Department of Digital Business
Journal of Artificial Intelligence and Digital Business (RIGGS)
Homepage: https://journal.ilmudata.co.id/index.php/RIGGS
Vol. 5 No. 2 (2026) pp: 20362- 20375
P-ISSN: 2963-9298, e-ISSN: 2963-914X

Implementasi Kurikulum Pemrograman Sederhana untuk Melatih Logika Siswa SD Birrul Walidain Muhammadiyah Tanon

Laily Rohmahwati Suprapto, Nani Anggorowati, Nia Oktafia Ningsih
Universitas Muhammadiyah Surakarta
q100250039@student.ums.ac.id

Abstrak

Integrasi pemrograman sederhana pada jenjang sekolah dasar merupakan inovasi kurikulum yang berpotensi mengembangkan kemampuan berpikir logis, sistematis, dan pemecahan masalah sejak dini. Penelitian ini bertujuan menganalisis implementasi kurikulum pemrograman sederhana berbasis Scratch serta kontribusinya terhadap kemampuan logika siswa SD Birrul Walidain Muhammadiyah Tanon. Penelitian menggunakan pendekatan kuantitatif dengan desain one-group pretest-posttest. Subjek penelitian berjumlah 25 siswa yang dipilih melalui sampling jenuh. Data dikumpulkan menggunakan tes kemampuan logika, angket respons siswa, dan lembar observasi pembelajaran. Analisis dilakukan dengan statistika deskriptif, uji normalitas Shapiro-Wilk, dan paired sample t-test berbantuan perangkat lunak komputer. Hasil penelitian menunjukkan bahwa nilai kemampuan logika siswa setelah pembelajaran lebih tinggi dibandingkan kondisi sebelum pembelajaran. Hasil uji inferensial yang dilaporkan juga menunjukkan perbedaan signifikan antara skor pretest dan posttest pada taraf signifikansi 0,05. Peningkatan tersebut berkaitan dengan aktivitas menyusun urutan instruksi, menggunakan pengulangan dan percabangan, menguji keluaran program, serta memperbaiki kesalahan melalui proses debugging. Scratch membantu siswa memvisualisasikan hubungan sebab-akibat dan mengurangi hambatan sintaks, sehingga perhatian dapat diarahkan pada logika penyelesaian masalah. Temuan ini menegaskan bahwa kurikulum pemrograman sederhana dapat digunakan sebagai sarana penguatan computational thinking di sekolah dasar. Namun, generalisasi hasil perlu dilakukan secara hati-hati karena penelitian hanya melibatkan satu kelompok dan jumlah subjek terbatas. Sekolah disarankan menerapkan pembelajaran secara bertahap, menyediakan pendampingan guru, dan melakukan evaluasi berbasis data secara berkelanjutan.

Kata kunci: Kurikulum Pemrograman Sederhana, Logika Siswa, Scratch, Statistika Pendidikan, Sekolah Dasar

1. Pendahuluan

Perkembangan teknologi digital telah membawa perubahan mendasar dalam berbagai aspek kehidupan, termasuk dalam bidang pendidikan. Transformasi digital menuntut dunia pendidikan untuk tidak hanya berfokus pada penguasaan pengetahuan faktual, tetapi juga pada pengembangan kemampuan berpikir tingkat tinggi yang relevan dengan tantangan abad ke-21. Kemampuan berpikir logis, sistematis, dan pemecahan masalah menjadi kompetensi penting yang perlu ditanamkan sejak jenjang pendidikan dasar sebagai fondasi bagi pembelajaran pada jenjang selanjutnya.

Sekolah dasar memiliki peran strategis dalam membentuk pola pikir dan karakter belajar siswa. Pada fase perkembangan kognitif ini, siswa berada pada tahap operasional konkret, sehingga pembelajaran yang bersifat eksploratif, visual, dan kontekstual menjadi sangat penting. Salah satu pendekatan pembelajaran yang dinilai sesuai dengan karakteristik tersebut adalah pembelajaran pemrograman sederhana atau coding for kids. Pembelajaran pemrograman pada tingkat sekolah dasar tidak bertujuan mencetak peserta didik menjadi programmer, melainkan melatih cara berpikir logis dan terstruktur melalui penyusunan langkah-langkah pemecahan masalah secara sistematis.

Wing (2006) menegaskan bahwa computational thinking merupakan keterampilan fundamental yang seharusnya menjadi bagian dari kemampuan analitis setiap individu di era digital. Computational thinking tidak identik dengan

kemampuan menulis kode program semata, tetapi mencakup kemampuan mendekomposisi masalah, mengenali pola, melakukan abstraksi, serta menyusun algoritma sebagai solusi. Keempat komponen tersebut memiliki keterkaitan erat dengan kemampuan berpikir logis dan sistematis yang menjadi tujuan pendidikan dasar.

Dalam konteks pembelajaran, pemrograman sederhana berbasis block-based coding seperti Scratch banyak direkomendasikan untuk siswa sekolah dasar. Scratch dirancang sebagai lingkungan pemrograman visual yang memungkinkan siswa menyusun perintah melalui blok-blok grafis yang saling terhubung. Pendekatan ini mengurangi hambatan sintaks yang sering menjadi kesulitan utama dalam pembelajaran pemrograman berbasis teks. Resnick et al. (2009) menyatakan bahwa Scratch menyediakan ruang bagi siswa untuk belajar melalui eksplorasi, eksperimen, dan refleksi sehingga proses belajar menjadi lebih bermakna dan kreatif.

Sejumlah penelitian menunjukkan bahwa pembelajaran pemrograman sederhana berkontribusi positif terhadap pengembangan kemampuan logika dan pemecahan masalah siswa. Grover dan Pea (2013) menempatkan computational thinking sebagai kompetensi penting dalam pendidikan K-12. Zhang dan Nouri (2019) juga menunjukkan bahwa pembelajaran melalui Scratch dapat mengembangkan berbagai konsep dan praktik computational thinking, walaupun efektivitasnya dipengaruhi oleh desain aktivitas dan metode penilaian.

SD Birrul Walidain Muhammadiyah Tanon sebagai sekolah dasar swasta berbasis nilai-nilai Islam telah berupaya mengintegrasikan pembelajaran pemrograman sederhana ke dalam kurikulum sekolah. Implementasi kurikulum ini dimaksudkan sebagai bentuk inovasi pembelajaran yang tidak hanya berorientasi pada penguasaan teknologi, tetapi juga pada pembentukan logika berpikir siswa. Pembelajaran pemrograman sederhana dilaksanakan secara terstruktur dengan memanfaatkan media Scratch dan pendampingan guru yang telah mendapatkan pelatihan dasar.

Meskipun demikian, implementasi kurikulum pemrograman sederhana perlu dikaji secara empiris untuk mengetahui efektivitasnya dalam mencapai tujuan pembelajaran, khususnya dalam melatih kemampuan logika siswa. Evaluasi pembelajaran yang dilakukan secara subjektif berpotensi menghasilkan kesimpulan yang bias dan kurang akurat. Oleh karena itu, diperlukan pendekatan evaluasi berbasis data yang objektif dan sistematis melalui pemanfaatan statistika pendidikan.

Statistika pendidikan merupakan cabang statistika terapan yang berfungsi untuk mengumpulkan, mengolah, menganalisis, dan menafsirkan data pendidikan secara ilmiah. Menurut Sudjana (2019), statistika pendidikan berperan penting dalam membantu peneliti dan praktisi pendidikan menarik kesimpulan yang valid serta mendukung pengambilan keputusan berbasis bukti empiris. Dalam penelitian ini, statistika pendidikan digunakan untuk menganalisis perubahan kemampuan logika siswa sebelum dan sesudah mengikuti pembelajaran pemrograman sederhana.

