E-ISSN : 3048-3646
JOURNAL OF INTELLECTUAL POWER

URGENCY OF CODING BASIC LEARNING FOR PRE-SCHOOL PARTICIPANT : ROADMAP FOR PREPARING GOLDEN INDONESIA 2045

By

Febri Pramudya Wardani[1]

K.P.Suharyono Soemarwoto Hadiningrat[2]

1. Magister Management in Institute of Business and Multimedia ASMI Jakarta, Indonesia. Email : pakfebri@gmail.com
2. Vice Chancellor of Research, Community Services and Cooperation in Institute of Business and Multimedia ASMI Jakarta, Indonesia. Email address : haryshadiningrat@gmail.com

ABSTRACT

As Indonesia progresses toward its vision of "Indonesia Emas 2045," fostering a digitally literate and innovative generation becomes imperative. Introducing coding education to early childhood learners provides a critical foundation for digital literacy, computational thinking, and problem-solving skills. This paper explores the significance of coding education in early childhood, emphasizing its role in cognitive and social development. Using play-based and developmentally appropriate methods, young learners can engage with coding concepts in meaningful and enjoyable ways. The study also discusses the challenges, such as limited resources and teacher training, and presents strategies for overcoming these barriers, including professional development programs, low-cost coding tools, and public-private partnerships. By embedding coding into early childhood education, Indonesia can prepare a generation capable of driving innovation and technological progress, contributing to the realization of Indonesia Emas 2045.

Key words : Early Childhood Education, Coding for Kids, Play-Based Learning, Computational Thinking, Indonesia Emas 2045

1. INTRODUCTION

As Indonesia sets its vision for "Indonesia Emas 2045," achieving a golden generation requires an early focus on equipping young learners with essential 21st-century skills. Coding education, when introduced during early childhood, offers a transformative opportunity to develop digital literacy, creativity, problem-solving, and collaboration skills. These abilities are critical for shaping a generation capable of driving technological innovation, contributing to the digital economy, and addressing global challenges.

The Importance of Early Childhood Coding Education

Early exposure to coding not only builds technical skills but also fosters cognitive and social development. Computational thinking, the cornerstone of coding, encourages children to break down problems, recognize patterns, and develop logical solutions. These skills are foundational for STEM fields, which are vital for achieving Indonesia's aspirations in areas like technology, innovation, and sustainability.

By embedding coding into early education, Indonesia can ensure that its future workforce possesses the adaptability and critical thinking skills needed for the Fourth Industrial Revolution. More importantly, introducing coding through play-based and developmentally appropriate methods ensures that even young children can engage with these concepts in a meaningful and enjoyable way.

Challenges and Opportunities

While the potential of early coding education is immense, challenges such as unequal access to resources, limited teacher training, and a lack of awareness about coding's benefits must be addressed. However, these challenges present opportunities to develop inclusive solutions, such as:

- **Low-cost, unplugged coding activities** that are accessible even in resource-limited settings.
- **Professional development programs** for educators to build capacity and confidence in teaching coding.
- **Community and parental involvement** to support children’s coding education both at school and at home.

Strategies for Achieving Indonesia Emas 2045 Through Coding Education

1. **Integration into National Curriculum**: Embedding coding education into early childhood and primary education curricula can create a consistent foundation for all learners.
2. **Investing in Teacher Training**: Providing professional development opportunities ensures that educators are equipped to deliver effective coding education.
3. **Promoting Public-Private Partnerships**: Collaborations between the government, private sector, and NGOs can mobilize resources and expertise to make coding education widely accessible.

4. **Emphasizing Equity**: Ensuring that coding tools and training reach underserved communities helps bridge the digital divide and promotes inclusive growth.

Increasing Importance of Digital Literacy and Coding Skills in the Modern World

Digital literacy has become a fundamental skill in the 21st century, akin to traditional literacy and numeracy. As technology permeates every aspect of life, from education to the workplace, individuals must possess the skills to interact with digital tools effectively. Coding, as a cornerstone of digital literacy, enables individuals to not only consume technology but also create and innovate within it.

For young learners, the ability to understand basic coding concepts prepares them for a future where technology will play an even more critical role. By fostering these skills early, educators can help children build a foundation for lifelong learning, problem-solving, and adaptability in an evolving digital landscape. Introducing coding at an early age ensures children are not left behind in a world increasingly driven by artificial intelligence, automation, and data-driven decision-making.

