Edcomtech: Jurnal Kajian Teknologi Pendidikan
Vol 9, No 2, Oktober 2024, pp. 66-78
https://doi.org/10.17977/um039v9i22024p66-78
pISSN: 2548-9879 (print) eISSN: 2599-2139 (online)
http://journal-fip.um.ac.id/index.php/edcomtech

Coding Learning Approaches For Early Childhood: A Systematic Review

Bayu Widyaswara Suwahyo[1*]
[1]*Fungsional Pengembang Teknologi Pembelajaran, Balai Besar Guru Penggerak Provinsi Jawa Timur*
**Email corresponding author: widbayu@gmail.com*

Article Info
Article history:
Received 01-03-2024
Revised 01-07-2024
Accepted 31-07-2024
Published 30-09-2024

How to cite: Suwahyo, B. W. (2024). Coding Learning Approaches For Early Childhood: A Systematic Review. *Edcomtech: Jurnal Kajian Teknologi Pendidikan, 9*(2), 66–78. https://doi.org/10.17977/um039v9i22024p66-78

Abstrak

Seiring meningkatnya peran literasi digital dalam pendidikan anak usia dini, diperlukan strategi yang tepat untuk memperkenalkan pemikiran komputasional dan pengodean kepada anak. Namun, masih terdapat keterbatasan strategi dan media pembelajaran yang sesuai dengan tahap perkembangan anak usia dini. Penelitian ini bertujuan untuk mengkaji secara sistematis hasil penelitian eksperimen terkait metode pembelajaran, media, dan faktor usia anak dalam pembelajaran pengodean pada pendidikan anak usia dini. Tinjauan sistematis ini menganalisis artikel yang dipublikasikan pada tahun 2020–2024 yang diperoleh dari empat basis data utama, yaitu Scopus, ScienceDirect, EBSCOhost, dan Web of Science. Dari proses seleksi berdasarkan kriteria inklusi dan eksklusi, terpilih 29 studi eksperimen untuk dianalisis lebih lanjut. Hasil kajian menunjukkan bahwa media pembelajaran konkret, seperti mainan pengodean fisik, efektif dalam meningkatkan pemahaman anak terhadap konsep pengodean. Strategi pembelajaran diklasifikasikan menjadi tiga pendekatan, yaitu plugged-in, unplugged, dan hibrida. Faktor usia berpengaruh signifikan terhadap keterlibatan dan pemahaman anak, khususnya pada pembelajaran yang bersifat instruksional. Meskipun pendekatan plugged-in dan unplugged sama-sama berdampak positif, pendekatan unplugged dinilai lebih efektif karena bersifat langsung, berbasis aktivitas, dan sesuai dengan karakteristik perkembangan anak usia dini. Tinjauan ini memberikan kontribusi penting bagi pengembangan pembelajaran pengodean pada pendidikan anak usia dini serta menjadi rujukan bagi pendidik dan pembuat kebijakan dalam merancang kurikulum yang responsif terhadap usia dan kesiapan digital anak.

Kata Kunci: *Pendidikan Anak Usia Dini, Pemikiran Komputasional, Pengodean, Pendekatan Pembelajaran, Media Pembelajaran.*

Abstract

As digital literacy becomes a core component of early childhood education, there's an increasing urgency to figure out how best to introduce computational thinking and programming to young learners. Despite growing interest, a clear gap remains: we still lack age-appropriate strategies and teaching tools that effectively support coding education at this developmental stage. This study tackles that gap through a systematic review of experimental research, focusing on how different teaching methods, learning media, and children's ages *influence their ability to grasp coding concepts in early education settings. Drawing on data from four major academic databases—Scopus (436 articles), ScienceDirect (284), EBSCOhost (107), and Web of Science (86)—the review screened publications from 2020 to 2024. After applying rigorous inclusion and exclusion criteria, 29 relevant experimental studies were selected for in-depth analysis. The findings highlight that tangible learning tools, such as physical coding toys, significantly enhance young children's comprehension. The teaching strategies reviewed fell into three broad categories: plugged-in (digital), unplugged (non-digital), and hybrid. Notably, age plays a critical role in how children engage with and understand coding, especially in instruction-heavy environments. While both plugged and unplugged approaches were beneficial, unplugged methods stood out as particularly effective thanks to their hands-on and developmentally appropriate nature. Overall, this review adds meaningful insights to the field of early coding education and offers practical guidance for educators and policymakers looking to build responsive, age-sensitive curricula that foster digital readiness from an early age.*

