Jurnal-JRI

by Dyah Nurul Afiyah

Submission date: 27-Jul-2022 10:31PM (UTC+0900)

Submission ID: 1875815838

File name: Riskans_-_COMPARATIVE_ANALYSIS_OF_PATHFINDING.docx (224.83K)

Word count: 3537

Character count: 17950

JURNAL RISET INFORMATIKA **P-ISSN: 2656-1743 |E-ISSN: 2656-1735**
Vol. 4, No. 3. June 2022 DOI: https://doi.org/10.34288/jri.v4i3.XXX

Accredited rank 3 (SINTA 3), excerpts from the decision of the Minister of RISTEK-BRIN No. 200/M/KPT/2020

COMPARATIVE ANALYSIS OF PATHFINDING ARTIFICIAL INTELLIGENCE USING DIJKSTRA AND A* ALGORITHMS BASED ON RPG MAKER MV

Riska Nurtantyo Sarbini[1], Irdam Ahmad[2], Romie Oktovianus Bura[3], Luhut Simbolon[4]

Defence Science
Universitas Pertahanan
riskanurtantyosarbini@gmail.com[1], irdam.ahmad@idu.ac.id[2], romiebura@idu.ac.id[3],
lsimbolon@427gmail.com[4]

Abstrak

Dalam sebagian besar game, kecerdasan buatan fungsi penemu jalan dibutuhkan untuk menemukan tercepat untuk dilalui hal tersebut penting untuk banyak permainan komputer, khususnya permainan Role Playing Game (RPG). Algoritma pathfinding yang diimplementasikan pada game ini adalah algoritma A dan algoritma djiksa. Tujuan dari penelitian ini adalah untuk menguji coba sistem kecerdasan buatan untuk melakukan pencarian rute menggunakan algoritma A* dan algoritma dijkstra berbasis RPG Maker MV. Hasil penelitian Dari waktu yang didapatkan, pada percobaan pada 8 titik dengan mekanisme Pathfinding menggunakan algoritma A* lebih cepat dalam menemukan rute terdekat dengan catatan waktu 00:07:54 sedangkan menggunakan algoritma Djikstra 00:35:26. Hal tersebut dikarenakan pemberian bobot berlipat pada titik yang tidak dapat dilalui hal tersebut menyebabkan proses perhitungan biaya jalan menjadi lebih cepat dan efisien. Catatan waktu yang dibutuhkan menunjukkan jarak antar titik pencarian.*

Kata kunci: Kecerdasan Buatan-1; Pathfinding-2; Algoritma Djikstra-3; Algoritma A-4*

Abstract

In most games, a pathfinding artificial intelligence is required for the fastest discovery to be traversed. It is an important thing for many video games, particularly for Role Playing Game (RPG). The algorithm pathfindings implemented in this game are A* and Dijkstra Algorithms. This study aims to test an artificial intelligence system for discovering routes using the A* and Dijkstra algorithms based on RPG Maker MV. The result showed that from the time obtained, in the experiment on 8 nodes using the Pathfinding mechanism of A* algorithm has faster result in discovering the nearest route with the time of 00:07:54. Whereas, Dijkstra Algorithm has a 00:35:26 time result. It indicates that the multiple weighting in the impassable nodes caused the cost calculation process becomes faster and more efficient. The time record needed represents the distance between the search nodes.

Keywords: Artificial Intelligence-1; Pathfinding-2; Dijkstra Algorithm-3; A* algorithm-4

INTRODUCTION

When artificial intelligence gets into various fields, especially game applications, it presents interesting experiences for the users (Zhao, M., 2020). Artificial Intelligence (AI) is used in games to provide more interesting and interactive experiences (Hammedi, S. et. al, 2020). Through the intelligent technology deep improvement, artificial intelligence (AI) has been the core technical for improving the capability in playing a game, and also as the main value of the game promotion that can give more deep experience in playing game (Tang, C. et. al, 2020). AI is the main component in a game and it needs to be careful development and adjusted regularly. The role affects toward capacity and memory used in a game. AI is an important component that often impacts the success or failure of a game.

