MOTIVECTION
Journal of Mechanical Electrical and Industrial Engineering

Volume: 7, Number: 2, 2025
ISSN: 2655 – 7215 [printed] ; 2685-2098 [online]
DOI: https://doi.org/10.46574/motivection.v7i2.445

IMEIRS

Implementation of Finite State Automata for 6-Axis Robot in the Screwing Process

Risfendra[1], Rapis Nursalam[1]*, Eko Purnama[2], Defri[3]

ABSTRACT

The screwing process in industrial product assembly demands high levels of precision, consistency, and time efficiency. This study presents the implementation of Finite State Automata (FSA) using the Nondeterministic Mealy Machine model to control a 6-axis EPSON robot. The system integrates a programmable logic controller (PLC), human-machine interface (HMI), sensors, pneumatic actuators, and a screwdriver tool. A state diagram is developed to define robot behavior, where each transition is triggered by real-time input signals such as sensor feedback. The robot follows a logical sequence: from retrieving tools and screws to performing the screwing operation. Experimental results show that the FSA-based system enables stable and repeatable performance, reduces error rates, and enhances operational safety through the use of predefined safe points. This approach simplifies programming logic, improves monitoring clarity, and offers scalability for integration in various industrial applications that involve repetitive, precision-based tasks.

Keywords
Finite State Automata, Mealy Machine, Screwing, Robot 6-Axis, Programmable Logic Controller, Screwdriver

[1] Department of Electrical Engineering, Faculty of Engineering, Universitas Negeri Padang
Jl. Prof. Dr. Hamka, Kampus UNP Air Tawar, Padang, Sumatera Barat
[2] PT UNPMES Inovasi Indonesia
Gedung Business Center Kampus UNP, Air Tawar, Padang
[3] PT MES Teknologi Indonesia
Kawasan Industri Tunas Bizpark Blok D No 15, Batam Centre, Batam
* Corresponding Author: risfendra@ft.unp.ac.id

Submitted : April 22, 2025. Accepted : June 23, 2025. Published : July 16, 2025

INTRODUCTION

The swift progress of science and technology has resulted in the expansion of information systems and the extensive implementation of automated industrial processes [1]. The advent of the Fourth Industrial Revolution (Industry 4.0) has catalyzed technical innovation that radically alters human interaction with machines and production settings [2][3]. In this environment, industrial robots have become indispensable for executing intricate and repetitive operations with exceptional precision and efficiency [4][5]. A task that can greatly benefit from robotic automation is the screwing operation, which can be performed with a 6-axis robot [6][7].

The screwing process is essential in the manufacturing sector, especially in the assembly of components necessitating precise and uniform torque application. To attain ideal screwing performance, a control system must effectively regulate the robot's sequence and operational conditions in a systematic manner [8][9]. A promising method for doing this is the implementation of Finite State Automata (FSA), which organizes the robot's activities into distinct states, each aligned with specified operational circumstances [10]. In previous research, Makrini et al. introduced a Hierarchical Finite State Machine (HFSM) for task allocation in human-robot collaborative assembly. HFSMs enable modular and scalable task modeling, encompassing primary tasks, subtasks, and elementary actions, hence facilitating flexible transitions and enhancing system flexibility [11]. This study concentrates on the direct implementation of Finite State Automata on a 6-axis robotic system, specifically regarding screwing operations.

Finite State Automata is a formal computational model characterized by a finite number of states, where transitions between states occur in response to specific input symbols, as defined by a transition function [12]. When applied to robotic control, FSA offers the advantage of explicitly mapping each operational step to a discrete state, improving the traceability and reliability of the system while minimizing the likelihood of human error [13]. There are two main types of FSA: Deterministic Finite Automata (DFA) and Nondeterministic Finite Automata (NFA) [14][15]. Within the NFA category, two widely used models are the Mealy Machine and the Moore Machine [16]. FSA can be formally represented as a 5-tuple (Q, Σ, δ, S, F), where Q denotes the set of states, Σ the input symbols, δ the transition function, S the initial state, and F the set of final states [12][14].

