



# **Implementation of Finite State Automata in an Amusement Park Automatic Ticket Selling Machine**

Fergie Joanda Kaunang<sup>1</sup>, Jacqueline Waworundeng<sup>2</sup>

<sup>1</sup> Fakultas Teknologi Informasi, Universitas Advent Indonesia

<sup>2</sup> Fakultas Ilmu Komputer, Universitas Klabat, Manado

*fergie.kaunang@unai.edu*

## **ABSTRACT**

**Introduction:** Amusement Park is a place that provides various attractions for entertainment purpose. People can enjoy games, rides such as roller coaster rides, merry-go-round, etc. Over the time, technology has grown. Many things that are usually done manually by humans are now being replaced by computers. With an automated ticket selling machine, the process of buying the ticket of an amusement park becomes easier for the user. Automata theory is a theoretical branch that has not been widely known to many yet plays essential role in the field of computer science. The main concept of automata theory itself is how to make machines works automatically. Therefore, this study aims to show the implementation of the concept of Automata theory in an amusement park automatic ticket selling machine. **Methods:** The method used to develop the application is formal methods known as Finite State Automata. We also used Sequential Linear method to design the application. Formal method is a mathematical modelling that links the production, development, and verification of software and hardware. In this application, the concept of Finite State Automata was applied to recognize and then capture the pattern on the process of ticket selling machine. **Results:** The result of this study shows that Finite State Automata can become one of the alternatives to design an automatic ticket selling machine for an amusement park by reading each input given by the user and then converts it to the language know by the Finite State Automata. **Discussion:** In this application, the payment method is by inserting a certain amount of cash into the machine and the machine will proceed to the next process. However, for future development another payment method can be added. For example, instead of inserting cash, user can choose to pay using debit as well as credit card.

**Keywords:** Automata Theory, Formal Methods, finite state automata, amusement park, automatic ticket selling machine.

## **INTRODUCTION**

The development of increasingly advanced technology, humans are now increasingly facilitated in doing their work. With this progress, the work carried out is more effective and efficient. Nowadays, lots of automation system implementation can be seen around us.

Automation system itself refers to a condition where devices, tools, as well as process are done without any manual intervention. Amusement Park is a place that provides various attractions for entertainment purpose. People can enjoy games, rides such as roller coaster rides, merry-go-round, etc. Over the time, technology has grown. Many things that are usually done manually by humans are now being replaced by computers. With an automated ticket selling machine, the process of buying the ticket of an amusement park becomes easier for the user. Automata theory is a theoretical branch that has not been widely known to many yet plays essential role in the field of computer science. The main concept of automata theory itself is how to make machines works automatically. This concept has been used in many aspects such as in completing the study card plan for college students, application to introduce the alphabets to toddlers, etc. Therefore, this study aims to show the implementation of the concept of Automata theory in an amusement park automatic ticket selling machine.

## **LITERATURE REVIEW**

Finite state automata is an abstract machine in the form of a mathematical model system with discrete inputs and outputs that can recognize the simplest language (regular language) and can be implemented in a real way. Finite State Automata is a mathematical model that receives input and output that has a finite number of states and can move from one state to another based on input and transition functions (Putra, 2017). Finite state machine (FSM) or finite state automata is a part of formal method which is a control system design methodology that describes the behavior or working principle of the system using the following three things: State, event and action (Rahadian, Suyatno, & Maharani, 2017). At one time in a significant period of time, the system will be in an active state. Meaning that the system can be in any active configurations. The system can switch or transition to another state if it gets input or certain events, both originating from external devices or components in the system itself. This state transition is generally also accompanied by actions taken by the system when responding to the input that occurs. These actions can be simple actions or involve a relatively complex set of processes (Setiawan, 2006). The working principle of Finite State Automata are as follows: (1) Receive input strings, (2) Read (scan substring) initial character with control depends on the initial state, (3) With control and initial character that has been read, the state will move to a new state, (4) The process continues until all strings are absorbed or read, (5) If the last state are in the set of predetermined final states, then the string is received or recognized by the Finite State Automata. If not, then the string is rejected or not recognized by finite state machine (Utdirartatmo, 2005). Formally the FSA is stated in 5 tuples, which are (Hopcroft, 2008):  $M = (Q, \Sigma, \delta, S, F)$  where:

$Q$  = The set of states

- $\Sigma$  = The set of input symbol
- $\delta$  = Transition Function
- S = Initial state/initial position
- F = Final state/accepting state

Transition diagram has been used to represent and visualize finite state automata as shown in Figure 1 (Linz, 2006). The circle represents the state, the label on the circle is the name of the state, the arrow indicates a transition or displacement state, a circle that is preceded by a bow without a label indicating state initial, and finally the double circle represents the final state.