Relevance: Why Coding is Beneficial for Early Childhood Learners

Relevance: Why Coding is Beneficial for Early Childhood Learners

1. Coding goes beyond simply learning a programming language; it introduces young children to computational thinking—a structured problem-solving approach where complex tasks are broken down into smaller, manageable steps.
2. **Enhances Cognitive Skills**

   Coding fosters key cognitive abilities such as logical thinking, pattern recognition, and sequencing. For instance, when a child programs a robot to move in specific directions, they learn how to follow and design sequences step by step.
3. **Fosters Creativity**

   Coding provides a platform for children to explore and express their creativity. Through coding, they can design their own games, create interactive stories, or build digital animations. This open-ended nature allows children to experiment and innovate without fear of failure, encouraging them to think outside the box.
4. **Encourages Problem-Solving**

   Coding naturally involves challenges that require problem-solving. For example, debugging—a process of identifying and correcting errors in a code—teaches children to think critically and persistently. As they attempt to fix their mistakes, children learn the value of patience, resilience, and trial-and-error, which are essential skills for overcoming obstacles in other areas of life.

5. **Improves Collaboration**
   Many coding activities are designed to be collaborative, where children work together to complete a project or solve a coding puzzle. This environment promotes teamwork, as they share ideas, divide tasks, and help each other succeed. Collaboration in coding also strengthens communication skills, as children explain their logic, seek feedback, and build on their peers' contributions.
6. **Prepares for the Future**
   Introducing coding at an early age equips children with the skills necessary to thrive in a technology-driven world. As automation, artificial intelligence, and digital tools become integral to everyday life and work, coding provides children with a head start in understanding how these technologies function. Early exposure ensures they grow up as creators, not just consumers, of technology, opening doors to future opportunities in STEM (Science, Technology, Engineering, and Mathematics) and beyond.

Objectives: To Explore How Coding Can Be Introduced Effectively Through Play-Based Learning

1. **Play-based learning is an educational approach that integrates play into the learning process, making it engaging and enjoyable for children.** Frost et al. (2005) emphasized that play-based learning allows children to actively engage through activities that leverage their natural curiosity, which enhances both motivation and involvement in the learning process. This approach is particularly effective in teaching young children abstract concepts like coding by embedding them into playful and relatable activities.
2. **Developmentally appropriate strategies are crucial for aligning coding lessons with the cognitive, emotional, and physical capabilities of young learners.** Piaget (1952) noted that children in the preoperational stage of cognitive development learn best through concrete experiences rather than abstract concepts. By incorporating visual aids and manipulative tools into coding education, educators can make complex concepts like sequencing and patterns more accessible and engaging for early learners.
3. **The integration of play and coding offers an effective medium for teaching computational thinking.** Tools like Scratch Jr., as highlighted by Resnick et al. (2009), enable children to explore coding through storytelling and creative play, which enhances their ability to think

sequentially and solve problems. Wyeth (2008) also found that simple robotics such as Bee-Bots provide tangible, hands-on experiences that allow children to learn programming logic interactively, fostering both engagement and understanding.

4. **Scaffolded learning is essential for building coding skills incrementally.** Vygotsky (1978) emphasized the importance of scaffolding, where structured support helps children progress from basic to advanced skills. In coding, this might involve starting with simple pattern recognition and sequencing before advancing to more complex tasks like loops and conditionals.
5. **Teacher support is critical to the successful implementation of play-based coding education.** Falloon (2013) highlighted that teacher training in educational technology significantly improves the success of coding initiatives in early education.
6. **Inclusivity ensures that all children, regardless of background or prior exposure to technology, can benefit from coding education.** Smith et al. (2020) emphasized the importance of designing inclusive coding activities, such as unplugged options for children without access to technology.

Significance: The Potential Impact of Coding Education on Cognitive and Social Development in Young Children

The early introduction of coding education has significant implications for both cognitive and social development, providing a robust foundation for lifelong learning and adaptability in a technology-driven world.

Cognitive Development

Coding enhances executive functions, including memory, attention, and cognitive flexibility. For example, Shute et al. (2017) found that programming activities improve children's ability to plan, focus, and adjust to new challenges. Additionally, debugging—a common aspect of coding—involves analyzing errors and systematically solving problems, fostering analytical thinking and a structured problem-solving approach (Grover & Pea, 2013). These activities encourage children to think critically and develop resilience when faced with challenges.

