TELKOMNIKA, Vol.14, No.2, June 2016, pp. 489~496
ISSN: 1693-6930, accredited A by DIKTI, Decree No: 58/DIKTI/Kep/2013
DOI: 10.12928/TELKOMNIKA.v14i1.3156

Segway Line Tracer Using Proportional-Integral-Derivative Controllers

Wijaya Kurniawan*[1], Mochammad Hannats Hanafi Ichsan[2], Eko Setiawan[3]
University Of Brawijaya, Jl. Veteran Malang, East Java, Phone: +62 341 551611/ Fax +62 341 565420
*Corresponding author, e-mail: wjaykurnia@ub.ac.id[1], hanas.hanafi@ub.ac.id[2], ekosetiawan@ub.ac.id[3]

Abstract

Intelligent control, sensors and hardware integration are expected to generate an efficient transportation system and and minimum effort, to carry goods from one location to another location. Line tracer used by robot to transport follow the path; it has a system that uses a light sensor to read the color from a line that represents the path to make specific direction. Segway is two wheeled transportation item that have an efficient energy used. Nowadays line tracer can only work if it has three or more wheels and segway can only work with riders. This research segway designed by lego robot, PID (Proportional, Integral, Derivative) control used to control an input from gyroscope sensor in form of elevation angle of the earth. The control system is expected to control two wheeled Segway to reach steady state rapidly. So the Segway would run without involving human or without rider.

***Keywords**: Segway, line tracer, gyroscope and PID*

Copyright © 2016 Universitas Ahmad Dahlan. All rights reserved.

1. Introduction

Two-wheeled vehicle that is commonly called a Segway, being a research topic that is growing, and Line Tracer is a robot that can walk on his own by reading the lines and paths have been determined and balance automatically based on changes and shifts in its balance point [1]. This study aims to expand research that has been done previously, which combines the two wheel balancing robot and make a two-wheeled robot can walk on his own by using Line Tracer [2]. Segway is expected to produce a design that can run itself (automatic) carries the goods without human assistance [3].

In the previous studies conducted, two-wheeled robot lego successfully created using the Model Predictive Control by using inverted pendulum [5]. Another study Q-Learning to make a robot can learn how to solve the problems finding the route by way of the robot itself after some time learning in via the same path [6].

This research will be conducted by making the design of the Segway that use the robot as research objects. Lego NXT is a robot which is used as a model Segway, and will be given a color sensor that uses a PID controller for the testing process.

Lego NXT robot will become a model and as a means of test the control algorithms that will be examined. PID control algorithm programmed with high-level language C and has nice performance to tracking stability [4]. The color sensor is a device that can distinguish colors, in this case color sensor used to make the robot recognize the color of the path that has color [5].

The suitability of the performance of the control algorithm, color sensors and performance the robot will be investigated in order Segway robot that uses the concept that runs on a track that has been provided and can run itself automatically without the driver. How to organize the performance of two motors, each of which is owned by both wheels, so the motor can alternately processed balance and make the shift places.

2. Research Method

Lego Mindstorms NXT kit is an package that contains parts such as Lego robots, sensors, actuators and a small computer - NXT Brick. In order to able to develop and upload the program to the NXT Brick required an algorithm. This research used Lego because it used 32-bit ARM microprocessor and the user can programmed it as well using the compiler LegoC. LegoC contains a complete development environment [7]. The reason for choosing LegoC is in

Received November 26, 2015; Revised March 22, 2016; Accepted April 6, 2016

addition to having a high processing speed is also very easy to work with and has many built-in features, such as data logging, gamepad support, debugging, Bluetooth wireless communication, sensor drivers and so on [8].

