[1] Sri Utami

[2] Anik Ghufron

A Critical Appraisal: Elementary School Coding Education Effectiveness with Scratch in Alignment with Cognitive Development, Analyzed through Piagetian Lens

Journal of Electrical Systems

Abstract: - The digitalization has revolutionized the education sector by integrating technology and bringing about significant changes. Computational Thinking (CT), an indispensable skill for the 21st century, has now become an integral part of school curricula in several countries worldwide. Focused on enhancing CT skills, programming education, especially through Scratch, Scratch simplifies programming concepts, making it suitable for elementary school children aged 7 to 11 years. Aligned with Piaget's concrete operational stage, emphasizing organized and rational thinking. Hence, this study aims to delve deeper into how Scratch programming education can meet the cognitive developmental needs of children in this phase. The research methodology used is a literature review, employing data from textbooks, journals, scholarly articles, and literature reviews that contain concepts central to the research focus. The data analysis involves collecting relevant research findings, organizing, and systematically recording information. The results indicate that Scratch not only provides a platform for learning programming but also supports the development of concrete operational cognitive skills taught in Piaget's theory. Through creativity in designing Scratch projects, children can apply and strengthen their understanding of concepts such as seriation, classification, reversibility, conservation, and the elimination of egocentrism. But, it's essential to consider an approach tailored to the developmental characteristics of children when designing programming education at the elementary level. This study is expected to provide concise guidance for educators and policymakers in formulating programming education that optimally fosters cognitive development in elementary school children.

Keywords: Coding Education, Scratch, Cognitive Development, Concrete Operational Stage, Elementary School, Piagetian Lens.

I. Introduction

In recent years, there has been a growing recognition of the importance of integrating coding education into elementary school curricula, acknowledging the transformative potential it holds for fostering critical thinking and problem-solving skills among young learners. This popularity has led governments, researchers, policymakers, and stakeholders worldwide to propose computing in compulsory. This topic has been the subject of attention even across international organizations such as the OECD and UNESCO, recognizing CT and coding as essential competencies required to meet future requirements for a successful life [1]. It is believed that programming is one of the most efcient ways to develop CT and cognitive skill [2]

One prominent platform that has gained widespread adoption for introducing coding concepts to elementary school students is Scratch. Scratch offers an visually engaging programming space where users can easily drag and stack blocks within the Scratch programming environment's script area. The user can observe the execution of each block, promoting a comprehension of intricate issues and encouraging the exploration of uncomplicated solutions [3].

The age range of elementary school children is approximately 7-12 years old, entering Piaget's theory of the concrete operational stage. In this stage, problem-solving and reasoning become lifelong skills. They can create logical structures explaining physical experiences, including solving abstract problems. At this stage, children can engage in logical thinking based on past experiences and concrete evidence, successfully completing tasks related to material conservation, transitive reasoning, and object classification.[4]

The in-depth exploration of the effects of Scratch learning on children has emerged as a crucial question, particularly in the context of their cognitive considerations. While Scratch learning is acknowledged to have positive impacts, As indicated by previous research conducted by Bahar, Scratch had a significant effect on children's computational thinking skills, which are a part of cognitive skills [5]. its alignment with the cognitive development needs of elementary school children and its implications remain an area with insufficient in-depth research focus.

It is important to tailor learning to the cognitive development of children in order to maximize understanding and information reception. The use of methods and approaches, as well as the selection of learning materials, that align with their developmental stages is crucial. Non-alignment may increase the risk of misunderstanding, boredom, or

[1,2] Yogyakarta State University,

1sri0073fipp.2022@student.uny.ac.id,2anikghufron@uny.ac.id

failure to comprehend the learning material, thereby potentially hindering academic development and cognitive skills in children.

Therefore, this study aims to delve deeper into the compatibility and alignment of coding education through the Scratch program with the cognitive developmental stages of elementary school children, referencing Jean Piaget's theory of cognitive development. This research is directed towards establishing a more comprehensive knowledge base regarding its potential implications on the cognitive development of children.

II. THEOEITICAL FRAMEWORK

A. Coding Education with Scratch

1. Coding education

Coding education is a learning process aimed at imparting knowledge and skills in computer programming, encompassing the creation, reading, and understanding of programming languages. Its primary objective is to enhance understanding of programming logic, algorithms, and data structures, enabling individuals to develop software applications and improve problem-solving abilities, logical thinking, and creativity.

