Bulletin of Computer Science and Electrical Engineering
Vol. 2, No. 2, December 2021, pp. 74~83
ISSN: 2722-7324, DOI: 10.25008/bcsee.v2i2.1146

Comparison Analysis of Breadth First Search and Depth Limited Search Algorithms in Sudoku Game

Tirsa Ninia Lina[1], Matheus Supriyanto Rumetna[2]
[1,2]Faculty of Computer Science, Victory University of Sorong, Indonesia

Article Info

Article history:

Received Dec 13, 2021
Revised Dec 20, 2021
Accepted Dec 30, 2021

Keywords:

Artificial Intelligence
Breadth First Search
Depth Limited Search
Sudoku Game
Information Technology

ABSTRACT

Sudoku is a game that sharpens the brain and is very well known. But the problem faced in this condition is how we can find a solution for the completion of this game. Problems with the Sudoku game can be solved by using the concept of Artificial Intelligence (AI). Some of the algorithms that can be used are the Breadth First Search (BFS) and Depth Limited Search (DLS) algorithms. The purpose of this research is to find a solution for Sudoku and make a comparative analysis of the search results of the two algorithms. The results obtained are application design in the form of a simulation of the completion of the Sudoku game problem with two algorithms. And it has been proven from the two algorithms that DLS is more efficient and faster than BFS. While BFS itself has advantages, in terms of a more structured and systematic search system that is able to find all possible numbers in each box. In this case, if a Sudoku question has more than one answer, then BFS will find it.

Corresponding Author:

Tirsa Ninia Lina,
Faculty of Computer Science,
Victory University of Sorong,
Basuki Rahmat Road, Km. 11, 5, Klasaman, Klawuyuk, Sorong City, West Papua 98416, Indonesia.
Email: tirsawp@gmail.com

1. INTRODUCTION

Artificial Intelligence (AI) is a part of computer science that studies how to make computers do jobs like and as well as humans do. At the beginning of its creation, the computer only functioned as a calculating tool. But along with the times, computer technology is increasingly being improved and until now AI is often used to find solutions or solutions [1],[2]. In the case of AI there is a space that contains all the states which can be called a state space. Conditions in the state space include: start state, goal state is a solution that is reached and needs to be checked whether it has reached the target, rules or rules that provide limitations on how to change a state into another state. The state is represented as a node, while the allowed steps or actions are represented by arcs [3],[4].

One of the popular puzzle games originating from Japan is Sudoku. Sudoku consists of 81 squares consisting of 9 columns and 9 rows. The big box is further divided into 9 partial squares, each of which consists of 3 x 3 squares (see Figure 1). Players may fill the boxes with numbers from 1 to 9. The condition is that there must be no repetition of numbers in one column, also in one row, and in any 3 x 3 partial box. As a starting point, some of the boxes have been filled with opening numbers. Players are welcome to continue. Unlike other number games, Sudoku does not

train arithmetic intelligence (the ability to add or subtract numbers), but Sudoku trains players' logical intelligence to complete Sudoku by filling all the boxes without violating the existing rules. A good Sudoku problem is one that only has exactly one answer. Sudoku is a puzzle game that deals with space and circumstances.

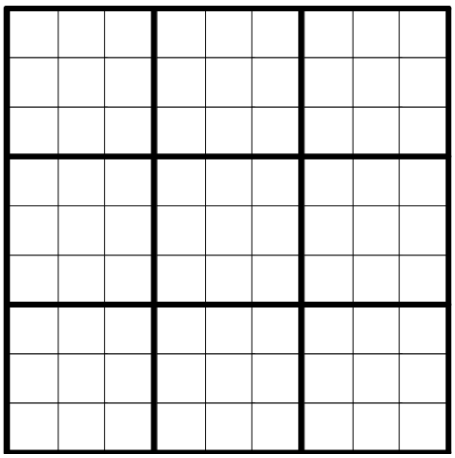

Figure 1. Sudoku game box

In AI there are several search techniques, such as the Breadth First Search (BFS) and Depth Limited Search (DLS) algorithms [2],[4],[5]. The Breadth First Search (BFS) algorithm is a search that broadens the solution space systematically and explores all the possibilities that exist at each level. And the Depth Limited Search (DLS) algorithm is a search that is carried out from the initial node in depth until the most recent node is found.

The advantage of the BFS algorithm is that there is no deadlock, if there is one solution then BFS will find it, and if there is more than one solution then the minimum solution will be found. The disadvantage is that it requires a lot of memory, because it stores all nodes in one tree, and it takes quite a long time, because it will test n levels to get a solution at the (n + 1) level. For more details, consider the illustration of BFS in Figure 2.

