Modeling financial statements for small and medium businesses in Worm-Made Fertilizer Using Finite State Automata (FSA)

Aviv Yuniar Rahman¹, Feddy Wanditya Setiawan², April Lia Hananto³, Bagus Setyawan⁴,

¹Department of Informatics Engineering, Universitas Widyagama Malang.
²Department of Mechanical Engineering, Politeknik Hasnur Barito Kuala.
³Department of Information Systems, Universitas Buana Perjuangan Karawang.

¹) E-mail: aviv@widyagama.ac.id.
²) E-mail: feddy@polihasnur.ac.id.
³) E-mail: aprilia@ubpkarawang.ac.id.
⁴) E-mail: bagussetyawanmm01@gmail.com

Abstract. Malang Raya is currently a city that has very rapid economic development. In its development, it began with several small businesses. It is various businesses in Malang such as the culinary, home industry, to livestock business. Currently, one example of a rapidly increasing business is small and medium-sized businesses. Small and medium businesses aim to attract the attention of customers in buying. In this business, there must be financial reports as support in processing business financial data. With this, a State Automata (FSA) finite is needed. Finite State Automata is a concept model that shows the design of an application. With the concept of Finite State Automata, it will introduce a pattern that is in the cash application system. In this test, it will make 3 designs to do the test results. In the three test results it is known that for the first test, it is explained that in the processing process that is carried out 5 times, there is an error in processing 3 times, and in the second test process that is done 5 times, there is an error in processing as much as 1 time. In the third test, this is the result of developing the first and second designs. From the results, it is known that in processing data 5 times, there are no errors in calculations. To test the speed in processing data is appropriate and significant enough to be used as a cash management application for a business.

Keyword: UMKM Fertilizer; Worm Fertilizer; Cash software; FSA

1. Introduction
The current economy in Malang Raya is getting increasingly rapid [1]. In the rapid development of the economy, it starts with several small businesses [2] which is an important point in improving the economy [3]. Various businesses in Malang, such as culinary, home industry, to the livestock business [4]. In this business, the average is currently in the home industry [5]. Small and medium businesses that are currently developing in Indonesia [6]. The method is calculated with a very simple effort. With the existence of small and medium businesses will provide access to small entrepreneurs to do
marketing [7]. And in this small and medium business aims to attract the attention of customers in buying. This provides an opportunity for small entrepreneurs [8]. In a small and medium business, using the medium of worms as fertilizer. This business is one of the businesses that is still not widely known by the public. In the business is the process of making a fertilizer made from worms. The business starts from the worm cultivation process to the process of making fertilizer with worms. This business can produce every day to be used as fertilizer. With these small and medium businesses can introduce a fertilizer product made from worms to make a tool as an alternative replacement.

2. Related Research
In this case, using a Finite State Automata (FSA). Finite State Automata is a concept model that shows the design of an application [9]. With the concept of Finite State Automata, it will introduce a pattern that is in the cash application system [10][11], to use a simple financial process [13] still is to introduce a pattern from point one to another point [12]. In designing the cash application, it was conducted in Sukun Urban Village, Malang. With the application of cash to find out in detail about the income and expenses made. In financial reports that are managed by the finance and management departments, the reports often use simple financial processes [13]. For this reason, a cash processing application is needed to support a financial process in processing a report [14]. Using cash processing applications can reduce the error rate in data entry. As well as reducing storage space, which should stack books in a place will become a file properly without using books again [15].

Previous researchers describe the notes of small and medium-sized businesses. Financial reports use manual notes on financial reporting in most areas. Management in a business is not going well. Also, it is the lack of background in their experience that makes the process even more delayed [16]. Then in subsequent research also shows that in financial management, they already understand the basic concepts in managing, but they do not store that data in one group. This will be a little detrimental to financial management. In addition, in this case, it will also make it difficult for financial managers to find data one by one [17]. And other studies have the same problems in the process of financial records. This causes an error in entering data, or financial calculation does not match the amount available. In the process of reporting both amounts and financial data also use manual records. This will slightly delay the time if the data in a financial need is in a different year. And in the process of manual records will also be a little inconvenient if the data in finance is written every day [18].