This study implements Finite State Automata using the Mealy Machine model on a 6-axis robot for automating the screwing process. By structuring the robot's movements as a series of defined states, each triggered by input from various sensors, the system facilitates clearer programming logic and operational transparency [10][12]. This research aims to explore whether applying NFA to robotic screwing processes can simplify operator oversight and enhance the clarity and reliability of robot control sequences [17].

METHOD

In this study, a system is designed to implement Finite State Automata on a 6-axis robot in the screwing process. FSA will describe the movement of the robot using a state diagram [18][19]. By using the FSA model of Mealy Machine, the output will depend on the input and the current state conditions [20]. The stages of designing this state diagram are identifying input, output, and state, designing the state diagram, transition function table, and output transition table in the screwing process [21].

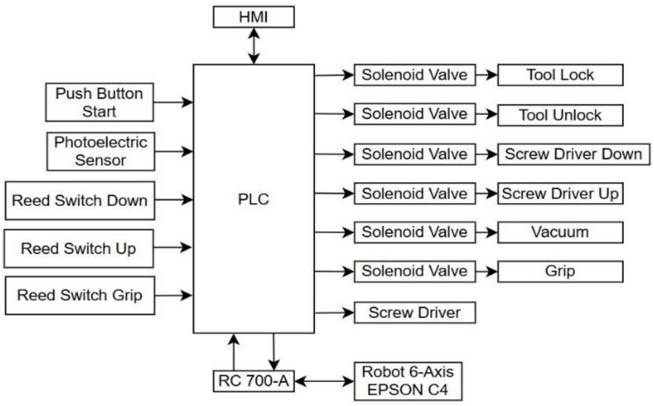

Figure 1. Block diagram

A block diagram is essential for the design and creation of a system. Figure 1 shows a block diagram illustrating the configuration of a system comprising input devices, processing units, and output devices. The utilized input devices include a push button start, a photoelectric sensor, and a reed switch sensor. The employed output devices include a solenoid valve for the pneumatic cylinder, a solenoid valve for vacuum control, a solenoid valve for tool locking and unlocking, a solenoid valve for screwdriver elevation and depression, and a solenoid valve for object gripping and screwdriver operation. The process device employs Programmable Logic Control (PLC) that interfaces with the Human Machine Interface (HMI) through Ethernet, facilitating input and output integration with a 6-axis EPSON C4 robot via the RC700-A controller.

The working principle of the system involves utilizing a 6-axis robot to perform the screwing process on a target object, with the assistance of an integrated screwdriver tool. The attachment of the screwdriver enables the robot to carry out the screwing operation automatically, without manual intervention. In operation, the robot retrieves a screw and fastens it onto the designated object by rotating the screwdriver to apply the required torque. To achieve precise movement and positioning, the robot must be programmed using RC+7 software. Initially, the robot starts from a predefined "Safe Point Tool" position. It then moves to the screwdriver station to pick up the tool, returns past the same safe point, and proceeds to the screw feeder to retrieve a screw. From there, the robot passes through the "Safe Point Screwing" before reaching the screwing station to perform the fastening task. Upon completion, the robot returns to the Safe Point Tool to enter standby mode. The toolpath is illustrated in Figure 2.