Keywords: *early childhood education, computational thinking, coding, learning approaches, teaching media.*

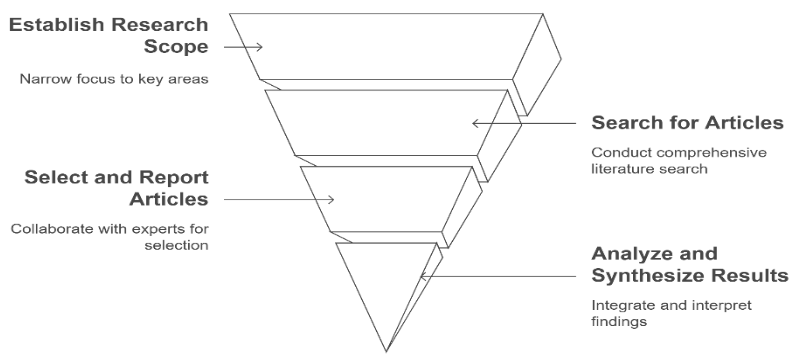
INTRODUCTION

As coding becomes essential in modern education, researchers have increasingly studied how young children develop problem-solving and programming skills. Coding is now seen as a fundamental literacy, enabling children to think logically, analyze problems, and create technology (Parandekar et al., 2023). Early exposure to coding is cost-effective and yields long-term benefits, enhancing cognitive flexibility, creativity, and resilience (Bers, 2020). Teaching coding at a young age also boosts collaboration, communication, problem-solving, and critical thinking skills (Pollarolo et al., 2024). While the benefits are widely acknowledged, debates continue regarding the most effective pedagogical approach—whether structured curricula or exploratory, play-based learning provide more meaningful engagement for young learners. At the same time, many early childhood educators face challenges in implementing coding education due to limited training, lack of resources, and the absence of clear guidelines (Zurnacı & Turan, 2024). However, many early childhood teachers face challenges due to limited training, resources, and guidelines (Lee et al., 2025). Coding has been integrated into education since the 1970s, with some researchers arguing it should be as fundamental as literacy (Papadakis, 2022). Despite significant efforts in some countries to promote coding in early childhood, its integration into formal settings remains a challenge (Kim & Jeong, 2023).

This study uses a systematic review to examine current methods for teaching coding to young children worldwide. Few studies have systematically reviewed coding applications in early childhood education. Pollarolo et al (2024) provide a review (2010–2022) on pedagogical strategies used by Early Childhood Education and Care (ECEC) teachers to support children's play with coding toys, exploring teachers' perspectives and the impact on children's cognitive and socio-emotional development. Rich et al (2024) reviewed over 300 coding tools for children and developed a classification framework, analyzing how these tools are used in elementary education. Macrides et al (2022) suggest that programming can be taught as a standalone subject or integrated into curricula with other subjects like music, art, and math, with storytelling and literacy-based activities offering developmentally appropriate ways to introduce programming. The study emphasizes the need for further research on integrated programming curricula and teacher training to better equip educators. Unlike previous studies that overlooked instructional design and media selection based on age, this literature review examines how age influences teaching strategies and the choice of educational media for coding instruction. It focuses exclusively on coding education in early childhood. By analyzing 29 experimental studies conducted between 2020 and 2024, this review explores the effectiveness of both digital (plugged-in) and hands-on (unplugged) learning methods while considering how age shapes children's engagement with coding. It also examines the impact of different teaching approaches, parental involvement, and inclusive education to ensure accessibility for all learners. As coding education for young children continues to grow, the best teaching strategies and curriculum structure remain unclear. By synthesizing insights from various studies, this review aims to inform preschool education policies and practices, helping educators design effective coding curricula that prepare young learners for the digital age.

METHOD

This systematic review, conducted from January 2020 to December 2024 following the PRISMA guidelines, aims to examine coding education for young children. The study follows several steps (Davis et al., 2014) : 1) defining the research scope based on the problem statement; 2) searching for relevant articles using predefined keywords; 3) selecting articles with expert collaboration; and 4) analyzing and synthesizing the results. PRISMA ensures a transparent, comprehensive review, comparing digital and non-digital learning approaches and considering age-related engagement. It also identifies research gaps, such as integrating coding into early childhood curricula and the need for teacher training. The study synthesizes experimental findings on computational thinking and programming, aiming to guide future educational practices and policies by highlighting gaps in existing literature (Kitchenham dkk., 2010). The review followed four key phases. First, the research scope was clearly defined, centered on the challenge of delivering effective coding instruction in early childhood education. Next, a comprehensive literature search was carried out using predefined keywords—such as “early childhood coding,”,“programming for preschool,” and “plugged and *unplugged learning”*—across academic databases including Scopus, ERIC, Web of Science, and Google Scholar.