The purpose of this case will introduce an application and explain the stages that exist in a cash application using the Finite State Automata method. Where in this Finite State Automata method can be known patterns and contents contained in a financial processing application And will get an overview of the process flow of the course of an application, starting from the entry process until the final result. This will make it easier for financial management to process data without having to do manual financial writing again.

3. Research Methods
This method will explain the stages that exist in a cash application. This method is designing patterns to find out the contents in the application. Thus the method can get an appropriate result in the cash application and to get an overview of the cash application system for small and medium businesses, on the whole, using the Finite State Automata method. To build this cash application, the step that must be prepared is to design and process data with an overview of each flow aimed at Figure 1. This explains for the application user, where when the application is run, the user will provide input in the "transaction" column. In that column, the user can enter information by what is being done. After that, it is in the "unit" view. Where in the column, there are all types of goods with units and prices following those in the business. Whereas in the column "Amount" the user must calculate how many amounts have been made in the sale. In the column "amount". After that, the user enters the income or expenditure and is checked by the system to start a calculation. In the calculation which will be checked later to be included in the overall balance or net balance.
3.1 System Design

In this system model uses finite state automata (FSA), where if it is formulated and taken as an example with the formula M in finite state automata (FSA), among others (Tm, ∑, δ, S, F) And can be interpreted for Tm = a step the set / set of states, ∑ = the set of an input, δ = the use and function of the transition, S = initial step F = final step / final destination. Figure 2 is the application design that illustrates a relationship between the initial state to the final state.
Figure 3. Second Model

Figure 4. Third Model

Table 1. Description of the First Model

| Tm0 | Initial State (Initial Capital) | Tm8 | Worm |
| Tm1 | Transaction                    | Tm9 | Land of Worms |
| Tm2 | Unit                           | Tm10| Tool Care |
| Tm3 | Total                          | Tm11| Electricity |
| Tm4 | Income                         | Tm12| Transportation |
| Tm5 | Spending                       | Tm13| Services |
| Tm6 | Seeds                          | Tm14| Ending Balance |
| Tm7 | Feed                           |
Table 2. Description of the Second Model

| Tm0 | Initial State (Initial Capital) |
| Tm1 | Transaction                     |
| Tm2 | Unit                           |
| Tm3 | Total                          |
| Tm4 | Income                         |
| Tm5 | Spending                       |
| Tm6 | Seeds                          |
| Tm7 | Feed                           |
| Tm8 | Worm                           |
| Tm9 | Land of Worms                  |
| Tm10| Tool Care                      |
| Tm11| Electricity                    |
| Tm12| Transportation                 |
| Tm13| Services                       |
| Tm14| Ending Balance                 |

Table 3. Description of the Third Model

| Tm0 | Initial State (Initial Capital) |
| Tm1 | Transaction                     |
| Tm2 | Unit                           |
| Tm3 | Total                          |
| Tm4 | Income                         |
| Tm5 | Spending                       |
| Tm6 | Seeds                          |
| Tm7 | Feed                           |
| Tm8 | Worm                           |
| Tm9 | Land of Worms                  |
| Tm10| Tool Care                      |
| Tm11| Electricity                    |
| Tm12| Transportation                 |
| Tm13| Services                       |
| Tm14| Ending Balance                 |

In the transition diagram shown in Figure 2, it can be explained if $M = (Tm, \Sigma, \delta, S, F)$ as follows:

$Tm = \{Tm0, Tm1, Tm2, Tm3, Tm4, Tm5, Tm6, Tm7, Tm8, Tm9, Tm10, Tm11, Tm12, Tm13, Tm14\}$

