Signal and Image Processing Letters
Vol. 3., No. 1, 2021, pp. 36-46
ISSN 2714-6677
https://simple.ascee.org/

IMU sensor based home search method using backtracking algorithm on hexapod robot in Indonesia fire extinguisher robot contest

Yesa Friti Maydevanti [a,1], Nuryono Satya Widodo [a,2,*]

[a] Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[1] yesa1700022041@webmail.uad.ac.id; [2] nuryono.sw@ee.uad.ac.id
* corresponding author

ARTICLE INFO

Keywords
IMU sensor
Backtracking algorithm
Indonesian fire fighting robot

ABSTRACT

Abstract-This paper describes the IMU sensor-based home search using a backtracking algorithm on a hexapod robot in the Indonesian fire fighting robot contest. The search for the homepage in this Indonesian legged robot contest is part of the robot's mission to achieve maximum points in the race. The algorithm discussed in this paper is a backtracking algorithm. Where the robot will return to the initial position by tracing solutions that meet the requirements. By using the IMU BNO055 sensor the robot will follow the direction that will depart then will guide the robot back to its home position. The results show that the backtracking algorithm is successfully implemented. So that the robot will still return home even though an error occurs when crossing the room intersection.

1. Introduction

The Indonesian Fire Extinguisher Robot Contest (KRPAI) is a national level robot contest held annually by the Directorate General of Higher Education (DIKTI). It was held for the first time since 2004 and has experienced many developments and improvements in terms of technology in the competition.[1] Universitas Ahmad Dahlan included one of the teams in the KRPAI division, namely ALJAZARI, in the Indonesian Robot Contest (KRI) held by KEMENDIKBUD using a 6-foot Robot or known as hexapods.

In the KRPAI division competition, ALJAZARI was able to carry out missions according to the rules in the competition by using the GY-26 compass sensor to determine direction, [2] however this sensor experienced interference when around buildings composed of steel rods because steel rods are a ferromagnetic material that can generate a magnetic field even though the trigger magnet has disappeared [3].

Therefore the BNO055 IMU Sensor is used which has the same working principle as the GY-26 compass to read the angle of view without being disturbed by a magnetic field, and can be used to find out where the robot is facing which is useful as a reference for knowing the starting position and direction of the robot in the hallway. By using the backtracking algorithm, the robot will return home without entering other rooms.

2. Research Method

2.1. System Design

2.1.1. Hardware Design

The KRPAI robot system has 1 main controller, namely Arduino Due, to control the robot's supporting sensors and 1 supporting controller, namely Open CM 9.04 A, to regulate the movement of the robot's legs. The overall block diagram of the KRPAI robot system can be seen in Fig. 1[2].

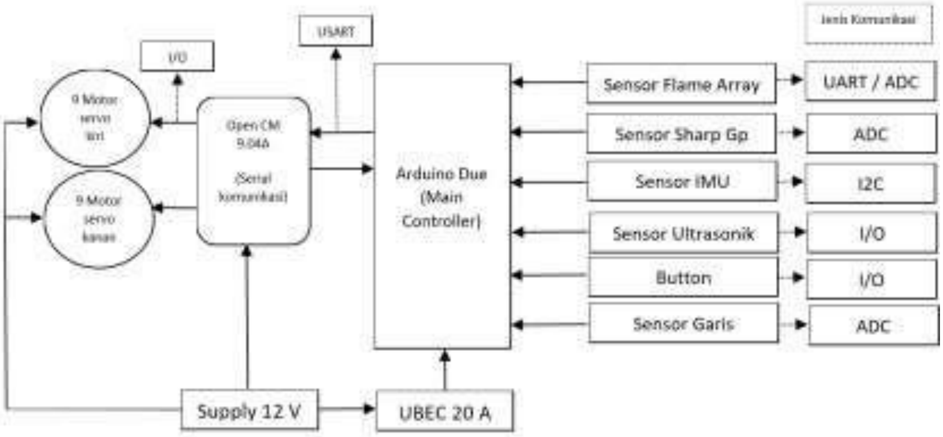


Fig. 1. Robot system block diagram

2.1.2. Software design

In designing this system, the Arduino IDE software is used as a tool to program the Arduino Due and Open CM which work as Master-Slave. Where Arduino is the main controller and sensor controller while open CM is the Dynamixel servo controller. Before doing system programming using the Arduino IDE software, a flowchart of the system to be designed is carried out.