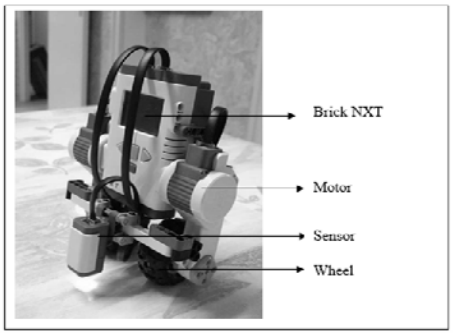

Figure 1. Lego Mindstorm NXT

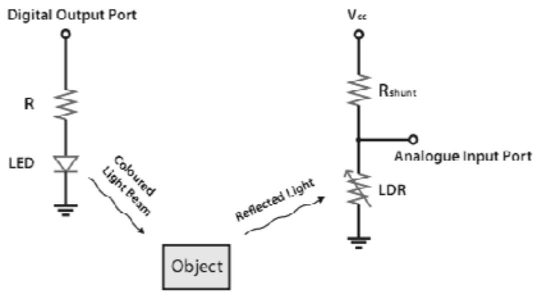

Figure 2. Colors circuit sensor

Lego has a color sensor to detect multiple colors include red, green, and blue [9]. Step in Figure 2 red diode is turned on for a few moments, the red light of an object and the object will reflect an certain light intensity characteristics. The intensity is detected by LDR which is on the active surface of the sensor.

This process is repeated on the other two diodes are green and blue so that it will obtain 3 values are then processed in the program. With a particular reference value that has been determined in previous calibration process, the color of an object can be determined or read by the sensor. In the diagram it can be seen in the image color detection process procedure below.

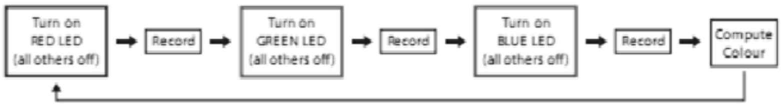
Turn on RED LED (all others off)
Record
Turn on GREEN LED (all others off)
Record
Turn on BLUE LED (all others off)
Record
Compute Colour

Figure 3. Color Detection Procedure

Although combine three colors on the LED will produces whitecolor, it does not mean 3 LEDs can be replaced with an white LED. The reason is that in order to identify the color of an object, it takes a minimum of 2 component signal. This was done by measuring the intensity of each frequency of the reflected light. In this case it is the color sensor circuit is only using three basic colors of red, greencolor, and blue [10].

The process of acquiring color sensor to read the signals, the signals collected at the digitized sensor analog input to a value in the range 0 - 1023.

Range signal area defined by the following equation:

$$S_{range} = \frac{LDR_{max}}{LDR_{max} + R_{shunt}} - \frac{LDR_{min}}{LDR_{min} + R_{shunt}} \tag{1}$$

In order to obtain a high-resolution sensor, the resistance of the shunt resistor is calculated to suit the characteristics of the LDR. The circuit consists of LDR and shunt resistor form a voltage divider circuit.

$$\frac{d}{d\,R_{shunt}} S_{range} = \frac{-LDR_{max}}{(LDR_{max} + R_{shunt})^2} - \frac{LDR_{min}}{LDR_{min} + R_{shunt}} \tag{2}$$

In order to search for the optimal value shunt resistor so that the area signal becomes as wide as possible then the above equation should given by differential against R shunt which is then filled a value of zero (0) in order to get the equation.

$$0 = \frac{d}{d\,R_{shunt}} S_{range} solve, R_{shunt} - \begin{bmatrix} (LDR_{max} * LDR_{min})^{\frac{1}{2}} \\ -(LDR_{max} * LDR_{min})^{\frac{1}{2}} \end{bmatrix} \quad (3)$$

The method used in this research, the movement of the motor is described in the following flowchart. At the beginning of the robot is executed, declared Δ T of 0.5 s and n are integers. The next process is to multiply Δ T with n to produce odd or even numbers. If the results of odd n the process performed by the sensor is reading lines right after the completion of the process of reading the right, the left sensor read lines. If given input lines are black and white environment, the reading process if it finds black lines, then the condition 1, if it does not find the color black (which in this case white) then the condition 0. If conditions are right sensor readings and sensor=1 left=0 then the motor right-back and left motor forward, and vice versa. However, if the sensor right and left=0 then both the motor will move forward.