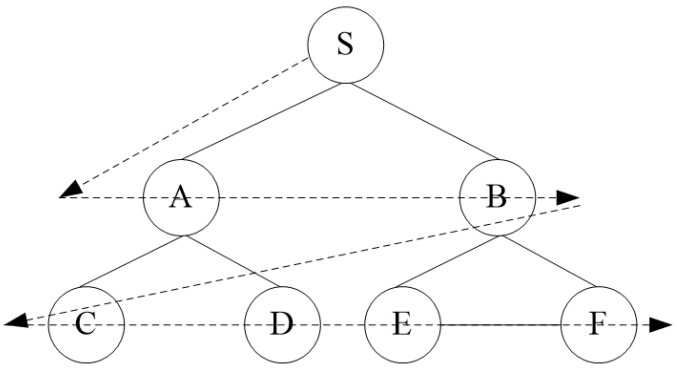

Figure 2. BFS algorithm

The advantage of the DLS algorithm is that it quickly reaches the depth of the search space. If it is known that the path to the solution of the problem will be long then DLS will not waste time performing a large number of 'shallow' states in the graph/tree problem. DLS is much more efficient for search space with multiple branches because it doesn't need to evaluate all nodes at a certain level in the open list. In addition, DLS requires relatively small memory because only the nodes on the active path are stored. The disadvantage is that before use, it must be known what the maximum level of a solution is. If the depth limit is too small than the solution depth level, then DLS cannot find the existing solution. That is, DLS can be incomplete if the depth limit is smaller than the solution level. For more details, consider the illustration of DLS in Figure 3.

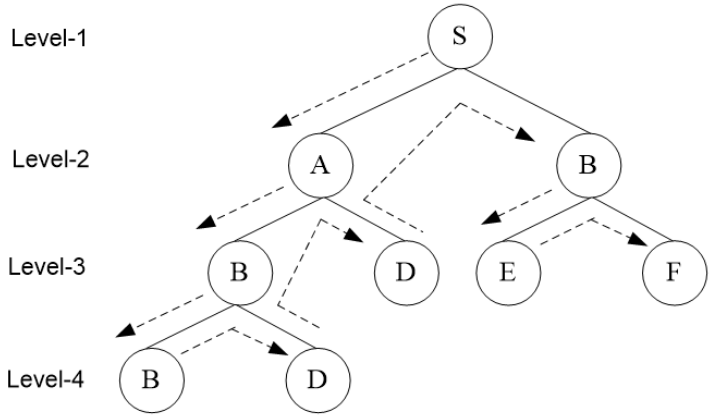

Figure 3. DLS algorithm

This research will create software that can be used to find solutions for the Sudoku game using BFS and DLS. In addition, a comparative analysis will be made regarding the time required to obtain a solution and the number of nodes stored and developed by these two algorithms.

2. RESEARCH METHOD

Research methods are divided into:

1) Data collection method

Collecting data by collecting literature, journals and readings related to research [6],[7],[8],[9],[10],[11],[12],[13],[14],[15],[16],[17],[18],[19],[20],[21].

2) Analysis Methods

Methods The analysis is carried out with the BFS and DLS algorithms [1],[3],[22],[23],[24],[25],[26],[27],[28],[29].

BFS is a search that is carried out by visiting each node systematically at each level until the goal state is found. Or in other words, a search that is carried out by visiting nodes at the same level so that the goal state is found [30],[31],[32],[33].

DLS is carried out from the initial node in depth to the most recent (dead-end) or until a solution is found. In other words, the branch or child node that is visited first. DLS limits the maximum level of a solution [34],[35],[36],[37],[38],[39],[40].

The detailed research method can be seen in Figure 4.