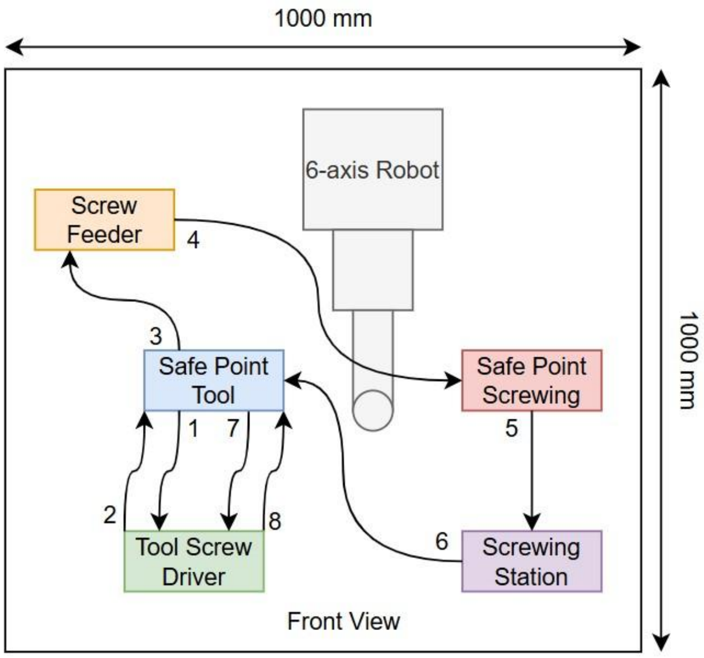

Figure 2. Movement sketch

Flowchart of the Screwing Process

The flowchart depicts the sequence of logical operations in the screwing process, including input signals, output actions, and interrelated robotic movements [22]. Each step in the flowchart represents a distinct phase in the process, encompassing the robot's initial standby position, screw acquisition, tightening operation, and return to the initial position in preparation for the subsequent cycle. State transitions are activated by real-time input conditions, including sensor signals, status updates from the programmable logic controller

(PLC), or feedback from the human-machine interface (HMI). These inputs guarantee that the robot adheres to a deterministic and traceable procedure. Figure 3 illustrates the comprehensive flowchart depicting the automated screwing procedure with the 6-axis robotic system.

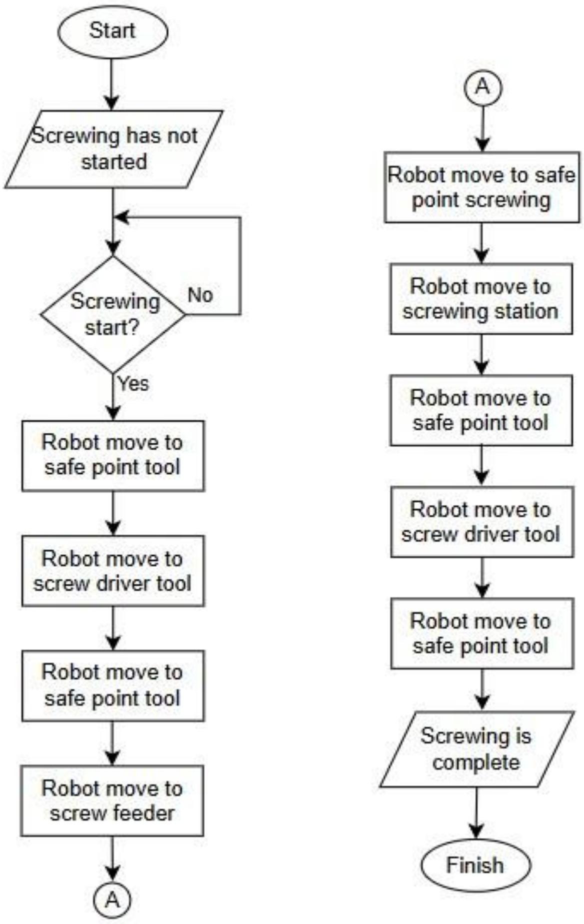

Figure 3. Flowchart

RESULT AND DISCUSSION

In this study, the technique used is to design a state diagram as a model of system behavior. The state diagram is used as a basis for defining tuples and state diagrams. The type of state diagram used is Mealy Machine with the concept of Nondeterministic Finite Automata (NFA), because Mealy Machine is able to produce output every time a state transition occurs, which depends on the input and previous state. The practical implications of this system are significant, especially for industries that rely on repetitive and precision-based tasks such as electronics assembly, automotive component installation, or even consumer product packaging.