The flowchart of the robot fire exit subroutine algorithm is shown in Fig. 2. The flowchart of the robot fire outdoor hallway subroutine algorithm is shown in Fig. 3. The flowchart of the robot's decision to search for home is shown in Fig. 4.

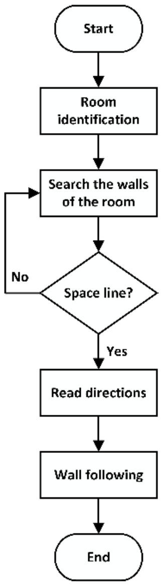


Fig. 2. Flowchart of the robot fire room exit subroutine algorithm

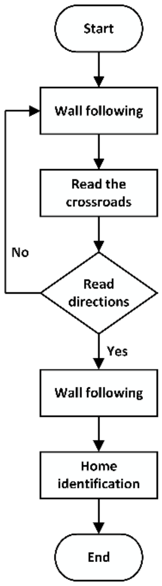


Fig. 3. Algorithm flowchart of the robot fire outdoor tunnel subroutine

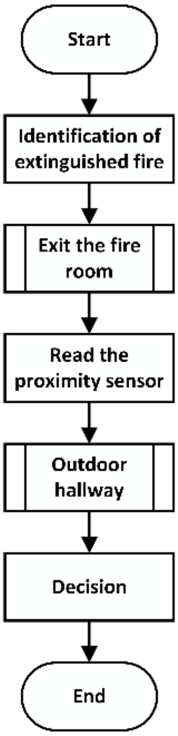

Start
Identification of extinguished fire
Exit the fire room
Read the proximity sensor
Outdoor hallway
Decision
End

Fig. 4. Flowchart of the robot's decision to search for home

In the flowchart of the exit subroutine of the robot fire room, the robot will first identify the room and then perform a wall following command or trace the walls of the room using a proximity sensor and look for the room line to determine whether the robot is in the room or not. Then when the robot finds the room line and exits to the hall of the room, the robot will read the direction and continue the command to trace the wall.

In the flow chart of the robot fire room outdoor tunnel subroutine, continuing from the fire room exit subroutine the robot will execute the wall following command then the robot will read the direction of the intersection. If the robot succeeds in reading the intersection, the robot will return to do wall following. Then the robot will read the directions and identify the home.

In this study the robot will use a backtracking algorithm where the robot will trace its way back to its original position using the IMU BNO055 sensor as a direction determinant.

2.2. IMU BNO055

BNO055 is a 9 DoF IMU sensor module that combines a 14-bit 3-axis accelerometer, a 16-bit 3-axis gyroscope, a 3-axis geomagnetic sensor, and a 32-bit ARM Cortex-M0 microcontroller in one compact module, so data generated can be accessed easily by the microcontroller via the serial interface.[4] The concept used by the robot to be able to rotate to a certain angle is by entering the desired angle, then reading the BNO055 sensor angle which is then sent to Arduino to then provide a signal to drive the motor [5].

2.3. Algorithm

The algorithm used is an algorithm that functions as a mapping algorithm and is a combination of the Depth First Search algorithm [6].

This algorithm is used to find a more concise solution, allowing a shorter solution search time [7].

The robot's return trip will take advantage of the living nodes that have been formed during the room search. The resulting knots will be simplified to eliminate dead knots (death notes) so that no more tracing the nodes that do not lead to a solution. Then the robot will walk home following the simplified node [8].