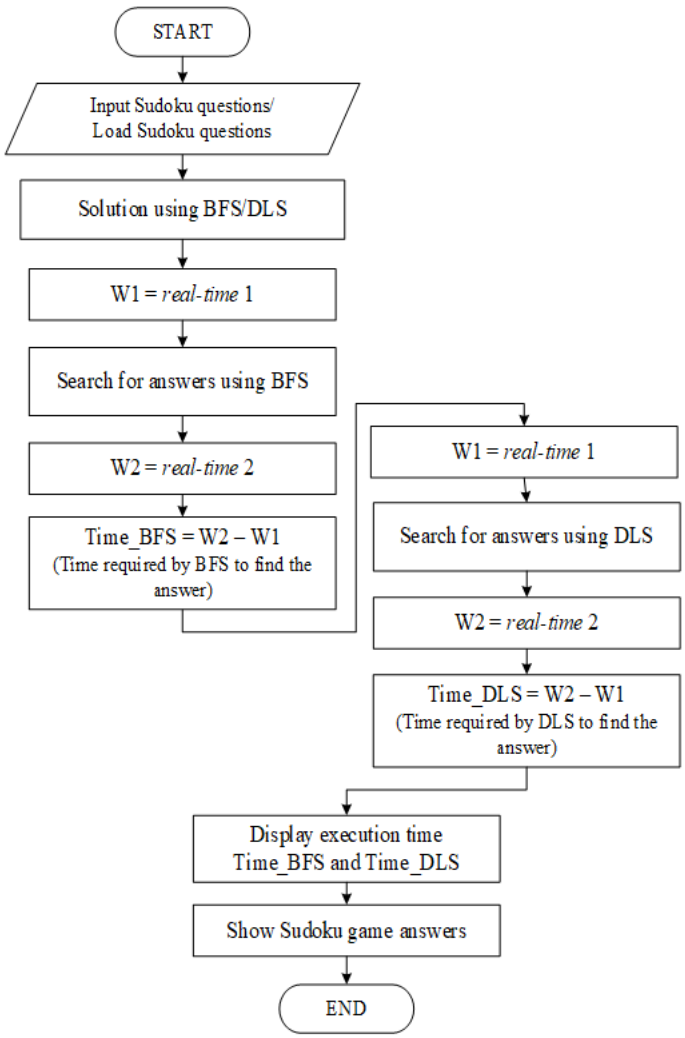

Figure 4. Flowchart research method

3. RESULTS AND DISCUSSION

For example, the Sudoku problem that you want to solve is shown in Figure 5 below.

	1			3		5		
	3		9		4	7		
8		9				4		
		6					7	
2			8		1			5
	8					2		
		7				8		4
		8	1		9		2	
		2		5			6	

Figure 5. Sudoku game questions

3.1. BFS to Complete Sudoku game

The process of finding Sudoku answers using the BFS is as follows [4],[7],[22],[41],[42]:

1) Search starts from an empty root node
2) From this root node, all possible numbers that can be entered in box-0 are developed without violating the Sudoku rules. Each of these possible numbers is entered into a single node at level-0.
3) If the next box is a Sudoku question box (a number that cannot be changed when working on a Sudoku problem), then the parent node does not need to develop child nodes and all parent nodes have the same child node, namely the node that contains the numbers in the question box.
4) Meanwhile, if the next box is the answer box, then develop all possible numbers that can be entered in the box without violating the Sudoku rules.
5) Continue developing and searching for nodes until level 80 is reached. The answer is a series of steps from node level-0 to node level-80.
6) If none of the nodes can reach the 80th level, then it is certain that the Sudoku problem has no solution.

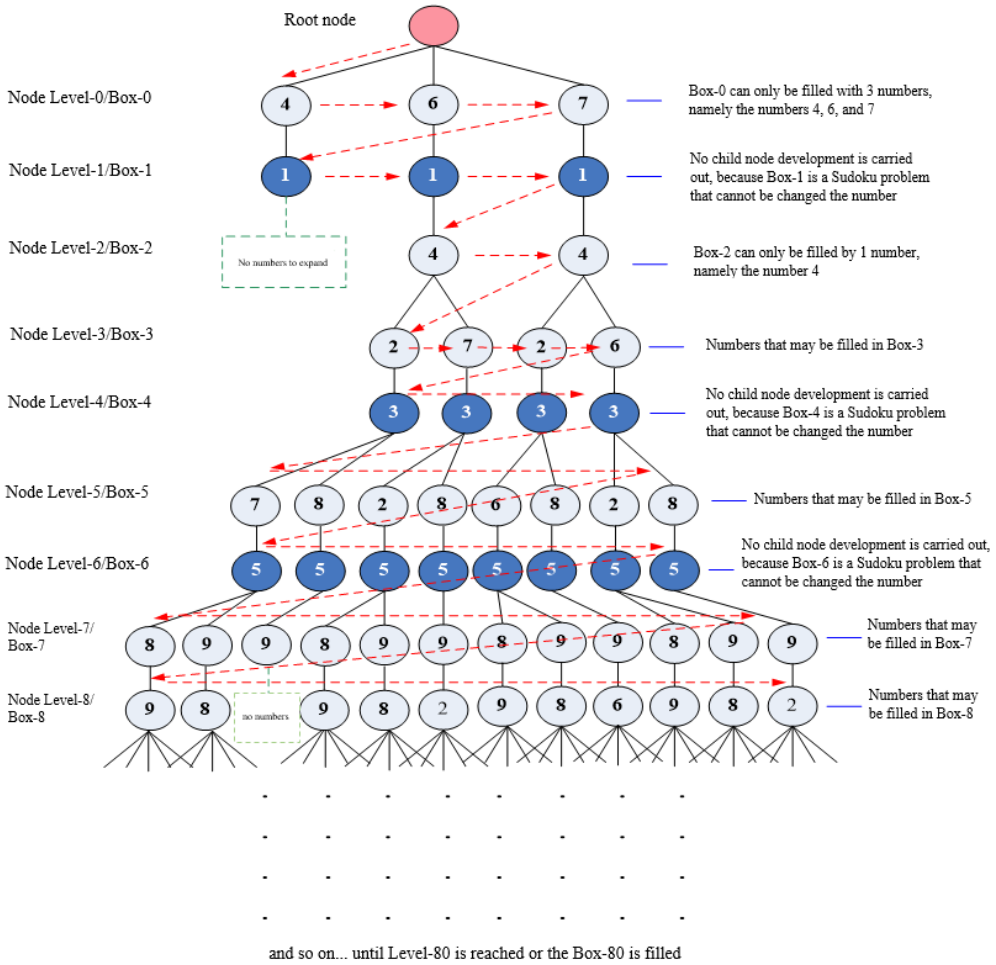

Figure 6. Sudoku game answer search process using BFS

Figure 6 shows that the search starts from an empty root node. The BFS lookup then looks for all possible numbers that can be placed in Box-0 and those numbers are 4, 6 and 7. These