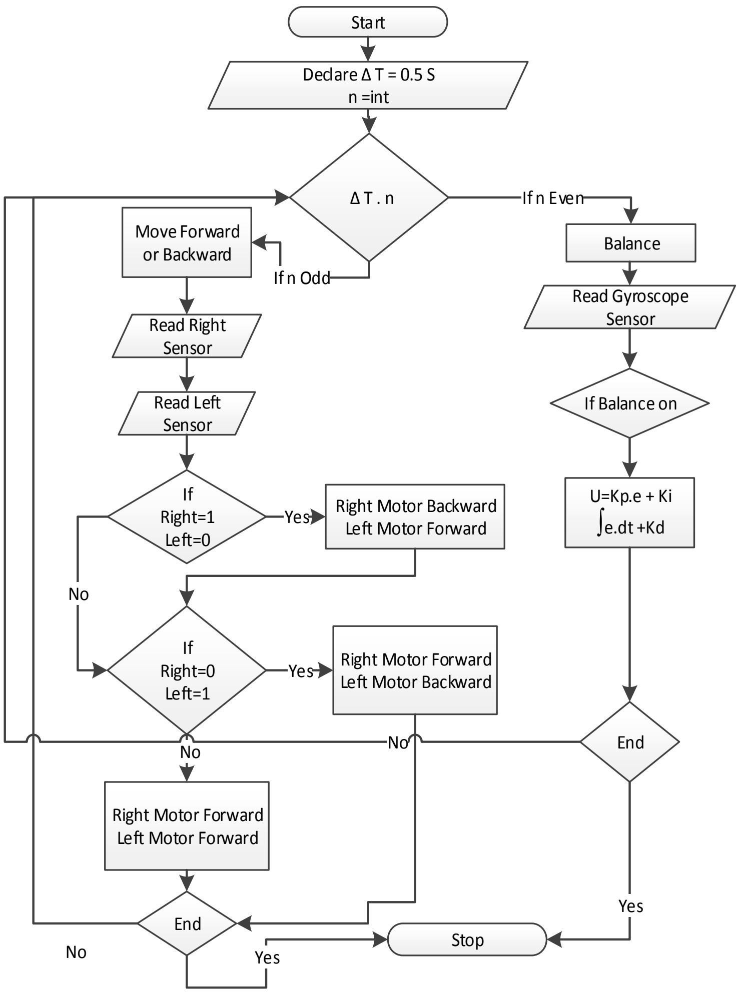

Figure 4. Algorithm which Implemented in Robot

In order to tune the system, required knowledge about the influence of of the PID controller transient response and steady parameters of the system that would be controlled. Such knowledge can be seen from the mathematical equations that of the PID controller [11]. The nature of PID parameters are:

P : effect on the transient state (settling time=Tss) and steady (steady state error=Ess).
I : improve steady state (steady state error=Ess) but usually worsen transient (arising overshoot=Mp).
D : improve the transient state of the system (settling time=Ts, overshoot=Mp) but no effect at all on the systems steady state.

Expressed control action as:

$$m(t) = K_p e(t) + \frac{K_p}{T_i}\int_0^t e(t)dt + K_p T_d \frac{de(t)}{\mathrm{dt}} \tag{4}$$

This type of controller used to improve the speed of response, prevent errors and maintain a stable steady state [5].

3. Results and Analysis

In this chapter, the robot will be tested three times. The first test is a test gyroscope sensor, this sensor are used to that the robot can stand up well. The second test is the test of a configuration value Ki, Kp and Kd with trial error method of. While the third test is a test against the color path and the environment, with the test expectation can be determined percentage rate of systems.