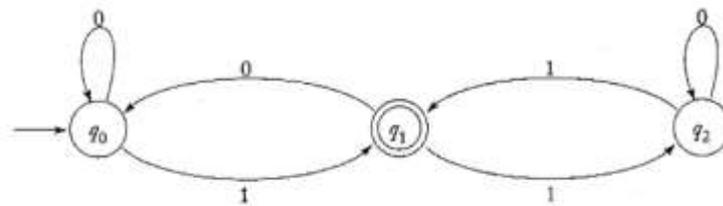


Figure 1: **Example of transition diagram for finite state automata**

An example of finite state system can be seen on the Farmer Crosses River problem. A farmer or a man (M) with a goat (G), wolf (W) and a box of cabbage (C) wants to cross a river. They are initially on the left bank of the river. There is boat that is large enough to carry two items which is the farmer and only one of the other three items. The farmer and his troupe wish to cross to the right side of the river, and he can bring each across, one at a time. However, if the farmer leaves the wolf and the goat alone on one shore, the goat will be eaten by the wolf. If the goat and the box of cabbage are left alone on the shore, the goat will eat the cabbage. The problem is how can the farmer and his entourage successfully cross the river without the goat or cabbage being eaten? Figure 1 (Hopcroft, 2008) shows the solution of the problem using finite state system.

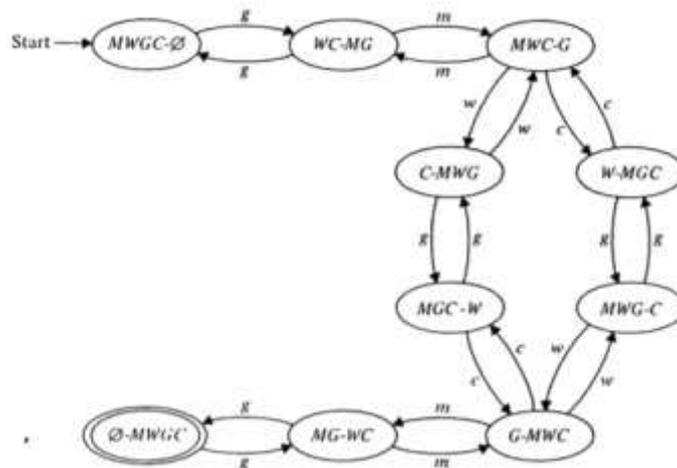


Figure 2: **Transition Diagram for wolf, goat, and box of cabbage problem**

Each oval or circle in Figure 1 represents the states in finite state machine. The label of each state represents the current situation of the problem. For example, the state label WC-MG indicates that the wolf and the box of cabbage are on the left bank, while the farmer and the goat are on the right bank. Therefore, the starting of the problem is the state named MWGC- $\emptyset$  and the final state is represented with double circle which is the  $\emptyset$ -MWGC state. The labelled arrows represent the transition each time the boat move. The labels 'm', 'c', 'g', and 'w' are the items carried during the movement of the boat accordingly which in finite automata machine they are the input symbols. There are two possible outputs of finite state machine which are 'accept' or 'error'. The accept output indicates that a final state has been reached. There are two solution for the wolf, goat and box of cabbage problem as can be seen by observing the paths from the starting state to the final state.

Finite state automata have two different types. The first type of automaton is the finite accepters which the operations are deterministic. This type of automaton is known as Deterministic Finite Automata (DFA). DFA is formally defined by quintuple as mentioned before. This DFA works in the following manner. Firstly, it is assumed to be in the starting state. The input mechanism that will be read by the machine is the leftmost symbol of the input string. Every time the automaton moves, the input mechanism increments one position to the right. Meaning that each move of the automaton consumes one input symbol. The automaton can only read one symbol on each step and the input mechanism can move only from left to right. The transitions from one state to another define the transition function (Linz, 2006).

The second type of automaton is non-deterministic finite state automata (NFA/NDFA). In NFA it is possible for one symbol to cause a transition to more than one condition and provides several possible movements so that the output cannot be ascertained. In addition, it is also possible to make a spontaneous transition or the  $\epsilon$  (epsilon) transition.

## **METHODS**

This research combined two methods to analyze and design the application. Formal method is a mathematical modeling used to bridge (formal specifications) the manufacture, development and verification of hardware and software, which functions in the initial design to test results (Jackson, 1998). The waterfall or sequential linear method as well as formal method. The waterfall method has several stages and must be done sequentially, namely Requirements, Design, Implementation, Verification, and Maintenance (Balaji & Murugaiyan, 2012). The reason we used waterfall model is because the requirements of this application are clear enough since this is a simple and short application.



