

DESAIN NILAI KONSTANTA PENGENDALI DAN JUMLAH SENSOR UNTUK OPTIMASI RESPON ROBOT PENGIKUT GARIS

S. Marwanto¹, Mirwan Ushada², Ali Parkhan³

¹ Fakultas Teknologi Industri, Universitas Islam Indonesia;

² Fakultas Teknologi Pertanian, Universitas Gadjah Mada Yogyakarta;

³ Fakultas Teknologi Industri, Universitas Islam Indonesia

ABSTRAK

Desain nilai konstanta pengendali PID dan jumlah sensor ditujukan untuk mengoptimalkan respon robot pengikut garis. Pengendali PID berupa konstanta proporsional (K_p), konstanta integral (K_i) dan konstanta derivatif (K_d). Respon robot pengikut garis berupa overshoot dan settling time. Optimalisasi respon robot pengikut garis berupa overshoot yang semakin pendek dan settling time yang semakin singkat.

Hasil eksperimen dengan metode Taguchi menunjukkan bahwa kondisi optimal dicapai pada seting parameter $A_1B_3C_3D_3$ dan diperoleh overshoot sebesar 5.300 cm serta settling time sebesar 3.833 detik. Untuk meminimalkan overshoot dan settling time dibutuhkan kombinasi level faktor $K_p = 100$, $K_i = 1$, $K_d = 14$ dan jumlah sensor garis 8 buah.

Kata-kata kunci : Pengendali PID, Robot pengikut garis, overshoot, settling time, Taguchi

ABSTRACT

The design of PID controllers constant value and the number of sensors intended to optimize the response of the line follower robot. PID controller consists of a proportional constant (K_p), integral constants (K_i), and derivative constant (K_d). Response from the line follower robot consists of overshoot and settling time. Optimizing the response of line follower robots are getting shorter form of overshoot and settling time.

The experimental results with the Taguchi method showed that the optimal conditions reached at $A_1B_3C_3D_3$ parameter settings and obtained 5,300 cm overshoot and settling time of 3,833 seconds. To minimize overshoot and settling time required level combination of factors $K_p = 100$, $K_i = 1$, $K_d = 14$ and the number of line sensors 8 pieces.

Keywords: PID controller, robot line follower, overshoot, settling time, Taguchi

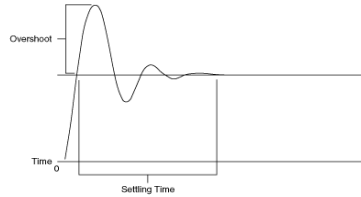
1. Pendahuluan

Robot pengikut garis merupakan salah satu bentuk robot bergerak otonom yang banyak dirancang baik untuk industri, pendidikan maupun penelitian [4];[5]. Penggunaan robot pengikut garis di industri secara luas untuk mengangkut berbagai macam material termasuk palet, rak, keranjang dan kontainer. Salah satu aplikasi yang paling populer adalah *handling* palet dimana proses perpindahan palet merupakan jenis pekerjaan berulang yang sangat umum di lingkungan manufaktur.

Dalam penelitian di bidang medis telah dibuat robot pengikut garis yang dapat membawa obat untuk pasien yang membutuhkan. Teknologi ini fokus pada penghantaran yang aman, tepat waktu, efisien dan efektif [5]. Selain itu, juga telah dibuat robot pengantar makanan untuk mengganti peran seorang pelayan di restoran. Robot ini memiliki kemampuan untuk mengantarkan makanan secara otomatis [4].

Lintasan yang harus dilalui oleh robot pengikut garis dapat berupa garis lurus, garis berbelok ataupun persimpangan yang dibentuk dari garis hitam diatas permukaan putih atau sebaliknya [7].

Robot pengikut garis dapat menyimpang ke kiri dan ke kanan secara bergantian dalam mengikuti lintasannya jika hanya diterapkan sistem kendali on-off[2]. Respon dari robot pengikut garis dapat diukur dari besar simpangan antara lintasan robot dengan arah robot yang disebut sebagai *overshoot* dan waktu yang dibutuhkan robot kembali ke lintasannya jika terjadi penyimpangan yang disebut sebagai *settling time* seperti pada Gambar 1.1.