Finite State Automata with Mealy Machines

The movement of the robot from one state to another is determined by the input received and the applicable transition function. The Finite State Automata approach with the Mealy Machine model on a 6-axis robot in the screw assembly process consists of a number of important components, namely the state set, initial state, final state, alphabet, and transition function. In its implementation, the system receives input, then runs the process based on the predetermined logic to produce output at the final stage. In FSA, the system is represented by a tuple consisting of five elements, namely with the formula M = (Q, Σ, δ, S, F). The description of each element is as follows: state sets (Q), input symbol sets (Σ), transition function (δ), initial state (S), final state sets (F). So it can be defined as follows:

Q: {S0, S1, S2, S3, S4, S5}
Σ: {A, B, C, D, E, F, G, H, I, J}
δ: transition function
S: {S0}
F: {S5}

Table 1 illustrates the states experienced by a 6-axis robot throughout the automated screw assembling procedure. The initial state S0 serves as the system's starting point, where the robot remains idle, awaiting a signal to commence operations. Upon receiving the input, the robot proceeds to S1 to access the safe point tool, designated as a secure location for acquiring or substituting work tools. The robot proceeds to S2 to retrieve the screwdriver tool. Upon successful acquisition of the tool, the robot proceeds to S3 to the screw feeder location to get the screw. Upon the completion of the screw, the robot retreats to S4, designated as the secure position for screwing, serving as a safe transitional point prior to the screwing process. Ultimately, the robot transitions to state S5, designated as the Screwing Station, where the screwing operation is executed.

Table 1. State sets

Code	Description
S0	Initial state
S1	Robot moves to Safe Point Tool
S2	Robot moves to Screwdriver Tool
S3	Robot moves to Screw Feeder
S4	Robot moves to Safe Point Screwing
S5	Robot moves to Screwing Station

Table 2 summarizes the input and output components of the automated screw assembly system using a 6-axis robot. Within the Mealy Machine model, both input and output play a central role in determining state transitions. Inputs are derived from several control signals, including the start push button and various sensors. These sensors include photoelectric sensors used to detect object presence, as well as reed switches that monitor the position and movement of pneumatic cylinders and the robot arm based on programmed logic. The outputs represent system responses that are executed as a result of specific state transitions. These include actions such as tool locking and unlocking, activation of the vacuum mechanism, vertical motion of the screwdriver (up/down), object gripping, and initiation of the screwing process itself. The clear mapping between input signals and output actions enables precise, repeatable control of the robotic screwing sequence.

Table 2. Input and output sets

Code	Input Description	Code	Output Description
A	Push Button Start	0	Output Off
B	Photoelectric Sensor	1	Finish Screwing
C	Reed Switch Down	a	Tool Lock
D	Reed Switch Up	b	Tool Unlock
E	Reed Switch Grip	c	Screwdriver Down

Code	Input Description	Code	Output Description
F	Robot arrives at Safe Point Tool	d	Screwdriver Up
G	Robot arrives at Screwdriver Tool	e	Vacuum
H	Robot arrives at Screw Feeder	f	Grip
I	Robot arrives at Safe Point Screwing	g	Screwdriver
J	Robot arrives at Screwing Station		

State Diagram

Based on the state set table and the input and output set table, a state diagram of the Mealy Machine model can be created. The input and output transitions in screwdriving with this 6-axis robot function to regulate the movement of the robot from one state to another, based on the input received and producing the appropriate output as shown in Figure 4.