Various AI techniques are used in a game, such as the use of pathfinding function, decision making, intelligent narrative technology, and character intelligence (Hammedi, S. et. al, 2020). Artificial Intelligence (AI) that is needed for pathfinding is assumed as an important thing in computer games, particularly in Role Playing Game. It has been the main research area in video games for some decades (Iskandar, U.A.S. et. al, 2020). Usually, it is used as the core of the artificial intelligence movement system in computer games.

In this situation, the algorithm pathfindings that are commonly implemented in a game are Dijkstra and A* Algorithms.

Dijkstra algorithm is one algorithm that is often used to solve the pathfinding problem using the principle of determining the first node to the next that keeps connecting until the target node. The basic of this algorithm is based on the bandwidth allocation of nodes (Waleed, S. et. al, 2017). This algorithm is used to discover the shortest way based on the smallest weight starting from the departure node to the others. As an example, the building and monument as the point and the road as the lines, so the Dijkstra will calculate entire lines by the smallest weight from the greedy algorithm. It includes the route finder algorithm used to solve the problem of the shortest way in one node source that has no negative side cost and produces shortest way from the tree. This algorithm is often used in a routing process (Wahyunngsih, D, et. al, 2018).

Similar to the Dijkstra algorithm, the A* algorithm is another pathfinding method used in this game to discover the shortest way to prevent static or dynamic obstacles (Sazaki, Y. et. al, 2017). A* algorithm is often used for heuristics finding of an optimal path on the track. "Heuristics prediction" h(x) provides the best route prediction through the knot. It visits the nodes in this order of heuristic estimate (Rachmawati, D. et. al 2018). In improving pathfinding ability, some researchers used A* algorithm in a Real-Time Strategy Game (Chen, J. H. et. al, 2013).

In this research, an improvement approach of Artificial Intelligence (AI) is proposed to analyze the Pathfinding using A* and Dijkstra algorithms. It aims to test the time effectiveness in artificial intelligence for pathfinding in a Role-Playing game using these two algorithms.

RESEARCH METHODS

The method used in this system improvement is System Development Life Cycle (SDLC). It is the process for designing, developing, testing the high quality software. The aim of SDLC is to provide the structured flow in assisting high quality software production, fulfill the user expectation, acquire the model of software life cycle, and compare its performance (Saravanan, T. et. al, 2020). One of the most important phase of SDLC is the quality assurance or testing phase (Sinha, A. et. al, 2021).

Types of research

This research is a kind of software engineering through SDLC phases of waterfall model. The phases are designing, analyzing, testing, implementing, and analyzing of observation data result (quantitative).

Time and Place of Research

This research was done on March 2022 and the software construction used computer laboratory PC desktop on the application was ready to be used. Next, installation step on the smartphone.

Research Target / Subject

Research target is to construct stable application on the smartphone then continue to test the location node search using pathfinding mechanism of A* and Dijkstra algorithms.

Procedure

Research procedures started from the application making and continue to analysis the nearest route search data that related to the duration needed to discover the nearest route, the explanation can be seen as follows:

The software engineering model used the SDLC with the waterfall model (Trivedi, P. et.al, 2013). Software making is the most important project management. Besides, the software process model is an important tool to get standards, especially in the game digital software making. The creation of a software model for the entire life cycle of software development (SDLC) by the Waterfall model must be efficient for the software team to get productivity (output) of 80% easily through the reduction of software development. In the end, it can increase the software process performance (Iqbal, M. et. al., 2009). The steps that get through the game-making involve designing, analyzing, designing, implementing, and testing. It can be seen in the figure 1 as follows :

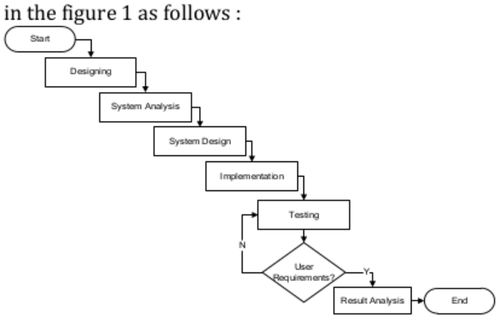

Figure 1. The steps of System Development using Waterfall Model

Explanation from Picture 1 as follows :

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v4i3.XXX
Accredited rank 3 (SINTA 3), excerpts from the decision of the Minister of RISTEK-BRIN No. 200/M/KPT/2020

1. Designing

In this step, system specification will be considered and constructed based on the user requirements. There are some tasks classification that must be done, as follows:

a. Collecting some information requirements related to the system that will be constructed, such as game observation.

b. Determining the objectives of the program with focusing on the specific problems to be solved, that is designing a stable game.

c. Determining the components inside the game which is related to the artificial intelligence ability using A* and Dijkstra algorithms.