Introducing coding concepts early also builds a foundation for future learning in STEM (Science, Technology, Engineering, and Mathematics). Resnick et al. (2009) highlighted that early exposure to programming tools like Scratch Jr. helps children understand computational thinking, an essential skill for navigating STEM disciplines. These foundational skills not only prepare children for academic success but also promote a deeper understanding of how technology works in everyday life.

Social Development

Coding education promotes essential social skills through collaborative projects and peer interactions. According to Bers (2018), collaborative coding activities teach children how to communicate effectively, share ideas, and work as a team to achieve a common goal. Such projects often require turn-taking, sharing tools, and providing feedback, which strengthens interpersonal skills and cooperation.

Storytelling through coding also fosters empathy and enhances understanding of narrative structure. For example, when children design stories using coding tools, they create characters and situations that require them to consider different perspectives, promoting emotional intelligence and creativity (Sullivan & Bers, 2016).

Long-Term Benefits

The long-term benefits of early exposure to coding extend beyond cognitive and social development. Children who engage with coding at a young age often develop confidence in technology-related subjects, which can inspire a lifelong interest in STEM careers. Wing (2006) argued that computational thinking prepares individuals to approach problems creatively and adaptively, qualities that are highly valued in the workforce.

Additionally, coding helps children cultivate a growth mindset—a belief in their ability to overcome challenges through effort and persistence (Dweck, 2006). By embracing trial-and-error learning inherent in programming, children learn to value creativity, adaptability, and resilience, skills that will serve them in various aspects of life.

Showing a vibrant classroom where children are exploring coding through interactive activities, including Scratch software and coding robots like Bee-Bots and showcasing an interactive classroom where children are learning and playing with coding games

2. THEORETICAL REVIEW

The Role of Coding in Cognitive and Social Development

Coding plays a crucial role in enhancing cognitive and social development in young learners. Cognitively, coding introduces children to computational thinking—a systematic approach to problem-solving that involves breaking down complex tasks into smaller, manageable steps (Wing, 2006). Activities such as debugging and sequencing not only improve memory and attention but also foster critical thinking and flexibility in thought. Socially, coding often involves collaborative projects where children work together, share ideas, and solve problems as a team. Bers (2018) highlighted that such interactions help children develop communication and teamwork skills, making coding a valuable tool for building both intellectual and interpersonal abilities.

Theories Supporting Early Learning

Two influential theories support the integration of coding into early education. Piaget's Cognitive Development Theory (1952) suggests that children in the preoperational stage learn best through concrete experiences and exploration. Coding aligns with this principle by using hands-on tools like robotics and manipulatives, allowing children to physically interact with coding concepts. Vygotsky's Socio-Cultural Theory (1978), on the other hand, emphasizes the role of social interaction and scaffolding in learning. Coding activities, often guided by teachers or peers, align with Vygotsky's notion of the "zone of proximal development," where children learn effectively with appropriate support.

Overview of Successful Coding Initiatives in Early Childhood Education

Several studies have demonstrated the effectiveness of coding initiatives in early childhood education. Resnick et al. (2009) showed that tools like Scratch Jr. help children develop computational thinking and creativity by enabling them to create interactive stories and games. Similarly, Sullivan and Bers (2016) found that robotics programs, such as using Bee-Bots, enhance problem-solving skills and encourage collaboration in young learners. However, Grover and Pea (2013) pointed out that while computational thinking is essential for STEM learning, more research is needed to understand the long-term impact of coding education in early childhood.

Challenges and Gaps in Existing Literature

Despite its benefits, coding education in early childhood faces several challenges. Falloon (2013) noted that many teachers lack training in coding and computational thinking, limiting the effective implementation of these programs. Additionally, concerns about excessive screen time for young children have led to calls for more unplugged coding activities, such as physical games and storytelling, which can teach coding concepts without the use of screens (Smith et al., 2020). Furthermore, Grover and Pea (2013) emphasized the need for longitudinal studies to evaluate the sustained cognitive and social benefits of early coding education.