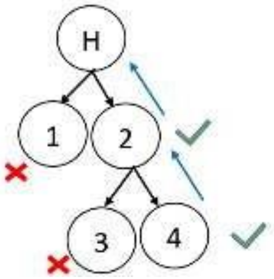


Fig. 5. Backtracking algorithm illustration

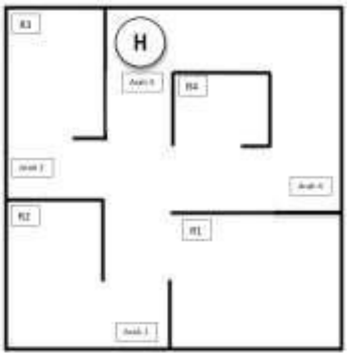


Fig. 6. Room numbering

In this research, the writer gives a certain code for naming the position of home, namely by numbering the rooms. The space line is determined by knowing the direction of the robot exiting the room after the fire fighting mission. By using the IMU as a determinant of direction and a distance sensor as a reference for conditions around the room. The goal is to facilitate the robot's decision making [9][10][11][12]. Room numbering can be seen in Fig. 6.

3. Result and Discussion

3.1. Sensor Testing

3.1.1. Sensor Calibration Test Results

The IMU calibration test is used to get the IMU value and setpoint value in the KRPAI arena, the way the tool works reads the directions at the intersection to get the IMU value which will be entered into the program and used by the robot to navigate. The IMU calibration can be seen in Fig. 7.

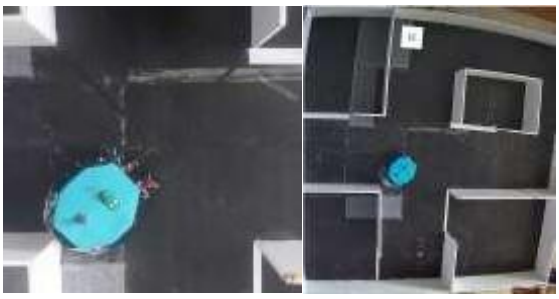


Fig. 7. BNO055 IMU calibration

In testing the IMU BNO05 sensor, calibration is carried out by turning on the robot and positioning it at an intersection facing one of the corners as shown in Fig. 7. Then the IMU BNO05 sensor will read angle values ranging from 0 degrees to 359 degrees. This is done so that the robot can determine the middle value of each direction. IMU calibration can be seen in Fig. 8.

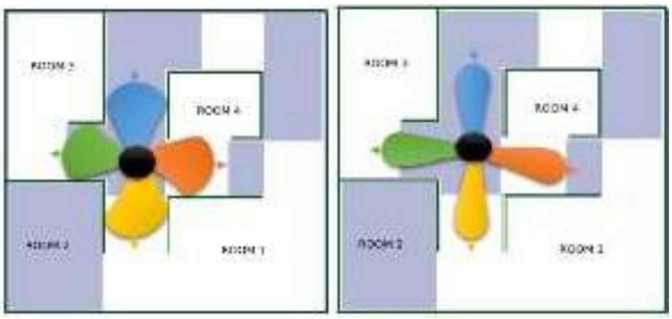


Fig. 8. IMU BNO05 calibration angle

The IMU BNO05 reading results can be seen in Table 1.

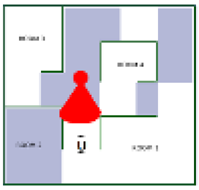
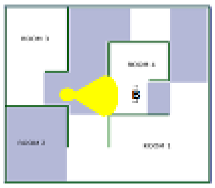
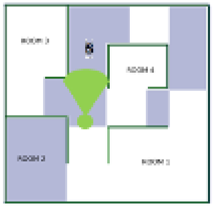
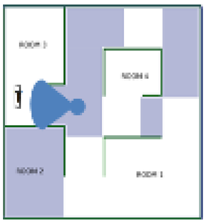
Table 1. IMU BNO05 Sensor Reading Results

No	Robot Direction (code)	IMU Value	Inverse Value
1	North (1)	268-359	313
2	West (2)	181-267	227
3	South (3)	91-181	135
4	East (4)	0-90	58

3.1.2. Testing the Results of Reading the IMU BNO05 Sensor Values on the Robot

This test is carried out to find out whether the robot has succeeded in reading the direction of the intersection on the track. The testing phase is the same as in the IMU calibration where the robot will assess the point 0 "Zero" as north. The results of this test can be seen on the LCD whether the robot is correct in reading the direction of the intersection at the KRPAI competition arena, the calibration of the robot is carried out before the robot is run. The results of reading the science value on the robot can be seen in Table 2.