2. System Analysis

The steps involve:

a. Identifying the problems of system description and running system explanation.

b. Identifying and process analyzing the requirements on the Role-Playing mechanism.

c. Identifying the need analysis, such as doing system requirements checklist for functional and non-functional needs towards the implementation of game artificial intelligence.

d. Identifying and analyzing of alternative solution on the system that will be constructed.

3. System Design

In this step, there are some activities as follows:

a. Application architecture design or sitemap.

b. Input and output designs involve identification and layout making.

c. Process design involve process identification and system process scenario, then continued in modelling.

d. Process design involve process identification and system process scenario, then modeled using DFD (Data Flow Diagram).

e. Data base design involve identification of table, entity, ERD making (Entity Relation Diagram).

f. Interface design involve interface identification and creating layout used.

g. Formulate the implementation of RPG through pathfinding mechanism.

4. Implementation

Database is used to save the input and output data from the game system. The database from this application uses RPG Maker MV. The design of database as the knowledge base is the storage basic concept and game storyline. Data base design starts from creating the tables, determining the keys in each table, relating one table to another, etc. use database application program from the RPG Maker MV program. This Game creation includes the database system and editor map.

5. Testing

The testing steps of this research are:

a. Verification: observing the suitability between design and result.

b. Validation: testing the game function suitability of RPG and the correlation between artificial intelligence utilization with pathfinding mechanism.

c. The testing is intended for acquiring the performance of pathfinding mechanism that has A* and Dijkstra algorithms function and observing the results.

Data, Instruments, and Data Collection Techniques

Processed data are HH:MM:SS which are obtained by the node route search result. There will be discovering and comparing processes using A* and Dijkstra algorithms. These aim to observe the level of time efficiency that is required in the discovery process. Technically, the comparison is through the author creating 8 nodes to be tested in an entire map on a mini-game of transportation and also the NPC that has pathfinding abilities based on A* and Dijkstra algorithms in the same game but different algorithms. These 8 nodes and NPC positions can be seen in Picture 2 as follows:

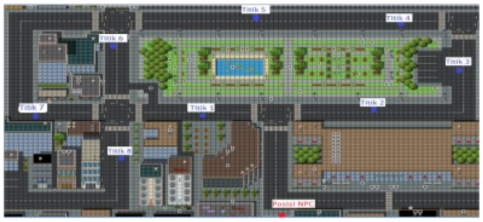

Figure 2. Analyze 8 nodes targets on map.

From the figure 2, it shows the red sentence is the first node of NPC and the blue one is 8 nodes that will be tested on the time required.

Data analysis technique

Quantitative research is the priority analysis that focused on numbers, starting from collecting data, interpreting the data obtained and presenting the result (Arikunto, 2006). In this research, the author used a quantitative technique to analyze the data through the time data required to reach the point target. 16 data will be obtained based on the distribution of 8 nodes using the Dijkstra algorithm and 8 nodes using A* algorithm.

RESULTS AND DISCUSSION

The ability of artificial intelligent using pathfinding method will be located on NPC in discovering Path, as the first and last nodes where NPC located on the coordinate of (22,51).

Then, it continues to initialization process of value discovery from the start node to the target (open_nodes, total_cost).