3.1. Gyroscope Sensor Test

First testing in this research is to examine the value of inputs obtained by Gyroscope sensor. Tests performed by putting a robot on the floor with the robot position "sleeping" as shown at Figure 5(a).

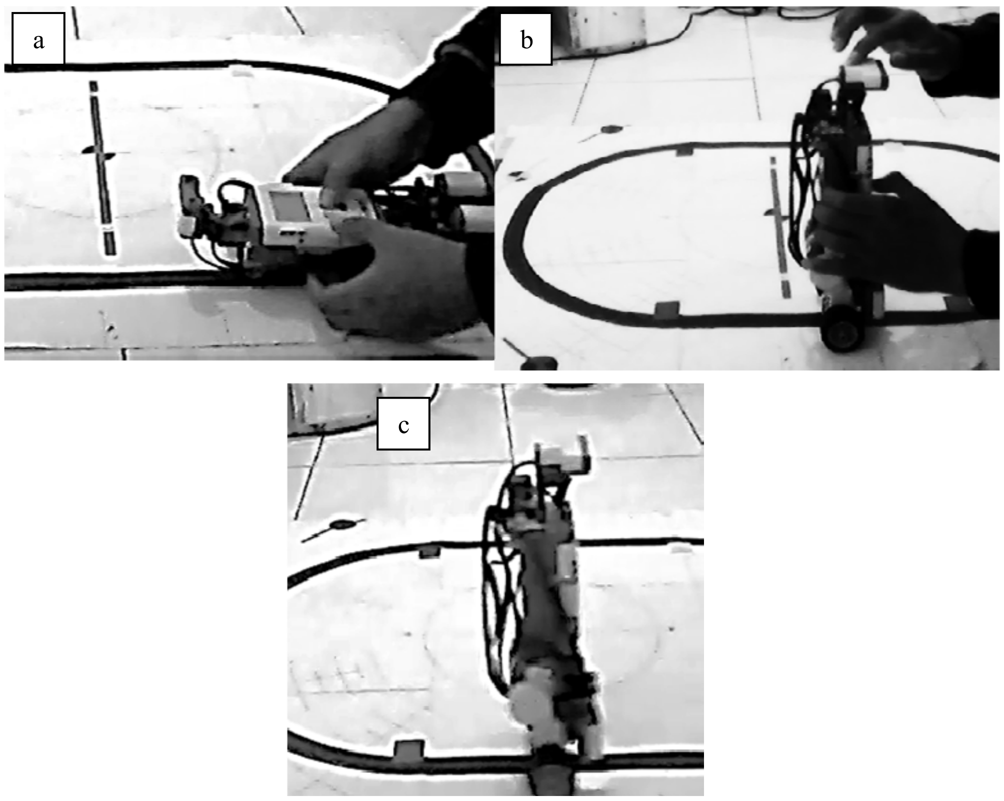

Figure 5. Robot at a) “sleeping” condition; b) “standing” condition with held; c) “standing” condition without being held

Then do the capture value by Gyroscope 50 times, after taking the value of the position of the bed 50 times the value obtained for the equilibrium, after the robot is placed in the position of "standing" Figure 5(b). or less than the angle of 90^o when the robot is being positioned in the "sleep" as in Figure 5(a).

Table 1. Gyroscope Sensor Count

Test number -	Gyroscope Value
1	594
2	593.5
3	599
--	--
48	595
49	599
50	592
Average	**596**

After calculation 50 times, the average was taken and the value results 596. This value is the value when the robot on sleep position became initial set value. After that the robot can stands as shown at Figure 5.c, robot can stands without being touched.