Picture 1. Research Methodology

No.	*Database*	*Number of Articles Retrieved*
1	Scopus	436

2	ScienceDirect	284
3	EBSCOhost	107
4	Web of Science	86
	Total	913

Table 1. Databased

Research Questions

The primary aim of this study is to gather experimental data on effective coding education methodologies for early childhood programs and the selection of supporting instructional media. The media choices are categorized into plugged-in and unplugged approaches, with age considered as a variable influencing learning and the use of coding tools. Additionally, the study aims to identify future research areas and gaps in existing literature. To achieve these objectives, the research addresses the following questions:

1. What coding learning approaches can be applied in early childhood education programs?
2. What are the most suitable instructional media for teaching coding to young children?
3. How does age influence coding instruction in early childhood education?

Search Process

In this study, a literature review on coding in preschool education was conducted using electronic databases. such as Scopus (436 articles) , ScienceDirect (284 articles), EBSCOhost (107 articles), and Web of Science (86 articles). The selected articles were drawn from the fields of psychology, sociology, and education, with English as the primary language of the articles. The combined keyword criteria used to search the database indexes included: 'early childhood', 'preschool', 'child', 'childhood', 'kindergarten', 'computer', 'programming', 'coding', 'robot', 'robots', 'plugged in', and 'unplugged'. Additionally, various combinations of these keywords were applied in the database search, utilizing "and" and "or" operators. Based on these keyword criteria, 913 articles from the period between 2020 and 2024 were identified, which were then moved on to the next phase of analysis

Selection and Reporting Process

The article selection process followed four stages. First, incomplete publications (e.g., abstracts, posters) and non-English articles without full-text access were excluded. Next, titles, keywords, and abstracts were screened to focus on computational thinking and programming in early childhood, excluding literature reviews and teacher education studies. In the third stage, full-text articles meeting the criteria were analyzed to determine their relevance, then categorized by teaching media, media type (plugged, unplugged, or both), and age range. Finally, 29 studies were selected after assessing methodology, sample, data collection, and analysis to answer research questions on coding education for young children.

Table 1. Inclusion and exclusion criteria

Kriteria	Inclusion Criteria	Exclusion criteria
Article attributes	Journal papers that have gone through peer review	Papers not reviewed through the peer review process, book chapters, conference papers/proceedings, unpublished works

Language	English	Papers written in other languages
Educational settings	Chidhood education	Non-childhood education
Konteks penelitian	Coding as a main research study	coding as a research supplement;
Study attributes	Empirical studies implementing instructional/intervention activities	Opinion papers, literature reviews, meta-analyses, conceptual studies, book review
Research focus	integrating coding into a learning environment	Excluding AI-based learning environments & not ("computer-based learning") and Not ("intelligent tutoring system")
Participants	childhood education students	Students were not directly involved in the research

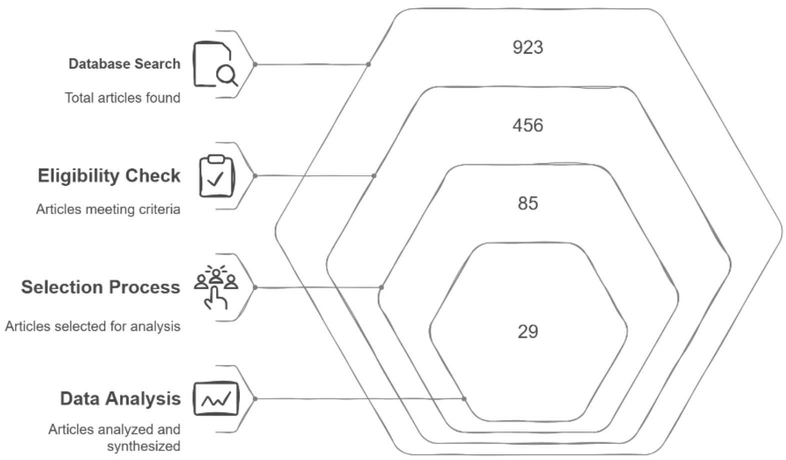


Figure 2. Selection and Reporting Process

RESULTS

The analysis of selected experimental studies revealed key findings in early childhood coding education. A major discovery is that computational thinking and programming skills are more effective when combined with motor activities, promoting active engagement with learning materials (Zurnacı & Turan, 2024). The study also highlights two main approaches: plugged-in, which uses digital tools like computers and tablets, and unplugged, which involves hands-on activities without digital devices. The effectiveness of each method is debated. Another finding emphasizes the importance of considering children's cognitive development, especially in the 3 to 6 age range, when skills like logical thinking, abstract reasoning, and problem-solving begin to grow (Piaget, 2013). Some studies tailor approaches to different age groups, offering insights into how developmental stages impact learning outcomes in computational thinking.

