E-ISSN : 2988-1986
https://ejournal.warunayama.org/kohesi

Kohesi: Jurnal Multidisiplin Saintek
Volume 7 No 1 Tahun 2025

IMPLEMENTASI METODE PI PADA MOTOR DC UNTUK PROSES PEMBENTUKAN BONGGOL JAGUNG

Alif Komaruddin[1], Supriatna Adhisuwignjo [2], Mila Fauziyah [3]
[1, 3] Jurusan Teknik Elektro, Politeknik Negeri Malang, Jalan Soekarno Hatta No.9 Malang, Indonesia.
Email: 825alif@gmail.com

Abstract

The process of producing corn cobs which are used as craft materials still has problems, especially regarding the tools used. The problem with forming corn cobs is that the method is still manual using a grinding machine, the production process is still less than standardized and there is still a lack of safety during the process of forming prism-shaped corn cobs (triangles and squares) to avoid unwanted work accidents. So technology is utilized on corn weevil tools using the PI method so that work safety can be increased. PI control is embedded in the ESP 32 microcontroller to regulate the speed of the DC motor as a suppressor in the process of forming corn cobs and an ultrasonic sensor to determine the size of corn cobs. By controlling the maximum response speed of 100 RPM on the DC motor and adjusting the distance during the pressing process, the corn cob formation process is faster, the shape of the corn cob is more precise, and avoids burning on the sanded corn cob. In this PI control, the Ziegler-Nichols 2 method is used to obtain the appropriate PI parameters. Tests were carried out with a Kp value of 0.255, a Ki value of 0.135 and setpoints of 40, 50, 60 RPM. From several experiments, stable results were obtained, namely at a set point of 50 RPM with a time delay of 1.7s, rise time 5s, peak time 3s, settling time 7s, max overshoot 2% and steady state error 2%.

Keywords: *Speed, Motor DC, Ultrasonic*

Abstrak

Proses produksi bonggol jagung yang dijadikan sebagai bahan kerajinan masih ada permasalahan khususnya pada alat yang dipakai. Dimana permasalahan Pembentukan bonggol jagung yang metodenya masih secara manual menggunakan mesin gerinda, proses produksinya masih kurang standart dan masih kurangnya keamanan pada saat proses pembentukan bonggol jagung berbentuk prisma (segitiga dan persegi) untuk menghindari kecelakaan kerja yang tidak diinginkan. Maka dilakukan pemanfaatan teknologi pada alat bonggol jagung dengan menggunakan metode PI dengan begitu keselamatan kerja bisa di tingkatkan. Kontrol PI ditanam di microkontroller ESP 32 untuk mengatur kecepatan motor DC sebagai penekan dalam proses pembentukan bonggol jagung dan sensor ultrasonik untuk menentukan ukuran bonggol jagung. Dengan mengontrol respon kecepatan maksimal 100 RPM pada motor DC serta mengatur jarak saat proses penekanan, menjadikan proses pembentukan bonggol jagung lebih cepat, bentuk bonggol jagung lebih presisi, dan

Article History
Received: Januari 2025
Reviewed: Januari 2025
Published: Januari 2025

Plagirism Checker No 234
Prefix DOI : Prefix DOI :
10.8734/Kohesi.v1i2.365
Copyright : Author
Publish by : Kohesi

menghindari terjadinya gosong pada bonggol jagung yang di amplas. Pada kontrol PI ini menggunakan metode Ziegler-Nichols 2 untuk mendapatkan parameter PI yang sesuai. Dilakukan pengujian dengan nilai Kp 0.255, nilai Ki 0.135 dan setpoint 40, 50, 60 RPM. Dari beberapa kali percobaaan didapatkan hasil yang stabil yaitu di setpoint 50 RPM dengan hasil Time delay 1.7s, rise time 5s, peak time 3s, settling time 7s, overshoot maks 2% dan error steady state 2%.
Kata kunci: Kecepatan, Motor DC, Ultrasonik