Gambar 1.1 *Overshoot* dan *settling time*

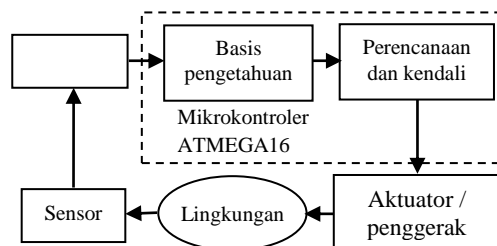
Pengendali proporsional, integral dan derivatif (PID) dapat mengatasi masalah pergerakan robot pengikut garis dengan *overshoot* yang panjang dan *settling time* yang lama [9]. Pengendali PID merupakan salah satu pengendali yang bertujuan memperbaiki kinerja suatu sistem, termasuk sistem kendali putaran motor dc [1]. Selisih antara lintasan dengan arah robot yang disebut sebagai kesalahan (*error*) diolah oleh pengendali PID untuk menghasilkan kecepatan yang sesuai pada masing-masing roda kiri dan kanan robot. Faktor penentu dari pengendali PID adalah pemilihan nilai kombinasi proporsional, integral dan derivatif. Pemilihan besaran proporsional, integral dan derivatif yang tepat akan menghasilkan respon sistem yang lebih baik [8].

Selain pengendali, jumlah sensor garis juga mempengaruhi respon dari robot pengikut garis. Robot pengikut garis dengan sensor garis yang lebih banyak akan lebih presisi dalam mendeteksi kondisi lintasan [6].

Pemilihan besaran konstanta proporsional, integral dan derivatif serta jumlah sensor garis yang dibutuhkan untuk mendapatkan respon robot pengikut garis yang lebih baik dan *smooth* menjadi penting dan perlu dilakukan eksperimen. Tujuan dari penelitian ini adalah untuk mendapatkan kombinasi konstanta pengendali dan jumlah sensor yang tepat untuk optimasi respon robot pengikut garis menggunakan metode Taguchi. Eksperimen Taguchi dilakukan dengan menentukan *settinglevel* terbaik dari faktor-faktor yang terkait agar nantinya respon robot pengikut garis dapat ditingkatkan. Kelebihan metode ini adalah mampu mengombinasikan hasil eksperimen melalui settingfaktor dan level yang optimal [10].

2. Tinjauan Pustaka

Sistem robot pengikut garis yang digunakan seperti pada gambar 2.1.



Gambar 2.1 Sistem robot pengikut garis

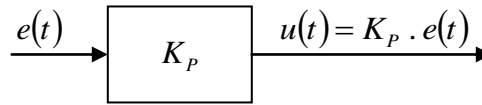
Sensor pendeteksi garis yang digunakan dalam robot pengikut garis mendasarkan pada prinsip pemantulan cahaya untuk membedakan warna garis dengan latar belakangnya [7]. Pada warna gelap penyerapan cahaya lebih besar daripada warna putih sehingga cahaya terpantul ke sensor menjadi lebih kecil.

Pengendali

Kualitas pengendali ditentukan oleh seberapa baik kinerja pengendali tersebut dalam melakukan proses pengendalian terutama pada bidang kendali yang tergolong sulit [3]. Keberadaan pengendali dalam sistem kendali mempunyai kontribusi yang besar terhadap perilaku sistem. Semakin cepat reaksi sistem mengikuti sinyal aktual dan semakin kecil kesalahan yang terjadi, semakin baiklah kinerja sistem kendali yang diterapkan [8].

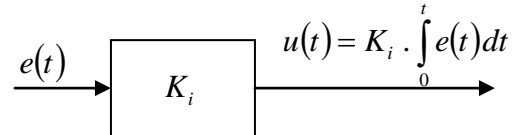
- a. Pengendali proporsional

Pengendali proporsional memiliki luaran yang sebanding/proporsional dengan besarnya sinyal kesalahan [1].



Gambar 2.2 Pengendali proporsional

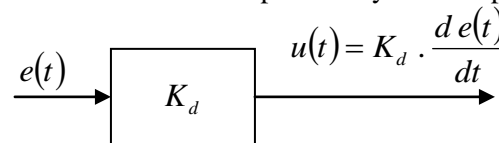
b. Pengendali integral; Pengendali integral berfungsi menghasilkan respon sistem yang memiliki kesalahan keadaan mantap nol.



Gambar 2.3 Pengendali integral

c. Pengendali derivatif

Luaran pengendali derivatif memiliki sifat seperti halnya suatu operasi derivatif.