Next, adding node lists to the Open list which expected to be traversed. It can be seen in Table 1 as follows :

Table 1. List of Open Nodes Traversed (Coordinate Nodes)

List of Open Nodes Traversed (Coordinate Nodes)								
22, 22	23, 22	24, 22	25, 22	26, 22	27, 22	28, 22	29, 22	30, 22
22, 23	23, 23	24, 23	25, 23	26, 23	27, 23	28, 23	29, 23	30, 23
22, 24	23, 24	24, 24	25, 24	26, 24	27, 24	28, 24	29, 24	30, 24
22, 25	23, 25	24, 25	25, 25	26, 25	27, 25	28, 25	29, 25	30, 25
22, 26	23, 26	24, 26	25, 26	26, 26	27, 26	28, 26	29, 26	30, 26
22, 27	23, 27	24, 27	25, 27	26, 27	27, 27	28, 27	29, 27	30, 27
22, 28	23, 28	24, 28	25, 28	26, 28	27, 28	28, 28	29, 28	30, 28
22, 29	23, 29	24, 29	25, 29	26, 29	27, 29	28, 29	29, 29	30, 29
22, 30	23, 30	24, 30	25, 30	26, 30	27, 30	28, 30	29, 30	30, 30
22, 31	23, 31	24, 31	25, 31	26, 31	27, 31	28, 31	29, 31	30, 31
22, 32	23, 32	24, 32	25, 32	26, 32	27, 32	28, 32	29, 32	30, 32
22, 33	23, 33	24, 33	25, 33	26, 33	27, 33	28, 33	29, 33	30, 33
22, 34	23, 34	24, 34	25, 34	26, 34	27, 34	28, 34	29, 34	30, 34
22, 35	23, 35	24, 35	25, 35	26, 35	27, 35	28, 35	29, 35	30, 35
22, 36	23, 36	24, 36	25, 36	26, 36	27, 36	28, 36	29, 36	30, 36
22, 37	23, 37	24, 37	25, 37	26, 37	27, 37	28, 37	29, 37	30, 37
22, 38	23, 38	24, 38	25, 38	26, 38	27, 38	28, 38	29, 38	30, 38
22, 39	23, 39	24, 39	25, 39	26, 39	27, 39	28, 39	29, 39	30, 39
22, 40	23, 40	24, 40	25, 40	26, 40	27, 40	28, 40	29, 40	30, 40
22, 41	23, 41	24, 41	25, 41	26, 41	27, 41	28, 41	29, 41	30, 41
22, 42	23, 42	24, 42	25, 42	26, 42	27, 42	28, 42	29, 42	30, 42
22, 43	23, 43	24, 43	25, 43	26, 43	27, 43	28, 43	29, 43	30, 43
22, 44	23, 44	24, 44	25, 44	26, 44	27, 44	28, 44	29, 44	30, 44
22, 45	23, 45	24, 45	25, 45	26, 45	27, 45	28, 45	29, 45	30, 45
22, 46	23, 46	24, 46	25, 46	26, 46	27, 46	28, 46	29, 46	30, 46
22, 47	23, 47	24, 47	25, 47	26, 47	27, 47	28, 47	29, 47	30, 47
22, 48	23, 48	24, 48	25, 48	26, 48	27, 48	28, 48	29, 48	30, 48
22, 49	23, 49	24, 49	25, 49	26, 49	27, 49	28, 49	29, 49	30, 49
22, 50	23, 50	24, 50	25, 50	26, 50	27, 50	28, 50	29, 50	30, 50
22, 51	23, 51	24, 51	25, 51	26, 51	27, 51	28, 51	29, 51	30, 51

The next process is getting the target nodes and create direction of MoveRoute (while path_x!=src_x|| path_y!=src_y), that is path = (2, 2, 2, 2, 2, 4, 4, 4, 4, 4, 4, 4, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 4,), where the values are given DOWN = 2, LEFT = 4, RIGHT = 6, UP = 8.

The experiment result of pathfinding to observe the level of time-efficient needed in the discovery process uses the comparison between A* and Dijkstra algorithms. Technically, the author will compare by creating 8 nodes that will be tested in the entire map of transportation mini-game and the NPC that has the pathfinding ability based on A* and Dijkstra algorithms in a similar game but different algorithms. The 8 nodes of NPC position can be seen in Picture 2 the result of Table 2 is as follows:

Table 2. Time results were obtained from the 8 target nodes using A* and Dijkstra algorithms.

Number.	Coordinate	Djikstra	A*
1	41, 31	00:00:23	00:00:06
2	76, 29	00:01:05	00:00:14
3	94, 18	00:05:09	00:00:55
4	82, 5	00:10:00	00:02:23
5	52, 3	<	00:03:03
6	22, 11	00:10:00	00:00:41
7	6, 31	<	00:00:17
8	24, 43	00:04:47	00:00:15
		00:02:03	
		00:01:59	
Total Time		00:35:26	00:07:54

From Table 2, the testing result of the 8 nodes experiment can be concluded that on the NPC using A* and Dijkstra algorithms methods in 4,5 nodes experiment, the discovery process needs longer time. The result may cause there is bigger terrain (obstacle) of a quite wide park among these nodes.