Table 2. Results of reading the IMU BNO05 value on the robot

No	Robot Position	Reading Robot Direction Results
1.		 North (1)
2.		 West (2)
3.		 South (3)
4.		 East (4)

3.1.3. Testing BNO055 Sensor Values and GY-26 Compass Sensors

The gy-26 compass sensor has problems when determining direction, especially when it is in a location with a large magnetic field, so it is replaced with the BNO05 sensor. The following is a comparison of the gy-26 and BNO05 compass sensors when in different locations.

The results of data retrieval of comparison values of sensor values around the main building of Universitas Ahmad Dahlan campus 4 are shown in Table 3.

Table 3. Comparison of sensor values around the main building of campus 4 of Universitas Ahmad Dahlan

No	Displacement every 10 cm	Robot direction							
		North		South		East		West	
		Compass GY-26	IMU BNO055	Compass GY-26	IMU BNO055	Compass GY-26	IMU BNO055	Compass GY-26	IMU BNO055
1	0 cm	275	0	93	180	0	90	174	270
2	10 cm	1	0	100	180	59	90	1	270
3	20 cm	89	0	0	180	120	90	95	270
4	30 cm	150	0	122	180	122	90	160	270
5	40 cm	100	0	179	180	1	90	122	270

The results of data collection for comparison of sensor values around the campus 3 building of Universitas Ahmad Dahlan are shown in Table 4.

Table 4. Comparison of sensor values around campus buildings 3

No	Displacement every 10 cm	Robot direction							
		North		South		East		West	
		Compass GY-26	IMU BNO055	Compass GY-26	IMU BNO055	Compass GY-26	IMU BNO055	Compass GY-26	IMU BNO055
1	0 cm	274	0	93	180	1	90	174	270
2	10 cm	274	0	92	180	2	90	175	270
3	20 cm	274	0	93	180	1	90	176	270
4	30 cm	275	0	93	180	1	90	175	270
5	40 cm	274	0	93	180	2	90	176	270

3.2. Testing the Results of Recognizing the Position of the Home Room

This test was carried out to identify every room in the KRPAI competition arena. The test results are obtained from reading the direction the robot is facing, and the distance between the robot and the surrounding walls. The results of the home room recognition test can be seen in Table 5.

Table 5. Home room test results

No	Robots Position	Robots Reading	Results
1			Return Room = Direction 1 Home Room = Direction 3
2			Fire Room = Direction 4 Home Room = Direction 3

No	Robots Position	Robots Reading	Results
3			Fire Room = Direction 1 Home Room = Direction 3
4			Fire Room = Direction 2 Home Room = Direction 3

3.3. Testing the Backtracking Algorithm in the KRPAI Arena

Testing the backtracking algorithm is carried out by paying attention to the direction of the robot's movement which is set to trace the aisle towards home. If on the way the robot is constrained by an error when reading the BNO05-IMU sensor, then conditioning is carried out to prevent entering another room with the consequence of a longer travel time. In this test using 1 home position which is the same as 2 departure rooms, namely room R1 and room R2. The placement of the 2 departure rooms that were tested can be seen in Fig. 9.

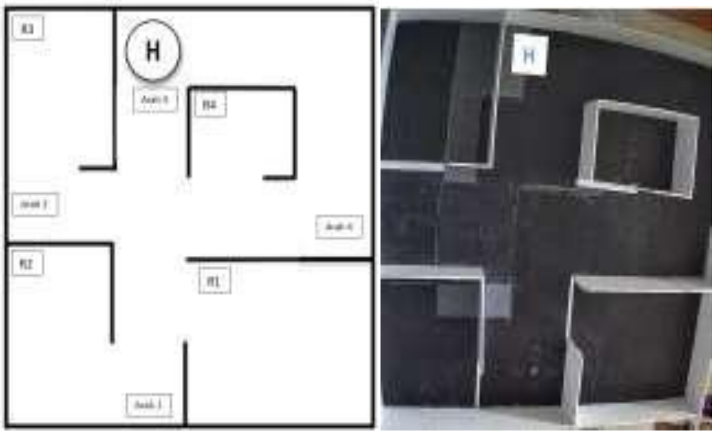


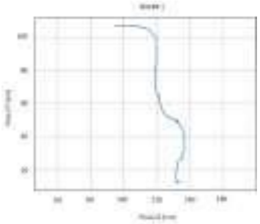
Fig. 9. The position of the room and home to be tested