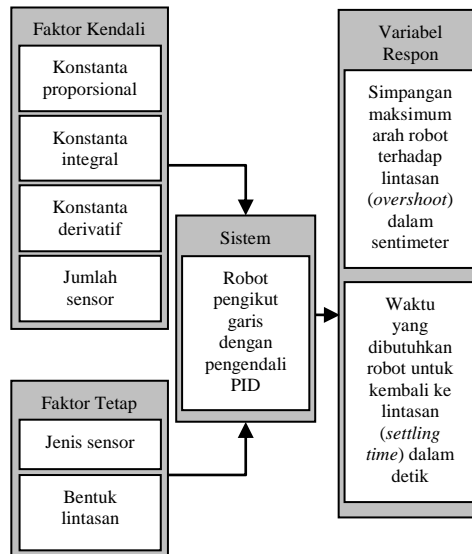


Gambar 2.4 Pengendali derivatif

d. Pengendali PID; Pengendali PID adalah sistem pengendali gabungan antara pengendali Proporsional, Integral dan Derivatif [3].

3. Metode Penelitian

a. Kerangka konsep



Gambar 3.1 Kerangka konsep

b. Variabel penelitian

1) Faktor kendali

Faktor kendali terdiri dari 4 faktor dengan 3 level, dapat dilihat pada tabel 3.1 bawah ini :

Tabel 3.1 Faktor dan level

Faktor kendali	Level		
	1	2	3
A.Konstanta	10	12	14
proporsional	0	0	0
B.Konstanta integral	0.1	0.5	1
C.Konstanta derivatif	6	10	14
D.Jumlah sensor	4	6	8

2) Faktor tetap

Faktor tetap pada penelitian ini adalah faktor-faktor yang dapat mempengaruhi respon robot pengikut garis tetapi dibuat tetap selama eksperimen, yaitu :

a) Jenis sensor

Jenis sensor garis yang digunakan adalah pasangan led infrared dan phototransistor karena dapat mengatasi gangguan cahaya di lingkungan sekitar robot pengikut garis [2].

b) Bentuk lintasan

Bentuk lintasan berupa garis lurus dan belokan 90^0 karena ruangan dalam rumah atau gedung-gedung berbentuk persegi sehingga sisi-sisi berbentuk lurus dan sudut 90^0 .

3) Variabel respon

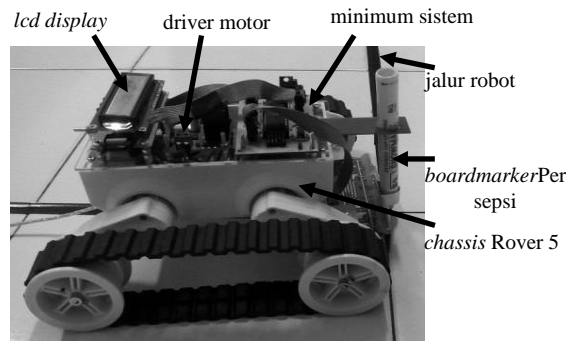
Variabel respon pada penelitian ini adalah :

a) Simpangan maksimum arah robot terhadap garis yang dilalui (*overshoot*) dalam sentimeter.

b) Waktu yang dibutuhkan robot untuk kembali ke jalurnya (*settling time*) dalam detik.

c. Tahapan penelitian

Obyek penelitian berupa robot pengikut garis disiapkan dengan cara merangkai modul minimum sistem, motor driver dan sensor garis diletakkan di *chassis* Rover 5.



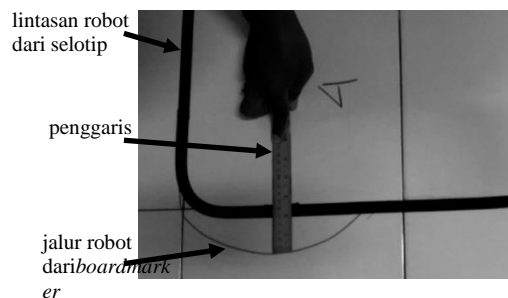
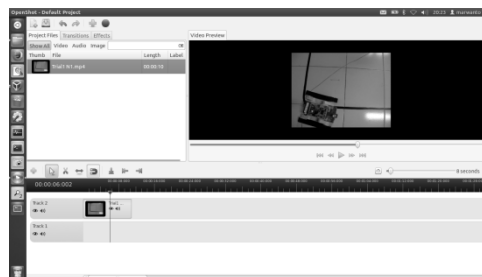
Gambar 3.2 Obyek penelitian

Camcoder disiapkan dengan memberikan tempat tepat di atas lantai agar dapat merekam gerakan robot dengan jelas seperti pada gambar 3.3.