3.2. Ki, Kp dan Kd Test and Analysis

Giving value for Kp, Ki and Kd is used to balance the robot through two wheels that is located on right and left of the robot, the three parameters are interconnected. The minimum value for Kp of 0.2 and maximum 0.6 is where the condition of the robot can stands even though the tilt. The smaller the value Ki increasingly leaning forward and if the greater the value of Kp robots increasingly leaning backward though it can stands. As for Ki given minimum value and a maximum value 3.8 3.2. The smaller the robot to balance the condition of the body more quickly, otherwise the greater the value Ki Kp response to the slower. While the value of Kd is to improve relations Kp and Ki in the body balance the robot. The following table shows the trial error input of each value to the equilibrium conditions of the robot.

Table 2. Robot Trial Against Kp, Ki and Kd Value

Trial number	Kp Value	Ki Value	Kd Value	Robot Condition Against -		
				Kp	Ki	Kd
1	0.2	3.2	0.003	Lean Back	Rapid	Slow Response
2	0.2	3.4	0.005	Lean Back	Rapid	Fast Response
3	0.2	3.6	0.003	Lean Back	Slow	Slow Response
4	0.2	3.8	0.005	Lean Back	Slow	Fast Response
5	0.2	3.2	0.003	Lean Back	Rapid	Slow Response
6	0.2	3.4	0.005	Lean Back	Rapid	Fast Response
7	0.2	3.6	0.003	Lean Back	Slow	Slow Response
8	0.2	3.8	0.005	Lean Back	Slow	Fast Response
9	0.4	3.2	0.003	Perpendicular	Rapid	Slow Response
10	0.4	3.4	0.005	Perpendicular	Rapid	Fast Response
11	0.4	3.6	0.003	Perpendicular	Slow	Slow Response
12	0.4	3.8	0.005	Perpendicular	Slow	Fast Response
13	0.4	3.2	0.003	Perpendicular	Rapid	Slow Response
14	0.4	3.4	0.005	Perpendicular	Rapid	Fast Response
15	0.4	3.6	0.003	Perpendicular`	Slow	Slow Response
16	0.4	3.8	0.005	Perpendicular	Slow	Fast Response
17	0.6	3.2	0.003	Lean Forward	Slow	Slow Response
18	0.6	3.4	0.005	Lean Forward	Slow	Fast Response
19	0.6	3.6	0.003	Lean Forward	Slow	Slow Response
20	0.6	3.8	0.005	Lean Forward	Slow	Fast Response
21	0.6	3.2	0.003	Lean Forward	Slow	Slow Response
22	0.6	3.4	0.005	Lean Forward	Slow	Fast Response
23	0.6	3.6	0.003	Lean Forward	Slow	Slow Response
24	0.6	3.8	0.005	Lean Forward	Slow	Fast Response

Description of robot conditions against the each parameter on Table 2:

The response to the Kp :

a. Lean Back : the angle between floor by robot 70^o-84^o
b. Perpendicular : the angle between floor by robot 85^o-95^o
c. Lean Forward : the angle between floor by robot 96^o-105^o
d. For another angle robot falls

The response to the Ki :

a. Rapid : equilibration process conditions robot is fast so that the movement of equilibration looks rough and it took 4 to 7 swing robot to equilibrate.
b. Slow : equilibration process is slow so that the movement of the robot condition equilibration smooth and only took 2 to 3 swing robot to equilibrate.

The response to the Kd

a. Fast Response : robot equilibration process response to Ki is fast
b. Slow Response : robot equilibration process response to Ki is slow

Based on test results of trial error by entering the respective value of the combination of Ki, Kp and Kd obtained the best combination of 0.4; 3.8; and 0.005. In these conditions the robot can stand up well. Stand well with the sense that the robot stands upright, good equilibration process and quick response.

3.2. Performance Testing

In the subsequent testing process, testing the performance of the color sensor. Color sensor is used to detect the color of the track under the robot to recognize the tracks. In the following image readings color configuration is done by robots. This test is used to read the success rate sensor readings of color when recognizing six colors including: black (color path), white (environment path) as well as red, yellow, green and blue is the color disturbance is located between the environment and pathways such as the Figure 6.