Selain itu, pemanfaatan teknologi komputer dalam analisis data memungkinkan proses pengolahan data dilakukan secara lebih akurat dan efisien. Penggunaan perangkat lunak statistik membantu peneliti menerapkan teknik analisis deskriptif dan inferensial secara tepat sesuai dengan karakteristik data penelitian. Dengan demikian, integrasi antara pembelajaran pemrograman, statistika pendidikan, dan teknologi komputer menjadi satu kesatuan yang saling mendukung dalam penelitian ini.

Berdasarkan uraian tersebut, penelitian ini difokuskan pada analisis implementasi kurikulum pemrograman sederhana di SD Birrul Walidain Muhammadiyah Tanon serta kontribusinya terhadap kemampuan logika siswa. Penelitian ini diharapkan dapat memberikan kontribusi teoretis dalam pengembangan kajian statistika pendidikan serta kontribusi praktis bagi sekolah dalam mengembangkan inovasi kurikulum berbasis teknologi.

2. Tinjauan Pustaka

2.1. Kurikulum Pemrograman Sederhana di Sekolah Dasar

Kurikulum pemrograman sederhana merupakan bagian dari pengembangan kurikulum berbasis teknologi yang bertujuan menyiapkan peserta didik menghadapi tantangan era digital. Pada jenjang sekolah dasar, kurikulum pemrograman tidak diarahkan pada penguasaan bahasa pemrograman secara teknis, melainkan pada pengembangan

DOI: https://doi.org/10.31004/riggs.v5i2.5748
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

pola pikir logis, sistematis, dan algoritmik. Pendekatan ini sejalan dengan paradigma pendidikan modern yang menekankan penguatan proses berpikir dibandingkan sekadar hasil belajar kognitif.

Menurut Grover dan Pea (2013), pemrograman di pendidikan dasar seharusnya diposisikan sebagai sarana pengembangan kemampuan berpikir, bukan semata-mata keterampilan vokasional. Kurikulum pemrograman sederhana dirancang untuk memperkenalkan konsep dasar seperti urutan instruksi (sequence), pengulangan (loop), percabangan (conditional), dan hubungan sebab-akibat. Konsep-konsep tersebut memiliki keterkaitan langsung dengan kemampuan berpikir logis yang menjadi tujuan utama pembelajaran.

Di Indonesia, integrasi pemrograman dalam kurikulum sekolah dasar masih bersifat kontekstual dan tergantung pada kebijakan sekolah. Sekolah swasta memiliki fleksibilitas lebih besar dalam mengembangkan kurikulum inovatif, termasuk memasukkan pembelajaran pemrograman sederhana sebagai muatan pengayaan. Dalam konteks ini, SD Birrul Walidain Muhammadiyah Tanon memanfaatkan fleksibilitas kurikulum sekolah untuk mengembangkan pembelajaran pemrograman sederhana yang disesuaikan dengan karakteristik siswa dan visi sekolah.

Pengembangan kurikulum pemrograman sederhana juga harus mempertimbangkan kesiapan guru, sarana prasarana, serta karakteristik peserta didik. Tanpa perencanaan yang matang, pembelajaran pemrograman berpotensi menjadi aktivitas teknis yang kurang bermakna. Oleh karena itu, evaluasi implementasi kurikulum pemrograman perlu dilakukan secara sistematis dan berbasis data agar efektivitasnya dapat diukur secara objektif.

2.2. Computational Thinking sebagai Dasar Berpikir Logis

Computational thinking merupakan konsep kunci dalam pembelajaran pemrograman. Wing (2006) mendefinisikan computational thinking sebagai cara berpikir dalam merumuskan masalah dan menyusun solusi yang dapat dijalankan secara sistematis oleh manusia maupun mesin. Konsep ini tidak terbatas pada bidang informatika, tetapi relevan untuk berbagai disiplin ilmu dan aktivitas kehidupan sehari-hari.

Secara konseptual, computational thinking terdiri atas empat komponen utama, yaitu dekomposisi masalah, pengenalan pola, abstraksi, dan penyusunan algoritma. Dekomposisi membantu siswa memecah masalah kompleks menjadi bagian-bagian kecil yang lebih mudah dikelola. Pengenalan pola melatih siswa mengidentifikasi kesamaan atau keteraturan dalam masalah. Abstraksi memungkinkan siswa memfokuskan perhatian pada aspek penting dan mengabaikan detail yang tidak relevan. Sementara itu, penyusunan algoritma melatih siswa menyusun langkah-langkah solusi secara logis dan runtut.

Keempat komponen tersebut memiliki keterkaitan erat dengan kemampuan berpikir logis. Kajian Tang et al. (2020) menunjukkan bahwa pengukuran computational thinking pada konteks pendidikan mencakup kemampuan pemrograman, pemecahan masalah, serta disposisi berpikir yang memerlukan instrumen penilaian beragam. Temuan tersebut menguatkan argumen bahwa pembelajaran pemrograman sederhana dapat menjadi media yang efektif untuk melatih logika siswa sekolah dasar apabila tujuan dan asesmennya dirancang secara selaras.

Dalam konteks statistika pendidikan, kemampuan berpikir logis juga berperan penting dalam memahami konsep data, pola, dan hubungan antarvariabel. Oleh karena itu, penguatan computational thinking sejak dini dapat memberikan dampak jangka panjang terhadap kesiapan siswa dalam mempelajari statistika dan matematika pada jenjang pendidikan selanjutnya.

2.3. Scratch sebagai Media Pembelajaran Pemrograman Sederhana

Scratch merupakan platform pemrograman visual berbasis block-based coding yang dikembangkan oleh MIT Media Lab. Scratch dirancang khusus untuk anak-anak dan pemula agar dapat belajar konsep pemrograman secara intuitif dan menyenangkan. Penggunaan blok-blok grafis yang dapat disusun secara visual memungkinkan siswa memahami logika program tanpa harus berhadapan dengan kompleksitas sintaks bahasa pemrograman tekstual.

Resnick et al. (2009) menjelaskan bahwa Scratch dikembangkan dengan prinsip low floor, high ceiling, and wide walls. Prinsip low floor memungkinkan siswa pemula segera memulai tanpa hambatan teknis yang besar. High ceiling

DOI: https://doi.org/10.31004/riggs.v5i2.5748
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

memberikan ruang bagi siswa mengembangkan proyek yang lebih kompleks seiring meningkatnya kemampuan. Sementara itu, wide walls memungkinkan beragam bentuk ekspresi kreatif sesuai minat siswa.

Sejumlah penelitian empiris menunjukkan potensi Scratch dalam pembelajaran sekolah dasar. Pérez-Marín et al. (2020) melaporkan peningkatan computational thinking setelah siswa sekolah dasar mengikuti pembelajaran konsep pemrograman menggunakan Scratch. Selain itu, Scratch memungkinkan guru mengamati proses berpikir siswa melalui alur program yang disusun sehingga evaluasi tidak hanya berfokus pada hasil akhir, tetapi juga pada proses.

Dalam penelitian ini, Scratch digunakan sebagai media utama pembelajaran pemrograman sederhana. Pemilihan Scratch didasarkan pada kesesuaiannya dengan karakteristik siswa sekolah dasar serta kemudahan penggunaannya dalam konteks pembelajaran di sekolah. Scratch juga mendukung pengumpulan data pembelajaran secara sistematis, yang selanjutnya dapat dianalisis menggunakan pendekatan statistika pendidikan.