numbers are then entered in the Level-0 child nodes. Then, the search and development of the node continues on the leftmost Level-0 node which contains the number 4. Because the next box (Box-1) is a Sudoku game question that cannot be changed its value, the Level-1 child node is not developed and is filled in immediately. by the number in Box-2, which is number 1. This also happens to other Level-1 nodes. The search then continues on the Level-1 node, from the leftmost node to the rightmost node and continues for nodes at the next level. If the BFS quest reaches the node-80, the Sudoku game answer has been found. However, if the BFS search does not find a way to reach node-80, then it is certain that the Sudoku game question does not have an answer.

3.2. DLS to Complete Sudoku game

The process of finding Sudoku answers using DLS is as follows[2],[3],[43],[44],[45],[46]:

1) The search starts from an empty root node.
2) From the root node, 1 possible number will be developed that can be entered in box-0 and does not violate the Sudoku rules. This number is a level-1 child node. Furthermore, the child nodes are developed from level-1, and so on.
3) If no child nodes can be expanded, because all numbers violate Sudoku rules, then backtrack to the parent node and develop another child node.
4) If the next box is a Sudoku problem (a number that cannot be changed when working on a Sudoku problem), then the parent node does not need to develop a child node and goes directly to the child node containing the Sudoku question and develops other child nodes.
5) Continue searching and expanding nodes until you reach the 80th level node. Sudoku's answer is a series of steps from node level-1 to node level-80.
6) If there is a backtrack on the root node, it can be ascertained that the Sudoku question does not have an answer.