Table 2. comparison of coding learning methods.

Learning Method	Advantages	Disadvantages

Unplugged Coding	Enhances computational thinking (CT), problem-solving, and sequencing skills without screens.	Limited exposure to real coding environments.
Project-Based Learning	Encourages creativity and problem-solving through play.	Requires guidance and structured activities.
Robotics & Tangible Programming	Hands-on learning, improves CT and executive function skills.	Can be expensive and require adult supervision.
Game-Based Learning	Interactive, engaging, increases motivation for young learners.	May lack depth in coding concepts.
Screen-Based Block Coding	Develops problem-solving and debugging skills through digital tools.	Requires devices, which may lead to screen time concerns.
Hybrid Approach (Unplugged + Robotics + Digital Tools)	Combines unplugged, robotics, and digital platforms for comprehensive learning.	Requires well-structured curriculum and resources.

Early Childhood Coding Learning Approach

The debate over the effectiveness of plugged-in (digital devices) versus unplugged (non-digital) activities in early childhood education remains unresolved. (Bers et al., 2014) explored teaching programming using two approaches: iPad-based and paper-and-pencil-based. Although they expected the screen-based method to yield better results, their study found no significant difference between the two. They attributed this to statistical flaws in their study and continued to advocate for computer-based programming education. In contrast, (Demir & Demir, 2021) argued that the unplugged approach offers advantages by focusing on physical activities and hands-on tasks without relying on digital devices. This method makes computational concepts easier for young learners to grasp and helps reduce screen time-related issues. (Wang et al., 2014) compared three programming teaching methods—Hybrid User Interface (HUI), Tangible User Interface (TUI), and Graphical User Interface (GUI)—and found that tangible programming activities (TUI) helped children better understand abstract concepts like loops. Similarly, (Relkin et al., 2021; Weipeng et al., 2022) showed that children using the KIBO robot (unplugged) outperformed those using ScratchJr (plugged-in) in developing computational thinking (CT) skills, suggesting that concrete, hands-on activities are more effective for young learners. (Kotsopoulos et al., 2022) supported the unplugged approach, showing that activity-based coding programs without digital devices help develop computational thinking and problem-solving skills in preschool children. Robotics-based applications also enhance creativity, critical thinking, and collaboration (Demir & Demir, 2021; Zurnacı & Turan, 2024). (Misirli & Komis, 2023) found that tangible robotics activities improve programming skills in children aged 3 to 6, while studies by (Critten et al., 2022) and (Papadakis, 2022) showed that using programmable robots like Bee-Bot enhances both computational thinking and spatial awareness in young children. (Blake-West & Bers, 2023) highlighted that ScratchJr, a developmentally appropriate tool for early childhood education, increases engagement and programming skills, with children often preferring certain movement blocks. (Pérez-Marín et al., 2022) also noted that children tend to choose code blocks based on preference rather than purpose. Finally, (Zurnacı & Turan, 2024) suggested that unplugged robot toys, which provide more concrete feedback and opportunities for trial and error, are particularly effective in teaching programming in early childhood education.

The debate over whether digital (plugged-in) or non-digital (unplugged) methods are more effective in early childhood education is still ongoing. While digital tools offer modern, engaging learning experiences, research suggests that unplugged approaches often work better for young children. Hands-on activities, like using tangible robots and physical play, help kids understand abstract concepts more easily, develop problem-solving skills, and improve memory and focus. Unplugged methods also reduce screen time, which can help prevent issues like attention problems or digital addiction. Ultimately, the best teaching approach should align with a child's developmental stage and provide meaningful, interactive learning experiences.

