TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 18, No. 5, October 2020, pp. 2517~2524
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v18i5.14882

Comparison of search algorithms in Javanese-Indonesian dictionary application

Yana Aditia Gerhana[1], Nur Lukman[2], Arief Fatchul Huda[3], Cecep Nurul Alam[4], Undang Syaripudin[5], Devi Novitasari[6]
[1-6]Department of Informatics, Faculty of Science and Technology,
Universitas Islam Negeri Sunan Gunung Djati, Indonesia
[1,4,5]Information Communication Technology, Asia e University Malaysia, Malaysia

Article Info

Article history:

Received Jun 9, 2019
Revised Apr 9, 2020
Accepted May 1, 2020

Keywords:

Boyer-Moore
Complexity text mining
Horspool
Knuth Morris Pratt
Performace
Searching

ABSTRACT

This study aims to compare the performance of Boyer-Moore, Knuth morris pratt, and Horspool algorithms in searching for the meaning of words in the Java-Indonesian dictionary search application in terms of accuracy and processing time. Performance Testing is used to test the performance of algorithm implementations in applications. The test results show that the Boyer Moore and Knuth Morris Pratt algorithms have an accuracy rate of 100%, and the Horspool algorithm 85.3%. While the processing time, Knuth Morris Pratt algorithm has the highest average speed level of 25ms, Horspool 39.9 ms, while the average speed of the Boyer Moore algorithm is 44.2 ms. While the complexity test results, the Boyer Moore algorithm has an overall number of n $26n^2$, Knuth Morris Pratt and Horspool $20n^2$ each.

Corresponding Author:

Nur Lukman,
Department of Informatics, Faculty of Science and Technology,
Universitas Islam Negeri Sunan Gunung Djati,
A. H. Nasution St. no. 105, Cibiru-Bandung, Indonesia.
Email: n.lukman@uinsgd.ac.id

1. INTRODUCTION

Search algorithm is one of the fundamental research studies in computer science [1-8], including its use in dictionaries. A dictionary is a tool used by someone to learn languages, both international, national, and regional languages. The process of searching vocabulary in a dictionary application requires time in the search process to issue a translation of the word being searched. The search process generally uses a string matching algorithm as a data search algorithm [9-13]. The purpose of using a string matching algorithm, also, to speed up the search process also aims to obtain the accuracy of search results. Several algorithms are belonging to this string algorithm, including the Boyer- Moore algorithm, Horspool algorithm, Knuth Morris Pratt algorithm, and others [14-16].

Research that explains the implementation and updating of matching strings has been discussed in previous studies [1, 4, 7, 12, 15, 17, 18]. Based on research that has been done before, this research tries to develop a Java-Indonesian language di ctionary application, by comparing the performance of several Boyer Moore string data search algorithms. Knuth Morris Pratt and the Horspool algorithm with the addition of the Speech To Text feature to the application. These three algorithms are the best string matching algorithm, which has different table shifts which can search data faster than other algorithms [3-5, 19, 20]. Performance

that is compared is the value of accuracy and average processing time of search results, so we know which string matching algorithm is the best to develop in future research.

2. RESEARCH METHOD

The performance testing method is used to test the performance of three string matching algorithms, namely the Boyer-Moore algorithm, the Knuth Morris Pratt algorithm, and the Horspool algorithm in the Indonesian-Javanese dictionary application. Performance is measured based on the level of accuracy and the level of speed in the search process time in the application. Performance analysis is carried out in several stages. The first stage is the input stage of the vocabulary or data string, the second stage is the process of searching for string data using one of the string algorithms. Figure 1 explains the stages of algorithm performance analysis. The stage is called pre-processing which is a phase of text mining which is very important [21, 22], pre-processing represents data to be more structured until the data is ready to be processed [23] according to the algorithm used. The last stage is the output stage of the calculation process of the algorithm used, in the form of a translation of the vocabulary or search string data.