$\Sigma = (1,0)$

$\delta = \delta(Tm0, Tm0,Tm1)=Tm0, \delta(\emptyset,Tm2)= Tm1, \delta(Tm2,Tm3)=Tm2, \delta(\emptyset,Tm4,Tm5)= Tm3, \delta(\emptyset,Tm6,Tm7,Tm8,Tm9)= Tm4, \delta(\emptyset,Tm4,Tm10Tm11,Tm12,Tm13)=Tm5, \delta(\emptyset, Tm14)=Tm6,$

$\delta(\emptyset, Tm14)=Tm7, \delta(\emptyset, Tm14)=Tm8, \delta(\emptyset, Tm14)=Tm9, \delta(\emptyset, Tm14)=Tm10, \delta(\emptyset, Tm14)=Tm11, \delta(\emptyset, Tm14)=Tm12, \delta(\emptyset, Tm14)=Tm13, \delta(Tm14, Tm14)=Tm14$

$S = Tm0$

$F = Tm14$

For example, "Seed sales". The design will begin with the first state, where the state is at (Tm0), then move again towards (Tm1). (Tm1) is filled with information from the transaction "seed sales". After completing the explanation in (Tm1), then move again to (Tm2). In this state (Tm2) is an explanation of the unit and price of the seeds sold. From (Tm2) will go to (Tm3), here will contain about how much from the sales transaction such as "Rp. 250,000 / 3 Kg". Then step again towards (Tm5) in this state will be checked whether there is income or expenditure. Because of the sale of seeds, there will be income in a process. Then just go straight to (Tm4). In (Tm4) this will be processed in what category the seedlings are sold. After completing the process in that category, selling seeds is in the category (Tm6) then goes directly to (Tm6). Inside (Tm6) this will be written in the form of nominal income, such as "Rp.750,000". When finished, continue to the final state (Tm14). In this final state, everything will be explained like "Seed sales Rp.750,000, 9 kg". After doing 5 times there were errors in processing 3 times in the first model, then continued testing in the second model. In the second design model, the following explanation is:

In the transition diagram shown in Figure 3, it can be explained if $M = (Tm, \Sigma, \delta, S, F)$ as follows:

$Tm = \{Tm0, Tm1, Tm2, Tm3, Tm4, Tm5, Tm6, Tm7, Tm8, Tm9, Tm10, Tm11, Tm12, Tm13, Tm14\}$

$\Sigma = (1,0)$

$\delta = \delta(Tm0, Tm0,Tm1)=Tm0, \delta(\emptyset,Tm2)= Tm1, \delta(Tm2,Tm3)=Tm2, \delta(\emptyset,Tm4,Tm5)= Tm3, \delta(\emptyset,Tm5)= Tm4, \delta(\emptyset, Tm6,Tm7,Tm8,Tm9)= Tm4, \delta(\emptyset, Tm4,Tm10Tm11,Tm12,Tm13)=Tm5, \delta(\emptyset, Tm14)=Tm6,$

$\delta(\emptyset, Tm14)=Tm7, \delta(\emptyset, Tm14)=Tm8, \delta(\emptyset, Tm14)=Tm9, \delta(\emptyset, Tm14)=Tm10, \delta(\emptyset, Tm14)=Tm11, \delta(\emptyset, Tm14)=Tm12, \delta(\emptyset, Tm14)=Tm13, \delta(Tm14, Tm14)=Tm14$
Table 4. First Model Transition Table

| δ | 0 | 1 |
|---|---|---|
| Tm0 | Tm0 | Tm0, Tm1 |
| Tm1 | Ø | Tm2 |
| Tm2 | Tm2 | Tm3 |
| Tm3 | Ø | Tm5 |
| Tm4 | Ø | Tm6, Tm7, Tm8, Tm9 |
| Tm5 | Ø | Tm4, Tm10, Tm11, Tm12, Tm1 |
| Tm6 | Ø | Tm14 |
| Tm7 | Ø | Tm14 |
| Tm8 | Ø | Tm14 |
| Tm9 | Ø | Tm14 |
| Tm10 | Ø | Tm14 |
| Tm11 | Ø | Tm14 |
| Tm12 | Ø | Tm14 |
| Tm13 | Ø | Tm14 |
| Tm14 | Tm14 | Tm14 |