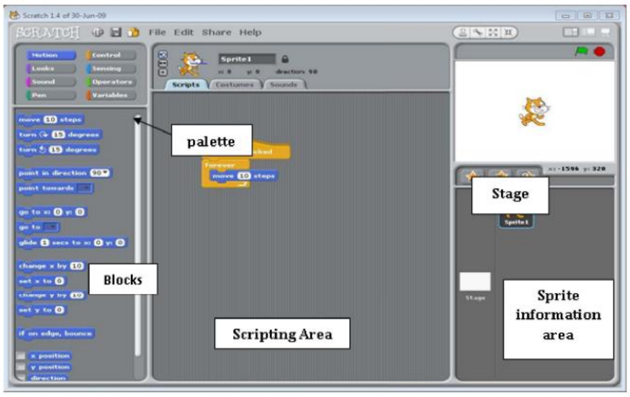
2. Scratch

Scratch is a visual programming language, empowers children to effortlessly create interactive content like stories, games, music, and art.[6] Developed by MIT's Media Lab, Scratch employs puzzle-piece-like code representation, ensuring syntactic correctness. Designed for all ages and backgrounds, Scratch serves as an introductory language, encouraging users to program interactive stories, games, animations, and simulations.[7] It eliminates the complexity of syntax and tedious program formats, allowing students to click and drag code blocks for seamless programming.[8] Using graphical programming blocks, Scratch offers a user-friendly approach to building diverse projects.[9]

3. What is learned from learning Scratch?

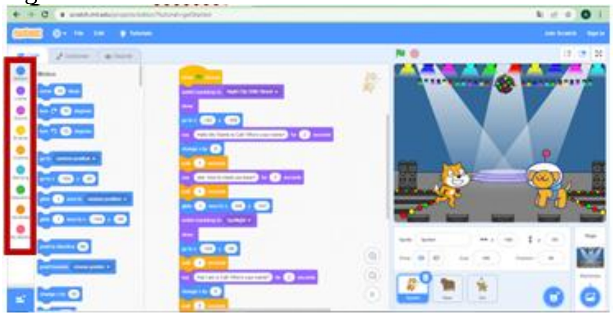
a. *Scratch Interface,* is the interface used in the Scratch programming environment, Scratch Interface includes :

Fig. 1: Scratch GUI (Graphical user interface)



1). *Blocks Palette:* It is a collection of programming blocks accessible to users for creating scripting. These blocks involve various functions, such as moving characters, playing sounds, and setting conditions.

Fig. 2: Block Pallete



Scratch blocks consist of nine different categories, each marked by eight distinct colors, with each color indicating a different function.

Table 1. Programming Block Categories The table below shows programming block categories:

Category		Notes
🟦	Motion	Controls sprite movement such as angle and position
🟪	Looks	Manages the appearance of the sprite
🟪	Sound	Plays audio and effects
🟨	Events	Handles events
🟧	Control	Conditional statements, loops, etc.
🟦	Sensing	Allows sprite to interact with its surrounding environment
🟩	Operators	Mathematical and comparison operators
🟧	Variables	Creates variables and lists
🟥	My Blocks	Enables the creation of functions without return values[10]

2). Script Area: An area where users assemble blocks to create scripting. Users can drag blocks from the palette to this area and arrange them in the desired sequence.
3). Stage: The place where users can visualize the project being executed. It includes characters (sprites) and backgrounds that can be customized by users.

b. Key Features of Scratch
1) *Diverse Programming Blocks*: Scratch provides various programming blocks to perform specific actions, such as changing sprite positions, playing sounds, or setting branching conditions.
2) *Costumes and Backdrops*: Users can alter the appearance of characters (sprites) and backgrounds using costumes and backdrops that can be imported or created. 3) *Sound and Music:* Scratch allows users to easily add sound effects or music to their projects.

c. Learning Process:
1) Drag-and-Drop Programming: Scratch uses a visual programming approach with drag-and-drop methods, making it easily understandable, especially for beginners.
2) Visual Representation: Each programming block has a clear visual representation, making it easy for users to understand programming logic.

d. Basic Programming Concepts Learned
1) *Execution Sequence:* Users understand the execution sequence of the program based on the arrangement of blocks in the script area.
2) *Branching and Looping:* Scratch supports branching (if-else) and looping concepts, enabling users to create programs with more complex logic.
3) *Variables and Data Storage:* Users can use variables to store and manage data in their projects.

e. Creativity and Expression:
1) *Animation and Game Projects:* Scratch allows users to create animation projects, simple games, or other creative interactions.
2) *Design Capability*: With costumes and backdrops, users can showcase their creativity in character and project background design.

f. Teaching Computational Concepts,
Computational Thinking: Scratch aids in the development of computational thinking, involving problem-solving, algorithm design, and logic understanding.