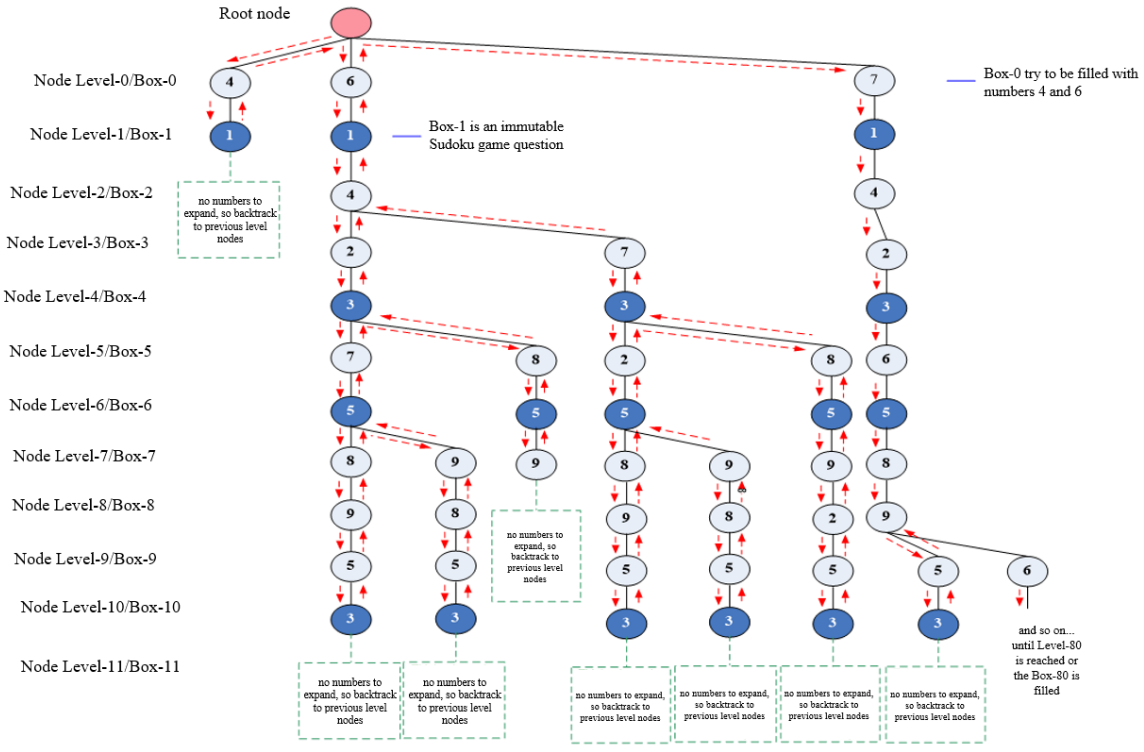

Figure 7. Sudoku game answer search process using DLS

Figure 7 shows that the search starts from the root node. The DLS search expands the Level-0 child node (Square-0) and tries to enter the first number from 1-9 which does not violate the rules of the Sudoku game. The number in question is number 4, so number 4 is placed in Box-0 or Level-0 node and traces that node. Then, the search develops child nodes of Level-0 nodes, namely Level-1 nodes. However, because Box-1 contains an immutable Sudoku game problem, the Level-1 node is immediately occupied by the number 1. The development of the child nodes continues at the Level-1 node. It turns out that no numbers can be placed on the Level-2 nodes because all placements of numbers 1-9 violate the rules of the Sudoku game. Therefore, the search returns (backtrack) from the Level-1 node to the Level-0 node. A search on the level-0 node cannot develop any other child nodes other than the number 1, because the level-1 node is a Sudoku game problem box. Therefore, the search backtracks again to the root node. On the root node, the search develops another child node. The next number that does not violate the rules of the Sudoku game is number 6. Place the number 6 on the Level-0 node and trace the node. The search develops a child node from the Level-0 node, namely the Level-1 node that contains the Sudoku game problem, namely number 1. Search for the node. Next, the search expands the child nodes of the Level-1 node and returns a node with the number 4 (because numbers 1, 2 and 3 violate the rules of the Sudoku game). The search traces the Level-2 node and continues on the next nodes. The answer is found when the search node reaches the Level-80 node. However, if there is a backtrack to the root node, it can be ascertained that the Sudoku game question does not have an answer.

The results of the search with BFS and DLS to find the Sudoku answer to the question in Figure 5 can be seen in Figure 8.

7	1	4	2	3	6	5	8	9
6	3	5	9	8	4	7	1	2
8	2	9	5	1	7	4	3	6
9	5	6	4	2	3	1	7	8
2	7	3	8	9	1	6	4	5
4	8	1	6	7	5	2	9	3
1	9	7	3	6	2	8	5	4
5	6	8	1	4	9	3	2	7
3	4	2	7	5	8	9	6	1

Figure 8. Sudoku game answers

We will compare the number of nodes developed and the time it takes for the two algorithms to come up with an answer for the Sudoku game. This comparison is done by testing the 3 questions that have been previously stored. The details can be seen in Table 1.

Table 1. Comparison results table

Questions name	Search	BFS	DLS
Question 1	Time	9.37391	0.60841
	Node	12867	1188
Question 2	Time	2.81128	2.49891
	Node	4700	4212
Question 3	Time	8.62366	1.49903
	Node	13201	2507

BFS search requires a higher number of nodes than DLS search to find a solution, because BFS search expands all possible numbers that can be placed in each box, from the first box to the last box, while DLS search expands the number possibilities in depth, or in other words DLS develops only 1 possible number in a box and immediately moves to the next box, expands again 1

possible number to the next and so on until all boxes are filled with numbers. If no number can be placed in an answer box because it violates the rules of permaSudoku, then DLS will backtrack to the previous box and change the number in the box with another number, before advancing again to the next box.

Because the number of nodes developed by BFS search for answers is more and more wasteful than DLS search, DLS search finds Sudoku game answers faster than BFS search. BFS has an advantage, in terms of a more structured and systematic search system that is able to find all possible numbers in each box. In this case, if a Sudoku game question has more than one answer, then BFS will find it. DLS search proved to be better than BFS search in the number of nodes required and the time it took to find Sudoku game answers.

4. CONCLUSION

After conducting the analysis, several conclusions can be drawn as follows:

1) In theory in the case of Sudoku game completion, DLS is more efficient and faster than BFS. While BFS has an advantage, in terms of a more structured and systematic search system that is able to find all possible numbers in each box. In this case, if a Sudoku game question has more than one answer, then BFS will find it.
2) Meanwhile, in the problems that have been searched for solutions, not all questions can be solved with these two algorithms or can not find answers.
3) The drawback of this research is that it is only limited to 3x3 levels for the Sudoku game, so we can add several existing levels, namely 6x6, 8x8, 12x12. Can also use Heuristic algorithms, A* algorithms, and others.

ACKNOWLEDGEMENTS

The researcher is grateful to those who have helped during this research, including the Information Systems Study Program, Faculty of Computer Science and the Research and Community Service Institute at Victory Sorong University.

REFERENCES

[1] A. Wibowo, . B., . L., and F. Fathurrahman, "Implementasi Algoritma Breadth First Search Dan Obstacle Detection Dalam Penelusuran Labirin Dinamis Menggunakan Robot Lego," *Ilmu Komput. dan Inf.*, vol. 4, no. 1, pp. 15–22, 2011, doi: 10.21609/jiki.v4i1.153.