δ(Ø, Tm14) = Tm7, δ(Ø, Tm14) = Tm8, δ(Ø, Tm14) = Tm9, δ(Ø, Tm14) = Tm10, δ(Ø, Tm14) = Tm11, δ(Ø, Tm14) = Tm12, δ(Ø, Tm14) = Tm13, δ(Tm14, Tm14) = Tm14

S = Tm0
F = Tm14

For example, "Sales of worms". The design will start at (Tm0), then move again towards (Tm1). From (Tm1) this is filled with information on the results of the transaction "worm sale". After completing the explanation in (Tm1), then move again to (Tm2). In this state (Tm2) is an explanation of the unit and price of the seeds sold.

From (Tm2) will go to (Tm3), here will contain about how much of the sales transaction is like "Rp. 300,000 / 2 kg". Then moving towards (Tm4) in this state will be checked whether there is income or expenditure. Because the sale of worms is included in the import process. After being in (Tm4) move again to (Tm5). In (Tm5) this will process all results in what category the worm sales. After completing the process in that category, the sales of worms are in the category (Tm8) then go directly to (Tm8). Inside (Tm8) this will be written in the form of nominal income, such as "Rp.600,000". When finished, continue to the final state (Tm14). In this final state, everything will be explained as "Sales of worms Rp.600,000.4 kg". After making 5 errors in processing 1 time in the second model, then continued testing in the third model. In the final testing process is the third test. In this third test is a form of development of the first and second testing. In an explanation like the following:

In the transition diagram shown in Figure 4, it can be explained if M = (Tm, Σ, δ, S, F) as follows:

Tm = {Tm0, Tm1, Tm2, Tm3, Tm4, Tm5, Tm6, Tm7, Tm8, Tm9, Tm10, Tm11, Tm12, Tm13, Tm14}
Σ = (1, 0)
δ = δ(Tm0, Tm0, Tm1) = Tm0, δ(Tm0, Tm2) = Tm1, δ(Tm2, Tm3) = Tm2, δ(Ø, Tm4, Tm5) = Tm3,
δ(Tm4, Tm6, Tm7, Tm8, Tm9) = Tm4, δ(Tm5, Tm6, Tm7, Tm10, Tm11, Tm12, Tm13) = Tm5,
δ(Ø, Tm14) = Tm6, δ(Ø, Tm14) = Tm7, δ(Ø, Tm14) = Tm8, δ(Ø, Tm14) = Tm9, δ(Ø, Tm14) = Tm10,
δ(Ø, Tm14) = Tm11, δ(Ø, Tm14) = Tm12, δ(Ø, Tm14) = Tm13, δ(Tm14, Tm14) = Tm14
S = Tm0
F = Tm14
Table 5. Second Model Transition Table

| \( \delta \) | 0 | 1 |
|-------------|---|---|
| Tm0         | Tm0, Tm1 | Tm0, Tm1 |
| Tm1         | Ø, Tm2   | Tm2   |
| Tm2         | Tm2, Tm3 | Tm3   |
| Tm3         | Ø, Tm4, Tm5 | Tm4, Tm5 |
| Tm4         | Ø, Tm5   | Tm5   |
| Tm5         | Ø, Tm6, Tm7, Tm8, Tm9, Tm10, Tm11, Tm12, Tm13 | Tm11, Tm12, Tm13 |
| Tm6         | Ø, Tm14  | Tm14  |
| Tm7         | Ø, Tm14  | Tm14  |
| Tm8         | Ø, Tm14  | Tm14  |
| Tm9         | Ø, Tm14  | Tm14  |
| Tm10        | Ø, Tm14  | Tm14  |
| Tm11        | Ø, Tm14  | Tm14  |
| Tm12        | Ø, Tm14  | Tm14  |
| Tm13        | Ø, Tm14  | Tm14  |
| Tm14        | Tm14     | Tm14  |