PENDAHULUAN

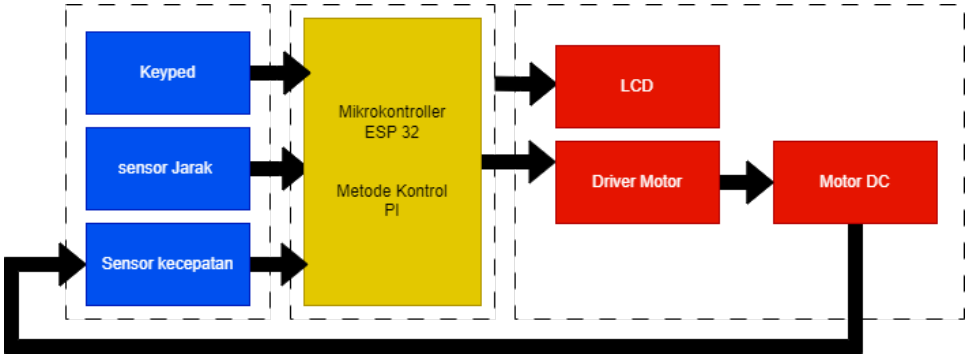
Motor DC adalah suatu perangkat yang mengubah energi listrik menjadi energi kinetik atau gerakan (motion)[1]. Motor DC ini juga dapat disebut sebagai Motor Arus Searah[2]. Motor DC memiliki dua terminal dan memerlukan tegangan arus searah atau DC (Direct Current) untuk dapat menggerakannya [3]. Motor Listrik DC biasanya digunakan pada perangkat-perangkat Elektronik dan listrik yang menggunakan sumber listrik DC seperti Vibrator Ponsel[4], Kipas DC[5], dan Bor Listrik DC[6]. Motor Listrik DC ini menghasilkan sejumlah putaran per menit atau biasanya dikenal dengan istilah RPM (Revolutions per minute) dan dapat dibuat berputar searah jarum jam[7] maupun berlawanan arah jarum jam apabila polaritas listrik yang diberikan pada Motor DC tersebut dibalikan[8]. Motor Listrik DC tersedia dalam berbagai ukuran rpm dan bentuk[9]. Kebanyakan Motor Listrik DC memberikan kecepatan rotasi sekitar 3000 rpm hingga 8000 rpm[10]. Motor DC memerlukan suplai tegangan arus searah pada kumparan medan untuk diubah menjadi energi gerak mekanik[11]. Kumparan medan pada motor DC disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar)[12].

Pada penelitian ini mesin penghalus dalam proses pembentukan bonggol jagung memanfaatkan putaran motor DC. Namun, salah satu kelemahan dari motor DC adalah tidak mampu mempertahankan kecepatannya dengan stabil bila terjadi perubahan beban. Apabila terjadi perubahan beban maka kecepatan motor DC akan menurun [13]. Oleh karena itu, pengendalian yang akurat dan efisien dari motor DC adalah hal yang sangat penting. Dalam penelitian sebelumnya telah dilakukan aksi kontrol untuk menstabilkan putaran kecepatan motor antara lain dengan pemanfaatan kontrol PI [14], kontrol PID [15], dan Kontrol Fuzzy [16]. Pada kasus ini metode otomasi yang paling efektif untuk mengendalikan motor DC adalah pengendalian PI.

Pengendalian PI telah menjadi pilihan dalam pengendalian motor induksi karena kemampuannya untuk menangani gangguan dan kesalahan dengan baik, serta kemudahan dalam pengaturan parameter yang sesuai [17]. Untuk mencapai kinerja yang optimal dalam sistem kendali, parameter kontrol seperti Kp dan Ki harus diatur dengan cermat. Tuning nilai konstanta pada kontrol PI dapat dilakukan dengan uji coba eksperimen atau penggunaan metode berbasis aturan, Ziegler-Nichols [18][19][20][21][22].