[2] S. Tarmiandi, E. Z. Astuti, and S. Astuti, "Implementasi Algoritma Breadth First Search Pada Pencarian Rute Terpendek Tempat Kos Di Semarang Tengah," in *Seminar Nasional Sistem Informasi dan Teknologi Informasi*, 2018, pp. 524–528, [Online]. Available: http://www.sisfotenika.stmikpontianak.ac.id/index.php/sensitek/article/view/298.

[3] S. Lailiyah, A. Yusnita, and T. A. Panotogomo, "Penerapan Algoritma Depth First Search Pada Sistem Pencarian Dokumen," in *SNITT*, 2017, pp. 174–179.

[4] B. Prasetiyo and M. R. Hidayah, "Penggunaan Metode Depth First Search (DFS) dan Breadth First Search (BFS) pada Strategi Game Kamen Rider Decade Versi 0.3," *Sci. J. Informatics*, vol. 1, no. 2, pp. 161–167, 2014, doi: 10.15294/sji.v1i2.4022.

[5] A. S. M. Lumenta, "Perbandingan Metode Pencarian Depth-First Search, Breadth-First Search Dan Best-First Search Pada Permainan 8-Puzzle," *e-journal Tek. Elektro dan Komput.*, pp. 1–6, 2014.

[6] A. Hasibuan, D. S. Tambunan, and A. Info, "Design and Development of An Automatic Door Gate Based on Internet of Things Using Arduino Uno," *Bull. Comput. Sci. Electr. Eng.*, vol. 2, no. 1, pp. 17–27, 2021, doi: 10.25008/bcsee.v2i1.1141.

[7] T. Espinoza-Cordero, K. Ortiz-Cotrina, and ..., "Implementation of Electronic Medical Records System EQUALI to Improve Patient Care," *Bull. Comput. ...*, vol. 2, no. 1, pp. 9–16, 2021, doi: 10.25008/bcsee.v2i1.1144.

[8] N. Nikhlis, A. Iriani, and K. D. Hartomo, "Soft System Methodology (SSM) Analysis to Increase the Number of Prospective Students," *INTENSIF J. Ilm. Penelit. dan Penerapan Teknol. Sist. Inf.*, vol. 4, no. 1, pp. 63–74, 2020, doi: 10.29407/intensif.v4i1.13552.

[9] M. Rumetna, Supriyanto *et al.*, "PENERAPAN METODE SIMPLEKS UNTUK MENGHASILKAN KEUNTUNGAN MAKSIMUM PADA PENJUAL BUAH PINANG," *J. Dedication To Papua*

Community2, vol. 2, no. 1, pp. 75–86, 2019.

[10] M. S. Rumetna and T. N. Lina, "Pelatihan menghitung hasil penjualan rokok selama masa pandemi covid-19 menggunakan metode simpleks dan software pom-qm," *J. Pendidik. Dan Pemberdaya. Masy.*, vol. 8, no. 1, pp. 69–77, 2021, [Online]. Available: https://ejournal.unsri.ac.id/index.php/jppm/article/view/14110/pdf.

[11] V. Ngamelubun *et al.*, "Optimalisasi Keuntungan Menggunakan Metode Simpleks Pada Produksi Batu Tela," *Ris. Komput.*, vol. 6, no. 5, pp. 484–491, 2019.

[12] M. S. Rumetna *et al.*, "MENGHITUNG KEUNTUNGAN MAKSIMAL DARI PENJUALAN ROTI ABON GULUNG DENGAN MENGGUNAKAN METODE SIMPLEKS DAN SOFTWARE POM-QM," *J. Jendela Ilmu*, vol. 1, no. 1, pp. 6–12, 2020.

[13] M. S. Rumetna, D. Manongga, and A. Iriani, "PENERAPAN KNOWLEDGE CAPTURE UNTUK PROMOSI FAKULTAS MENGGUNAKAN SOFT SYSTEM METHODOLOGY (SSM) (STUDI KASUS : FAKULTAS TEKNIK , UNIVERSITAS VICTORY SORONG)," in *Prosiding Seminar Nasional Geotik*, 2017, pp. 106–116.

[14] M. S. Rumetna, T. N. Lina, R. R. Pakpahan, Y. Ferdinandus, F. S. Pormes, and J. E. Lopulalan, "Implementing Knowledge Management System to Improve Effectiveness of Faculty Activities," in *Bukittinggi International Conference on Education*, 2020, doi: 10.4108/eai.14-9-2020.2305670.

[15] R. Ong *et al.*, "Maksimalisasi Keuntungan Pada Usaha Dagang Martabak Sucipto Menggunakan Metode Simpleks Dan POM-QM," *Ris. Komput.*, vol. 6, no. 4, pp. 434–441, 2019.