Educational Media for Coding in Early Childhood

Learning tools created to foster coding skills fall into two main categories: screen-based technology, which operates via devices such as computers, tablets, or smartphones, and tangible technology, which involves direct physical manipulation without the need for separate screen-based software (Bers et al., 2014). More specifically, screen-based programming tools for early childhood education (ages 3-6) can be divided into three types: a) robots with programming software controlled through screens, b) Web-based games and platforms intended to teach programming principles, and c) Tools for animation and game development. For example, screen-based robots like Lego WeDo 2.0 and Dash & Dot use physical electronic components, while online platforms (such as Code.org, LightbotJr, and Kodable) and tools like ScratchJr do not rely on physical components. (Lavigne et al., 2020) describe web-based applications and tools such as ScratchJr as 'virtual kits.' In related studies, Lego WeDo was used with its own visual programming software, while digital applications like Robopad, Daisy the Dinosaur, and Code.org were also explored. ScratchJr, an animation development tool, was featured in several studies, including those that compared its effectiveness with physical programming toys and web-based games. KIBO was implemented in six interventions, Bee-Bot in five, Code-A-Pillar in two, and Code & Go Robot Mouse in. Additionally, three other programming toys identified include the Arduino Robot Car, the ALERT floor robot kit, and TurtleBot (Misirli & Komis, 2023; Weipeng et al., 2022; Yang et al., 2022)

In summary, the inconsistency in technology classification highlights a challenge in educational research. The rapid advancement of technology, including both new tools and improvements to existing ones, often leads to overlapping definitions and categories. For instance, tools like Lego WeDo might be categorized as screen-based technology because they use visual software, yet they also feature physical components, placing them in the tangible technology category. Nevertheless, this diversity offers significant opportunities. A more organized classification system would help researchers assess tool effectiveness more easily, while educators could select the most appropriate technology to meet children's learning needs. Furthermore, these tools offer the chance to cultivate programming skills from an early age, laying an important foundation for future learning.

Learning tools for developing computational thinking and coding skills generally fall into two categories: screen-based technology, which uses devices like computers and tablets, and tangible technology, which allows hands-on interaction without relying on screens. However, classifying these tools isn't always straightforward. Some, like Lego WeDo, use both digital programming software and physical components, making it difficult to place them in just one category. This lack of clear classification can be a challenge in educational research, as it complicates the process of evaluating and comparing different tools. Despite this, the wide range of available technologies provides valuable learning opportunities for young children. A more organized classification system could help researchers study these tools more effectively and assist educators in selecting the best options for early learners. Ultimately, these tools help build foundational programming skills from an early age, setting the stage for future learning.

Age In Coding Learning

Although not explicitly mentioned in the studies reviewed, Age plays a crucial role in programming (Papadakis, 2022). However, it remains crucial for educators and researchers to consider age when designing teaching practices. While younger children may not grasp abstract concepts as easily, early exposure to programming can enhance long-term skill development (Bati, 2022). Thus, teaching methods should be tailored to the developmental stages of students to ensure effective learning. (Pérez-Marín et al., 2022) also found that Preschool children aged 3 to 5 are able to successfully complete programming tasks. In a different study (Bers et al., 2014) included In their research, they focused on 3-year-olds, using the KIBO robot kit, which is specifically designed for young children. The results showed that children were able to acquire advanced coding and computational thinking skills through robotics, proving that learning can begin at a very early age. However, while early coding is possible, researchers observed that children who were successful in problem-solving tasks with four instructions might face challenges with tasks involving five instructions (Critten et al., 2022; Kotsopoulos et al., 2022). These findings align with those of (Yang et al., 2022), which indicate that both knowledge and robot programming skills improve as children age. In a study with 60 children ranging from preschool to second grade, no significant differences were observed between kindergarten students. However, a notable difference in problem-solving abilities was found between preschoolers and second-grade children. Preschoolers performed poorly on the Solve-It 1 task (a simple task that required them to arrange four commands correctly). They also struggled with the more complex Solve-It 2 task, which involved arranging five commands correctly. The researchers believe these findings are linked to children's working memory and attention span, and this assumption seems to be valid. (Bati, 2022) suggest that young children do not have the memory capacity to retain five instructions at the same time. In line with this, (Ploog & Wiktorski, 2024) found that 5-year-olds perform better than younger children in terms of memory retention. The conclusion from this discussion is that young children (ages 3-4) can successfully complete programming and coding tasks with a limited number of instructions, while older children (ages 5-6) are able to manage tasks involving five or more instructions. This raises the question: can programming applications help develop working memory in children? According to (Gerosa et al., 2022), the answer is no, as their study on whether programming activities improve children's math skills, Spatial awareness and working memory were assessed, but no improvement was observed in working memory. However, improvements were seen in their math skills and spatial awareness. This finding is important because it highlights the need to consider working memory when designing programming programs and computational thinking instruction for early childhood education. Additionally, (Lavigne et al., 2020) showed that using a coding curriculum with coded card media reduces Cognitive load is considered by taking into account the cognitive development aspects related to understanding and following instructions. Put simply, computational applications that use concrete objects can help make abstract concepts more understandable for children. (Lin et al., 2020) discovered that age plays a crucial role in programming, as kindergarten children process information differently than elementary school students. Their research demonstrated that using robotics programs with design-based, concrete media can support cognitive development in children during the pre-operational, transitional, and concrete operational stages. The findings showed a progressive increase in scores across these developmental stages, with statistically significant differences. Children in the concrete operational stage displayed more effective programming skills than those in the pre-operational stage, as they were able to use practical feedback and structured reasoning to complete tasks more accurately. In a separate study by (Saxena et al., 2020), Using the Bee-Bot in an unplugged problem-solving activity with two age groups (K1: 3-4 years old, K2: 4-5 years old) revealed differences in instruction comprehension. While K1 students could complete the activity, they struggled with instructions like "turn left" or "turn right." Researchers recommended using simpler tasks with two to four steps for younger children to support their language and computational thinking (CT) development. The study highlights the importance of age and working memory in teaching CT and coding, suggesting a tailored approach based on cognitive abilities for effective learning.