METODE

Prinsip Kerja Alat

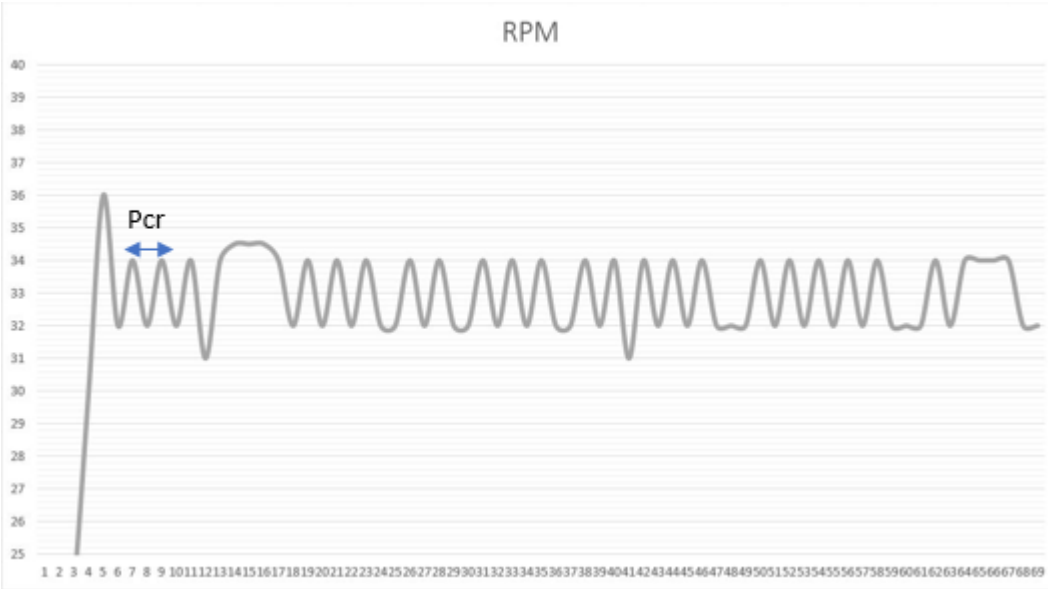


Gambar 1 Blok Diagram Sistem

Pertama bonggol jagung dimasukkan ke dalam penjepit. Kemudian bonggol jagung akan didorong oleh motor DC. Keypad digunakan untuk memasukkan nilai set point kecepatan putar motor DC. LCD digunakan untuk menampilan setpoint dan kecepatan putar motor DC serta besar arus yang tercapai pada saat itu. Setelah setpoint dimasukkan maka mikrokontroller ESP 32 akan mengolah data dengan membandingkan pembacaan kecepatan putar motor DC oleh sensor kecepatan dengan setpoint yang telah ditentukan. Kemudian motor dan sensor kecepatan akan membaca kecepatan motor DC yang sedang berjalan. Kecepatan motor tersebut akan distabilkan dan disesuaikan kecepatannya oleh Kontrol PI sesuai dengan setpoint yang diberikan. ESP 32 akan membandingakan nilai kesalahan atau error kecepatan yang dibaca oleh sensor FC03IR kemudian secara otomatis akan mengeluarkan perintah untuk menyesuaikan motor sesuai dengan setpoint yang diberikan.

Perancangan PI

Pada metode ini dilakukan dengan cara meningkatkan nilai Kp dari 0 ke nilai kritis Kcr dimana *output* pertama menunjukan osilasi berkelanjutan Setelah mendapatkan nilai Kcr kemudian menentukan nilai Pcr. Nilai Pcr adalah jarak antara puncak gelombang. Untuk mencari nilai Kp, Ti dengan metode *ziegler- nichlos* 2. Gambar 2 Merupakan respon sistem berosilasi saat diberikan nilai Kcr = 0.5



Gambar 2 Respon Sistem Saat nilai Kcr = 0.5

Dari hasil penyetelan nilai Kp, didapatkan respon berosilasi saat diberi nilai Kcr = 0,5, dan didapatkan t1 = 7 dan t2 = 9. Sehingga nilai Pcr didapatkan Seperti pada persamaan 1.

$$Pcr = t2 - t1 \quad (1)$$

$$= 9 - 7$$
$$= 2$$

Sehingga untuk mencari nilai Kp, dan Ki didapatkan seperti persamaan 1, 2, 3, 4

- Mencari Nilai Kp

$$Kp = 0{,}45\ x\ Kcr \quad (2)$$
$$= 0.6 \text{ x } 0.5$$
$$= 0.225$$

- Mencari Nilai Ki

$$Ti = \frac{1}{1.2}\ x\ Pcr \quad (3)$$
$$= \frac{1}{1.2} \text{ x } 2$$
$$= 1{,}667$$
$$Ki = {}^{Kp}/_{Ti} \quad (4)$$
$$= {}^{0{,}225}/_{1.667}$$
$$= 0{,}135$$

Keterangan :
Kcr = Konstanta *critical*
Pcr = Periode *critical*
Kp = Konstanta *proportional*
Ki = Kontansta *integral*
Hasil nilai Kp, Ki, dan Kd setelah dihitung dengan metode *Ziegler- nichols* 2 adalah Kp = 0.225, Ki = 0.135. Selanjutnya akan dimasukan kedalam program mikrokontroller.