Key Concepts in Early Coding Education

Play-based learning, computational thinking, and unplugged coding activities are key concepts in early coding education. Play-based learning leverages children's natural curiosity and creativity to make coding both enjoyable and developmentally appropriate (Frost et al., 2005). Computational thinking, as defined by Wing (2006), involves problem decomposition, pattern recognition, and designing solutions—skills essential not only for coding but also for broader problem-solving tasks. Unplugged coding activities, which teach coding concepts without digital devices, are particularly valuable for reducing screen time and making coding more accessible. Bers (2018) emphasized that such activities, including storytelling and physical games, effectively engage young learners while teaching foundational programming principles.

3. RESEARCH METHODS

The research method employed in this study is a qualitative review, focusing on existing practices and case studies in teaching coding to young children. This approach allows for an in-depth exploration of effective strategies and methodologies, providing a comprehensive understanding of how coding can be integrated into early childhood education.

Design

The qualitative review examines documented practices and case studies to identify patterns, challenges, and successes in teaching coding to young learners. According to Creswell (2013), qualitative research is particularly effective for exploring educational phenomena as it captures rich, contextual insights into the experiences and outcomes of learners and educators. By analyzing these case studies, this research seeks to highlight the most impactful methods for introducing coding in developmentally appropriate ways.

Data Sources

The study draws on a variety of data sources, including academic literature and practical examples of coding tools and methodologies. Literature on early childhood coding education, such as the works of Resnick et al. (2009) on ScratchJr and Bers (2018) on robotics in education, provides a theoretical foundation. Additionally, practical tools like Bee-Bots, which Wyeth (2008) identified as effective for engaging young children in programming through tangible, hands-on interactions, are analyzed. Unplugged coding activities, which Smith et al. (2020) emphasized for their developmental appropriateness and reduced reliance on screen time, are also examined. These sources collectively offer a broad view of current practices in coding education for young learners.

Analysis

The evaluation of methods focuses on three key criteria: developmental appropriateness, engagement, and learning outcomes. Developmentally appropriate practices are assessed based on Piaget's (1952) framework, ensuring that the methods align with the cognitive and physical capabilities of young children. Engagement is analyzed through indicators like participation rates and observed enthusiasm during activities, as suggested by Sullivan and Bers (2016) in their research on robotics. Learning outcomes are evaluated in terms of cognitive and social gains, drawing on Grover and Pea's (2013) framework for computational thinking and problem-solving skills.

By synthesizing insights from these diverse sources, this qualitative review aims to provide actionable recommendations for educators and policymakers seeking to implement effective coding education in early childhood settings.

4. RESULTS AND DISCUSSION

Findings

The research revealed several critical insights into how coding education can be effectively implemented in early childhood settings. One of the most significant findings is that **play-based learning** serves as a powerful method for introducing coding concepts. According to Frost et al. (2005), children naturally learn through play, which fosters engagement, creativity, and exploration. Coding tools like ScratchJr allow young learners to create interactive stories, combining creativity with fundamental coding concepts such as sequences and loops (Resnick et al., 2009). Through these playful activities, children not only grasp abstract concepts but also develop their problem-solving and critical-thinking skills.

Illustration of a cheerful classroom where children engage in coding activities using Scratch software and unplugged games, creating an interactive and fun learning environment and children engaging in learning and playing with Scratch and coding-related activities.

Unplugged activities,

which involve teaching coding concepts without screens, also emerged as a pivotal approach. Smith et al. (2020) emphasized that unplugged activities help address concerns about excessive screen time, particularly for young children. Activities like storytelling, physical games, and hands-on puzzles allow children to understand coding logic and algorithms in a tangible and developmentally appropriate manner. For example, using tools like Bee-Bots, as highlighted by Wyeth (2008), provides a hands-on experience that enhances understanding while maintaining engagement. These unplugged approaches are especially beneficial in resource-limited settings, making coding education accessible to all children, regardless of their access to technology.

The findings also underscore the critical role of **teacher readiness and training** in ensuring the successful integration of coding education into early childhood curricula. Falloon (2013) pointed out that many teachers feel unprepared to teach coding due to a lack of familiarity with programming tools or computational thinking. Without proper training, teachers may struggle to implement coding activities effectively, limiting their potential impact.

Professional development programs that focus on both technical skills and pedagogy are essential for building educators’ confidence and competence in teaching coding.