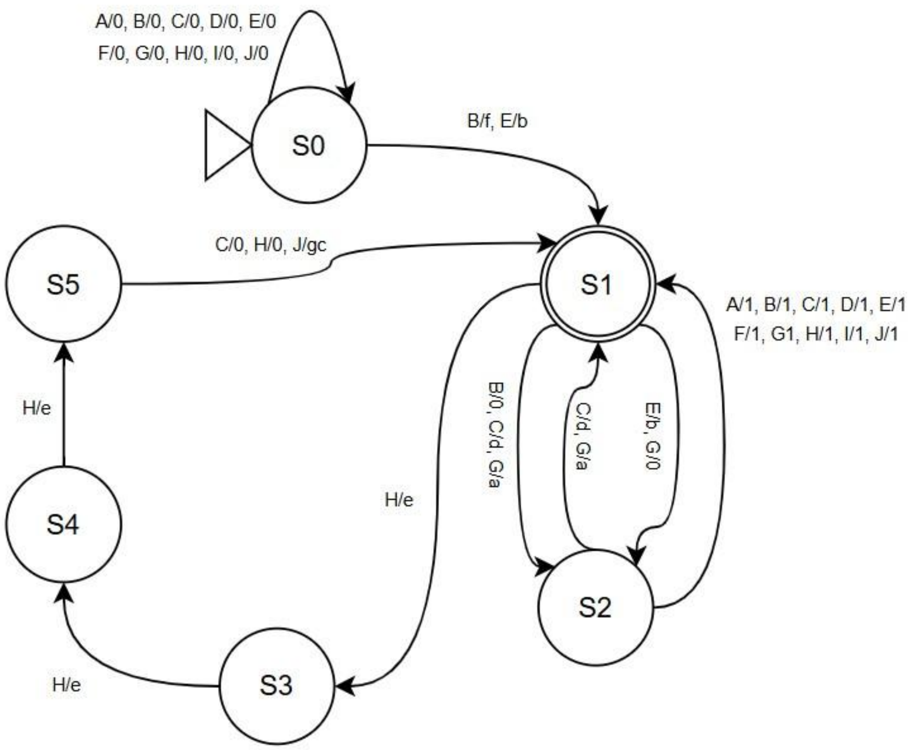

Figure 4. State diagram design

In state S0, the robot awaits input from the push button labeled A. Upon receiving this input, the robot transitions to state S1, resulting in an output of off (0). In state S1, if the photoelectric sensor (B) identifies an object, the robot transitions to state S2. The generated output consists of tool unlock (b) and grip (f). Upon the robot's arrival at the designated safe point tool (F), the resultant output is the elevation of the screwdriver (d). In state S2, activation of the reed switch down sensor (C) prompts the robot to revert to state S1, with the output being tool lock (a) for tool installation. Upon arriving at state S3, if the robot accesses the screw feeder (H), it transitions to state S4 with the output designated as vacuum (e). Upon attaining state S4, the robot proceeds to the screwing station (S5) to execute the screwing operation. The generated output activates the screwdriver (g) and the screwdriver down (c). The final state is S5; upon completion of the screwing operation (B), the robot reverts to state S1, with the output being tool unlock (b), signifying the completion of the screwing task.

Table 3. Transition function

δ	A	B	C	D	E	F	G	H	I	J
S0	S1	-	-	-	-	-	-	-	-	-
S1	-	S2	-	-	S2	S2, S3	-	-	-	-
S2	-	-	S1	-	-	-	S1	-	-	-
S3	-	-	-	S4	-	-	-	S4	-	-
S4	-	-	-	-	-	-	-	-	S5	-
S5	-	-	S1	-	-	-	-	-	-	S1

Table 3 illustrates the transition function δ of the system, determined by the combination of the present state and the received inputs. For instance, from state S0, upon receiving input A (press button start), the system transitions to state S1, whereas other inputs do not initiate a transition. From state S1, inputs B, E, and F will transition the robot to either S2 or S3 contingent upon the sensor circumstances. Similarly, in state S2, inputs C and G will return the robot to S1. The transition from S3 to S4 is activated by input D or H, whereas the transfer from S4 to S5 is initiated by input I. Ultimately, from state S5, inputs C or J will return the system to S1.