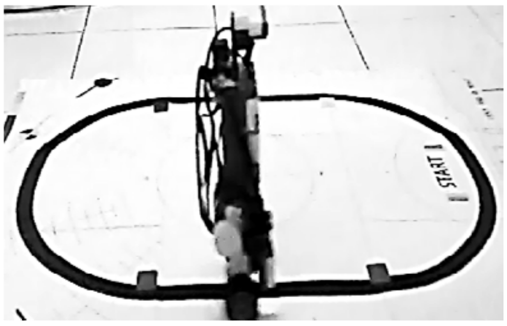

Figure 6. Robot and the Path

Path Specification :

a. 1 Path=2 Straight Pat + 4 Cure (2 Left and 2 Right Curve)
b. 1 Straight Path=31cm
c. 1 Curve Path=21cm
d. Color Disturbance, represent with a box, each box $2cm^2$

Tests carried out ten times a round to the right, and ten times round to the left. Testing is done each for the main color (the color line and environment), and color disturbance (red, green, blue, and red. To test the color of the line, has four kinds of testing that curves to the right, turn left, and on track straight, each of which tested between conditions of by time.

Condition, represents the condition of the robot when to recognize the line, giving the parameter is if the robot to recognize the path well and can walk it will be represented by parameter "path", if the robot does not recognize the path properly (out of line) then it will be represented by parameter "fail". While the length of time represents the time used in the parameter "work", for the parameter " fail" time is not recorded.

Table 3. Test by Path

Test Number	Right Curve		Left Curve		Straight	
	Kondisi	**Waktu**	**Kondisi**	**Waktu**	**Kondisi**	**Waktu**
1	Work	00.00.47	Work	00.01.00	Work	00.00.36
2	Work	00.00.44	Work	00.00.51	Work	00.00.20
3	Work	00.00.42	Work	00.00.51	Work	00.00.15
4	Fail	-	Work	00.00.46	Work	00.00.21
5	Work	00.00.44	Fail	-	Work	00.00.16
6	Work	00.00.41	Work	00.01.35	Work	00.00.20
7	Fail	-	Work	00.00.54	Work	00.00.23
8	Work	00.00.43	Work	00.00.48	Work	00.00.21
9	Work	00.00.44	Work	00.00.46	Work	00.00.22
10	Fail	-	Work	00.00.42	Work	00.00.23
Success Rate	**70%**		**90%**		**100%**	
Average Success Rate	**00.00.43,57**		**00.00.54,77**		**00.00.21,7**	

Results of testing on the Right curves along the 29 cm have a success rate of 70% takes an average of 43.57 seconds, while when curves Left has a success rate of 90% and an average time of 54.77 seconds. For the test on a straight track has a 100% success by an average time of 21.7 seconds.

Tests for color disturbance were done 10 times to the right or left. The success of the detection sensor to detect is if the disturbance is reading disturbance that disturbance is at the right or left. Disruption if the delay is more than 5 seconds (Fail / *), if the delay is not more than 5 seconds without any disturbance. Parameter dash (-) in the table is when not testing. If the test is done on the disorder on the Right, then on the left there is no disturbance and otherwise.

Table 4. Trial Against Disturbance Color

Trial	Red		Yellow		Blue		Green	
	Right	**Left**	**Right**	**Left**	**Right**	**Left**	**Right**	**Left**
1	00.00.01	-	00.00.01	-	00.00.29*	-	00.00.01	-
2	00.00.02	-	00.00.01	-	00.00.55*	-	00.00.01	-
3	00.00.01	-	00.00.02	-	00.00.04	-	00.00.01	-
4	00.00.02	-	00.00.03	-	Fail	-	00.00.01	-
5	00.00.01	-	00.00.01	-	00.02.20*	-	00.00.02	-
6	-	00.00.01	-	00.00.01	-	00.00.54*	-	00.00.54
7	-	00.00.01	-	00.00.02	-	Fail	-	00.04.12
8	-	00.00.02	-	00.00.01	-	00.01.07*	-	00.01.07*
9	-	00.00.01	-	00.00.02	-	00.00.02	-	00.00.45
10	-	00.00.01	-	00.00.02	-	00.04.12*	-	Fail
Success Rate	100%	100%	100%	100%	20%	20%	100%	60%
success rate against the color	100%		100%		20%		80%	
Average Success Rate	75%							