[16] M. S. Rumetna, T. N. Lina, L. R. Tauran, T. Patty, A. Malak, and K. Yawan, "Penerapan Metode Simpleks pada Usaha Dagang Bintang Tiurma," *J. Innov. Inf. Technol. Appl.*, vol. 2, no. 01, pp. 28–36, 2020.

[17] M. S. Rumetna, "AUDIT LINGKUNGAN DAN PENGENDALIAN TEKNOLOGI INFORMASI PADA PT.XYZ," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 9, no. 2, pp. 753–768, 2018.

[18] M. S. Rumetna *et al.*, "BERBASIS WEBSITE PADA PERUSAHAN CENDRAWASIH WIPUTRA MANDIRI KOTA SORONG DESIGN OF A WEBSITE-BASED DEMAND INFORMATION SYSTEM IN CENDRAWASIH WIPUTRA MANDIRI COMPANY," *Elektro Luceat*, vol. 7, no. 1, pp. 10–19, 2021.

[19] M. S. Rumetna, M. Pieter, and M. Manurung, "APLIKASI PENGENALAN KARAKTER ALFANUMERIK MENGGUNAKAN ALGORITMA HAMMING DISTANCE," *Pros. SNATIF*, no. 4, pp. 77–84, 2017, [Online]. Available: https://media.neliti.com/media/publications/173678-ID-aplikasi-pengenalan-karakter-alfanumerik.pdf.

[20] M. S. Rumetna *et al.*, "PENDAMPINGAN DAN PELATIHAN PENERAPAN METODE SIMPLEKS PADA USAHA DAGANG BINTANG TIURMA," *J. Abdimas Bina Bangsa*, vol. 01, no. 02, pp. 205–214, 2020.

[21] P. Pratiwi *et al.*, "Pengembangan Aplikasi Mobile Augmented Reality untuk Mendukung Pengenalan Koleksi Museum," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 1, p. 147, 2018, doi: 10.25126/jtiik.201853891.

[22] H. Agung and N. Yunus Marselinus, "Implementasi Metode Breadth First Search danVikor pada Aplikasi Diagnosa Kerusakan Hardware Komputer," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 5, no. 2, pp. 46–53, 2016, doi: 10.32736/sisfokom.v5i2.38.

[23] M. S. Rumetna *et al.*, "Mengoptimalilasi keterbatasan sumber daya untuk memaksimalkan keuntungan penjualan es kelapa muda menggunakan metode simpleks dan software pom-qm," *Pengabdi. Masy.*, vol. 02, no. 02, pp. 136–149, 2019.

[24] T. N. Lina and M. S. Rumetna, "Analysis of Land Use Change in Bantul Regency Using Geoprocessing Technique," in *International Conference of Computer Science and Engineering Technology(ICCSET)*, 2018, pp. 506–512, doi: 10.4108/eai.24-10-2018.2280499.

[25] M. S. Rumetna and I. Sembiring, "PEMANFAATAN CLOUD COMPUTING BAGI USAHA KECIL MENENGAH (UKM)," in *Prosiding Seminar Nasional Geotik*, 2017, no. ISSN:2580-8796, pp. 1–9.

[26] M. S. Rumetna, "Audit Lingkungan Dan Pengendalian Teknologi Informasi Pada Pt. Xyz," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 9, no. 2, pp. 753–768, 2018, doi: 10.24176/simet.v9i2.2294.

[27] M. S. Rumetna and T. N. Lina, "Forecasting Number of Covid-19 Positive Patients in Sorong City Using the Moving Average and Exponential Smoothing Methods," *IJICS (International J. Informatics Comput. Sci.*, vol. 5, no. 1, pp. 37–43, 2021, doi: 10.30865/ijics.v5i1.2908.

[28] M. S. Rumetna, T. N. Lina, T. P. Sari, P. Mugu, A. Assem, and R. Sianturi, "Optimasi Jumlah Produksi Roti Menggunakan Program Linear Dan Software POM-QM," *Comput. Based Inf. Syst. J.*,