Table 6. Third Model Transition Table

| \( \delta \) | 0 | 1 |
|-------------|---|---|
| Tm0         | Tm0, Tm1 | Tm0, Tm1 |
| Tm1         | Tm1, Tm2 | Tm2   |
| Tm2         | Tm2, Tm3 | Tm3   |
| Tm3         | Ø, Tm4, Tm5 | Tm4, Tm5 |
| Tm4         | Tm4, Tm6, Tm7, Tm8, Tm9 | Tm6, Tm7, Tm8, Tm9 |
| Tm5         | Tm5, Tm6, Tm7, Tm8, Tm9, Tm10, Tm11, Tm12, Tm13 | Tm12, Tm13 |
| Tm6         | Ø, Tm14  | Tm14  |
| Tm7         | Ø, Tm14  | Tm14  |
| Tm8         | Ø, Tm14  | Tm14  |
| Tm9         | Ø, Tm14  | Tm14  |
| Tm10        | Ø, Tm14  | Tm14  |
| Tm11        | Ø, Tm14  | Tm14  |
| Tm12        | Ø, Tm14  | Tm14  |
| Tm13        | Ø, Tm14  | Tm14  |
| Tm14        | Tm14     | Tm14  |

For example, "Payment of services". The design will begin with the first state, where the state is at (Tm0), then move again towards (Tm1). (Tm1) is filled with information from the transaction "payment for services". After completing the explanation in (Tm1), then move again to (Tm2). This state is a unified explanation of the type of transaction. From (Tm2) will go to (Tm3), here will contain about how much from the transaction such as "Rp. 200,000 / week". Then step toward (Tm5) in this state will be checked whether there is income or expenditure. Due to payment for services, there is an expense in a process. In (Tm5) this will be processed in what category the service payment is. After completion of the process, payment for services is in the category (Tm13) and goes directly to (Tm13). Inside (Tm13) this will be written in the form of the nominal, such as "Rp.800,000". When finished, continue to the final state (Tm14). In this final state, everything will be explained as "Payment of services Rp.800,000, 4 weeks (1 month)". After doing it 5 times, there is no error in this model. And in this third testing process can be used for a financial data processing application because the level of speed and data processing is appropriate and known without the slightest error.
Table 7. Comparison of Model Designs

|   | Model 1 | Model 2 | Model 3 |
|---|---------|---------|---------|
| δ | 0       | 1       | 0       | 1       | 0       | 1       |
| Tm0| processed | processed | processed | processed | processed | processed |
| Tm1| not processed | processed | not processed | processed | processed | processed |
| Tm2| processed | processed | processed | processed | processed | processed |
| Tm3| not processed | processed | not processed | processed | processed | processed |
| Tm4| not processed | processed | not processed | processed | processed | processed |
| Tm5| not processed | processed | not processed | processed | processed | processed |
| Tm6| not processed | processed | not processed | processed | not processed | processed |
| Tm7| not processed | processed | not processed | processed | not processed | processed |
| Tm8| not processed | processed | not processed | processed | not processed | processed |
| Tm9| not processed | processed | not processed | processed | not processed | processed |
| Tm10| not processed | processed | not processed | processed | not processed | processed |
| Tm11| not processed | processed | not processed | processed | not processed | processed |
| Tm12| not processed | processed | not processed | processed | not processed | processed |
| Tm13| not processed | processed | not processed | processed | not processed | processed |
| Tm14| processed | processed | processed | processed | processed | processed |