2.4. Statistika Pendidikan dalam Evaluasi Pembelajaran

Statistika pendidikan merupakan alat utama dalam mengevaluasi efektivitas pembelajaran secara objektif. Statistika memungkinkan peneliti mengubah data mentah hasil belajar menjadi informasi yang bermakna melalui proses pengolahan dan analisis data. Menurut Sudjana (2019), statistika pendidikan berfungsi untuk mendeskripsikan fenomena pendidikan dan menguji hipotesis secara ilmiah.

Statistika deskriptif digunakan untuk menggambarkan karakteristik data, seperti nilai rata-rata, persentase, dan standar deviasi. Statistik ini memberikan gambaran umum mengenai kemampuan logika siswa sebelum dan sesudah mengikuti pembelajaran pemrograman sederhana. Sementara itu, statistika inferensial digunakan untuk menguji apakah perbedaan yang terjadi bersifat signifikan secara statistik atau hanya terjadi secara kebetulan.

Dalam penelitian kuantitatif pendidikan, uji perbedaan rata-rata seperti paired sample t-test sering digunakan untuk menganalisis perubahan kemampuan siswa sebelum dan sesudah perlakuan pembelajaran. Sugiyono (2021) menegaskan bahwa penggunaan uji statistik yang tepat akan meningkatkan validitas kesimpulan penelitian. Oleh karena itu, pemilihan teknik analisis statistik dalam penelitian ini disesuaikan dengan desain penelitian dan karakteristik data.

Pemanfaatan perangkat lunak statistik berbasis komputer juga menjadi bagian penting dalam penelitian ini. Teknologi komputer memungkinkan analisis data dilakukan secara lebih akurat, efisien, dan transparan. Dengan demikian, integrasi antara pembelajaran pemrograman, statistika pendidikan, dan teknologi komputer mencerminkan pendekatan pembelajaran dan penelitian yang relevan dengan tuntutan era digital.

3. Metode Penelitian

3.1. Pendekatan dan Desain Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan desain deskriptif dan inferensial. Pendekatan kuantitatif dipilih karena penelitian bertujuan mengukur secara objektif pengaruh implementasi kurikulum pemrograman sederhana terhadap kemampuan logika siswa melalui data numerik yang dapat dianalisis secara statistik. Desain deskriptif digunakan untuk menggambarkan karakteristik kemampuan logika siswa, sedangkan desain inferensial digunakan untuk menguji perbedaan kemampuan logika siswa sebelum dan sesudah mengikuti pembelajaran pemrograman sederhana.

Desain penelitian yang digunakan adalah pretest–posttest satu kelompok (one group pretest–posttest design). Desain ini memungkinkan peneliti membandingkan kondisi awal kemampuan logika siswa dengan kondisi setelah diberikan perlakuan pembelajaran pemrograman sederhana. Meskipun desain ini tidak melibatkan kelompok kontrol, analisis statistik inferensial tetap digunakan untuk menguji signifikansi perubahan yang terjadi.

3.2. Subjek dan Lokasi Penelitian

DOI: https://doi.org/10.31004/riggs.v5i2.5748
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

Penelitian dilaksanakan di SD Birrul Walidain Muhammadiyah Tanon, sebuah sekolah dasar swasta yang telah mengintegrasikan pembelajaran pemrograman sederhana dalam kurikulum pengembangan sekolah. Subjek penelitian adalah 25 siswa yang mengikuti pembelajaran pemrograman sederhana berbasis Scratch pada semester berjalan.

Teknik pengambilan sampel yang digunakan adalah sampling jenuh, yaitu seluruh anggota populasi dijadikan sebagai sampel penelitian. Teknik ini dipilih karena jumlah populasi relatif kecil dan seluruh siswa memiliki karakteristik pembelajaran yang homogen. Dengan menggunakan sampling jenuh, hasil penelitian diharapkan dapat merepresentasikan kondisi riil implementasi pembelajaran pemrograman sederhana di sekolah tersebut.

3.3. Variabel Penelitian

Penelitian ini melibatkan dua variabel utama, yaitu:

1. Variabel bebas (independent variable): implementasi kurikulum pemrograman sederhana berbasis Scratch.
2. Variabel terikat (dependent variable): kemampuan logika siswa.

Kemampuan logika siswa didefinisikan sebagai kemampuan berpikir sistematis dalam menyusun langkah-langkah pemecahan masalah, mengenali pola, dan memahami hubungan sebab-akibat dalam konteks pembelajaran pemrograman sederhana.

3.4. Teknik dan Instrumen Pengumpulan Data

Pengumpulan data dalam penelitian ini dilakukan menggunakan beberapa instrumen, yaitu:

1. Tes Kemampuan Logika

   Tes digunakan untuk mengukur kemampuan logika siswa sebelum (pretest) dan sesudah (posttest) mengikuti pembelajaran pemrograman sederhana. Tes disusun dalam bentuk soal objektif dan soal berbasis pemecahan masalah sederhana yang mengukur aspek urutan logika, pengulangan, dan pengambilan keputusan.

2. Angket Respons Siswa

   Angket digunakan untuk memperoleh data pendukung mengenai respons dan persepsi siswa terhadap pembelajaran pemrograman sederhana. Angket disusun menggunakan skala Likert dengan empat kategori penilaian.

3. Lembar Observasi Pembelajaran

   Observasi dilakukan untuk mencatat aktivitas siswa dan keterlibatan mereka selama proses pembelajaran berlangsung. Data observasi digunakan sebagai informasi pendukung dalam menafsirkan hasil analisis statistik.

Sebelum digunakan, instrumen penelitian diuji validitas dan reliabilitasnya secara terbatas untuk memastikan bahwa instrumen mampu mengukur variabel yang diteliti secara konsisten dan akurat.

3.5. Teknik Analisis Data

Analisis data dilakukan menggunakan statistika pendidikan berbantuan komputer. Tahapan analisis data meliputi:

1. Statistik Deskriptif

   Statistik deskriptif digunakan untuk menggambarkan kemampuan logika siswa melalui nilai rata-rata, skor minimum dan maksimum, serta standar deviasi. Analisis ini memberikan gambaran umum mengenai kondisi awal dan akhir kemampuan logika siswa.

2. Uji Prasyarat Analisis

DOI: https://doi.org/10.31004/riggs.v5i2.5748
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

Sebelum dilakukan analisis inferensial, data diuji prasyarat normalitas untuk memastikan bahwa data memenuhi asumsi penggunaan uji parametrik. Uji normalitas dilakukan menggunakan uji Shapiro–Wilk dengan taraf signifikansi 0,05.

3. Statistik Inferensial

   Untuk menguji perbedaan kemampuan logika siswa sebelum dan sesudah pembelajaran, digunakan **uji *paired sample t-test***. Uji ini dipilih karena data berasal dari subjek yang sama dengan dua kali pengukuran. Kriteria pengambilan keputusan didasarkan pada nilai signifikansi (p-value) dengan taraf signifikansi 0,05.

Seluruh analisis data dilakukan menggunakan perangkat lunak statistik berbasis komputer guna meminimalkan kesalahan perhitungan dan meningkatkan akurasi hasil analisis.

3.6. Prosedur Penelitian

Prosedur penelitian dilaksanakan melalui beberapa tahapan, yaitu:

1. Pelaksanaan pretest kemampuan logika siswa.
2. Implementasi pembelajaran pemrograman sederhana berbasis Scratch.
3. Pelaksanaan posttest kemampuan logika siswa.
4. Pengumpulan data angket dan observasi.
5. Analisis data statistik dan penarikan kesimpulan.

4. Hasil Penelitian dan Pembahasan

4.1. Hasil Analisis Statistik Deskriptif

Analisis statistik deskriptif dilakukan untuk memperoleh gambaran umum mengenai kemampuan logika siswa sebelum dan sesudah mengikuti pembelajaran pemrograman sederhana berbasis Scratch. Data yang dianalisis berasal dari hasil tes kemampuan logika yang diberikan pada tahap pretest dan posttest kepada 25 siswa SD Birrul Walidain Muhammadiyah Tanon.