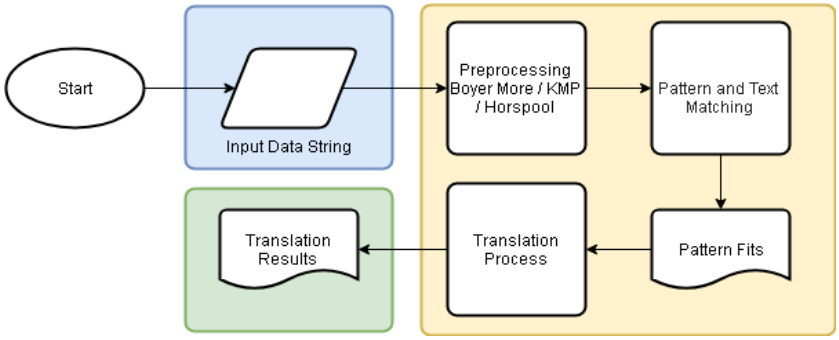

Figure 1. Sistem flowchart running

2.1. Boyer-Moore algorithm

The Boyer-Moore algorithm becomes one of the most frequently used string-lookup algorithms or is implemented into a document or data search feature in the database because it is considered the most efficient in typical applications and is best compared to other string search algorithms [24, 25]. The Boyer-Moore algorithm starts matching a character from the right direction of the pattern or the right-to-left direction of the text [16, 26, 27]. Adequately systematic, the stages that the Boyer-Moore algorithm performed at the time of matching the following strings [26, 28]:

a. Boyer-Moore's algorithm starts matching the pattern at the beginning of the text.
b. The Boyer-Moore algorithm will match from right-to-left to match the pattern character characters with the characters in the matched text until one of the conditions is met.

The searching of data in Boyer Moore algorithm can be seen on the pattern to search the word “ALA” on “BURUK ALA”.

a. Align the pattern of ALA, matched with BURUK ALA

Text : B U R U K A L A
Pattern : A L A

b. Determine the shift table BmBc and BmGs

Tables 1 and 2 explain the Boyer Moore algorithm in searching data that can be seen from the search for the word you want to search (patterns). The BmBc value in Table 1 is obtained from the results of enumeration starting from the string and then to the initial string, starting at the 0th index. Then record the characters that have been found [26, 28, 29]. Table 2 explains the process of finding BmBC values. The enumerator value will be added by 1 found this character has never been found before. Then back to the previous position, the character "A" because the character "A" has been seen previously, the value of the transfer is 1 [26, 28, 29].

c. Make the iteration table to the pattern matching with text

Table 3 explains the iteration pattern of text matching in the Boyer-Moore algorithm. Iteration in the Boyer-Moor algorithm stops at the 4th iteration, meaning that the search for the word ALA in BURUK

ALA text is found in the 4th iteration. Iteration is carried out based on the value of the shifts in the BmBc and BmGs tables and compares which shift value is highest between the two [26, 28, 29].

Table 1. BmBc value

Table BmBc			
Index	0	1	2
Pattern	A	L	A
BmBc	0	1	0

Table 2. BmGs value

Table BmGs			
Index	0	1	2
Pattern	A	L	A
BmBc	3	3	1

Table 3. Boyer Moore algorithm alteration scheme

Indeks	0	1	2	3	4	5	6	7	8
1	B	U	R	U	K		A	L	A
	A	L	A						
2	B	U	R	U	K		A	L	A
			A	L	A				
3	B	U	R	U	K		A	L	A
				A	L	A			
4	B	U	R	U	K		A	L	A
Data Found							A	L	A

2.2. Knuth Morris Pratt algorithm (KMP)

The KMP algorithm has a different shift than the Boyer-Moore algorithm. Broadly speaking the stages in the KMP algorithm, when performing string matching are [16, 20, 22, 30]:

a. Algorithms KMP starts matching pattern at the beginning of the text
b. The KMP algorithm performs a shift or matching pattern character with text characters from left to right by matching characters per character until one of the conditions is met.

The searching of data in Knuth Morris Pratt algorithm can be seen on the pattern to search the word "ALA" on "BURUK ALA".

a. Align the pattern of ALA, matched with BURUK ALA

Text : B U R U K A L A
Pattern : A L A

b. Determine the value of data fringe

In Table 4, the initial boundary value will always be 0. The boundary value is calculated based on a character pattern only. If anyone appreciates the character in the pattern, then the edge value 1 and so on will increase in value with the value j, and I shift to the next pattern character [16, 20, 22, 30].

Table 4. Pattern edge

Table Pattern edge			
J	0	1	2
P(j)	A	L	A
B(b)	0	0	1

c. Make the iteration table to the pattern matching with text

Table 5 explains the iteration pattern matching text on the KMP algorithm. The iteration of the KMP algorithm is similar to the Boyer Moore algorithm; the iteration stops at the 4th iteration. In contrast to the KMP Algorithm, the interaction process is not based on BmBc and BmGs tables but based on values from the edges of the pattern.