The NPC Dijkstra algorithm of the 4,5 nodes experiment is stopped (not continued). It is because of spending a too long time. The experiment process stopped at 10 minutes because it is not efficient and will disturb the game application. It might be caused by the bigger terrain (obstacle) of a quite wide park among these nodes. The graphic comparison lines can be seen in Picture 3 as follows :

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v4i3.XXX
Accredited rank 3 (SINTA 3), excerpts from the decision of the Minister of RISTEK-BRIN No. 200/M/KPT/2020

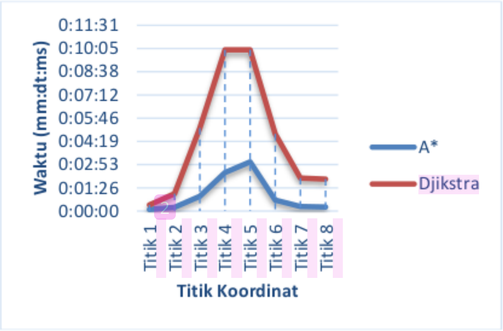

Figure 3. Graphic Comparison Using A* and Dijkstra algorithms.

Nodes 1,2 is the nearest nodes of NPC where the obstacle level is only other NPC transportation and some terrain nodes. The A* algorithm NPC needs some seconds to find the target coordinate. A similar case is also found on Dijkstra algorithm NPC.

The process nodes of 3,4,5,6 need a longer time due to the nearest nodes with the city park in a game. Park has quite a lot of terrain of object NPC character or transportation and the objects of plants, chair, pool, and trees. Meanwhile, the use of the Dijkstra algorithm needs a very long time so the process stopped at 10 minutes. It is because of spending a very long time.

7,8 nodes are not quite close to the park, but it has a long route. Dijkstra algorithm needs a longer time rather than the A* algorithm.

CONCLUSIONS AND SUGGESTIONS

Conclusion

According to the time obtained, the experiment of 8 nodes using A* algorithm Pathfinding mechanism has faster route in discovering the nearest route, with the time record of 00:07:54, whereas Dijkstra algorithm with the time record of 00:35:26. The reason is because multiple weighing on the nodes cannot be traversed. It causes the road cost calculation process is faster and more efficient.

Time record needed by NPC does not represent the distance between NPC and the discovery nodes, but it is the route discovery process to the target. It can be seen on the Lag process in a game before finding.

Suggestion

Technically, the game based RPG Maker MV application, where the artificial intelligence implemented is based on the author source code, may produce different result if testing to the other author. It needs improvement using other based application, such as unity3D, game maker, and so on.

REFERENCES

Zhao, M. (2020). Analysis on the Connection Between Nonplayer Character And Artificial Intelligence. *2020 International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI)*, pp. 105-108. doi: 10.1109/ICHCI51889.2020.00030.

Hammedi, S., Essalmi, F., Jemni, M., & Qaffas, A. A. (2020). An investigation of AI in games: educational intelligent games vs non-educational games. *2020 International Multi-Conference on: "Organization of Knowledge and Advanced Technologies" (OCTA)*, pp. 1-4. doi: 10.1109/OCTA49274.2020.9151738.

Tang, C., Wang, Z., Sima, X., & Zhang, L. (2020). Research on Artificial Intelligence Algorithm and Its Application in Games. *2020 2nd International Conference on Artificial Intelligence and Advanced Manufacture (AIAM)*, pp. 386-389. doi: 10.1109/AIAM50918.2020.00085.

Iskandar, U. A. S., Diah, N. M., & Ismail, M. (2020). Identifying Artificial Intelligence Pathfinding Algorithms for Platformer Games. *2020 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, pp. 74-80. doi: 10.1109/I2CACIS49202.2020.9140177.