Hasil analisis menunjukkan bahwa nilai rata-rata kemampuan logika siswa pada pretest berada pada kategori sedang. Hal ini mengindikasikan bahwa sebelum implementasi pembelajaran pemrograman sederhana, sebagian besar siswa telah memiliki kemampuan logika dasar, namun belum berkembang secara optimal dalam hal penyusunan langkah algoritmik dan pemecahan masalah terstruktur. Standar deviasi pada pretest menunjukkan variasi kemampuan logika yang relatif moderat antar siswa.

Setelah siswa mengikuti pembelajaran pemrograman sederhana, nilai rata-rata posttest menunjukkan peningkatan yang cukup signifikan. Peningkatan ini tidak hanya terlihat pada nilai rata-rata, tetapi juga pada penurunan standar deviasi, yang mengindikasikan bahwa kemampuan logika siswa menjadi lebih merata. Dengan kata lain, pembelajaran pemrograman sederhana tidak hanya meningkatkan kemampuan logika secara umum, tetapi juga membantu mengurangi kesenjangan kemampuan antar siswa.

Hasil statistik deskriptif ini memberikan indikasi awal bahwa implementasi kurikulum pemrograman sederhana berkontribusi positif terhadap pengembangan kemampuan logika siswa. Namun demikian, untuk memastikan bahwa peningkatan tersebut bersifat signifikan secara statistik, diperlukan analisis lanjutan menggunakan statistika inferensial.

4.2. Uji Prasyarat Analisis

Sebelum dilakukan uji inferensial, data hasil pretest dan posttest diuji prasyarat normalitas menggunakan uji Shapiro–Wilk. Uji ini dipilih karena jumlah sampel penelitian kurang dari 50 dan dinilai lebih sensitif dalam mendeteksi distribusi data.

DOI: https://doi.org/10.31004/riggs.v5i2.5748
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

Hasil uji normalitas menunjukkan bahwa nilai signifikansi data pretest dan posttest berada di atas taraf signifikansi 0,05. Dengan demikian, dapat disimpulkan bahwa data kemampuan logika siswa berdistribusi normal dan memenuhi asumsi untuk dilakukan analisis menggunakan uji parametrik. Pemenuhan asumsi ini menjadi dasar yang penting dalam menentukan teknik analisis statistik yang tepat agar kesimpulan yang dihasilkan bersifat valid.

4.3. Hasil Uji Statistik Inferensial

Untuk menguji perbedaan kemampuan logika siswa sebelum dan sesudah mengikuti pembelajaran pemrograman sederhana, digunakan uji paired sample t-test. Uji ini sesuai karena pengukuran dilakukan terhadap subjek yang sama dalam dua kondisi yang berbeda, yaitu sebelum dan sesudah perlakuan pembelajaran.

Hasil uji paired sample t-test menunjukkan nilai signifikansi (p-value) lebih kecil dari 0,05. Hal ini berarti terdapat perbedaan yang signifikan secara statistik antara kemampuan logika siswa sebelum dan sesudah mengikuti pembelajaran pemrograman sederhana. Dengan demikian, hipotesis yang menyatakan bahwa implementasi kurikulum pemrograman sederhana berpengaruh terhadap kemampuan logika siswa dapat diterima.

Secara statistik, hasil ini menunjukkan bahwa peningkatan kemampuan logika siswa bukan terjadi secara kebetulan, melainkan sebagai dampak dari perlakuan pembelajaran yang diberikan. Temuan ini memperkuat hasil analisis deskriptif dan memberikan dasar empiris yang kuat mengenai efektivitas pembelajaran pemrograman sederhana di sekolah dasar.

4.4. Pembahasan

4.4.1 Makna Peningkatan Kemampuan Logika Siswa

Peningkatan kemampuan logika setelah pembelajaran perlu dimaknai sebagai perubahan pada cara siswa mengorganisasi langkah penyelesaian masalah, bukan sekadar bertambahnya pengetahuan mengenai tombol atau fitur dalam aplikasi Scratch. Dalam aktivitas pemrograman, siswa harus menentukan tujuan, memecah tugas menjadi bagian yang lebih kecil, memilih perintah yang sesuai, dan mengurutkan perintah tersebut agar menghasilkan keluaran yang diharapkan. Rangkaian aktivitas ini menempatkan logika sebagai inti proses belajar. Ketika suatu sprite tidak bergerak, suara tidak muncul, atau alur cerita berhenti sebelum selesai, siswa memperoleh umpan balik langsung bahwa terdapat ketidaksesuaian antara rencana dan implementasi. Mereka kemudian meninjau kembali urutan blok, menguji dugaan, dan memperbaiki bagian yang bermasalah. Proses tersebut konsisten dengan gagasan computational thinking sebagai kemampuan merumuskan masalah dan menyusun solusi yang dapat dijalankan secara sistematis (Wing, 2006; Grover & Pea, 2013).

Dalam konteks siswa sekolah dasar, perubahan tersebut penting karena kemampuan logika masih berkembang melalui pengalaman konkret dan representasi visual. Blok perintah pada Scratch memberi bentuk yang dapat dilihat dan dimanipulasi terhadap konsep abstrak seperti urutan, kondisi, dan pengulangan. Dengan menyeret, menyusun, serta menjalankan blok, siswa dapat mengamati konsekuensi dari setiap keputusan. Hubungan antara tindakan dan keluaran menjadi lebih mudah dipahami dibandingkan apabila siswa langsung diminta menulis sintaks berbasis teks. Oleh karena itu, peningkatan skor posttest dapat dijelaskan melalui bertambahnya pengalaman siswa dalam membuat hubungan sebab-akibat, mengenali pola kesalahan, dan menyusun prosedur yang runtut. Temuan ini sejalan dengan kajian Zhang dan Nouri (2019) yang menunjukkan bahwa pembelajaran menggunakan Scratch dapat memfasilitasi beragam dimensi computational thinking pada peserta didik usia sekolah.

Meskipun hasil inferensial yang dilaporkan menunjukkan perbedaan signifikan, signifikansi statistik tidak boleh dipahami sebagai satu-satunya indikator keberhasilan. Efektivitas pembelajaran juga perlu dilihat dari perubahan perilaku belajar, kualitas strategi penyelesaian, kemampuan menjelaskan alasan pemilihan blok, dan kemandirian ketika menghadapi kesalahan. Data angket dan observasi yang telah dikumpulkan dalam penelitian ini berpotensi memperkaya interpretasi hasil tes. Ketika temuan kuantitatif didukung oleh meningkatnya keterlibatan, ketekunan, serta kemampuan siswa menjelaskan alur program, kesimpulan mengenai kontribusi pembelajaran menjadi lebih kuat. Pendekatan multi-instrumen ini relevan dengan rekomendasi Tang et al. (2020), yang menekankan bahwa penilaian computational thinking sebaiknya tidak hanya bergantung pada satu bentuk tes.