Thus, Scratch provides an educational and immersive environment to help users, especially children, grasp fundamental programming concepts while enabling them to express their creativity.

B. Jean Piaget's Cognitive Development Theory

Cognition literally means "to know". Knowledge can be accepted wisdom or thinking of our mind memories which are formed from the manipulation of various raw input notions through our five senses. This knowledge is applied or used for taking action towards achievement of goals and therefore it is the basis of the cognitive development procedure. Cognitive development process is the formation and composition of thinking processes. Cognitive development includes information processing, intelligence, way of thinking, logic, analysis, language development, and recall.[11]

Piaget's cognitive development theory is one that elucidates how children adapt to and interpret objects and events in their surroundings. It delineates how children learn the characteristics and functions of objects, such as toys, furniture, and food, as well as social entities like themselves, parents, and friends. In Piaget's view (1952), cognitive abilities or development result from the interplay of brain and nervous system development and experiences that assist individuals in adapting to their environment.[12]. The primary shift during this stage is that a child's thinking becomes less reliant on perception and more oriented towards logic. Children exhibit an enhanced ability to comprehend conservation tasks during this period. They can concentrate on various aspects of a stimulus, a skill recognized as 'decentering.' As concrete operations take effect, children grasp the concept of 'reversibility' [13].

Piaget (1964) argued that due to genetic similarities and shared experiences among humans, a certain degree of uniformity in their cognitive development can be expected. Therefore, he formulated four stages of cognitive development that unfold during childhood and adolescence. Each stage is marked by the emergence of new intellectual abilities, enabling individuals to comprehend an increasingly complex world. These stages are:

Table 2. Stages of cognitive development by Jean Piaget

Stages	**Age**	**Abilities**
Sensorimotor	0-2 Years	Indicates the concept of object permanence, which is the psychological ability to understand that an object continues to exist, even if not visible or relevant to current activities. However, at this stage, object permanence is not fully developed.
Preoperational	2-7 Years	Development of the ability to use symbols that represent objects in the surrounding environment. Thinking is still egocentric and centered.
Concrete Operational	7-11 Years	Capable of logical thinking. Can concretely focus on more than one dimension simultaneously and can also connect these dimensions to one another. Reduced egocentrism. Unable to think abstractly.
Formal Operational	11 Years-Adult	Capable of abstract thinking and can analyze problems scientifically, followed by problem-solving.

C. The Concrete Operational Stage

In Piaget's cognitive development theory, the concrete operational stage typically occurs around the age of 7 to 11 years, largely aligning with elementary school levels. This is The third of the four developmental stages, emerges between the ages of seven and twelve. It is characterized by the utilization of adequate logic. Key processes during this stage include:

1. Seriation: the ability to arrange objects based on size, shape, or other characteristics.
2. Classification: the ability to name and identify a series of objects based on their appearance, size, or other characteristics, including the notion that a series of objects can include others in that sequence. Children no longer have the logical limitation of animism (the belief that all objects are living and have feelings).
3. Decentering: children begin considering multiple aspects of a problem to solve it. For example, they no longer perceive a wide but short cup as having less content than a tall but narrow cup.
4. Reversibility: children start understanding that quantities or objects can be changed and then returned to their original state. For instance, they can quickly determine that 4+4 equals 8, and 8-4 equals 4, the previous total.
5. Conservation: understanding that the quantity, length, or number of objects is unrelated to the arrangement or appearance of those objects. For instance, if given two cups of the same size and filled with the same amount of liquid, they will know that pouring the liquid into a differently sized glass will not change the amount.
6. Elimination of Egocentrism: the ability to see things from another person's perspective (even when that person is thinking incorrectly).[12]

III. Research Methods

The research employed in this study is a literature review. The literature review method involves activities related to library data collection, reading, note-taking, and managing research materials [14].

Literature review is a necessary activity in academic research, particularly aiming to develop theoretical and practical aspects. It is conducted by researchers to find a foundation for theory, framework, and formulate preliminary assumptions, also known as research hypotheses. Researchers can then categorize, allocate, organize, and use various literature in their field. Conducting a literature review provides researchers with a broader and deeper understanding of the issues to be investigated.