vol. 09, no. 01, pp. 42–49, 2021.
[29] M. S. Rumetna *et al.*, "OPTIMALISASI PENJUALAN NOKEN KULIT KAYU MENGGUNAKAN METODE SIMPLEKS DAN SOFTWARE POM-QM," *Comput. Based Inf. Syst. J.*, vol. 08, no. 02, pp. 37–45, 2020.
[30] M. S. Rumetna, E. Sediyono, and K. D. Hartomo, "Analisis Perubahan Tata Guna Lahan di Kabupaten Bantul Menggunakan Metode Global Moran's I," *J. Buana Inform.*, vol. 8, no. 4, pp. 225–234, 2017, doi: 10.24002/jbi.v8i4.1446.
[31] L. Sarmin *et al.*, "PENERAPAN METODE SIMPLEKS UNTUK MENGHITUNG KEUNTUNGAN MAKSIMUM PADA PENGRAJIN GELANG BESI PUTIH DI PASAR REMU SORONG," *J. KUADAS*, vol. 1, no. 2, pp. 1–7, 2018.
[32] M. S. Rumetna, T. N. Lina, L. Simarmata, L. Parabang, A. Joseph, and Y. Batfin, "Pemanfaatan POM-QM Untuk Menghitung Keuntungan Maksimum UKM Aneka Cipta Rasa (ACR) Menggunakan Metode Simpleks," in *GEOTIK*, 2019, pp. 12–22.
[33] T. N. Lina, B. S. Marlissa, M. S. Rumetna, and J. E. Lopulalan, "Penerapan Metode Simpleks Untuk Meningkatkan Keuntungan Produksi," *Ris. Komput.*, vol. 7, no. 3, pp. 459–468, 2020, doi: 10.30865/jurikom.v7i3.2204.
[34] M. S. Rumetna, "PEMANFAATAN SISTEM INFORMASI GEOGRAFI UNTUK DETEKSI DAERAH RAWAN LONGSOR DI KECAMATAN SIDOMUKTI, SALATIGA," *KUADAS*, vol. 1, no. 1, 2018.
[35] M. S. Rumetna and T. N. Lina, "Sistem Informasi Kampung Wisata Arborek Dengan Metode Waterfall," *Informatics Educ. Prof.*, vol. 5, no. 1, pp. 31–40, 2020.
[36] M. S. Rumetna, T. N. Lina, and A. B. Santoso, "RANCANG BANGUN APLIKASI KOPERASI SIMPAN PINJAM MENGGUNAKAN METODE RESEARCH AND DEVELOPMENT," *Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 11, no. 1, pp. 119–128, 2020.
[37] T. N. Lina *et al.*, "PENERAPAN METODE SIMPLEKS DALAM OPTIMALISASI KEUNTUNGAN HASIL PRODUKSI LEMON CINA DAN DAUN JERUK PURUT," *Elektro Luceat*, vol. 6, no. 1, 2020.
[38] M. S. Rumetna and T. N. Lina, "Pelatihan Penggunaan Sistem Inventory Data Barang Pada Gudang CV Tanaya," *JPM J. Pengabdi. Masy.*, vol. 1, no. 1, pp. 11–17, 2020.
[39] T. N. Lina *et al.*, "SISTEM INFORMASI E-ARSIP BERBASIS WEB (STUDI KASUS : PT HALEYORA POWERINDO CABANG SORONG)," *J. Jendela Ilmu*, vol. 1, no. 1, pp. 1–5, 2020.
[40] M. S. Rumetna, T. N. Lina, and J. E. Lopulalan, "A knowledge management system conceptual model for the sorong COVID-19 task force," *Int. J. Informatics Vis.*, vol. 4, no. 4, pp. 195–200, 2020, doi: 10.30630/joiv.4.4.418.
[41] M. S. Rumetna, E. E. Renny, and T. N. Lina, "Designing an Information System for Inventory Forecasting," *Int. J. Adv. Data Inf. Syst.*, vol. 1, no. 2, pp. 80–88, 2020, doi: 10.25008/ijadis.v1i2.187.
[42] M. S. Rumetna *et al.*, "Optimasi Pendapatan Pembuatan Spanduk dan Baliho Menggunakan Metode Simpleks (Studi Kasus : Usaha Percetakan Shiau Printing)," *J. Ris. Komput.*, vol. 7, no. 2, pp. 278–284, 2020, doi: 10.30865/jurikom.v7i2.1922.
[43] M. S. Rumetna, T. N. Lina, T. Aponno, A. Palisoa, and F. Singgir, "Penerapan Metode Simpleks Dan Software POM- QM Untuk Optimalisasi Hasil Penjualan Pentolan Bakso," *Ilm. Manaj. Inform. dan Komput.*, vol. 02, no. 03, pp. 143–149, 2018.
[44] T. N. Lina *et al.*, "PREMIUM DAN PERTALITE MENGGUNAKAN METODE MAXIMIZATION OF PROFIT ON PREMIUM AND PERTALITE BUSINESSES USING SIMPLEX METHODS AND POM-QM," *Elektro Luceat*, vol. 7, no. 1, pp. 1–9, 2021.
[45] M. S. Rumetna, "Pemanfaatan Cloud Computing Pada Dunia Bisnis: Studi Literatur," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 3, pp. 305–314, 2018, doi: 10.25126/jtiik.201853595.
[46] M. S. Rumetna, "KOMBINASI GNU PRIVACY GUARD DAN HAMMING DISTANCE UNTUK KEAMANAN EMAIL SERTA JALUR SERTIFIKASI COMBINATION OF GNU PRIVACY GUARD AND HAMMING DISTANCE FOR EMAIL SECURITY AND CERTIFICATION PATHS," *Elektro Luceat [November]*, vol. 7, no. 2, pp. 151–160, 2021.