DOI: https://doi.org/10.31004/riggs.v5i2.5748
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

4.4.2 Peran Sequence, Loop, dan Conditional dalam Pembentukan Logika

Konsep sequence atau urutan menjadi fondasi pertama yang menghubungkan pemrograman dengan kemampuan berpikir logis. Siswa belajar bahwa perubahan posisi satu blok dapat mengubah seluruh perilaku program. Sebuah perintah bergerak yang ditempatkan sebelum perintah mengarah ke tujuan dapat menghasilkan keluaran berbeda dibandingkan ketika kedua blok disusun dalam urutan yang benar. Pengalaman ini melatih siswa memahami bahwa penyelesaian masalah membutuhkan prosedur yang jelas dan tidak dapat dilakukan secara acak. Di luar konteks pemrograman, keterampilan menyusun urutan juga relevan terhadap kegiatan membaca instruksi, menyelesaikan operasi matematika, melakukan percobaan sains, dan merencanakan aktivitas sehari-hari. Dengan demikian, pembelajaran Scratch berpotensi memberikan pengalaman lintas konteks selama guru membantu siswa merefleksikan hubungan antara urutan program dan urutan pemecahan masalah.

Konsep loop atau pengulangan memperkenalkan efisiensi berpikir. Siswa tidak hanya diminta membuat program yang berjalan, tetapi juga mencari cara yang lebih ringkas dan terstruktur. Ketika sebuah gerakan yang sama harus dilakukan berkali-kali, penggunaan blok pengulangan membantu siswa mengenali pola dan mengganti rangkaian perintah berulang dengan satu struktur yang lebih sederhana. Proses ini melatih pengenalan pola, generalisasi, dan efisiensi algoritma. Brennan dan Resnick (2012) menempatkan loop sebagai salah satu konsep komputasional penting karena penggunaannya menunjukkan bahwa siswa mulai memahami keteraturan yang dapat direpresentasikan secara formal. Kemampuan menemukan pola berulang juga mendukung pembelajaran matematika, khususnya pada pola bilangan, operasi berulang, dan hubungan antarvariabel.

Sementara itu, conditional atau percabangan melatih siswa membuat keputusan berdasarkan kondisi. Struktur jika-maka menuntut siswa mengidentifikasi informasi yang relevan, merumuskan syarat, dan memprediksi konsekuensi. Misalnya, sebuah karakter hanya memperoleh poin apabila menyentuh objek tertentu, atau berpindah layar apabila memenuhi kondisi yang ditentukan. Aktivitas tersebut membantu siswa memahami bahwa satu masalah dapat memiliki keluaran berbeda bergantung pada situasi. Kombinasi sequence, loop, dan conditional membentuk dasar algoritmik yang semakin kompleks. Apabila diajarkan secara bertahap melalui contoh kontekstual, ketiga konsep tersebut dapat memperkuat penalaran logis tanpa menjadikan pembelajaran terlalu teknis. Guru perlu memastikan bahwa siswa tidak sekadar meniru susunan blok, tetapi mampu menjelaskan fungsi dan alasan penggunaan setiap struktur.

4.4.3 Scratch sebagai Representasi Visual dan Pengurang Hambatan Sintaks

Keunggulan utama Scratch bagi siswa pemula terletak pada representasi visual berbasis blok. Dalam bahasa pemrograman tekstual, kesalahan tanda baca, penulisan perintah, atau struktur sintaks dapat menghentikan program meskipun ide penyelesaian siswa sudah benar. Kondisi tersebut berpotensi meningkatkan beban kognitif dan mengalihkan perhatian dari logika ke aspek teknis. Scratch mengurangi hambatan tersebut melalui blok yang hanya dapat disusun dalam kombinasi tertentu. Bentuk blok memberi petunjuk visual mengenai hubungan antarperintah, sehingga siswa dapat memusatkan perhatian pada urutan tindakan, kondisi, dan hasil. Prinsip lingkungan Scratch yang mudah dimasuki oleh pemula tetapi tetap menyediakan ruang bagi proyek kompleks mendukung pembelajaran bertahap dan kreatif (Resnick et al., 2009).

Visualisasi juga memudahkan guru mengamati proses berpikir siswa. Ketika program ditampilkan di layar, guru dapat menelusuri rangkaian blok bersama siswa dan mengajukan pertanyaan seperti mengapa blok tertentu ditempatkan pada posisi tersebut, kondisi apa yang diperiksa, atau bagian mana yang harus diperbaiki. Pertanyaan semacam ini mengubah pembelajaran dari kegiatan mengikuti tutorial menjadi dialog penalaran. Siswa didorong untuk mengungkapkan strategi, membandingkan alternatif, dan memberikan alasan terhadap solusi yang dipilih. Proses verbal tersebut penting karena kemampuan menjelaskan pemikiran merupakan indikator bahwa siswa memahami konsep, bukan hanya berhasil menghasilkan animasi.

Namun, kemudahan visual tidak otomatis menjamin berkembangnya logika. Siswa dapat membuat proyek dengan cara mencoba berbagai blok tanpa perencanaan, menyalin contoh, atau bergantung penuh pada bantuan guru. Karena itu, Scratch perlu ditempatkan sebagai media, bukan tujuan akhir pembelajaran. Setiap aktivitas sebaiknya memiliki persoalan yang jelas, kriteria keberhasilan, kesempatan melakukan prediksi sebelum menjalankan program, dan tahap refleksi setelah proyek selesai. Guru dapat menggunakan lembar perencanaan sederhana berisi tujuan, input, proses,

DOI: https://doi.org/10.31004/riggs.v5i2.5748
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

keluaran, serta kemungkinan kesalahan. Pendekatan tersebut membantu menghubungkan tindakan visual dengan struktur berpikir yang lebih eksplisit dan mencegah pembelajaran berubah menjadi aktivitas bermain tanpa tujuan kognitif.

4.4.4 Computational Thinking, Debugging, dan Ketekunan

Debugging merupakan bagian pembelajaran yang sangat relevan dengan pembentukan logika. Ketika program tidak bekerja sesuai rencana, siswa harus mengenali adanya kesalahan, menentukan lokasi kemungkinan kesalahan, menguji bagian program, dan mengevaluasi hasil perbaikan. Tahapan tersebut menyerupai proses pemecahan masalah ilmiah karena melibatkan observasi, penyusunan dugaan, pengujian, dan revisi. Kesalahan tidak lagi dipandang sebagai kegagalan akhir, tetapi sebagai informasi untuk memperbaiki solusi. Sikap ini penting pada jenjang sekolah dasar karena dapat membangun ketekunan dan keberanian mencoba. Pembelajaran yang memberi ruang aman untuk melakukan kesalahan membantu siswa memahami bahwa penyelesaian masalah sering memerlukan beberapa kali percobaan.

Dalam kerangka computational thinking, debugging berkaitan dengan praktik pengujian dan penyempurnaan artefak. Brennan dan Resnick (2012) menjelaskan bahwa proses komputasional tidak hanya mencakup konsep, tetapi juga praktik seperti menguji, memperbaiki, menggunakan kembali, dan melakukan abstraksi. Temuan Fagerlund et al. (2022) menunjukkan bahwa aktivitas pemrograman Scratch pada siswa sekolah dasar melibatkan proses perencanaan, iterasi, kolaborasi, dan debugging yang saling berkaitan. Karena itu, evaluasi kurikulum sebaiknya tidak hanya menilai apakah proyek selesai, melainkan juga bagaimana siswa bereaksi ketika menemui masalah dan strategi apa yang digunakan untuk mengatasinya.

Guru dapat memperkuat manfaat debugging dengan menggunakan pertanyaan penuntun daripada langsung memberikan jawaban. Pertanyaan seperti 'bagian mana yang sudah berjalan?', 'apa yang seharusnya terjadi?', dan 'blok mana yang mengendalikan tindakan tersebut?' membantu siswa mempersempit masalah. Strategi lain adalah meminta siswa menjalankan program langkah demi langkah, menonaktifkan sebagian blok, atau membandingkan proyek yang bekerja dengan proyek yang bermasalah. Kebiasaan ini melatih analisis sistematis dan mengurangi pola coba-coba tanpa arah. Dalam jangka panjang, keterampilan menghadapi kesalahan secara terstruktur dapat ditransfer ke tugas akademik lain, meskipun transfer tersebut perlu diteliti secara khusus dan tidak dapat diasumsikan hanya berdasarkan peningkatan skor pada satu jenis tes.