Table 5. Knuth Morris pratt algorithm alteration scheme

Indeks	0	1	2	3	4	5	6	7	8
1	B	U	R	U	K		A	L	A
	A	L	A						
2	B	U	R	U	K		A	L	A
			A	L	A				
3	B	U	R	U	K		A	L	A
				A	L	A			
4	B	U	R	U	K		A	L	A
Data Found							A	L	A

2.3. Horspool algorithm

The Horspool algorithm is one of the string search algorithms, which is a simplification of the Boyer-Moore algorithm [15, 16, 31, 32]. The Horspool algorithm has a simpler shifting compared to the Boyer-Moore algorithm. The Boyer-Moore algorithm has two shifting process functions, i.e., bad-character shift and good-suffix shift, so the Horspool algorithm only uses one panning that is bad-character shifting [33]. The function preprocess pattern in the Horspool algorithm is by performing a jump based on "bad-character" or based on the character mismatch in the pattern found in the text. Panning the Horspool algorithm uses the rightmost character in the current text window to determine the shift distance to be performed. The pattern will shift to the far right of the window until a match between the pattern character and the text. The searching of data in Horspool algorithm can be seen on the pattern to search the word “ALA” on “BURUK ALA”.

a. Align the pattern of ALA, matched with BURUK ALA

Text : B U R U K A L A

Pattern : A L A

b. Determine the shift table BmBc

Table 6 explains the initial interpretation of the Horspool algorithm. The matching process starts at the 0th index or the character "A". Perform the previous position, and the enumerator value will be added 1 if this character has never been found before, Step back to the previous position which is the character "A" because the character "A" has been found before then the replacement value is 1. The final scheme of the Horspool algorithm matching process is examined in Table 7. The last iteration determines the character 0 (blank) in the text does not match the character A in the pattern, so the matching process removes because 0 (empty) does not match.

Table 6. BmBc value

Table BmBc			
Index	0	1	2
Pattern	A	L	A
BmBs	0	1	0

Table 7. Horspool algorithm literacy schema

Indeks	0	1	2	3	4	5	6	7	8
1	B	U	R	U	K		A	L	A
	A	L	A						
2	B	U	R	U	K		A	L	A
			A	L	A				
3	B	U	R	U	K	?	A	L	A
Data Not Found					A	L	A		

3. RESULTS AND ANALYSIS

Data used in the form of a vocabulary that will be translated into the Javanese language derived from the existing Javanese dictionary. In the research that is being done this observation is done in the form of direct observation of the use of Javanese language in the community that began to be shifted based on a journal or article that writes directly about the alignment The use of Javanese language among the younger generation and several journals that proves some matching string algorithms that can be used for comparison.

Figure 2 explains the test results of the three algorithms. Based on the results of testing of 1500 vocabularies with 400 experiments, the accuracy of the Horspool algorithm is lower than the KMP and Boyer-Moor algorithms, with an accuracy rate of 85.3%, while the KMP and Boyer-Moor algorithms are 100% respectively. Mathematically the test results are explained as follows:

Accuracy level = (Number of successful samples/total number of samples) x 100%

Accuracy of Boyer-Moore = $\left(\frac{400}{400}\right) x100\%$= 100%

Accuracy of Akurasi KMP = $\left(\frac{400}{400}\right) x100\%$ = 100%

Accuracy of Akurasi Horspool = $\left(\frac{341}{400}\right) x100\%$ = 85.3%

Figure 3 explains the results of testing the time of the third algorithm search process. The speed of the algorithm is tested based on the processing time in searching vocabulary. Based on the test results, the algorithm that has the fastest speed is Knuth-Morris-Pratt with an average of 25 ms, Horspool 39.9 ms, and Boyer Moore 44.2 ms. Figure 4 explains in detail the comparison graph of the speed of each algorithm in searching each Javanese vocabulary into Indonesian where blue indicates Boyer Moore's algorithm, red indicates Knuth Morris Pratt's algorithm, and green indicates Horspool's algorithm.