In this study, the literature review is carried out by intermediary researchers after determining the research topic and formulating the problem statement, before researchers go into the field to collect the necessary data [15]. Data for this research are sourced from textbooks, journals, scholarly articles, and literature reviews containing the concepts under investigation.

Data analysis begins by sequentially considering research findings from the most relevant to sufficiently relevant. The analysis also takes into account the starting year of the research, beginning with the most recent and gradually moving backward to earlier years. Researchers read the abstracts of each study first to assess whether the issues discussed align with what they aim to solve in their research. Subsequently, researchers note important and relevant sections related to the research problem. To avoid plagiarism, researchers also record information sources and include them in the bibliography. If the information comes from others' ideas or research results, researchers make systematic notes, quotations, or information to easily locate them if needed in the future [15].

IV. Result

The research indicates that Scratch is positively associated with the concrete operational stage in Piaget's cognitive development theory, particularly in elementary school children (7-11 years old). Scratch contributes to the development of operational and logical thinking skills during this stage. The intuitive graphic interface of Scratch facilitates the understanding of programming concepts without the need for traditional code writing, considered effective in supporting learning during the concrete operational stage. Furthermore, Scratch, designed with attention to children's learning characteristics, also provides positive support for the development of creativity and problem-solving skills, contributing to improved logical thinking and understanding abstract concepts in a concrete context.

V. Discussion

Utilizing Scratch for learning in elementary school children proves to be highly pertinent to Piaget's cognitive development theory, specifically aligning with the concrete operational stage. This educational approach corresponds with several crucial characteristics associated with this developmental phase. The following elucidates its relevance to key aspects of the concrete operational stage:

Seriation and Classification, Through Scratch, children can develop seriation skills by arranging programming blocks sequentially. They can also use classification concepts to group objects or characters in their projects based on specific attributes.

Seriation and classification, as fundamental cognitive skills, can be developed in children through Scratch, a programming platform. Seriation skills are formed as children arrange programming blocks sequentially, aiding their understanding of the importance of sequencing in programming and laying the foundation for complex problem-solving. Additionally, Scratch facilitates the concept of classification by allowing children to group objects or characters in their projects based on specific attributes, enhancing computational thinking and stimulating creativity.

For example, a child can use Scratch to create a game where the main character must overcome obstacles. In developing seriation skills, they arrange programming blocks to control the character's movements, such as jumping and avoiding obstacles, deepening their understanding of sequencing. Simultaneously, in the context of classification, the child can use Scratch to categorize obstacles based on height or type, creating categories like "High Obstacles" and "Low Obstacles." This process not only builds an enjoyable game but also intuitively and creatively sharpens the child's seriation and classification skills."

Decentering, In Scratch, children are encouraged to consider multiple aspects of a problem. They have to think comprehensively when designing programs, including how characters interact, move, and respond to input.

Decentering in the context of Scratch refers to the cognitive skill of encouraging children to consider multiple aspects of a problem or task. It involves broadening one's perspective beyond a single focal point and thinking comprehensively about various elements of a situation. In the case of Scratch programming, decentering prompts children to go beyond merely coding individual actions or commands for characters; instead, they are encouraged to think holistically about the entire program's design. This includes considering how characters interact with each other, how they move within the program, and how they respond to different inputs or stimuli.

For example, when a child is creating a game in Scratch, decentering would involve thinking about not only how the main character moves but also how it interacts with other characters, objects, or elements in the game. It may involve considering the consequences of specific actions or events within the program and anticipating how the characters should respond. By promoting decentering, Scratch encourages a more comprehensive and interconnected approach to programming, fostering a deeper understanding of the relationships and dynamics within the created digital environment.

Reversibility, Scratch allows children to design programs involving the concept of reversibility, where actions can be undone or repeated. For example, they can create animations where characters move forward and backward.

Reversibility, within the context of Scratch, refers to the ability for children to design programs where actions can be undone or repeated. It involves incorporating the concept that certain processes or movements can be reversed, allowing for flexibility and experimentation in programming. Scratch provides a user-friendly platform that allows children to easily implement the idea of reversibility in their coding projects.