Results of testing to detect tampering colors for each color of both right and left each color (right + left / 2). For red color disturbance can be detected with accuracy of 100%, 100% yellow, 20% blue and green colors can be distinguished by the level of accuracy of 80%.

4. Conclusion

After doing some testing on the robot, it was concluded in this study. PID algorithm successfully implemented on the robot so that robot doesn’t fall. PID is used to set the robot equilibrium, in this research the best combination value of Ki, Kp and Kd are used to adjust the balance of each robot is 0.4; 3.8; and 0.005. At the a straight path, success rate the robot in passing along a straight line 31 cm is 100% by an average travel time 00.00.21,7 seconds, whereas when tested under conditions curves to the right has a success rate of 70% by average travel time 00.00.43,57 and curves to the left by 90% by average time 00.00.54,77. The robot can distinguish between colors disturbance against the color path, for the red and yellow color has a success rate that is very good at 100%, for the color hurdle green has a success rate which is pretty good by 80%, while for the blue color is not good only for 20%, and so that the average success rate is 75%. So it can be categorized as working well, the robot doesn't fall with the combination value of PID and and robots that work well when passing obstacles colors except blue color.

References

[1] Saputro BY, Amri D, Jayanti SD. *Comparison of Control Methods PD, PI, and PID on Two Wheeled Self Balancing Robot*. EECSI 2014. Yogyakarta, Indonesia. 2014.

[2] Jung T, Kim HW, Jung S. *Implementation and Control of Balancing Line Tracer Using Vision*. International Conference on Ubiquitous Robots and Ambient Intelligence. Incheon. 2011; 8: 858-862.

[3] Nguyen HG, Kogut G, Barua R, Burmeister A. *A Segway RMO-Based Robotic Transport System*. SPIE Proc. Philadhelphia. 2004: 5609.

[4] Li L, Xie J, Li W. Fuzzy Adaptive PID Control of a New Hydraulic Erecting Mechanism. *TELKOMNIKA Telecommunication Computing Electronics and Control.* 2013; 11(4): 715-724

[5] Canale M, Brunet SC. *A Lego Mindstorm NXT experiment for Model Predictive Control Education*. European Control Conference. Zurich. 2013: 2549-2554.

[6] Ricardo VCA, Enrique HP, Gabriel HAM. *A Line follower robot implementation using Lego's Mindstorms Kit and Q-Learning*. ACTA Universitaria. Guanajuato. 22: 113-118.

[7] Oliveira G, Silva R, Lira T, Reis LP. *Environment Mapping using the Lego Mindstorms NXT and leJOS NXJ*. Universidade do Porto. Porto.

[8] Henryranu BP, Nurwarsito H, Kurniawan W. One-Time Password Implementation on Lego Mindstorms NXT. *TELKOMNIKA Telecommunication Computing Electronics and Control*. 2014; 12(3): 689-694.

[9] Koncanda M, Wilke MB, Ballantine DS. Using Lego Mindstorms NXT Robotics Kits as a Spectrophometric Instrument. *International Journal on Smart Sensing and Intelligent Systems.* 2010; 3(3): 2010.

[10] Trussel M. Engaging with the NXT Prototype Board. Spring Term. Swiss Federal Institute Technology of Zurich. 2009.

[11] Gunterus, F. Falsafah Dasar Sistem Pengendalian Proses. Jakarta: PT Elex Media Komputindo. 1994.