Age plays a key role in teaching coding to young children. Research shows that even preschoolers (ages 3-4) can complete simple programming tasks, but older children (ages 5-6) are better equipped to handle more complex instructions due to their developing working memory and cognitive abilities. While programming activities can enhance skills like math and spatial awareness, they don’t necessarily improve working memory, which is essential for following multi-step instructions. This highlights the importance of designing age-appropriate learning experiences. By using hands-on, tangible tools, educators can help young children better understand abstract concepts and build a strong foundation for coding in a way that aligns with their developmental stage.

DISCUSSION

This study aims to summarize the experimental findings related to computational thinking and programming in early childhood education, concentrating on elements such as the learning approaches used, appropriate learning media, and the comparison between plugged-in and unplugged methods, as well as age. The literature reviewed primarily focuses on robotics tools like KIBO and block-based coding platforms such as ScratchJ. As mentioned earlier, there is continued debate regarding the effectiveness of plugged-in versus unplugged applications. Research indicates that using ScratchJr enhances children's experiences in areas such as self-expression, numerical and spatial reasoning, creative thinking, and storytelling (Blake-West & Bers, 2023). Other studies show that using robotic toys like Bee-Bot enables children to engage with tools and explore interactions in subject areas Such as science, engineering, math, and literacy (Critten et al., 2022; Papadakis, 2022), while also introducing them to computer programming (Misirli & Komis, 2023). It is concluded that unplugged applications are more effective in teaching computational thinking and programming skills in early childhood education because they provide more concrete experiences. In line with this, (Bers et al., 2014) demonstrated that using concrete media, such as off-screen robot features, creates a collaborative environment where children can actively work together. This setting also encourages children to share materials, leading to a more interactive learning experience and supporting the development of their social skills. Additionally, (Blake-West & Bers, 2023) It was emphasized that ScratchJr, designed for tablet use, does not offer as tangible an experience for children, despite providing a similar coding experience through programming steps on a computer screen. To address this, (Blake-West & Bers, 2023) developed the "ScratchJr Collaborative Project Guide," which enables multiple children to collaborate on a project. However, (Misirli & Komis, 2023) emphasized that no concrete evidence was found to support that programming with an iPad or tablet is more effective than using pen and paper. They concluded that the quality of the learning experience itself is more important than the teaching environment, particularly when focusing on technology for early childhood coding.

Having the right tools for computer education is important, but what matters more is creating learning experiences that fit naturally into a child's environment. This depends on teachers who are knowledgeable and prepared to implement these methods effectively. According to (Barr et al., 2011), Successfully integrating computational thinking into early childhood education requires policy changes and adequate teacher resources. This study confirms that young children can acquire computational thinking and programming skills, with research now focusing on the ideal learning environment. While both plugged and unplugged activities are valuable, unplugged methods have proven more effective, offering practical and budget-friendly benefits

CONCLUSION

This review highlights that young children are capable of learning programming and developing computational thinking (CT) skills through both digital (plugged-in) and non-digital (unplugged) methods. However, unplugged approaches—like robotics and hands-on activities—tend to be more effective for younger learners, as they offer tangible, age-appropriate experiences that align with children's cognitive development. Age is a significant factor in learning success. As children grow older, they can better manage complex tasks and multi-step instructions thanks to their developing memory and thinking skills. For this reason, coding activities should be tailored to match children's developmental stages and cognitive abilities. Ultimately, while having access to the right tools is helpful, what matters most is the quality of the learning experience and the teacher’s readiness to guide it. A balanced, developmentally appropriate approach—supported by strong teacher training and clear education policies—is key to successfully integrating coding into early childhood education.