For instance, a child can create animations in Scratch wherein characters move both forward and backward. This ability to reverse actions is not only engaging but also introduces the concept of undoing or repeating steps in a sequence. In practical terms, this means that if a child has programmed a character to move in a certain direction, they can also include instructions for the character to move in the opposite direction, effectively reversing the initial action. This feature not only adds dynamism to the programming experience but also helps children understand the principle of reversibility, a valuable concept in both programming and problem-solving contexts.

Conservation, Children can grasp the concept of conservation through programming. For instance, in creating a game, they can establish rules that the number of points or lives remains consistent even as the appearance of the game changes.

Conservation, when applied to programming in the context of platforms like Scratch, refers to a child's ability to understand that certain properties or quantities can remain constant despite changes in appearance or arrangement. In other words, they can appreciate that certain elements, such as points or lives in a game, can be conserved or maintained even when the visual aspects of the game are altered.

For example, when children create a game using Scratch, they can establish rules in their code that ensure the conservation of points or lives. This means that, regardless of how the game's graphics or scenarios evolve, the fundamental rules governing the scoring system or the number of lives for the player remain unchanged. This introduces children to the concept of conservation in a dynamic and interactive way, fostering an understanding that certain attributes can persist and be preserved despite external transformations. Through programming, children learn not only the technical aspects of coding but also abstract concepts like conservation, which has broader applications in various fields of learning and problem-solving.

Elimination of Egocentrism, In creating Scratch projects, children learn to think from the perspective of users or potential viewers. They need to consider how their projects will be understood by others, eliminating egocentrism in presenting ideas.

Eliminating egocentrism in Scratch projects teaches children to consider user perspectives, fostering an awareness of diverse viewpoints. Encouraged to think about clarity, user interface, and design from the user's standpoint, children develop effective communication skills, ensuring accessibility for a broader audience. For instance, when creating an interactive story, a child eliminates egocentrism by ensuring clear elements and an engaging user experience for those unfamiliar with the storyline. This process not only enhances programming skills but also cultivates empathy and communication beyond an individual perspective.

Thus, Scratch not only provides a platform for learning programming but also supports the development of concrete operational cognitive skills taught in Piaget's theory. Through creativity in designing Scratch projects, children can apply and strengthen their understanding of concepts such as seriation, classification, reversibility, conservation, and the elimination of egocentrism.

VI. FUTURE WORK

Future research should focus on the prolonged impact of elementary school coding education through Scratch on cognitive development. Extended longitudinal studies would provide insights into how early coding skills continue to shape cognitive abilities. Diversifying participants across socio-economic backgrounds and educational settings would enhance external validity. Exploring alternative cognitive development theories, beyond Piagetian perspectives, would offer a more comprehensive understanding. Comparative studies with other coding platforms and the inclusion of control groups can contribute valuable insights. Additionally, investigating the role of teacher proficiency and pedagogical strategies in coding education effectiveness would optimize instructional approaches. These avenues would advance our understanding of the dynamic interplay between coding education, cognitive development, and evolving educational technology.

VII. Limitation

The literature review on "Elementary School Coding Education Effectiveness with Scratch in Alignment with Cognitive Development, Analyzed through Piagetian Lens" faces certain limitations. Firstly, the review's scope may be confined by the availability and currency of existing literature, potentially missing out on recent developments in the field of coding education or alternative cognitive development theories. Additionally, the reliance on the Piagetian lens for analysis may oversimplify the nuanced and multifaceted nature of cognitive development, as contemporary educational practices often draw on a broader range of cognitive theories that were not extensively explored in this review. The generalizability of findings may also be constrained by the heterogeneity in study methodologies, participant characteristics, and educational contexts across the reviewed literature, making it challenging to derive universally applicable conclusions.

Moreover, the potential bias in the selection of literature, stemming from publication bias or language limitations, could influence the comprehensiveness of the review. The absence of a systematic review methodology may further limit the robustness of the synthesis and the establishment of causal relationships between coding education with Scratch and cognitive development. Recognizing and addressing these limitations in future research would contribute to a more thorough and nuanced understanding of the effectiveness of elementary school coding education in alignment with cognitive development.

Conclusion

In conclusion, the critical appraisal of elementary school coding education effectiveness with Scratch, aligned with cognitive development and analyzed through a Piagetian lens, sheds light on the positive correlation between the Scratch platform and the concrete operational stage in Piaget's cognitive development theory. The study highlights the suitability of Scratch for teaching coding concepts to elementary school students, particularly during the concrete operational stage (typically occurring between ages 7 to 11), where children begin to develop operational and logical thinking skills.