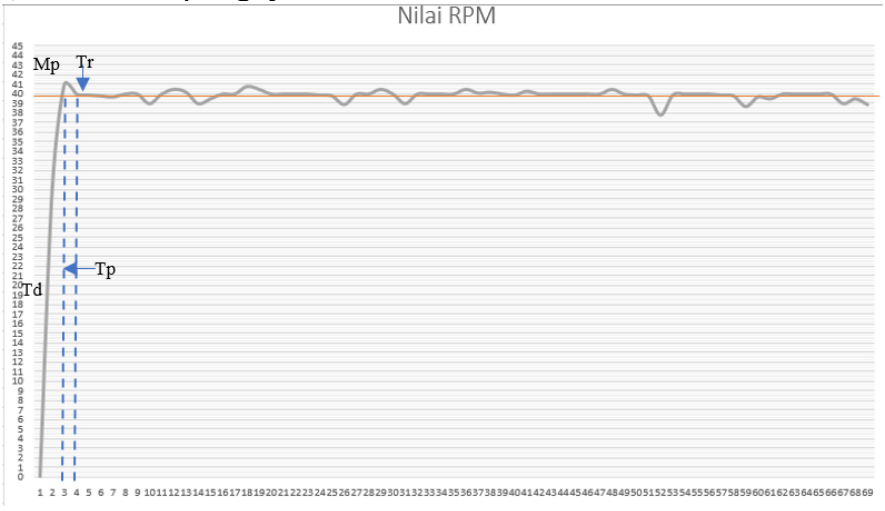
HASIL DAN DISKUSI

Pengujian Sistem Keseluruhan

Hasil pengujian keseluruhan yaitu menggunakan bonggol jagung yang berbentuk persegi dan segitiga dilakukan pengujian sebanyak 3 kali. Pengujian 1 setpoint 40 RPM, pengujian 2 setpoint 50 RPM, pengujian 3 Setpoint 60 RPM.

Pengujian 1

Gambar 3 menunjukkan grafik respon kecepatan dengan setpoint 40 RPM dengan nilai Kp = 0.255, Ki = 0.135 pengujian 1



Tp

Gambar 3 Kp = 0.255, Ki = 0.135 dengan *Setpoint* 40 RPM

Dari hasil respon yang ditunjukkan pada gambar 3 dapat dianalisa respon sistem yang dihasilkan yaitu seperti berikut:

a. *Delay time* (td) merupakan waktu yang diperlukan respon untuk mencapai 50% dari *setpoint*. Pada grafik yang ditunjukkan gambar 3 nilai td = 1,8 detik
b. *Rise time* (tr) merupakan waktu yang dibutuhkan sistem untuk mencapai nilai *setpoint* mulai dari t = 0 hingga respon mencapai sumbu *setpoint* yang pertama. Pada grafik yang ditunjukkan gambar 3 nilai Tr = 4 detik.
c. *Peak time* (tp) merupakan waktu puncak yang diperlukan respon menuju titik puncak pertama dari *overshoot*. Pada grafik yang ditunjukkan gambar 3 nilai Tp = 3 detik.
d. *Settling time* (ts) merupakan waktu yang diperlukan sistem untuk mencapai dan tetap berada dalam kriteria toleransi (2%) setelah *rise time*. Pada grafik yang ditunjukkan gambar 3 nilai Ts = 8 detik
e. *Overshoot* maks (Mp) merupakan hasil perbandingan antara nilai maksimum respon (*Overshoot*) yang melebihi nilai *steady state* dimana nilai tersebut dikonversi dalam bentuk persentase. Adapun berdasarkan persamaan 4.1 maka dapat menentukan nilai persentasi *overshoot* yaitu:

$$\%Os = \frac{Stp - Ssp}{Ssp} \times 100\% \qquad (4.1)$$

Dimana:

%Os = Persentase osilasi
Stp = Nilai RPM saat puncak
Ssp = Nilai RPM saat *setpoint*

f.

$$\%Os = \frac{41 - 40}{40} \times 100\%$$
$$\%Os = \frac{1}{40} \times 100\%$$
$$\%Os = 2.5\%$$

g. *Error steady state* dapat ditentukan dengan mencari nilai puncak tertinggi dan terendah dari respon sistem. Berikut ini adalah rumus untuk mennetukan nilai *error* atas dan bawah dari *steady state* sistem seperti pada persamaan 4.1.