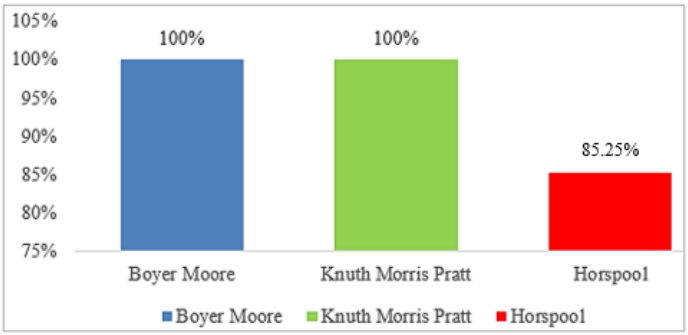

Figure 2. Accuracy algorithm

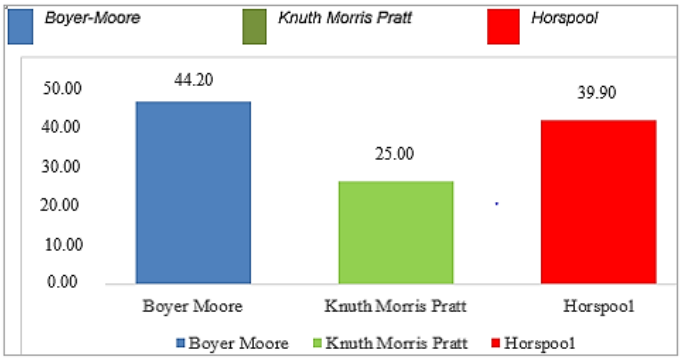

Figure 3. Average value of comparison algorithm at match speed level

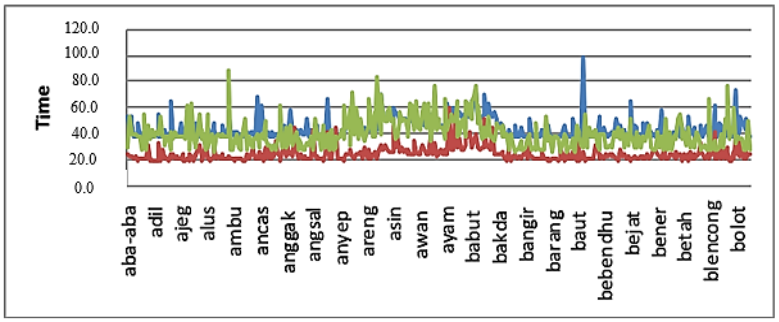

Figure 4. Time execution algorithm

Testing algorithms based on the complexity of the algorithm, testing is conducted based on the efficiency of how much space (space) and the time it takes for the algorithm to run every step by step in the algorithm. After testing the three algorithms, obtained the result that Boyer Moore's algorithm has an overall n total of 11n, with the amount of n^2 as much as $26n^2$. The algorithm of Knuth Morris Pratt algorithm has an overall n of 8n and an amount of n^2 as much as $20n^2$, and the Horspool algorithm has a value of n as much as 10n and value of n^2 as much as $20n^2$. Figure 5 explains that Knuth Morris Pratt's algorithm has

the best time efficiency with a sum value of N^2 and the least n values compared to the Horspool algorithm and Boyer Moore's algorithm, which means the time the algorithm process is Has the most rapid processing efficiency rates compared to another algorithm.

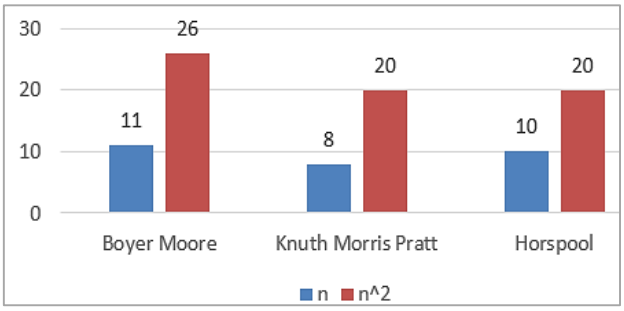

Figure 5. Comparison of algorithm complexity testing

4. CONCLUSION

Based on the test results, Boyer-Moor and KMP have a higher level of accuracy compared to the Horspool algorithm. While the average processing time of KMP is better than Horspool and Boyer-Moor. The processing time is directly proportional to the efficiency produced by KMP, which is better than the Boyer-Moor and Horspool algorithm, with the least number of n^2 and n values.