Robot pengikut garis diprogram dengan menggunakan bahasa pemrograman Bascom AVR (*Basic Compiler AVR*). Kepekaan sensor garis diatur melalui software dengan memanfaatkan fasilitas *getadc* dari program Bascom AVR.

Gambar 3.4 Posisi *camcoder*

Untuk mendapatkan data *overshoot*, diukur simpangan maksimum antara lintasan robot dari selotip isolasi dengan jalur robot dari hasil *boardmarker*.

Gambar 3.5 Pengukuran *overshoot*Gambar 3.6 *OpenShot Video Editor*

Untuk mendapatkan data *settling time*, hasil rekaman dari *camcoder* dipindahkan ke laptop dan diputar menggunakan program *OpenShot Video Editor* untuk dilihat waktu robot dari awal menyimpang lintasan sampai kembali ke lintasannya.

Pada penelitian ini faktor kendali terdiri dari 4 faktor (A,B,C,D) dengan 3 level (1,2 dan 3) sehingga matriks orthogonal yang digunakan adalah $L_9 (3^4)$.

4. Hasil dan Pembahasan

Data hasil pengamatan *overshoot* dan *settling time* untuk robot pengikut garis tanpa kendali PID dan jumlah sensor 2 buah disajikan dalam tabel 4.1.

Tabel 4.1 *Overshoot* dan *settling time* robot tanpa PID, 2 sensor garis

Replikasi	<i>Overshoot</i> (cm)	<i>Settling time</i> (s)
1	8.0	6.70
2	9.0	7.47
3	8.0	7.00
rerata	8.333	7.057

Data hasil pengamatan *overshoot* dan *settling time* untuk robot pengikut garis menggunakan kendali PID dengan memvariasikan faktor kendali konstanta proporsional (A), konstanta integral (B), konstanta derivatif (C) dan jumlah sensor (D) masing-masing dengan 3 level faktor disajikan dalam tabel 4.2 dan 4.3.

Tabel 4.2 Hasil pengamatan *overshoot* (sentimeter)

Trial	Faktor Kendali				<i>Overshoot</i> (sentimeter)			Rerata
	A	B	C	D	Y_1	Y_2	Y_3	\bar{Y}
1	1	1	1	1	8.0	8.5	6.5	7.667
2	1	2	2	2	5.9	5.8	6.0	5.900
3	1	3	3	3	5.5	5.4	5.0	5.300
4	2	1	2	3	6.3	8.0	6.6	6.967
5	2	2	3	1	7.5	6.0	6.3	6.600
6	2	3	1	2	8.0	9.5	8.2	8.567
7	3	1	3	2	6.0	7.6	8.1	7.233
8	3	2	1	3	5.1	6.0	7.0	6.033
9	3	3	2	1	5.0	5.6	6.7	5.767

Tabel 4.3. Hasil pengamatan *settling time* (detik)

Trial	Faktor Kendali				<i>Settling time</i> (sentimeter)			Rerata
	A	B	C	D	Y_1	Y_2	Y_3	\bar{Y}
1	1	1	1	1	5.30	5.70	4.90	5.300
2	1	2	2	2	4.87	4.20	3.90	4.322
3	1	3	3	3	4.07	3.83	3.60	3.833
4	2	1	2	3	5.00	3.90	4.90	4.600
5	2	2	3	1	4.70	3.90	4.47	4.356
6	2	3	1	2	5.77	6.03	6.00	5.932
7	3	1	3	2	4.83	5.67	5.00	5.167
8	3	2	1	3	3.53	3.67	4.70	3.967
9	3	3	2	1	3.63	3.83	4.73	4.067

Pengujian normalitas data digunakan uji Kolmogorov Smirnov dengan software IBM®SPSS® Statistics 20 diperoleh kesimpulan bahwa data *overshoot* dan *settling time* berdistribusi normal.

Uji homogenitas menggunakan Uji Bartlett diperoleh kesimpulan bahwa data *overshoot* dan *settling time* bersifat homogen.

Untuk mengetahui faktor utama yang berpengaruh secara signifikan terhadap variabel respon digunakan Analisis Variansi (ANOVA) dan diperoleh kesimpulan bahwa semua faktor kendali berpengaruh secara signifikan terhadap nilai respon *overshoot* dan *settling time*.

Berdasar perhitungan SNR dan efek tiap faktor diperoleh nilai optimal tiap faktor untuk respon *overshoot* seperti pada tabel 4.4 dan untuk respon *overshoot* seperti pada tabel 4.5.