4.4.5 Keterlibatan, Kreativitas, dan Motivasi Belajar

Scratch memungkinkan siswa membuat cerita interaktif, permainan, animasi, dan simulasi sederhana. Ruang kreativitas ini memberi kesempatan kepada siswa untuk menghubungkan konsep pemrograman dengan minat dan pengalaman mereka. Ketika siswa memiliki pilihan terhadap karakter, latar, atau tema proyek, mereka cenderung memiliki rasa kepemilikan yang lebih kuat terhadap hasil belajar. Keterlibatan emosional tersebut dapat memperpanjang waktu siswa bertahan dalam menyelesaikan masalah. Studi Pérez-Marín et al. (2020) menunjukkan bahwa pembelajaran Scratch yang dirancang dengan pendekatan pedagogis yang sesuai dapat meningkatkan penguasaan konsep pemrograman dan computational thinking pada siswa pendidikan dasar.

Motivasi dalam pembelajaran pemrograman tidak hanya berasal dari tampilan menarik, tetapi juga dari siklus umpan balik yang cepat. Setelah blok dijalankan, siswa segera melihat apakah karakter bergerak, suara berbunyi, atau skor berubah. Umpan balik langsung membantu siswa menghubungkan perintah dengan akibat dan mendorong percobaan berikutnya. Akan tetapi, guru perlu menjaga keseimbangan antara kebebasan berkreasi dan pencapaian kompetensi. Proyek yang terlalu terbuka dapat membuat sebagian siswa kehilangan arah, sedangkan langkah yang terlalu terstruktur dapat mengurangi eksplorasi. Pendekatan bertahap dapat dimulai dari proyek contoh, dilanjutkan modifikasi, dan diakhiri proyek mandiri dengan kriteria yang jelas.

Respons positif siswa yang diperoleh melalui angket sebaiknya dianalisis sebagai data pendukung, bukan sebagai bukti tunggal efektivitas. Siswa dapat menyukai aktivitas karena media baru dan menarik, tetapi kesenangan tidak selalu identik dengan peningkatan kemampuan. Oleh karena itu, data motivasi perlu dibaca bersama hasil tes, observasi, dan kualitas artefak. Apabila siswa menunjukkan keterlibatan tinggi sekaligus mampu menjelaskan algoritma dan

DOI: https://doi.org/10.31004/riggs.v5i2.5748
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

memperbaiki kesalahan, maka motivasi berfungsi sebagai pendukung proses kognitif. Sebaliknya, apabila keterlibatan hanya terlihat pada aspek dekorasi proyek, guru perlu mengarahkan kembali perhatian pada tujuan logika dan pemecahan masalah.

4.4.6 Peran Guru dan Scaffolding dalam Implementasi Kurikulum

Keberhasilan kurikulum pemrograman sederhana sangat dipengaruhi oleh peran guru sebagai perancang pengalaman belajar. Guru tidak harus menjadi programmer profesional, tetapi perlu memahami konsep dasar, tujuan computational thinking, strategi mendampingi debugging, dan cara menilai proses. Dukungan guru penting terutama ketika siswa pertama kali menghadapi konsep abstrak. Scaffolding dapat diberikan melalui demonstrasi singkat, contoh konkret, kartu blok, pertanyaan penuntun, atau pembagian tugas menjadi tahapan kecil. Dukungan tersebut kemudian dikurangi secara bertahap ketika siswa mulai mandiri.

Kurikulum yang efektif perlu memuat urutan kompetensi yang realistis. Tahap awal dapat berfokus pada peristiwa dan sequence, dilanjutkan loop, conditional, variabel sederhana, dan penggabungan beberapa konsep dalam proyek. Setiap tahap sebaiknya memiliki tujuan terukur serta aktivitas latihan dan refleksi. Lye dan Koh (2014) menekankan pentingnya desain pembelajaran dan lingkungan konstruktif dalam pengembangan computational thinking melalui pemrograman. Dengan demikian, keberhasilan tidak cukup ditentukan oleh ketersediaan perangkat atau aplikasi, tetapi oleh kualitas tugas, interaksi kelas, dan umpan balik.

Pengembangan profesional guru juga perlu direncanakan secara berkelanjutan. Pelatihan satu kali mungkin membantu pengenalan alat, tetapi belum tentu cukup untuk membangun kemampuan mengintegrasikan computational thinking ke pembelajaran. Guru membutuhkan kesempatan mencoba proyek, mendiskusikan kesulitan, mengamati contoh pembelajaran, dan mengevaluasi pekerjaan siswa. Kajian Liu et al. (2024) menunjukkan bahwa dukungan terhadap guru merupakan komponen penting dalam integrasi computational thinking pada kelas K-12. Bagi SD Birrul Walidain Muhammadiyah Tanon, forum berbagi praktik antarguru dan dokumentasi modul dapat membantu menjaga konsistensi implementasi ketika kurikulum diperluas ke kelas atau tahun ajaran berikutnya.

4.4.7 Kolaborasi dan Pair Programming

Pembelajaran pemrograman dapat dilaksanakan secara individual maupun berpasangan. Dalam konteks keterbatasan perangkat, pair programming menjadi strategi yang relevan sekaligus membuka peluang pengembangan kemampuan komunikasi. Dua siswa dapat berbagi peran sebagai pengendali perangkat dan peninjau solusi, kemudian bertukar peran secara berkala. Siswa yang mengendalikan komputer bertanggung jawab menyusun blok, sedangkan pasangannya mengamati, memberi saran, dan memeriksa kesesuaian dengan rencana. Interaksi ini mendorong siswa mengungkapkan alasan dan mendengarkan alternatif penyelesaian.

Wei et al. (2021) menemukan bahwa pendekatan pair programming parsial dapat meningkatkan keterampilan computational thinking dan efikasi diri siswa sekolah dasar. Meski demikian, kerja berpasangan perlu dikelola agar tidak didominasi satu siswa. Guru dapat menetapkan aturan pergantian peran, memberikan lembar tanggung jawab, dan mengamati kualitas percakapan. Pasangan juga perlu disusun dengan mempertimbangkan karakteristik siswa, bukan hanya kemampuan akademik. Tujuannya adalah menciptakan interaksi produktif, bukan sekadar membagi penggunaan komputer.

Hasil penelitian ini yang menunjukkan peningkatan kemampuan logika dapat menjadi dasar untuk mengembangkan aktivitas kolaboratif pada implementasi berikutnya. Proyek kelompok kecil dapat meminta siswa merancang cerita, membagi bagian program, menguji proyek, dan mempresentasikan algoritma. Penilaian tetap perlu membedakan kontribusi individu dan hasil kelompok. Guru dapat menggunakan pertanyaan lisan, catatan refleksi, atau tugas singkat individual setelah proyek untuk memastikan setiap siswa memahami konsep. Dengan cara tersebut, kolaborasi tidak hanya menyelesaikan masalah keterbatasan sarana, tetapi menjadi strategi pedagogis untuk memperdalam penalaran.

4.4.8 Statistika Pendidikan dan Evaluasi Berbasis Data

DOI: https://doi.org/10.31004/riggs.v5i2.5748
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

Pemanfaatan statistika pendidikan dalam penelitian ini memberikan dasar yang lebih objektif untuk menilai perubahan kemampuan siswa. Statistik deskriptif membantu menggambarkan kecenderungan nilai dan variasi antar siswa, sedangkan uji inferensial digunakan untuk menilai apakah perbedaan sebelum dan sesudah pembelajaran cukup kuat secara statistik. Penggunaan perangkat lunak komputer mengurangi kesalahan perhitungan, tetapi tidak menggantikan keputusan metodologis peneliti. Pemilihan uji, pemeriksaan asumsi, kualitas instrumen, dan interpretasi hasil tetap menentukan validitas kesimpulan.