REFERENCES

[1] M. Abdullahi, M. A. Ngadi, S. I. Dishing, S. M. Abdulhamid, and B. I. eel Ahmad, "An efficient symbiotic organisms search algorithm with chaotic optimization strategy for multi-objective task scheduling problems in cloud computing environment," *J. Netw. Comput. Appl.*, vol. 133, pp. 60-74, 2019.
[2] G. Li, H. Zhou, X. Jing, G. Tian, and L. Li, "An intelligent wheel position searching algorithm for cutting tool grooves with diverse machining precision requirements," *Int. J. Mach. Tools Manuf.*, vol. 122, pp. 149-160, 2017.
[3] M. S. Sanaj and P. M. Joe Prathap, "Nature inspired chaotic squirrel search algorithm (CSSA) for multi objective task scheduling in an IAAS cloud computing atmosphere," *Eng. Sci. Technol. an Int. J.*, 2019.
[4] J. Brandão, "A memory-based iterated local search algorithm for the multi-depot open vehicle routing problem," *Eur. J. Oper. Res.*, vol. 284, no. 2, pp. 559-571, 2020.
[5] S. Makhdoomi and A. Askarzadeh, "Optimizing operation of a photovoltaic/diesel generator hybrid energy system with pumped hydro storage by a modified crow search algorithm," *J. Energy Storage*, vol. 27, 2020.
[6] M. Yarlagadda, K. Gangadhara Rao, and A. Srikrishna, "Frequent itemset-based feature selection and Rider Moth Search Algorithm for document clustering," *J. King Saud Univ. -Comput. Inf. Sci.*, 2019.
[7] P. H. Xiao, B. W. Li, and S. B. Wang, "A coupled DEM-FTSM approach with a new cost-effective searching algorithm of particles for gas-solid flow with heat transfer," *Int. J. Therm. Sci.*, vol. 143, pp. 52-3, 2019.
[8] E. H. Mohamed and E. M. Shokry, "QSST: A Quranic Semantic Search Tool based on word embedding," *J. King Saud Univ.–Comput. Inf. Sci.*, 2020.
[9] S. T. Klein and D. Shapira, "The String-to-Dictionary Matching Problem," *Proceedings of the Data Compression Conference*, vol. 55, no. 11, pp. 143-152, 2015.
[10] S. Das and K. Kapoor, "Weighted approximate parameterized string matching," *AKCE Int. J. Graphs Comb.*, vol. 14, no. 1, pp. 1-12, 2017.
[11] G. Didier and L. Tichit, "Designing optimal- and fast-on-average pattern matching algorithms," *arXiv:1604.08860v4*, 2016.
[12] C. Ryu and K. Park, "Improved pattern-scan-order algorithms for string matching," *J. Discret. Algorithms*, 2018.
[13] M. Xu, X. Jia, M. Pickering, and A. J. Plaza, "Cloud removal based on sparse representation via multitemporal dictionary learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 5, pp. 2998–3006, 2016.
[14] M. Bhagya Sri, R. Bhavsar, and P. Narooka, "String Matching Algorithms," *Int. J. Eng. Comput. Sci.*, vol. 7, no. 3, pp. 23769-23772, 2018.
[15] C. C. Hoong and M. A. Ameedeen, "Boyer-moore horspool algorithm used in content management system of data fast searching," *Adv. Sci. Lett.*, vol. 23, no. 11, pp. 11387-90, 2017.
[16] D. Gurung, U. K. Chakraborty, and P. Sharma, "Intelligent Predictive String Search Algorithm," *Procedia Computer Science*, vol. 79, pp. 161-169, 2016.
[17] W. R. King, "Knowledge Management and Organizational Learning," *Annals of Information Systems 4*, pp. 3-13, 2009.
[18] M. Morales, S. Scherer, and R. Levitan, "A Cross-modal Review of Indicators for Depression Detection Systems," *Proceedings of the Fourth Workshop on Computational Linguistics and Clinical Psychology-From Linguistic Signal to Clinical Reality*, 2017.