Testing is carried out with an image processing system, namely marking objects to help observe the robot's movement patterns while walking. Where the video will be taken first using the camera that is right above the arena. The video will then be processed using the Python application and Open CV to get tracking and plotting results for a running robot. The results of the red and green line drawings are the direction of the robot's movement towards home. While the blue line in plotting functions as a line in the image field marker.

3.3.1. Testing the Return to Home Robot R1 and Home

The return to home test is carried out with the departure room in room 1 and the home room in direction 3. The results of the robot's travel time test can be seen in Table 6.

Table 6. Testing the robot return to home R1 and Home

No	Configuration	Testing	Plotting
	Room 1		
1	Home Direction		

Testing									
Test 1		Test 2		Test 3		Test 4		Test 5	
Result	Time (s)	Result	Time (s)	Result	Time (s)	Result	Time (s)	Result	Time (s)
v	9	v	8	v	10	x	x	x	x

Decision triggers in room 1 towards home can be seen in Table 7.

Table 7. Trigger room 1 decision

No.	Reference	Action	Description
1	The robot detects the nearest right wall	Wall right	Exit room 1
2	The robot detects the intersection and the robot detects direction 3	Straight - left wall	Pass through the crossroad into the home hall
3	The robot detects direction 3 as home and the robot detects the distance from the front sensor and the right sensor	Enter home - Stop	The robot made it home

From Table 7 it can be seen that out of 5 trials the robot failed home 2 times. This can happen due to detection failures from the IMU BNO055 and the proximity sensor, so preventive measures are taken using the backtracking algorithm.

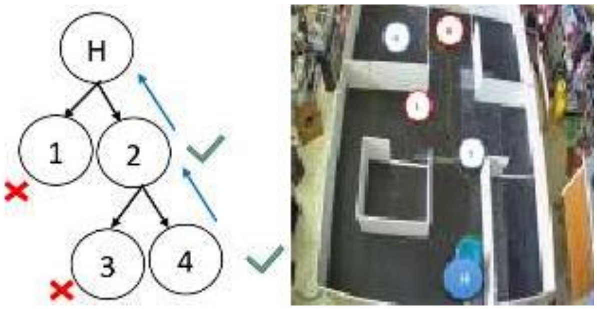


Fig. 10. Space Node 1 Backtracking Algorithm Space Node 1 Backtracking Algorithm

The results of testing the backtracking algorithm in room 1 towards home can be seen in Table 8

Table 8. Testing the return to home robot R1 and Home with the backtracking algorithm

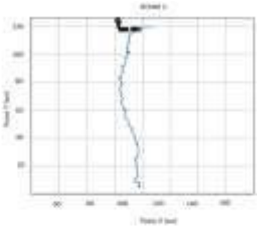
No.	Reference	Action	Description
1	The robot detects the nearest right wall	Wall right	Exit room 1
2	The robot failed to detect the intersection and failed to detect direction 3	Wall right	Pass through the crossroad into hall 4
3	The robot detects direction 4 and the robot detects the nearest front sensor distance	Stop 1 second - turn right - wall right	The robot's position returns to 3
4	The robot detects the line	Stop 1 second - reverse - turn left - straight 1 second	The position of the robot avoids entering the room
5	Robot detects intersection and detects direction 3	Stop 1 second - wall right	Pass through the intersection towards home
6	The robot detects direction 3 as home and the robot detects the distance with the front sensor and the right sensor	Entered home - stopped	The robot made it home

In Fig. 10 the use of the backtracking algorithm on the robot will make the robot trace live nodes starting from code 4 where the robot completes the blackout mission in room 1 then goes to code 2 detects the intersection to return home then completes the end of the mission at code H or home. However, as shown in Table 8, a robot that fails to come home will take a dead node, namely code 1, where the robot fails to detect the intersection and makes a decision on the right wall.