The output transition of the screwdriving process, as represented by the Mealy Machine model, illustrates the output response resulting from specific combinations of states and inputs. In state S0, all inputs yield an output of 0 (output off), signifying that the system remains inactive. In state S1, inputs B, E, F, and G yield outputs f (grip), b (tool unlock), and a (tool lock), accompanied by signal 1, which signifies the completion of the screwdriving process. In state S2, inputs C, E, and G activate outputs d (screwdriver raised), b (tool unlocked), and a (tool locked), based on the robot's actions. States S3 and S4 generate outputs, specifically a (tool lock) and e (vacuum), upon detecting inputs H or G. In state S5, the output from input J indicates the screwdriving operation with outputs g and c (screwdriver and screwdriver down) as seen in Table 4.

Table 4. Output Transition

δ	A	B	C	D	E	F	G	H	I	J
S0	0	0	0	0	0	0	0	0	0	0
S1	0, 1	f, 1	0, 1	0, 1	b, 0, 1	0, 1	0, a, 1	0, 1	0, 1	0, 1
S2	0	f	d	0	0, b	0	a, 0	0	0	0
S3	0	f	d	0	0	0	a	e	0	0
S4	0	f	d	0	0	0	a	e	0	0
S5	0	f	0	0	0	0	a	e	0	g, c

Screwing Process Testing using 6-Axis Robot

The 6-axis robot was pre-programmed using RC+7 software and integrated with a programmable logic controller (PLC) to coordinate movement in accordance with the screwing process, as illustrated in Figure 1. The robot's motion path is depicted in the schematic shown in Figure 2, which highlights eight directional movements indicated by arrows and five key movement points throughout the operation. This configuration ensures that the robot follows a structured and repeatable trajectory, moving from the initial standby position through each operational stage—tool pickup, screw acquisition, screwing, and return to standby—based on predefined safe points. The combined use of PLC, RC+7 software, and real-time sensor feedback enables accurate execution of each screwing cycle.

Figure 5. Robot movement to screwdriver tool

The robot first assumes a standby position, which functions as its default condition upon system activation. Upon pressing the push button, the robot proceeds to the specified Safe Point Tool. Upon attaining this position, it initiates the tool unlock mechanism to disengage the lock on the tool changer. Concurrently, if the photoelectric sensor identifies an object at the screwing station, the gripping mechanism activates to immobilize the object. Simultaneously, the robot advances to the screwdriver station to obtain the tool. Upon arrival, the tool unlock is disabled, and the tool lock is enabled to secure the screwdriver. The screwdriver is in an operational state, ready for the screwing task. The sequence of actions is depicted in Figure 5.

Figure 6. Robot movement to screw feeder

After that, the robot will move towards the screw feeder through the safe point tool to take the available screw and then activate the vacuum to suck the screw into the hole provided in the tool as seen in Figure 6.

Figure 7. Robot movement to screwing station

The robot advances to the screwing station by traversing the specified "Safe Point Screwing" to reduce the likelihood of collisions with adjacent components. This safety mechanism is included into the Finite State Automata (FSA)-based control system, ensuring that each transition between states incorporates a predetermined safe point prior to achieving the operational aim. The operation at the screwing station commences with the activation of the screwdriver, which rotates and subsequently descends to secure the screw, as depicted in Figure 7. Upon finishing the screwing duty, the robot withdraws the screwdriver and restores it to its allocated holder. The robot subsequently returns to the Safe Point Tool, entering standby mode in anticipation of the next screwing cycle.

CONCLUSION AND RECOMMENDATION

Conclusion

This study shows that the application of Finite State Automata (FSA) with the Mealy Machine model on a 6-axis robot in the screwing process can regulate the movement sequence and system response. Through the design of state sets, inputs, outputs, and transition functions, the system is able to respond to each condition in real-time and run the screwing process automatically. The integration between PLC and 6axis robot successfully performs robot movements according to the logical sequence in the state diagram. The system also takes into account safe points in robot movement to avoid collisions and ensure work safety. The implementation results show that the system can work on input, and is able to complete the screwing process repeatedly.