Sazaki, Y., Primanita, A., & Syahroyni, M. (2017). Pathfinding car racing game using dynamic pathfinding algorithm and algorithm A. *2017 3rd International Conference on Wireless and Telematics (ICWT)*, pp. 164-169. doi: 10.1109/ICWT.2017.8284160.

Waleed, S., Faizan, M., Iqbal, M., & Anis, M. I. (2017). Demonstration of single link failure recovery using Bellman Ford and Dijikstra algorithm in SDN. *2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT)*, pp. 1-4. doi: 10.1109/ICIEECT.2017.7916533.

Wahyuningsih, D., & Syahreza, E. (2018). Shortest Path Search Futsal Field Location With Dijkstra Algorithm. *Indonesian Journal of Computing and Cybernetics Systems*, 12(2), 161-170.

Parapat, M. N., Kusbianto, D., & Rahmad, C. (2017). Rancang Bangun Aplikasi Pencarian Rute Terpendek Jasa Kiriman Barang Berbasis

Mobile Dengan Metode Algoritma Dijkstra. *Jurnal Informatika Polinema,* 3(3), 15-21. DOI:10.33795/jip.v3i3.28.

Rachmawati, D., & Gustin, L. (2020). Analysis of Dijkstra's Algorithm and A* Algorithm in Shortest Path Problem. *Journal Physics: Conference Series*, 15(6), 12-18.

Chen, J. H., Shih, T. K., & Chen, J. Y. (2012). To develop the ubiquitous adventure RPG (role play game) game-based learning system. *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2973-2978. doi: 10.1109/ICSMC.2012.6378247.

Saravanan, T., Jha, S., Sabharwal G., & Narayan, S. Comparative Analysis of Software Life Cycle Models. *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pp. 906-909. doi: 10.1109/ICACCCN51052.2020.9362931.

Sinha, A., & Das, P. (2021). Agile Methodology Vs. Traditional Waterfall SDLC: A case study on Quality Assurance process in Software Industry. *2021 5th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)*, pp. 1-4. doi: 10.1109/IEMENTech53263.2021.9614779.

Trivedi, P. & Sharma, A. (2013). A comparative study between iterative waterfall and incremental software development life cycle model for optimizing the resources using computer simulation. *2013 2nd International Conference on Information Management in the Knowledge Economy*, pp. 188-194.

Iqbal, M., & Rizwan, M. (2009) Application of 80/20 rule in software engineering Waterfall Model. *2009 International Conference on Information and Communication Technologies*, pp. 223-228. doi: 10.1109/ICICT.2009.5267186.

Arikunto, S. (2006). Suatu Pendekatan Praktik (Revisi VI). Jakarta: PT Rineka Cipta.

Jurnal-JRI

ORIGINALITY REPORT

10%	8%	3%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	pdfs.semanticscholar.org Internet Source	3%
2	media.neliti.com Internet Source	2%
3	Sanda Hammedi, Fathi Essalmi, Mohamed Jemni, Alaa A. Qaffas. "An investigation of AI in games: educational intelligent games vs non-educational games", 2020 International Multi-Conference on: "Organization of Knowledge and Advanced Technologies" (OCTA), 2020 Publication	1%
4	www.researchgate.net Internet Source	1%
5	Umar Affandi Shahrin Iskandar, Norizan Mat Diah, Marina Ismail. "Identifying Artificial Intelligence Pathfinding Algorithms for Platformer Games", 2020 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), 2020 Publication	1%

6	www.simplilearn.com Internet Source	<1 %
7	Rully Pramudita, Hery Heryanto, Rahmadya Trias Handayanto, Didik Setiyadi, Rita Wahyuni Arifin, Nadya Safitri. "Shortest Path Calculation Algorithms for Geographic Information Systems", 2019 Fourth International Conference on Informatics and Computing (ICIC), 2019 Publication	<1 %
8	Bima Septiantoro, Rudy Agus Gemilang Gultom, Amarulla Octavian. "Pengaruh Industri Media Nasional terhadap Media Warfare", Jurnal Studi Komunikasi dan Media, 2018 Publication	<1 %
9	baixardoc.com Internet Source	<1 %
10	download.garuda.kemdikbud.go.id Internet Source	<1 %
11	download.garuda.ristekdikti.go.id Internet Source	<1 %
12	www.igi-global.com Internet Source	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		