3.3.2. Testing the Return to Home Robot R1 and Home

The return to home test is carried out with the departure room in room 2 and the home room in direction 3. The results of the robot's travel time test can be seen in Table 9.

Table 9. Testing the robot return to home R2 and Home

No	Configuration	Testing	Plotting
1	Room 2 Home Direction		

Testing									
Test 1		Test 2		Test 3		Test 4		Test 5	
Result	Time (s)	Result	Time (s)	Result	Time (s)	Result	Time (s)	Result	Time (s)
v	9	v	9	v	10	x	x	x	x

Decision triggers in room 2 towards home can be seen in Table 10.

Table 10. Trigger room 2 decision

No.	Reference	Action	Description
1	The robot detects the nearest left wall	Wall left	Exit room 2
2	The robot detects the intersection and the robot detects direction 3	Stop 1 second - go straight - wall left	Pass through the crossroad into the home hall
3	The robot detects direction 3 as home, the sensor detects the distance from the front sensor and the right sensor	Entered home - stopped	The robot made it home

From Table 10 it can be seen that out of 5 trials the robot failed home 2 times. Where this can happen due to failed detection from the IMU and the proximity sensor. Therefore, preventive measures can be taken using the backtracking algorithm as shown in Fig. 11.

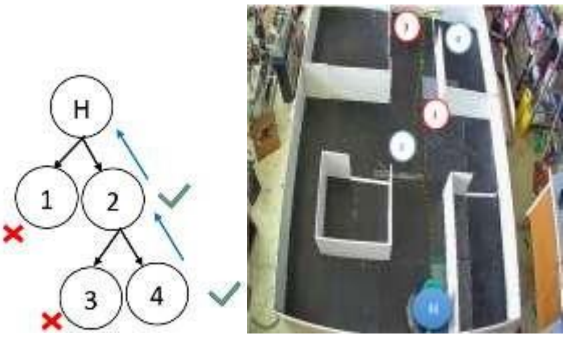
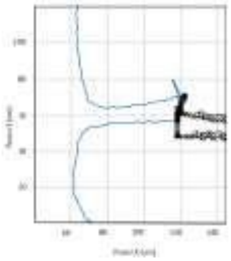


Fig. 11. Space Vertex 2 Backtracking Algorithm

In Fig. 11 the use of the backtracking algorithm on the robot will make the robot trace live nodes starting from code 4 where the robot completes the blackout mission in room 2 then goes to code 2 detects the intersection to return home then completes the end of the mission at code H or home. However, as shown in Table 9, a robot that fails to come home will take a dead node, namely code 1, where the robot fails to detect the intersection and makes a decision on the left wall. The results of testing the backtracking algorithm in room 2 towards home can be seen in Table 12.

Table 11. Testing the return to home robot R2 and Home with the backtracking algorithm.

No	Configuration	Testing	Plotting
1	Room 2 Home Room		

Testing					
Test 1		Test 2		Test 3	
Result	Time (s)	Result	Time (s)	Result	Time (s)
Succeed	13	Succeed	13	Succeed	14

From the results of testing the algorithm, it can be seen that the experimental time for the robot to trace the aisle when it fails to detect the intersection is not much different from the experimental results when the robot succeeds in detecting the intersection, as shown in Table 12. Trigger decision prevention robots that fail to detect intersections can be seen in Table 12.