While unplugged methods have shown strong potential for teaching coding to young children, current research faces several limitations, such as small participant groups, unclear distinctions between types of educational tools, and a lack of focus on long-term effects.

Although age and cognitive development—especially working memory—are recognized as important factors, they are not yet fully understood within this context. Future studies should aim to create clearer classifications of learning tools, explore how early coding education shapes development over time, and examine how teaching strategies, teacher readiness, and age-appropriate resources impact children's learning and growth.

REFERENCES

Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology*, *38*(6), 20–23.

Bati, K. (2022). A systematic literature review regarding computational thinking and programming in early childhood education. *Education and Information Technologies*, *27*(2), 2059–2082. https://doi.org/10.1007/s10639-021-10700-2

Bers, M. U. (2020). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. https://doi.org/10.4324/9781003022602

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, *72*, 145–157. https://doi.org/10.1016/j.compedu.2013.10.020

Blake-West, J. C., & Bers, M. U. (2023). ScratchJr design in practice: Low floor, high ceiling. *International Journal of Child-Computer Interaction*, *37*, 100601. https://doi.org/10.1016/j.ijcci.2023.100601

Critten, V., Hagon, H., & Messer, D. (2022). Can Pre-school Children Learn Programming and Coding Through Guided Play Activities? A Case Study in Computational Thinking. *Early Childhood Education Journal*, *50*(6), 969–981. https://doi.org/10.1007/s10643-021-01236-8

Davis, J., Mengersen, K., Bennett, S., & Mazerolle, L. (2014). Viewing systematic reviews and meta-analysis in social research through different lenses. *SpringerPlus*, *3*(1), 511. https://doi.org/10.1186/2193-1801-3-511

Demir, B. E., & Demir, F. (2021). Coding, Robotics and Computational Thinking in Preschool Education: The Design of Magne-Board. *Avrupa Bilim ve Teknoloji Dergisi*, (23), 52–61. https://doi.org/10.31590/ejosat.842483

Gerosa, A., Koleszar, V., Tejera, G., Gómez-Sena, L., & Carboni, A. (2022). Educational Robotics Intervention to Foster Computational Thinking in Preschoolers: Effects of Children's Task Engagement. *Frontiers in Psychology*, *13*. https://doi.org/10.3389/fpsyg.2022.904761

Kim, Y., & Jeong, H. I. (2023). Adopting Coding in Early Childhood Education: Teachers' Perspectives. *IEEE Transactions on Learning Technologies*, *16*(4), 473–483. https://doi.org/10.1109/TLT.2023.3264681

Kitchenham, B., Pretorius, R., Budgen, D., Pearl Brereton, O., Turner, M., Niazi, M., & Linkman, S. (2010). Systematic literature reviews in software engineering – A tertiary study. *Information and Software Technology*, *52*(8), 792–805. https://doi.org/10.1016/j.infsof.2010.03.006

Kotsopoulos, D., Floyd, L., Dickson, B. A., Nelson, V., & Makosz, S. (2022). Noticing and Naming Computational Thinking During Play. *Early Childhood Education Journal*, *50*(4), 699–708. https://doi.org/10.1007/s10643-021-01188-z

Lavigne, H. J., Lewis-Presser, A., & Rosenfeld, D. (2020). An exploratory approach for investigating the integration of computational thinking and mathematics for preschool children. *Journal of Digital Learning in Teacher Education*, *36*(1), 63–77. https://doi.org/10.1080/21532974.2019.1693940

Lee, J., Yunus, S., & Lee, J. O. (2025). Investigating Children's Programming Skills Through Play with Robots (KIBO). *Early Childhood Education Journal*, *53*(1), 109–117. https://doi.org/10.1007/s10643-023-01563-y