The intuitive graphical interface of Scratch facilitates the creation of programming blocks, allowing students to build programs without the need for traditional code writing. The platform's design, tailored to the characteristics of children's learning, supports the enhancement of logical thinking, creativity, and problem-solving skills. However, it is crucial to recognize individual variations in cognitive development and consider the social and learning environment's impact.

While the study underscores the effectiveness of Scratch in elementary school coding education, future research may delve deeper into the social and environmental factors influencing the teaching efficacy. Furthermore, exploring variations in individual learning capacities and how instructional approaches can be customized to meet diverse needs could be a valuable avenue for future investigations in the realm of coding education.

REFERENCES

[1] S. Papadakis, “The Impact of Coding Apps to Support Young Children in Computational Thinking and Computational Fluency . A Literature Review,” vol. 6, no. June, pp. 1–12, 2021, doi: 10.3389/feduc.2021.657895.

[2] B. Jiang and Z. Li, “Effect of Scratch on computational thinking skills of Chinese primary school students,” *J. Comput. Educ.*, vol. 8, no. 4 PG-505–525, pp. 505–525, 2021, doi: 10.1007/s40692-021-00190-z.

[3] R. Vinayakumar, K. P. Soman, and P. Menon, “Fractal Geometry: Enhancing Computational Thinking with MIT Scratch,” *2018 9th Int. Conf. Comput. Commun. Netw. Technol.*, no. Query date: 2023-11-03 23:32:392 cites: PG-, 2018, doi: 10.1109/icccnt.2018.8494172.

[4] B. Lefa, “THE PIAGET THEORY OF COGNITIVE DEVELOPMENT : AN EDUCATIONAL,” no. September, 2014.

[5] N. Bahar, “The effect of scratch on children’s English Language and cognitive development,” 2021.

[6] J. C. Olabe, M. A. Olabe, X. Basogain, and ..., “Programming and robotics with Scratch in primary education,” *Educ. a* ..., no. Query date: 2023-11-03 23:30:3538 cites: https://scholar.google.com/scholar?cites=16363861432599734309&as_sdt=2005&sciodt=2007&hl=en PG-, 2011, [Online]. Available: https://www.researchgate.net/profile/Carlos-Castano-2/publication/268255644_Programming_and_Robotics_with_Scratch_in_Primary_Education/links/5630de6908ae0530378ce280/Programming-and-Robotics-with-Scratch-in-Primary-Education.pdfhttps://www.researchgate.ne NS -

[7] F. B. Flórez, R. Casallas, M. Hernández, A. Reyes, S. Restrepo, and G. Danies, “Changing a Generation’s Way of Thinking: Teaching Computational Thinking Through Programming,” *Rev. Educ. Res.*, vol. 87, no. 4 PG-834–860, pp. 834–860, 2017, doi: 10.3102/0034654317710096.

[8] X. Wang and Y. Yin, “Research on Scratch Programming Teaching Design for the Cultivation of Computational Thinking,” *Adv. J. Eng.*, no. Query date: 2023-11-03 23:32:39 PG-30-36, pp. 30–36, 2023,

doi: 10.55571/aje.2023017.

[9] R. Harimurti, A. Qoiriah, E. Ekohariadi, and M. Munoto, "Implementation of Computational Thinking Concepts in ICT Learning Using Scratch Programming," *Proc. Int. Conf. Indones. Tech. Vocat. Educ. Assoc. (APTEKINDO 2018)*, no. Query date: 2023-11-03 23:32:391 cites: PG-, 2018, doi: 10.2991/aptekindo-18.2018.23.

[10] S. Buddies, "Scratch User Guide: Installing & Getting Started with Scratch," 2023.

[11] G. S. Rehman and K. Ullah, "DEVELOPMENT THEORY : AN IMPLICATION IN LEARNING," vol. 31, no. 1.

[12] A. Thahir, "PSIKOLOGI PERKEMBANGAN".

[13] P. Sanghvi, "Piaget ' s theory of cognitive development : a review," vol. 7, no. 2, pp. 90–96, 2020.

[14] M. Zed, *Metode Penelitian Kepustakaan*. Jakarta: Yayasan Pustaka Obor Indonesia, 2014.

[15] H. Darmadi, *Metode Penelitian Pendidikan*. Bnadung: Alfabeta, 2011.