Table 12. Trigger room 2 decisions with the backtracking algorithm

No.	Reference	Action	Description
1	The robot detects the nearest left wall	Wall left	Exit room 2
2	The robot detects the intersection and the robot fails to detect direction 3	Wall left	The position of the robot traces the hall of room 3
3	The robot detects the line and detects the direction 3	Stop 1 second - reverse - turn right - wall left	The position of the robot avoids entering the room
4	The robot detects direction 3 as home and the robot detects the distance of the front sensor and the right sensor	Entered home - stopped	The robot made it home

4. Conclusion

Based on testing the IMU sensor-based home search method with the backtracking algorithm on the hexapod robot in the Indonesian fire-fighting robot contest made by researchers, it can be concluded that the algorithm that works so that the robot can return home can work well. The test results show that a robot that cannot read the intersection when it returns home will experience an error and cannot complete the mission. For this reason, the use of the backtracking algorithm can prevent the robot from returning home.

References

[1] Kemendigbud, “Petunjuk Pelaksanaan Kontes Robot Indonesia,” Kontes Robot Indones., pp. 1–4, 2020. http://repositori.kemdikbud.go.id/id/eprint/18017

[2] Y. Suwandono, “Algoritma Return To Home Dengan Look Up Table Pada Robot Hexapod Untuk Kontes Robot Pemadam Api,” Skripsi tidak dipublikasi. Yogyakarta : Universitas Ahmad Dahlan, 2019.

[3] Y. B. Zebua, “Analisis Struktur Kristal Dan Sifat Magnetik Strontium Ferrite (SrFe12O19) Powder Yang Dibuat Pada Komposisi Stoikiometri Dan Non Stoikiometri,” Skripsi tidak dipublikasi. Medan: Universitas Sumatera Utara, 2019.

[4] Z. Salsabila, “Implementasi Behaviour Based Robotics Pada Hexapod Robot Berkaki,” Skripsi tidak dipublikasi. Yogyakarta: Universitas Ahmad Dahlan, 2021.

[5] A. Febriawan, W. S. Aji, K. Iv, and J. R. Road, “Rotating Control on Robots Indonesian Abu Robot Contest with PID and IMUBNO055 Controls Kendali Berputar pada Robot Kontes Robot Abu Indonesia dengan kendali PID dan IMUBNO055,” vol. 2, no. 1, pp. 14–23, 2020, doi: 10.12928/biste.v2i1.987.

[6] H. Umari and S. Mukhopadhyay, "Autonomous robotic exploration based on multiple rapidly-exploring randomized trees," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, 2017, pp. 1396-1402, doi: 10.1109/IROS.2017.8202319.

[7] N. A. Hasanah, L. Atikah, D. Herumurti and A. A. Yunanto, "A Comparative Study: Ant Colony Optimization Algorithm and Backtracking Algorithm for Sudoku Game," *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Semarang, Indonesia, 2020, pp. 548-553, doi: 10.1109/iSemantic50169.2020.9234267.

[8] A. S. Hidayatullah, A. N. Jati and C. Setianingsih, "Realization of depth first search algorithm on line maze solver robot," *2017 International Conference on Control, Electronics, Renewable Energy and Communications (ICCREC)*, Yogyakarta, Indonesia, 2017, pp. 247-251, doi: 10.1109/ICCEREC.2017.8226690.

[9] S. Agnisarman, S. Lopes, K. C. Madathil, K. Piratla and A. Gramopadhye, “A survey of automation-enabled human-in-the-loop systems for infrastructure visual inspection,” *Automation in Construction*, vol. 97, pp. 52-76, 2019.

[10] K. P. Cheng, R. E. Mohan, N. H. K. Nhan and A. V. Le, "Graph Theory-Based Approach to Accomplish Complete Coverage Path Planning Tasks for Reconfigurable Robots," in *IEEE Access*, vol. 7, pp. 94642-94657, 2019, doi: 10.1109/ACCESS.2019.2928467.

[11] J. Bidot, L. Karlsson, F. Lagriffoul and A. Saffiotti, “Geometric backtracking for combined task and motion planning in robotic systems,” *Artificial Intelligence*, vol. 247, pp. 229-265, 2017.

[12] A. V. Le, P. C. Ku, T. Than Tun, N. Huu Khanh Nhan, Y. Shi and R. E. Mohan, “Realization energy optimization of complete path planning in differential drive based self-reconfigurable floor cleaning robot,” *Energies*, vol. 12, no. 6, pp. 1136, 2019.