., & Rutherford, T. (2024). Scaffolding expertise: Evaluating scaffolds for block-based coding among experts and novices. *ACM Conference Proceedings*.

DOI: https://doi.org/10.17509/e.v25i2.99888
p-ISSN 0852-1190 e-ISSN 2502-0781