Tabel 4.4 Nilai optimal tiap faktor untuk respon *overshoot*

Level	Faktor			
	A	B	C	D
1	-15.885	-17.302	-17.370	-16.487
2	-17.341	-15.844	-15.870	-17.118
3	-16.073	-16.153	-16.059	-15.695
Selisih	1.456	1.458	1.500	1.423
Rangking	3	2	1	4

Tabel 4.5 Nilai optimal tiap faktor untuk respon *settling time*

Level	Faktor			
	A	B	C	D
1	-12.979	-14.031	-14.004	-13.184
2	-13.859	-12.534	-12.767	-14.168
3	-12.858	-13.131	-12.925	-12.344
Selisih	1.002	1.497	1.236	1.824
Rangking	4	2	3	1

Dari tabel 4.4 diperoleh kombinasi level faktor optimal untuk *overshoot* adalah $A_1B_2C_2D_3$, artinya untuk minimasi *overshoot* dibutuhkan kombinasi level faktor $K_p = 100$, $K_i = 0.5$, $K_d = 10$ dan jumlah sensor = 8 buah. Dari tabel 4.5 diperoleh kombinasi level faktor optimal adalah $A_3B_2C_2D_3$, artinya untuk minimasi *settling time* dibutuhkan kombinasi level faktor $K_p = 140$, $K_i = 0.5$, $K_d = 14$ dan jumlah sensor = 8 buah.

Kombinasi level faktor optimal yang diperoleh untuk kedua respon tidak sama yaitu $A_1B_2C_2D_3$ untuk respon *overshoot* dan $A_3B_2C_2D_3$ untuk respon *settling time* sehingga perlu dilakukan optimasi multirespon MRSN (*Multi Respon Signal to Noise*).

Prediksi

Sebelum melakukan optimasi multirespon MRSN dilakukan dulu prediksi dengan menentukan model matematis dari data hasil eksperimen yang telah dilakukan. Model matematis yang digunakan adalah model regresi linear dan dilakukan dengan menggunakan *software* SPSS.

Dari *software* SPSS diperoleh persamaan matematis respon *overshoot* dan *settling time* untuk masing-masing replikasi sebagai berikut :

Untuk respon *overshoot* :

$$Y_1 = 12.683 - 0.028A - 0.639B - 0.088C - 0.300D$$

$$Y_2 = 10.509 - 0.004A - 1.205B - 0.208C - 0.058D$$

$$Y_3 = 4.057 + 0.036A - 0.445B - 0.096C - 0.075D$$

Untuk respon *settling time* :

$$Y_1 = 8.124 - 0.019A - 0.580B - 0.042C - 0.086D$$

$$Y_2 = 7.205 + 0.005A - 0.509B - 0.083C - 0.169D$$

$$Y_3 = 4.234 + 0.017A - 0.133B - 0.105C - 0.075D$$

Dari persamaan-persamaan tersebut disusun prediksi untuk 81 trial (3^4) masing-masing respon.

MRSN

Langkah-langkah yang sistematis dalam melakukan eksperimen multirespon dengan menggunakan *Multi Respon Signal to Noise* (MRSN) terdiri dari beberapa tahapan, yaitu :

a. Menghitung *Quality Loss* (L_{ij}) untuk setiap trial. Untuk karakteristik kualitas *Smaller-the-better* (STB) digunakan rumus :

$$L_{ij} = k \frac{1}{n_i} \sum_{k=1}^{n_i} y_{ijk}^2$$

dengan koefisien *quality loss overshoot* 10,906.404 dan koefisien *quality loss settling time* 5,495.330.

b. Menentukan *Multi Respon Signal to Noise* (MRSN).

1) Menentukan *Quality Loss* maksimum (L_i^*) untuk tiap respon.

Dari Tabel MRSN diperoleh *Quality Loss* maksimum untuk *overshoot* sebesar 805,219.795 dan *Quality Loss* maksimum untuk *settling time* sebesar 193,463.740.

2) Normalisasi *Quality Loss* (C_{ij}) setiap eksperimen menggunakan rumus :

$$C_{ij} = \frac{L_{ij}}{L_i^*}$$

3) Menghitung *Total Normalized Quality Loss* (TNQL) setiap eksperimen dengan rumus :

$$TNQL_j = \sum_{i=1}^m w_i C_{ij}$$

dengan bobot *settling time* (w_1) = 0.562641 dan bobot *overshoot* (w_2) = 0.437359.