Recommendation

Suggestions for the future so that the implementation of finite state automata is not only on 6-axis robots but can also be developed for various other types of robots, such as SCARA, delta robots, or even collaborative robotic arms (cobots) used in light industry. In addition, this approach also has the potential to be applied to other automated machines in production lines or smart devices in everyday life that require logic-based process control such as vending machines, games, and integration with other technologies such as vision sensors, IoT, and machine learning.

REFERENCES

[1] A. Tahar, P. B. Setiadi, and S. Rahayu, "Strategi Pengembangan Sumber Daya Manusia dalam Menghadapi Era Revolusi Industri 4.0 Menuju Era Society 5.0," *J. Pendidik. Tambusai*, vol. 6, no. 2, pp. 12380–12381, 2022, doi: 10.31004/jptam.v6i2.4428.

[2] L. H. Adha, Z. Asyhadie, and R. Kusuma, "Digitalisasi Industri Dan Pengaruhnya Terhadap Ketenagakerjaan Dan Hubungan Kerja Di Indonesia," *J. Kompil. Huk.*, vol. 5, no. 2, pp. 267–205, Dec. 2020, doi: 10.29303/jkh.v5i2.49.

[3] Sukardi, H. Setyawan, Risfendra, Usmeldi, and D. T. P. Yanto, "Effectiveness of Robotic Technology in Vocational Education: A Meta-Analysis," *Int. J. Inf. Educ. Technol.*, vol. 14, no. 4, pp. 521–532, 2024, doi: 10.18178/ijiet.2024.14.4.2073.

[4] A. Wiyono, I. Rokhim, and A. Sumardi, "Robot Lengan Untuk Proses Perakitan Miniatur Body Mobil Sebagai Media Pembelajaran Robotika," *Semin. Nas. Teknol. Dan Ris. Terap.*, vol. 1, no. September, pp. 65–72, 2019.

[5] G. Sandhika Putra and Risfendra, "Sistem kontrol lengan robot berbasis Arduino dengan aplikasi GUI," *JTEIN J. Tek. Elektro Indones.*, vol. 3, no. 2, pp. 387–398, 2022, doi: 10.24036/jtein.v3i2.270.

[6] M. Sága, V. Bulej, N. Čuboňova, I. Kuric, I. Virgala, and M. Eberth, "Case study: Performance analysis and development of robotized screwing application with integrated vision sensing system for automotive industry," *Int. J. Adv. Robot. Syst.*, vol. 17, no. 3, pp. 1–23, 2020, doi: 10.1177/1729881420923997.

[7] R. Naboni, A. Kunic, A. Kramberger, and C. Schlette, "Design, simulation and robotic assembly of reversible timber structures," *Constr. Robot.*, vol. 5, no. 1, pp. 13–22, 2021, doi: 10.1007/s41693-020-00052-7.

[8] K. Anja, R. Naboni, A. Kramberger, and C. Schlette, "Design and assembly automation of the Robotic Reversible Timber," 2021, doi: 10.1016/j.autcon.2020.103531.

[9] M. Siska, I. Siregar, A. Saputra, M. Juliana, and M. T. Afifudin, "Kecerdasan Buatan dan Big Data dalam Industri Manufaktur: Sebuah Tinjauan Sistematis," *Nusant. Technol. Eng. Rev.*, vol. 1, no. 1, pp. 41–53, 2023, doi: 10.55732/nter.v1i1.1119.

[10] R. Balogh and D. Obdržálek, "Using Finite State Machines in Introductory Robotics," *Adv. Intell. Syst. Comput.*, vol. 829, no. April, pp. 85–91, 2019, doi: 10.1007/978-3-319-97085-1_9.