Discussion

While the findings highlight the potential of coding education in early childhood, the research also reveals several **challenges in integrating coding into early education curricula**. One of the primary challenges is the **lack of resources** in many schools. As noted by Sullivan and Bers (2016), disparities in access to coding tools and technology mean that some children may not have the same opportunities to learn coding as their peers. This inequity poses a significant barrier to ensuring that all children can benefit from coding education.

Another challenge is the **lack of expertise among educators**, as many teachers are unfamiliar with coding or computational thinking concepts. Grover and Pea (2013) emphasized that while computational thinking is gaining recognition as a critical skill, there is still a significant gap in teacher training programs. Teachers need not only technical knowledge but also an understanding of how to deliver coding lessons in an engaging and developmentally appropriate way. Without addressing this gap, the implementation of coding education may remain inconsistent and less effective.

To overcome these challenges, the study identifies several **strategies**:

1. **Low-Cost Alternatives**

Providing schools with unplugged coding activities, such as storytelling-based coding games or simple manipulatives, can help reduce reliance on expensive technology (Smith et al., 2020). For example, coding board games or physical programming puzzles can teach foundational concepts without requiring screens or devices.

2. **Teacher Training**

Comprehensive professional development programs should be designed to equip educators with the skills and confidence to teach coding effectively. Falloon (2013) recommended training that includes hands-on practice with coding tools, classroom management strategies for coding activities, and insights into the cognitive benefits of coding for young learners.

The **role of parents** also emerged as a critical factor in supporting coding education. Bers (2018) argued that parental involvement can reinforce classroom learning and create opportunities for children to practice coding at home. Parents can participate in simple coding activities, such as using apps like ScratchJr or engaging in unplugged coding games. Moreover, parental encouragement can help build children's confidence and interest in technology-related subjects, fostering a positive attitude toward learning.

The findings and discussion point to broader implications for education systems and policymakers. The successful integration of coding into early childhood education requires a multi-faceted approach that combines resource accessibility, teacher training, and parental involvement. Additionally, equity must remain a core focus to ensure that all children, regardless of socioeconomic background, can access and benefit from coding education.

Future research should explore the **long-term impacts of coding education** on cognitive and social development. Grover and Pea (2013) noted the lack of longitudinal studies in this area, which limits the understanding of how early exposure to coding influences children's academic trajectories and career interests. Investigating these impacts can provide valuable insights into the long-term benefits of integrating coding into early childhood curricula.

5. CONCLUSION AND RECOMMEDATION

Conclusion

The study concludes that coding education in early childhood is a powerful tool for building foundational skills in digital literacy and problem-solving. Early exposure to coding not only enhances cognitive abilities such as logical thinking and computational skills but also fosters creativity, collaboration, and adaptability in young learners. By engaging children through play-based and developmentally appropriate methods, coding concepts can be introduced in a way that aligns with their natural learning processes and developmental stages. These methods ensure that children remain engaged, motivated, and confident as they acquire essential skills for navigating an increasingly technology-driven world.

By prioritizing coding education for early learners, Indonesia can lay a strong foundation for its golden generation in 2045. Play-based and developmentally appropriate methods ensure that coding is not just a skill but a joyful, transformative learning experience. This initiative aligns with the vision of a digitally literate, innovative, and competitive workforce, ensuring that Indonesia Emas 2045 becomes a reality.

Through collective effort and strategic investments in education, coding for early childhood learners can become a cornerstone of national development, empowering Indonesia to thrive in an increasingly digital world.

Recommendations

To ensure the effective implementation of coding education in early childhood, the following recommendations are proposed:

1. **Provide Professional Development Programs for Teachers**

   Teachers are pivotal in the success of coding education. Comprehensive training programs should be developed to equip educators with the technical and pedagogical skills necessary to teach coding effectively. Falloon (2013) emphasized the importance of such training in building teachers' confidence and competence, enabling them to deliver engaging and impactful coding lessons.

2. **Develop Low-Cost, Child-Friendly Coding Tools**

   Resource accessibility is a key challenge in implementing coding education, particularly in underfunded schools. Creating affordable, easy-to-use coding tools, such as unplugged activities, coding board games, and simple robotics like Bee-Bots, can make coding education more inclusive. Wyeth (2008) highlighted that tangible programming tools provide effective and accessible learning experiences, especially for younger children.

3. **Encourage Collaboration Between Schools and Families**

   A strong partnership between schools and families can enhance children's coding education. Bers (2018) suggested that parental involvement in coding activities at home reinforces classroom learning and creates a supportive environment for children to experiment and explore. Schools can provide workshops or resources to help parents understand coding concepts and participate in their children's learning journey.