4) Menghitung MRSN ratio setiap eksperimen menggunakan rumus :

$$MRSN_j = -10 \log(TNQL_j)$$

c. Menentukan kombinasi level faktor yang optimal berdasarkan MRSN terbesar.

Berdasar Tabel MRSN, nilai MRSN terbesar yaitu 3.957 pada trial ke-27 dengan kombinasi level faktor $A_1 B_3 C_3 D_3$ (level faktor $K_p = 100$; $K_i = 1$; $K_d = 14$ dan jumlah sensor = 8).

Kombinasi level faktor $A_1 B_3 C_3 D_3$ masuk dalam ortogonal array (sudah dieksperimenkan) sehingga tidak diperlukan eksperimen konfirmasi.

Uji Beda

Uji beda dilakukan untuk membandingkan kualitas antara kondisi optimal hasil perhitungan MRSN dengan kondisi robot pengikut garis tanpa pengendali PID dan jumlah sensor garis 2 buah.

Dari uji beda dapat disimpulkan bahwa terdapat peningkatan kualitas *overshoot* dan *settling time*.

5. Kesimpulan

Berdasarkan pada eksperimen robot pengikut garis yang telah dilaksanakan dengan menggunakan metode Taguchi multi respon (*Multi Respon Signal to Noise Ratio*) untuk mendapatkan nilai optimal respon robot berupa minimalisasi nilai *overshoot* dan *settling time*, diperoleh kombinasi level faktor konstanta pengendali berupa konstanta proporsional (K_p) sebesar 100, konstanta integral (K_i) sebesar 1 dan konstanta derivatif (K_d) sebesar 14 serta jumlah sensor 8 buah. Dengan kombinasi level faktor konstanta pengendali tersebut diperoleh minimum *overshoot* sebesar 5.300 sentimeter dan *settling time* sebesar 3.833 detik.

Daftar Pustaka

- [1] Ahmad Sahrul Romadhon, Muhammad Fuad. 2013. "Perancangan Sistem Kontrol Gerakan Pada Robot Line Tracer". *Jurnal Ilmiah Mikrotek Vol. 1, No.1 2013*.
- [2] Anirudh Sunil Nath, Aravind Kumar, Tarun Malik. 2013. "Implementation of PID Control to Reduce Wobbling in a Line Following Robot". *IJRET: International Journal of Research in Engineering and Technology, Volume: 02 Issue: 10, Oktober 2013*.
- [3] Arman Jaya, Irianto, Amin Setiadji. 2010. "Implementasi Kontroler PID Pada AVR (Automatic Voltage Regulator) Untuk Pengaturan Tegangan Eksitasi Generator Sinkron 3 Fasa".
- [4] Daisy A.N Janis, David Pang, ST., MT, J. O. Wuwung ST., MT. 2014. "Rancang Bangun Robot Pengantar Makanan Line follower". *e-journal Teknik Elektro dan Komputer*.

- [5] Deepak Punetha, Neeraj Kumar, Vartika Mehta. 2013. “Development and Applications of Line Following Robot Based Health Care Management System” **International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Volume 2, Issue 8, August 2013.**
- [6] Ilham Wha’is Abdhan Saquro. 2013. “Aplikasi Sistem Multiplexer Untuk Penambahan Jumlah Sensor Pada Robot Line Follower Berbasis Atmega32”.
- [7] Islam, M.S. &Rahman, M.A. 2013. “Design and Fabrication of Line Follower Robot” **Asian Journal of Applied Science and Engineering, Volume 2, No 2 (2013).**
- [8] Manoj Gogoi. 2010. “Proportional-Integral-Derivative (PID) Controller Design For Robust Stability of Arbitrary Order Plants With Time-Delay and Additive Uncertainty”.
- [9] Mohammad Rokonuzzaman, S.M. Ferdous, Sayedus Salehin, S.M.A.A. Hasnine, Feroz Ahmed. 2010. “Smooth Track-keeping and Real Time Obstacle Detection Algorithm with PID Controller for a Wheeled Line Following Robot”. **International Conference on Mechanical, Industrial and Energy Engineering 2010, 23-24 December 2010.**
- [10] Nasri Widha Setyanto, Lely Riawati, Rio Prasetyo Lukodono. 2014. “Desain Eksperimen Taguchi untuk Meningkatkan Kualitas Pupuk Organik Berbahan Baku Kotoran Kelinci”. **JEMIS, Vol. 2, No. 2, Tahun 2014.**