Figure 3: **The Waterfall Model Life Cycle**

In the first stage, the requirements stage we determined the requirements such as the function and the purpose of the application. In this stage, we also analyze the requirements using the formal method which will be explain in detail later. The design stage is when we develop the design of the application based on the previous requirements in the first stage. The result of this stage is the architectural design of the system. The next stage is the implementation stage. In this stage based on the input and design from the previous stage we will develop a program as the implementation of the previous stage. In verification stage, testing is done to capture any problems before the installation of the system. Finally, in maintenance stage we are going to do some modification if needed.

## RESULTS

Figure 4 shows the flowchart of how the ticket selling machine works. The initial state of the machine is idle. Meaning that, it will do nothing if there is no input given to the machine. The input of this machine the nominal of the money. When user input some money into the selling machine, it will then ask the user to input the amount of tickets the user wants to buy. The machine will calculate the nominal or amount of the money and the amount of tickets. It will then check if the total money is not enough then it will return a message to the user that the money inserted is not enough and ask the user to input another amount of ticket. If the total money is enough or more than the amount they need to pay, the machine will print out the tickets and then gives some change if any.

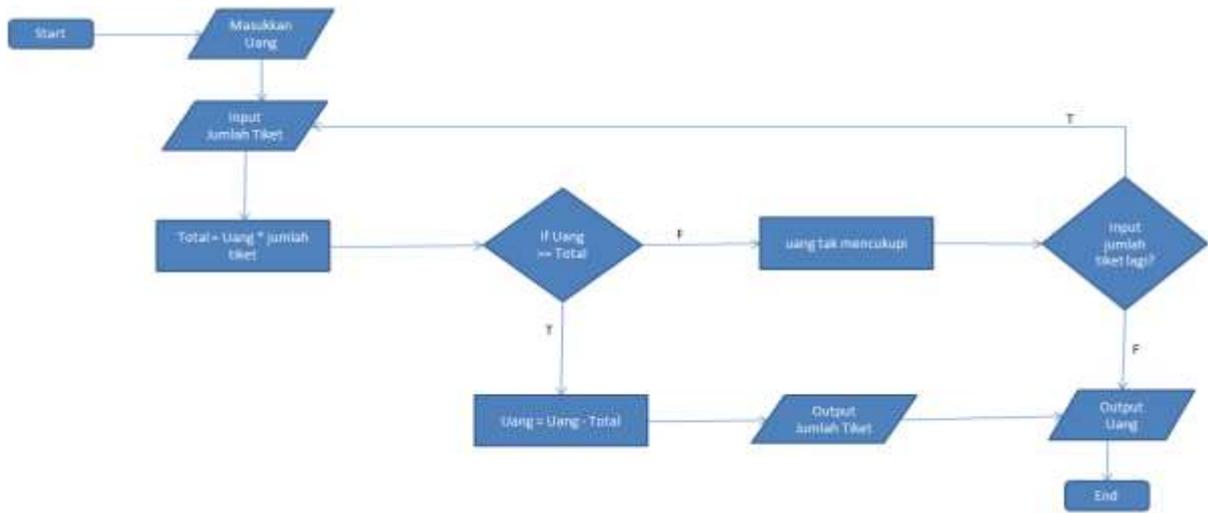


Figure 4: Flowchart of ticket selling machine

Based on this flowchart, we build a state transition diagram using the finite state automata concepts. The state transition diagram is provided in Figure 5.