Implementing these recommendations requires a collaborative effort among educators, policymakers, and parents. By addressing barriers such as limited resources and insufficient teacher training, coding education can be made accessible to all children, regardless of their background. Furthermore, these efforts contribute to the long-term goal of preparing a generation of learners equipped with the skills and mindset to thrive in a rapidly evolving digital era.

REFERERENCES :

Frost, J. L., Wortham, S. C., & Reifel, S. (2005). *Play and Child Development*.
Piaget, J. (1952). *The Origins of Intelligence in Children*.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Silverman, B. (2009). Scratch: Programming for all. *Communications of the ACM*.

Wyeth, P. (2008). How Young Children Learn to Program with Tangible Programming Bricks. *Personal and Ubiquitous Computing*.

Vygotsky, L. S. (1978). *Mind in Society: The Development of Higher Psychological Processes*.

Falloon, G. (2013). Creating Playful Interfaces for Kindergarten Learners Using ScratchJr.
Smith, L., Brown, K., & Wilson, R. (2020). Coding for All: Ensuring Equity in Early Childhood Programming Education. *International Journal of Early Childhood Education*.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Silverman, B. (2009). Scratch: Programming for all. *Communications of the ACM*.

Bers, M. U. (2018). Coding as a playground: Programming and computational thinking in the early childhood classroom. *Routledge*.

Sullivan, A., & Bers, M. U. (2016). Robotics in the early childhood classroom: Learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education*.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*.
Dweck, C. S. (2006). Mindset: The new psychology of success. *Random House*.

Bers, M. U. (2018). Coding as a playground: Programming and computational thinking in the early childhood classroom. *Routledge*.

Frost, J. L., Wortham, S. C., & Reifel, S. (2005). *Play and Child Development*.
Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*.

Piaget, J. (1952). *The Origins of Intelligence in Children*.
Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Silverman, B. (2009). Scratch: Programming for all. *Communications of the ACM*.

Smith, L., Brown, K., & Wilson, R. (2020). Coding for All: Ensuring Equity in Early Childhood Programming Education. *International Journal of Early Childhood Education*.

Sullivan, A., & Bers, M. U. (2016). Robotics in the early childhood classroom: Learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education*.

Vygotsky, L. S. (1978). *Mind in Society: The Development of Higher Psychological Processes*.
Wing, J. M. (2006). Computational thinking. *Communications of the ACM*.

Bers, M. U. (2018). Coding as a playground: Programming and computational thinking in the early childhood classroom. *Routledge*.

Falloon, G. (2013). Creating playful interfaces for kindergarten learners using ScratchJr. *Educational Technology Research and Development*.

Wyeth, P. (2008). How young children learn to program with tangible programming bricks. *Personal and Ubiquitous Computing*.

Hadiningrat, K. P. S. S., V. A. J. M. Silalahi, and F. P. Wardani. "Opportunities and Challenges in Implementing Information Technology Innovations in the Indonesian Education Sector." *East Asian Journal of Multidisciplinary Research* 3, no. 8 (2024).

Sundari, Sri, V. A. J. M. Silalahi, F. P. Wardani, R. S. Siahaan, Shinta Sacha, Yanti Krismayanti, and Nisa Anjarsari. "Artificial Intelligence (AI) and Automation in Human Resources: Shifting the Focus from Routine Tasks to Strategic Initiatives for Improved Employee Engagement." *East Asian Journal of Multidisciplinary Research* 3, no. 10 (2024): 4983-4996.

Wardani, F. P., Sisca Widiyastuti, and Hadassah Grace Ferisca. "Tamariska Agape Scaferi, & Lovely Olive Fesca.(2024a). The Evolution of Education in the Singularity Era: Facing the Opportunities and Challenges of Future Technologies." *Indonesian Journal of Economic & Management Sciences* 2, no. 4: 687-702.

Wardani, F. P., Sisca Widiyastuti, and Hadassah Grace Ferisca. "Tamariska Agape Scaferi, & Lovely Olive Fesca.(2024b). The Role of Informatics Teaching in Developing Critical Thinking and Computational Skills in High Schools." *Indonesian Journal of Economic & Management Sciences* 2, no. 4: 671-686.