Pada desain pretest-posttest satu kelompok, paired sample t-test tepat digunakan ketika skor perbedaan memenuhi asumsi normalitas. Namun, pelaporan hasil idealnya mencakup nilai rata-rata, standar deviasi, selisih rata-rata, nilai t, derajat kebebasan, p-value, interval kepercayaan, dan ukuran efek. Informasi tersebut membantu pembaca membedakan signifikansi statistik dari signifikansi praktis. Dengan sampel yang kecil, p-value saja belum memberikan gambaran lengkap mengenai besar perubahan. Ukuran efek seperti Cohen's d untuk data berpasangan dapat digunakan untuk menunjukkan kekuatan peningkatan, sedangkan interval kepercayaan menunjukkan ketepatan estimasi.

Evaluasi berbasis data juga bermanfaat bagi sekolah setelah penelitian selesai. Guru dapat melakukan asesmen diagnostik awal, memantau kemajuan per topik, dan mengidentifikasi siswa yang membutuhkan dukungan. Data tidak harus selalu dianalisis dengan teknik kompleks; distribusi skor, persentase ketuntasan, kesalahan yang sering muncul, dan perkembangan kualitas proyek dapat memberikan informasi praktis. Prinsip utamanya adalah menggunakan data untuk memperbaiki pembelajaran, bukan sekadar menghasilkan nilai. Tang et al. (2020) menegaskan bahwa asesmen computational thinking memiliki bentuk beragam dan harus disesuaikan dengan konstruk yang hendak diukur.

4.4.9 Validitas Pengukuran dan Triangulasi Data

Kemampuan logika merupakan konstruk yang luas sehingga instrumen tes perlu memiliki hubungan yang jelas dengan indikator penelitian. Soal sebaiknya mewakili kemampuan menyusun urutan, mengenali pola, memilih kondisi, memprediksi keluaran, dan menemukan kesalahan. Validitas isi dapat diperkuat melalui penilaian ahli, sedangkan reliabilitas menunjukkan konsistensi pengukuran. Apabila instrumen belum dilaporkan secara rinci, penelitian berikutnya perlu menyertakan kisi-kisi, contoh soal, prosedur penskoran, dan hasil uji kualitas instrumen agar temuan dapat ditelaah dan direplikasi.

Selain tes, artefak Scratch dapat menjadi sumber data yang kaya. Struktur program dapat dianalisis berdasarkan penggunaan event, sequence, loop, conditional, variabel, sinkronisasi, dan strategi debugging. Akan tetapi, jumlah blok yang banyak tidak selalu menunjukkan kualitas logika yang tinggi. Program sederhana dapat lebih efektif apabila menggunakan struktur yang tepat. Karena itu, rubrik penilaian perlu mempertimbangkan ketepatan, efisiensi, keterbacaan, dan kemampuan siswa menjelaskan solusi. Kerangka Brennan dan Resnick (2012) dapat membantu membedakan konsep komputasional, praktik komputasional, dan perspektif komputasional.

Triangulasi antara skor tes, observasi, angket, wawancara singkat, dan artefak program akan menghasilkan pemahaman yang lebih utuh. Misalnya, siswa yang memperoleh peningkatan skor tetapi masih sangat bergantung pada bantuan guru memerlukan interpretasi berbeda dari siswa yang mampu menyelesaikan proyek secara mandiri. Demikian pula, respons angket yang tinggi perlu dibandingkan dengan keterlibatan nyata selama pembelajaran. Triangulasi tidak bertujuan membuat semua data selalu sama, melainkan menjelaskan mengapa hasil dapat berbeda dan apa maknanya bagi pelaksanaan kurikulum.

4.4.10 Pemerataan Akses dan Kesiapan Sarana

Implementasi pemrograman di sekolah dasar memerlukan perhatian terhadap pemerataan akses. Perbedaan pengalaman menggunakan komputer dapat memengaruhi kecepatan siswa beradaptasi dengan Scratch. Siswa yang telah terbiasa menggunakan perangkat mungkin lebih cepat menavigasi antarmuka, tetapi belum tentu memiliki logika lebih baik. Guru perlu memisahkan keterampilan operasional, seperti menggunakan mouse dan menyimpan berkas, dari kemampuan computational thinking. Sesi orientasi singkat dapat membantu menyamakan kemampuan dasar sebelum pembelajaran inti dimulai.

DOI: https://doi.org/10.31004/riggs.v5i2.5748
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

Keterbatasan jumlah komputer dapat diatasi melalui pembelajaran berpasangan, rotasi stasiun, aktivitas unplugged, dan penggunaan lembar perencanaan. Aktivitas unplugged dapat memperkenalkan algoritma menggunakan kartu instruksi, permainan arah, atau simulasi kondisi sebelum konsep diterapkan pada Scratch. Dağ et al. (2023) menunjukkan bahwa kegiatan coding tanpa komputer juga dapat mendukung peningkatan computational thinking siswa sekolah dasar. Pendekatan campuran ini mengurangi ketergantungan pada perangkat sekaligus memperkuat pemahaman konsep.

Sekolah juga perlu mempertimbangkan keberlanjutan sarana, keamanan akun, penyimpanan proyek, dan perlindungan data siswa. Proyek yang dibagikan secara daring harus mengikuti kebijakan sekolah dan pendampingan orang dewasa. Penggunaan identitas pribadi, gambar siswa, atau informasi lokasi perlu dibatasi. Literasi digital dalam kurikulum pemrograman sebaiknya mencakup etika penggunaan konten, penghargaan terhadap karya orang lain, dan kebiasaan menjaga privasi. Dengan demikian, inovasi kurikulum tidak hanya mengembangkan kemampuan teknis dan logis, tetapi juga membentuk perilaku digital yang bertanggung jawab.

4.4.11 Implikasi Kurikulum bagi SD Birrul Walidain Muhammadiyah Tanon

Hasil penelitian memberikan dasar awal bagi sekolah untuk melanjutkan kurikulum pemrograman sederhana secara terencana. Implementasi dapat disusun dalam unit pembelajaran berjenjang dengan durasi yang cukup, target kompetensi yang jelas, dan proyek yang dekat dengan kehidupan siswa. Pada tahap awal, siswa dapat membuat animasi urutan kegiatan atau cerita nilai-nilai karakter. Tahap berikutnya dapat menggunakan loop dan conditional dalam permainan edukatif, simulasi matematika, atau kuis sederhana. Integrasi konteks sekolah membuat pemrograman tidak berdiri sebagai mata pelajaran yang terpisah, melainkan menjadi alat untuk mengekspresikan pemahaman.

Kurikulum juga dapat dihubungkan dengan tujuan pendidikan karakter. Proyek berpasangan mengajarkan komunikasi dan tanggung jawab, debugging melatih kesabaran, sedangkan presentasi proyek mendorong kepercayaan diri. Dalam konteks sekolah berbasis nilai Islam, tema proyek dapat diarahkan pada kebiasaan baik, kepedulian sosial, lingkungan, atau cerita keteladanan tanpa mengurangi fokus pada konsep pemrograman. Guru perlu memastikan bahwa penilaian tetap berpusat pada kemampuan menyusun solusi, bukan hanya keindahan tampilan.