3.2 Results and Discussion

In the three designs can be explained starting from the transition diagram in Figure 3, it can be seen that the transition table is addressed in Table 4. With the transition table in Figure 4, it can be explained that (Tm0) can move past Tm0 and Tm1. Then (Tm1) can only be through Tm2. (Tm2) can return to Tm2 and head to Tm3. (Tm3) can only go to Tm5. After that (Tm4) can pass Tm6, Tm7, Tm8, Tm9. (Tm5) can go to Tm4, Tm10 Tm11, Tm12, Tm13. Whereas (Tm6, Tm7, Tm8, Tm9, Tm10, Tm11, Tm12, Tm13) can only pass one destination, Tm14. And for the last Tm14 is the state finish / final destination. And in the second modeling transition diagram in Figure 4, it can be seen that the transition table is aimed at table 5. With the transition table in Figure 5, it can be explained that (Tm0) can move past Tm0 and Tm1. Then (Tm1) can only be through Tm2. (Tm2) can return to Tm2 and head to Tm3. (Tm3) can go to Tm4 and Tm5. After that (Tm4) can only pass Tm5. (Tm5) can go to Tm6, Tm7, Tm8, Tm9. (Tm6) can go to Tm10, Tm11, Tm12, Tm13. Whereas (Tm6, Tm7, Tm8, Tm9, Tm10, Tm11, Tm12, Tm13) can only pass one destination, Tm14. And for the last Tm14 is the state finish / final destination. And in the modeling transition diagram to Figure 3, we can see the transition table is aimed at Table 6. With the transition table in Figure 6, it can be explained that (Tm0) can move past Tm0 and Tm1. Then (Tm1) can only be through Tm2. (Tm2) can return to Tm2 and head to Tm3. (Tm3) can go to Tm4 and Tm5. After that (Tm4) can pass Tm6, Tm7, Tm8, Tm9. (Tm5) can go to Tm6, Tm7, Tm8, Tm9, Tm10, Tm11, Tm12, Tm13. Whereas (Tm6, Tm7, Tm8, Tm9, Tm10, Tm11, Tm12, Tm13) can only pass one destination, Tm14. And for the last Tm14 is the state finish / final destination. It can be seen that the faster the process of design work in the model, the faster the process of entering data in a job. It can be seen in Table 7, and model 3 can be explained that the design can provide a high level of speed in processing data. With this, it can explain some of the designs in a plot and easy to understand. In this design, it can reduce one's error rate in processing larger data.

4. Conclusions and Future Work

Of the three cash processing application models, the test results are known for the results of the first test explaining that in the processing process that was done 5 times there were errors in processing 3 times, and in the second testing process that was done 5 times there was an error in processing as much as 1 time. In the third test, this is the result of developing the first and second designs. From the results, it is known that in processing data 5 times, there are no errors in calculations. And in the third model has several different things, such as the flow of the process, the features provided to the processing speed in processing data. In this difference, the third model is the most superior and can
reduce the error rate in processing and processing data. In the third design, the model can also do for large data and have a high degree of accuracy compared to other models.

5. Acknowledgements
We thank the Sub Directorate for Community Empowerment of the Directorate of Research and Community Service (DRPM), for grants given to the Application of Appropriate Technology to the Community (PPTTG) Year 2019. Furthermore, we thank the liquid worm processing group for cooperation in the community service process. Hopefully, this cooperation will continue the following year.