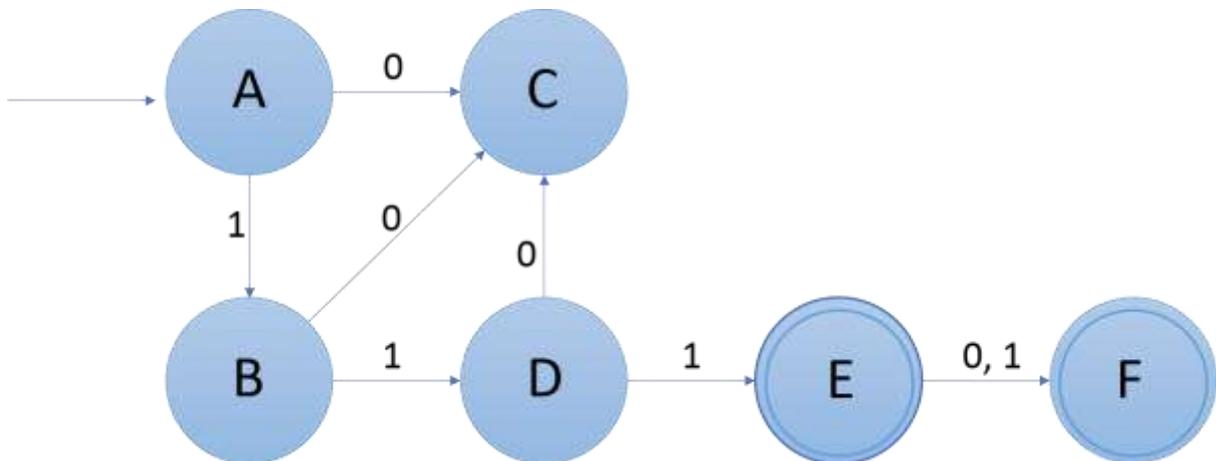


Figure 5: The State Transition Diagram

Figure 5 shows the process of purchasing the tickets from the selling machine. The formal definitions are as follows:

$$Q = \{A, B, C, D, E, F\}$$

$$\Sigma = \{0, 1\}$$

$$\delta = \{(A,1)B, (A,0)C, (B,1)D, (B,0)C, (D,1)E, (D,0)C, (E,0)F, (E,1)F\}$$

$$S = A$$

$$F = \{E, F\}$$

Table 1: **State name and its description**

| State | Description          |
|-------|----------------------|
| A     | Initial State (Idle) |
| B     | Amount of Money      |
| C     | Exit/Cancel          |
| D     | Amount of Ticket(s)  |
| E     | Prints Ticket(s)     |
| F     | Give change          |

Table 2: **The input symbols description**

| Input Symbols | Description  |
|---------------|--------------|
| 0             | Not Selected |
| 1             | Selected     |

Table 1 provides the description of each state while Table 2 shows the meaning of the input symbols. For example, assume the machine is in the initial state, when it receives or read 1, meaning that the process will continue to another state in this case state B, and so on. The double circle states are final or accepting states which in this problem the machine will accept successfully when it reaches either state E or state F.



Figure 6: **Interface design of initial state of the application**

The interface design of the application can be seen in Figure 6, 7, 8. Figure 6 shows the interface when nothing is inserted into the machine. Figure 7 shows the interface when the money is inserted into the machine and the amount of ticket has been chosen. Figure 8 is the a dialog box when the user has been successfully finish the transaction.



Figure 7: **Interface design after the money has been inserted and amount of tickets has been chosen**



Figure 8: **Interface design after the transaction is complete.**

## **DISCUSSION**

The aim of this study is to show the implementation of the concept of Automata theory in an amusement park automatic ticket selling machine. There are many methodologies can be used as the solution, however the results of this study show that Finite State Automata can be one of the methods to analyze how a ticket selling machine works. Using formal method we can defined a problem as well as design the solution of the problem. This shows that Automata Theory is an important field of study for it can be used to solve problems specifically in an automated system.

However, the analysis of this study is very simple. The configuration we used to implement the ticket selling machine basic logic is very limited. In this application, the payment method is by inserting a certain amount of cash into the machine and the machine will proceed to the next process. However, for future development another payment method can be added. For example, instead of inserting cash, user can choose to pay using debit as well as credit card. Other than that, another configuration can be added, such as the development of the basic logic used to develop the finite state automata analysis. For example, the state transition diagram can be expanded and have the calculation of the amount of money inserted and the amount of tickets.

### **Conclusion**

Based on the results, it can be concluded that Finite State Automata can be used as a logical basis in designing automatic ticket selling machines and making it easier for consumers to buy tickets in an amusement park.

## **REFERENCES**

- Balaji, S., & Murugaiyan, M. S. (2012). Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. *International Journal of Information Technology and Business Management*, 2(1), 26-30.
- Hopcroft, J. E. (2008). Introduction to automata theory, languages, and computation: Pearson Education India.
- Linz, P. (2006). An introduction to formal languages and automata: Jones & Bartlett Learning.
- Putra, Y. M. R. (2017). Sentence Analysis with Artificial Intelligence Machine Learning Using Finite State Automata. *PROXIES*, 1(1), 1-6.
- Rahadian, M. F., Suyatno, A., & Maharani, S. (2017). Penerapan Metode Finite State Machine Pada Game “The Relationship”.
- Setiawan, I. (2006). Perancangan Software Embedded System Berbasis FSM. Semarang: Universitas Diponegoro.
- Utdirartatmo, F. (2005). *Teori Bahasa dan Otomata*. Graha Ilmu, Yogyakarta, Ed 2.