[19] A. Kusnadi and A. K. Wicaksono, "Comparison of Horspool Algorithm and Zhu-Takaoka Algorithm in Desktop-Based String Search," *J. Ultim. Comput.*, vol. 9, no. 1, pp. 12-16, 2017.
[20] W. Astuti, "Analysis of String Matching in These Title Using Knuth-Morris-Pratt (KMP) Algorithm," *Ilk. J. Ilm.*, vol. 9, no. 2, pp. 167-172, 2017.
[21] S. Vijayarani, M. J. Ilamathi, M. Nithya, A. Professor, and M. P. Research Scholar, "Preprocessing Techniques for Text Mining -An Overview," *Int. J. Comput. Sci. Commun. Networks*, vol. 5, no. 1, pp. 7-16, 2015.
[22] M. Bhagya Sri, R. Bhavsar, and P. Narooka, "String Matching Algorithms," *Int. J. Eng. Comput. Sci.*, pp. 23769-23772, 2018.
[23] C. Slamet, A. R. Atmadja, D. S. Maylawati, R. S. Lestari, W. Darmalaksana, and M. A. Ramdhani, "Automated Text Summarization for Indonesian Article Using Vector Space Model," *IOP Conference Series: Materials Science and Engineering*, 2018.
[24] R. Y. Tsarev, A. S. Chernigovskiy, E. A. Tsareva, V. V. Brezitskaya, A. Y. Nikiforov, and N. A. Smirnov, "Combined string searching algorithm based on knuth-morris-pratt and boyer-moore algorithms," *IOP Conference Series: Materials Science and Engineering*, 2016.
[25] R. Fitriyanto, A. Yudhana, and S. Sunardi, "Implementation SHA512 Hash Function and Boyer-Moore String Matching Algorithm for Jpeg/exif Message Digest Compilation," *J. Online Inform.*, vol. 4, no. 1, pp. 16-23, 2019.
[26] F. T. Waruwu and R. Mandala, "Comparison of Knuth Morris Pratt and Boyer Moore Algorithms in Matching Strings in Nias Language Dictionary Application," *J. Ilm. INFOTEK*, vol. 1, no. 1, 2016.
[27] rachmad fitriyanto, A. Yudhana, and S. Sunardi, "Implementation SHA512 Hash Function and Boyer-Moore String Matching Algorithm for Jpeg/exif Message Digest Compilation," *J. Online Inform.*, vol. 4, no. 1, pp. 16-23, 2019.
[28] J. Bhandari and A. Kumar, "String Matching Rules Used by Variants of Boter-Moore," *J. Glob. Res. Comput. Sci.*, vol. 5, no. 1, pp. 8-11, 2014.
[29] R. F. Rahmat, D. F. Prayoga, D. Gunawan, and O. S. Sitompul, "Boyer-Moore Algorithm in Retrieving Deleted Short Message Service in Android Platform," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 308, no. 1, 2018.
[30] L. S. Riza, M. I. Firmansyah, H. Siregar, D. Budiana, and A. Rosales-Pérez, "Determining strategies on playing badminton using the Knuth-Morris-Pratt algorithm," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 16, no. 6, pp. 2763-2770, 2018.
[31] R. A. Baeza-Yates and M. Régnier, "Average running time of the Boyer-Moore-Horspool algorithm," *Theor. Comput. Sci.*, vol. 92, no. 1, pp. 19-31, 1992.
[32] L. Jun, Z. Zhuo, M. Juan, and L. Xingfeng, "Multi-pattern Matching Methods Based on Numerical Computation," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 11, no. 3, pp. 1497-1505, 2013.
[33] A. Kusnadi and A. K. Wicaksono, "Perbandingan Algoritma Horspool dan Algoritma Zhu-Takaoka dalam Pencarian String Berbasis Desktop," *J. Ultim. Comput.*, vol. 9, no. 1, pp. 12-16, 2017.

BIOGRAPHIES OF AUTHORS

Yana Aditia Gerhana is a lecturer from Informatic Engineering Faculty of Science and Technology UIN Sunan Gunung Djati Bandung in Indonesia, specialized in artificial intelligence information sistem and software engineering.

Nur Lukman is a lecturer from Informatic Engineering Faculty of Science and Technology UIN Sunan Gunung Djati Bandung in Indonesia, specialized in Information System and Computer Science.

Arief Fatchul Huda is a lecturer from Mathematics Faculty of Science and Technology UIN Sunan Gunung Djati Bandung in Indonesia, specialized in Image Processing and Spatio Temporal Analysis.

Cecep Nurul Alam is a lecturer from Informatic Engineering Faculty of Science and Technology UIN Sunan Gunung Djati Bandung in Indonesia, specialized Information System and Computer Science.

Undang Syaripudin is a lecturer from Informatic Engineering Faculty of Science and Technology UIN Sunan Gunung Djati Bandung in Indonesia, specialized Information System and Computer Science.

Devi Novitasari is a Bachelor of Engineering of Sunan Gunung Djati State Islamic University In 2018. After her bachelor degree, she gained considerable experience. Currently, she work in company property in bandung as digital marketing and data administration.