6. Reference
[1] A. Y. Rahman, S. Sumpeno, and M. H. Purnomo, “Arca Detection and Matching Using Scale Invariant Feature Transform (SIFT) Method of Stereo Camera,” Proc. - 2017 Int. Conf. Soft Comput. Intell. Syst. Inf. Technol. Build. Intell. Through IOT Big Data, ICSIIT 2017, pp. 66–71, 2017.
[2] A. L. Hananto and A. Y. Rahman, “User Experience Measurement On Go-Jek Mobile App In Malang City,” 2018 Third Int. Conf. Informatics Comput., pp. 1–6, 2018.
[3] A. Y. Rahman, S. Sumpeno, and M. H. Purnomo, “Video Minor Stroke Extraction Using Learning Vector Quantization,” pp. 1–7, 2017.
[4] S. Setiawidayat and A. Y. Rahman, “New method for obtaining Peak Value R and the duration of each cycle of Electrocardiogram,” 3rd Int. Conf. Sustain. Inf. Eng. Technol. SIET 2018 - Proc., pp. 77–81, 2018.
[5] A. Trianni, E. Cagno, A. Neri, and M. Howard, “Measuring industrial sustainability performance: Empirical evidence from Italian and German manufacturing small and medium enterprises,” J. Clean. Prod., vol. 229, pp. 1355–1376, 2019.
[6] D. Xiang, J. Chen, D. Tripe, and N. Zhang, “Family firms, sustainable innovation and financing cost: Evidence from Chinese hi-tech small and medium-sized enterprises,” Technol. Forecast. Soc. Change, vol. 144, no. February, pp. 499–511, 2019.
[7] R. Turner and A. Ledwith, “Project Management in Small to Medium-Sized Enterprises: Fitting the Practices to the Needs of the Firm to Deliver Benefit,” J. Small Bus. Manag., vol. 56, no. 3, pp. 475–493, 2018.
[8] J. M. M. Stoffers, B. I. J. M. Van der Heijden, and E. A. G. M. Jacobs, “Employability and innovative work behaviour in small and medium-sized enterprises,” Int. J. Hum. Resour. Manag., vol. 5192, no. January, pp. 1–28, 2018.
[9] B. Öztürk and B. Can, “Turkish lexicon expansion by using finite state automata,” Turkish J. Electr. Eng. Comput. Sci., vol. 27, no. 2, pp. 1012–1027, 2019.
[10] N. Karmani, H. Soussou, and A. M. Alimi, “Tunisian Arabic Chat Alphabet Transliteration Using Probabilistic Finite State Transducers,” Int. Arab J. Inf. Technol., vol. 16, no. 2, 2016.
[11] R. Rakhmavati, Irianto, F. D. Murdianto, A. Luthfi, and A. Y. Rahman, “Thermal Optimization on Incubator using Fuzzy Inference System based IoT,” 2019 Int. Conf. Artif. Intell. Inf. Technol., pp. 464–468, 2019.
[12] A. Naumovs, M. Dimitrijevs, and A. Yakaryılmaz, “The minimal probabilistic and quantum finite automata recognizing uncountably many languages with fixed cutpoints,” pp. 1–11, 2019.
[13] M. Lokanan and V. Tran, “Detecting anomalies in financial statements using machine learning algorithm The case of Vietnamese listed firms,” 2019.
[14] E. Murwaningsari, “The Relationship of Corporate Governance, Corporate Social Responsibilities and Corporate Financial Performance In One Continuum,” pp. 78–97, 2019.
[15] L. Wei, G. Li, X. Zhu, and J. Li, “Discovering bank risk factors from financial statements based on a new semi-supervised text mining algorithm,” Account. Financ., 2019.
[16] J. Jansson, J. Nilsson, F. Modig, and G. Hed Vall, “Commitment to Sustainability in Small and Medium-Sized Enterprises: The Influence of Strategic Orientations and Management Values,”
[17] L. Mura, A. Ključnikov, M. Tvaronavičienė, and A. Androniceanu, “Development trends in human resource management in small and medium enterprises in the Visegrad Group,” Acta Polytech. Hungarica, vol. 14, no. 7, pp. 105–122, 2017.

[18] C. Kunaka and M. N. Moos, “Evaluating mentoring outcomes from the perspective of entrepreneurs and small business owners,” South. African J. Entrep. Small Bus. Manag., vol. 11, no. 1, pp. 1–11, 2019.