Agar program berkelanjutan, sekolah dapat membuat peta kompetensi per jenjang, bank proyek, rubrik penilaian, dan dokumentasi perkembangan siswa. Evaluasi tahunan dapat membandingkan ketercapaian kompetensi, kendala sarana, kesiapan guru, serta respons siswa dan orang tua. Temuan penelitian ini menjadi baseline, tetapi keputusan perlu diperbarui berdasarkan data implementasi berikutnya. Kajian sistematis Giannakoulas dan Xinogalos (2024) menunjukkan bahwa aktivitas pemrograman dan permainan edukatif secara umum memberikan hasil positif bagi computational thinking siswa sekolah dasar, namun efektivitasnya bergantung pada desain, sasaran keterampilan, dan metode evaluasi.

4.4.12 Keterbatasan Penelitian dan Arah Penelitian Selanjutnya

Interpretasi hasil perlu mempertimbangkan bahwa desain one-group pretest-posttest tidak melibatkan kelompok kontrol. Peningkatan skor dapat berkaitan dengan pembelajaran, tetapi pengaruh lain seperti pengalaman mengerjakan pretest, perkembangan alami, aktivitas belajar di luar penelitian, atau antusiasme terhadap media baru belum dapat dipisahkan sepenuhnya. Jumlah subjek yang hanya 25 siswa dan berasal dari satu sekolah juga membatasi generalisasi. Temuan lebih tepat dipahami sebagai bukti awal pada konteks SD Birrul Walidain Muhammadiyah Tanon.

Penelitian berikutnya dapat menggunakan desain kuasi-eksperimen dengan kelompok pembanding, sampel lebih besar, dan pengukuran tindak lanjut. Kelompok pembanding dapat memperoleh pembelajaran logika tanpa Scratch atau menggunakan strategi berbeda, sehingga kontribusi khusus kurikulum dapat dinilai lebih kuat. Pengukuran beberapa minggu setelah pembelajaran diperlukan untuk mengetahui apakah peningkatan bertahan. Peneliti juga dapat membandingkan pembelajaran individual, pair programming, dan pendekatan unplugged untuk menentukan strategi yang paling sesuai dengan kondisi sekolah.

Pelaporan kuantitatif perlu dilengkapi dengan data numerik terperinci dan ukuran efek. Analisis kualitatif terhadap percakapan siswa, catatan debugging, serta struktur proyek dapat menjelaskan mekanisme perubahan. Penelitian juga

DOI: https://doi.org/10.31004/riggs.v5i2.5748
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

dapat menguji apakah kemampuan yang diperoleh berpindah ke tugas matematika, sains, atau pemecahan masalah umum. Studi Zhang dan Nouri (2019) serta Tang et al. (2020) menunjukkan bahwa bidang computational thinking masih menghadapi tantangan dalam definisi dan pengukuran. Karena itu, penelitian lanjutan harus menyatakan indikator secara eksplisit dan menggunakan instrumen yang sesuai agar klaim mengenai peningkatan logika dapat dipertanggungjawabkan secara ilmiah.

5. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan, dapat disimpulkan bahwa implementasi kurikulum pemrograman sederhana berbasis Scratch di SD Birrul Walidain Muhammadiyah Tanon efektif dalam melatih kemampuan logika siswa sekolah dasar. Hasil analisis statistik deskriptif menunjukkan adanya peningkatan nilai rata-rata kemampuan logika siswa setelah mengikuti pembelajaran pemrograman sederhana. Peningkatan tersebut diperkuat oleh hasil uji statistik inferensial menggunakan *paired sample t-test* yang menunjukkan perbedaan signifikan antara kemampuan logika siswa sebelum dan sesudah perlakuan pembelajaran. Temuan ini menunjukkan bahwa pembelajaran pemrograman sederhana tidak hanya berperan sebagai sarana pengenalan teknologi, tetapi juga sebagai media pembelajaran yang mampu mengembangkan kemampuan berpikir logis, sistematis, dan algoritmik siswa. Aktivitas penyusunan blok program, pengujian logika alur, serta perbaikan kesalahan program mendorong siswa untuk berpikir runtut dan reflektif dalam memecahkan masalah. Dari perspektif statistika pendidikan, penelitian ini menegaskan pentingnya evaluasi pembelajaran berbasis data dalam menilai efektivitas inovasi kurikulum. Penggunaan statistik deskriptif dan inferensial berbantuan komputer memungkinkan peneliti menarik kesimpulan secara objektif dan ilmiah, sehingga hasil penelitian dapat dijadikan dasar pengambilan keputusan pendidikan yang lebih akurat. Berdasarkan hasil penelitian, disarankan agar sekolah dasar dapat mempertimbangkan pengembangan kurikulum pemrograman sederhana secara berkelanjutan sebagai bagian dari inovasi pembelajaran. Penelitian selanjutnya disarankan menggunakan desain eksperimen dengan kelompok kontrol serta jumlah sampel yang lebih besar agar hasil penelitian memiliki daya generalisasi yang lebih luas.

Referensi

Bers, M. U. (2020). Coding as a playground: Programming and computational thinking in the early childhood classroom (2nd ed.). Routledge.

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

Dağ, F., Şumuer, E., & Durdu, L. (2023). The effect of an unplugged coding course on primary school students' improvement in their computational thinking skills. Journal of Computer Assisted Learning, 39(6), 1902–1918. https://doi.org/10.1111/jcal.12850

Fagerlund, J., Vesisenaho, M., & Häkkinen, P. (2022). Fourth grade students' computational thinking in pair programming with Scratch: A holistic case analysis. International Journal of Child-Computer Interaction, 33, 100511. https://doi.org/10.1016/j.ijcci.2022.100511

Giannakoulas, A., & Xinogalos, S. (2024). Studying the effects of educational games on cultivating computational thinking skills to primary school students: A systematic literature review. Journal of Computers in Education, 11, 1283–1325. https://doi.org/10.1007/s40692-023-00300-z

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

Liu, Z., Gearty, Z., Richard, E., Orrill, C. H., Kayumova, S., & Balasubramanian, R. (2024). Bringing computational thinking into classrooms: A systematic review on supporting teachers in integrating computational thinking into K-12 classrooms. International Journal of STEM Education, 11, 51. https://doi.org/10.1186/s40594-024-00510-6

Lodi, M., & Martini, S. (2021). Computational thinking, between Papert and Wing. Science & Education, 30, 883–908. https://doi.org/10.1007/s11191-021-00202-5

Louka, K., & Papadakis, S. (2024). Enhancing computational thinking in early childhood education through ScratchJr integration. Heliyon, 10(10), e30482. https://doi.org/10.1016/j.heliyon.2024.e30482

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

Nordby, S. K., Bjerke, A. H., & Mifsud, L. (2022). Computational thinking in the primary mathematics classroom: A systematic review. Digital Experiences in Mathematics Education, 8, 27–49. https://doi.org/10.1007/s40751-022-00102-5

Pérez-Marín, D., Hijón-Neira, R., Bacelo, A., & Pizarro, C. (2020). Can computational thinking be improved by using a methodology based on metaphors and Scratch to teach computer programming to children? Computers in Human Behavior, 105, 105849. https://doi.org/10.1016/j.chb.2018.12.027

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

DOI: https://doi.org/10.31004/riggs.v5i2.5748
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

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

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. Computers & Education, 148, 103798. https://doi.org/10.1016/j.compedu.2019.103798

Wei, X., Lin, L., Meng, N., Tan, W., Kong, S.-C., & Kinshuk. (2021). The effectiveness of partial pair programming on elementary school students' computational thinking skills and self-efficacy. Computers & Education, 160, 104023. https://doi.org/10.1016/j.compedu.2020.104023

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

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

DOI: https://doi.org/10.31004/riggs.v5i2.5748
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)