Lin, S.-Y., Chien, S.-Y., Hsiao, C.-L., Hsia, C.-H., & Chao, K.-M. (2020). Enhancing Computational Thinking Capability of Preschool Children by Game-based Smart Toys. *Electronic Commerce Research and Applications*, *44*, 101011. https://doi.org/10.1016/j.elerap.2020.101011

Macrides, E., Miliou, O., & Angeli, C. (2022). Programming in early childhood education: A systematic review. *International Journal of Child-Computer Interaction*, *32*, 100396. https://doi.org/10.1016/j.ijcci.2021.100396

Misirli, A., & Komis, V. (2023). Computational thinking in early childhood education: The impact of programming a tangible robot on developing debugging knowledge. *Early Childhood Research Quarterly*, *65*, 139–158. https://doi.org/10.1016/j.ecresq.2023.05.014

Papadakis, S. (2022). Apps to Promote Computational Thinking and Coding Skills to Young Age Children: A Pedagogical Challenge for the 21st Century Learners. *Educational Process: International Journal (EDUPIJ)*, *11*(1), 7–13. https://doi.org/DOI:10.22521/edupij.2022.111.1

Parandekar, S., Patarakin, E., & Yayla, G. (2023). A Modern Aspect of Instrumental Literacy: Coding. Dalam M. Dobryakova, I. Froumin, K. Barannikov, G. Moss, I. Remorenko, & J. Hautamäki (Ed.), *Key Competences and New Literacies: From Slogans to School Reality* (hlm. 367–390). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-23281-7_13

Pérez-Marín, D., Hijón-Neira, R., & Pizarro, C. (2022). Coding in early years education: Which factors influence the skills of sequencing and plotting a route, and to what extent? *International Journal of Early Years Education*, *30*(4), 969–985. https://doi.org/10.1080/09669760.2022.2037076

Piaget, J. (2013). *Play, dreams and imitation in childhood* (Vol. 25). Routledge. https://doi.org/10.4324/9781315009698

Ploog, M., & Wiktorski, T. (2024). The intentions of the designers of digital educational tools in early childhood education. *International Journal of Child-Computer Interaction*, *40*, 100653. https://doi.org/10.1016/j.ijcci.2024.100653

Pollarolo, E., Papavlasopoulou, S., Granone, F., & Reikerås, E. (2024). Play with Coding Toys in Early Childhood Education and Care: Teachers' Pedagogical Strategies, Views and Impact on Children's Development. A Systematic Literature Review. *Entertainment Computing*, *50*, 100637. https://doi.org/10.1016/j.entcom.2024.100637

Relkin, E., de Ruiter, L. E., & Bers, M. U. (2021). Learning to code and the acquisition of computational thinking by young children. *Computers & Education*, *169*, 104222. https://doi.org/10.1016/j.compedu.2021.104222

Rich, P. J., Bartholomew, S., Daniel, D., Dinsmoor, K., Nielsen, M., Reynolds, C., … Yauney, J. (2024). Trends in tools used to teach computational thinking through elementary coding. *Journal of Research on Technology in Education*, *56*(3), 269–290. https://doi.org/10.1080/15391523.2022.2121345

Saxena, A., Lo, C. K., Hew, K. F., & Wong, G. K. W. (2020). Designing Unplugged and Plugged Activities to Cultivate Computational Thinking: An Exploratory Study in Early Childhood

Education. *The Asia-Pacific Education Researcher*, *29*(1), 55–66. https://doi.org/10.1007/s40299-019-00478-w

Wang, D., Wang, T., & Liu, Z. (2014). A tangible programming tool for children to cultivate computational thinking. *The Scientific World Journal*, *2014*. https://doi.org/10.1155/2014/428080

Weipeng Yang, Ng, D. T. K., & Gao, H. (2022). Robot programming versus block play in early childhood education: Effects on computational thinking, sequencing ability, and self-regulation. *British Journal of Educational Technology*, *53*(6), 1817–1841. https://doi.org/10.1111/bjet.13215

Yang, W., Ng, D. T. K., & Gao, H. (2022). Robot programming versus block play in early childhood education: Effects on computational thinking, sequencing ability, and self-regulation. *British Journal of Educational Technology*, *53*(6), 1817–1841. https://doi.org/10.1111/bjet.13215

Zurnacı, B., & Turan, Z. (2024). Educational robotics or unplugged coding activities in kindergartens?: Comparison of the effects on pre-school children's computational thinking and executive function skills. *Thinking Skills and Creativity*, *53*, 101576. https://doi.org/10.1016/j.tsc.2024.101576