[11] I. El Makrini, M. Omidi, F. Fusaro, E. Lamon, A. Ajoudani, and B. Vandcrborght, "A Hierarchical Finite-State Machine-Based Task Allocation Framework for Human-Robot Collaborative Assembly Tasks," *IEEE Int. Conf. Intell. Robot. Syst.*, vol. 2022-October, pp. 10238–10244, 2022, doi: 10.1109/IROS47612.2022.9981618.

[12] A. S. Maulana, "Implementasi Finite State Automata (FSA) dengan Simulasi Vending Machine pada Aplikasi Android," *J. Edukasi Elektro*, vol. 3, no. 2, pp. 110–120, 2020, doi: 10.21831/jee.v3i2.28332.

[13] M. F. Rahadian, A. Suyatno, and S. Maharani, "Penerapan Metode Finite State Machine Pada Game 'The Relationship,'" *Inform. Mulawarman J. Ilm. Ilmu Komput.*, vol. 11, no. 1, p. 14, 2016, doi: 10.30872/jim.v11i1.198.

[14] R. Agastya Nugraha, Yanto, A. Mulyani, and W. Gata, "IJCIT (Indonesian Journal on Computer and Information Technology) Desain Vending Machine Rujak Buah Dengan Finite State Automata," *IJCIT (Indonesian J. Comput. Inf. Technol.*, vol. 5, no. 2, pp. 198–207, 2020, doi: 10.31294/ijcit.v5i2.8169.

[15] S. Prithi and S. Sumathi, "A technical research survey on bio-inspired intelligent optimization grouping algorithms for finite state automata in intrusion detection system," *Int. J. Eng. Sci. Technol.*, vol. 16, no. 2, pp. 48–67, 2024, doi: 10.4314/ijest.v16i2.6.

[16] A. D. Marzooqa, F. S. Yumna, F. L. Hafizh, and T. Setiadi, "Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi) Implementasi Vending Machine Ramen dengan Menggunakan," vol. 9, no. September, pp. 896–905, 2025.
[17] A. P. Giovani, F. Zamachsari, E. D. Agustono, M. I. Prasetya, and W. Gata, "Implementasi Finite State Automata Dalam Siklus Pembelajaran Magister Ilmu Komputer STMIK Nusa Mandiri," *CESS (Journal Comput. Eng. Syst. Sci.*, vol. 5, no. 2, p. 221, 2020, doi: 10.24114/cess.v5i2.16696.
[18] M. H. Jihad, T. Andromeda, and S. Sudjadi, "Aplikasi Pengendalian Arm Manipulator 4 Dof Menggunakan Algoritma Finite State Machine Sebagai Pemindah Sayuran Pascapanen Terozonisasi," *Transient*, vol. 7, no. 3, p. 782, 2019, doi: 10.14710/transient.7.3.782-788.
[19] Y. Zhan and M. S. Hsiao, "Formal validation for natural language programming using hierarchical finite State Automata," *ICAART 2021 - Proc. 13th Int. Conf. Agents Artif. Intell.*, vol. 1, no. Icaart, pp. 506–518, 2021, doi: 10.5220/0010388905060515.
[20] E. Engler, M. Baldauf, F. S. Torres, and S. Brusch, "Finite state machine modelling to facilitate the resilience of infrastructures: Reflections," *Infrastructures*, vol. 5, no. 3, 2020, doi: 10.3390/infrastructures5030024.
[21] B. Sutjiono and I. Nyoman, "Improvement Proses Screwing pada Lini Kaleng Kopi di PT Sinar Djaja Can," vol. 6, no. 1, pp. 15–20, 2018.
[22] N. Khesya, "Mengenal Flowchart Dan Pseudocode Dalam Algoritma Dan Pemrograman," *Definitions*, 2020, doi: 10.32388/tf77dy.

This page is intentionally left blank