$$ess = \frac{puncak - setpoint}{setpoint} \times 100\% \qquad (4.1)$$

Dimana:

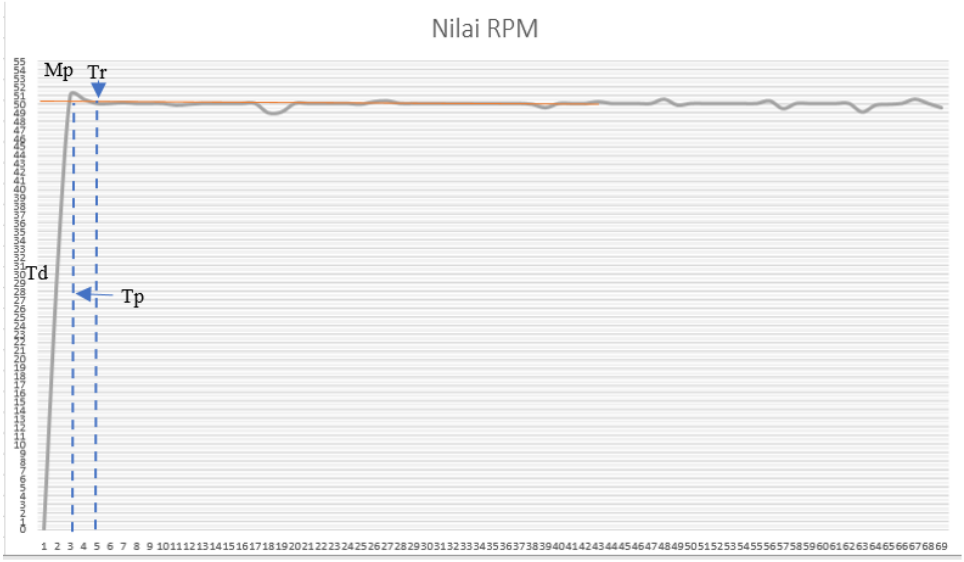
ess = *Error steady state*
puncak = Puncak atas atau bawah
setpoint = Nilai RPM saat *setpoint*

h.

$$ess\ (atas) = \frac{41 - 40}{40} \times 100\%$$
$$= 2.5\%$$
$$ess\ (bawah) = \frac{37.8 - 40}{40} \times 100\%$$
$$= -5.5\%$$

Pengujian 2

Gambar 4 menunjukkan grafik respon kecepatan dengan setpoint 50 RPM dengan nilai Kp = 0.255, Ki = 0.135 pengujian 2.



Gambar 4 Kp = 0.255, Ki = 0.135 dengan Setpoint 50 RPM

Dari hasil respon yang ditunjukkan pada gambar 4 dapat dianalisa respon sistem yang dihasilkan yaitu seperti berikut:

a. *Delay time* (td) merupakan waktu yang diperlukan respon untuk mencapai 50% dari *setpoint*. Pada grafik yang ditunjukkan gambar 4 nilai td = 1,7 detik
b. *Rise time* (tr) merupakan waktu yang dibutuhkan sistem untuk mencapai nilai *setpoint* mulai dari t = 0 hingga respon mencapai sumbu *setpoint* yang pertama. Pada grafik yang ditunjukkan gambar 4 nilai Tr = 5 detik.
c. *Peak time* (tp) merupakan waktu puncak yang diperlukan respon menuju titik puncak pertama dari *overshoot*. Pada grafik yang ditunjukkan gambar 4 nilai Tp = 3 detik.
d. *Settling time* (ts) merupakan waktu yang diperlukan sistem untuk mencapai dan tetap berada dalam kriteria toleransi (2%) setelah *rise time*. Pada grafik yang ditunjukkan gambar 4 nilai Ts = 7 detik
e. *Overshoot* maks (Mp) merupakan hasil perbandingan antara nilai maksimum respon (*Overshoot*) yang melebihi nilai *steady state* dimana nilai tersebut dikonversi dalam bentuk persentase. Adapun berdasarkan persamaan 4.1 maka dapat menentukan nilai persentasi *overshoot* yaitu:

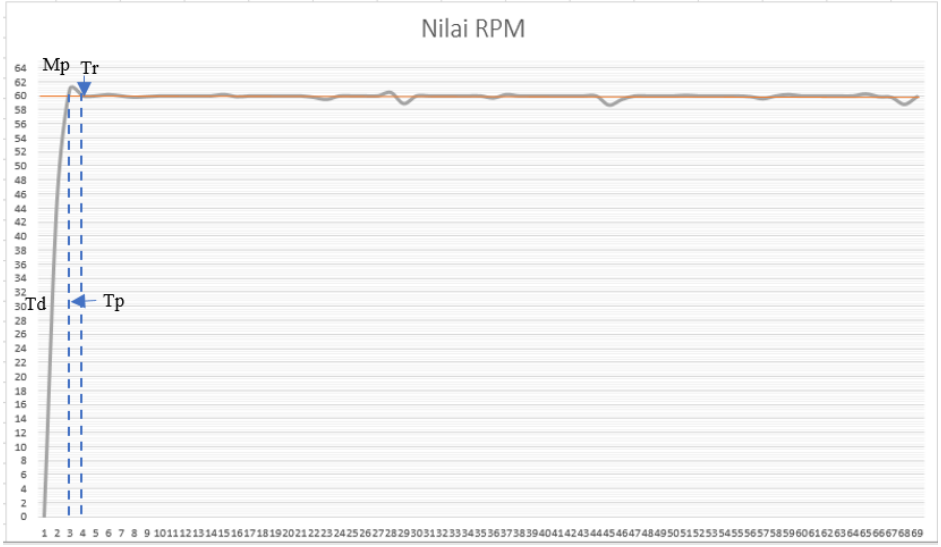
$$\%Os = \frac{51-50}{50} \times 100\%$$
$$\%Os = \frac{1}{50} \times 100\%$$
$$\%Os = 2\%$$

f. *Error steady state* dapat ditentukan dengan mencari nilai puncak tertinggi dan terendah dari respon sistem. Berikut ini adalah rumus untuk mennetukan nilai *error* atas dan bawah dari *steady state* sistem seperti pada persamaan 4.4.

$$ess\ (atas) = \frac{51-50}{50} \times 100\%$$
$$= \quad 2\%$$
$$ess\ (bawah) = \frac{48.9-50}{50} \times 100\%$$
$$= \quad -2.2\%$$

Pengujian 3

Gambar 5 menunjukkan grafik respon kecepatan dengan setpoint 60 RPM dengan nilai Kp = 0.255, Ki = 0.135 pengujian 3.



Gambar 5 Kp = 0.255, Ki = 0.135 dengan *Setpoint* 60 RPM

Dari hasil respon yang ditunjukkan pada gambar 5 dapat dianalisa respon sistem yang dihasilkan yaitu seperti berikut:

a. *Delay time* (td) merupakan waktu yang diperlukan respon untuk mencapai 50% dari *setpoint*. Pada grafik yang ditunjukkan gambar 5 nilai td = 1,7 detik
b. *Rise time* (tr) merupakan waktu yang dibutuhkan sistem untuk mencapai nilai *setpoint* mulai dari t = 0 hingga respon mencapai sumbu *setpoint* yang pertama. Pada grafik yang ditunjukkan gambar 5 nilai Tr = 4 detik.
c. *Peak time* (tp) merupakan waktu puncak yang diperlukan respon menuju titik puncak pertama dari *overshoot*. Pada grafik yang ditunjukkan gambar 5 nilai Tp = 3 detik.
d. *Settling time* (ts) merupakan waktu yang diperlukan sistem untuk mencapai dan tetap berada dalam kriteria toleransi (2%) setelah *rise time*. Pada grafik yang ditunjukkan gambar 5 nilai Ts = 6 detik
e. *Overshoot* maks (Mp) merupakan hasil perbandingan antara nilai maksimum respon (*Overshoot*) yang melebihi nilai *steady state* dimana nilai tersebut dikonversi dalam bentuk persentase. Adapun berdasarkan Persamaan 4.1 maka dapat menentukan nilai persentasi *overshoot* yaitu:

$$\%Os = \frac{61 - 60}{60} \times 100\%$$
$$\%Os = \frac{1}{60} \times 100\%$$
$$\%Os = 1.6\%$$

f. *Error steady state* dapat ditentukan dengan mencari nilai puncak tertinggi dan terendah dari respon sistem. Berikut ini adalah rumus untuk mennetukan nilai *error* atas dan bawah dari *steady state* sistem seperti pada Persamaan 4.4.

$$ess\ (atas) = \frac{61 - 60}{60} \times 100\%$$
$$= \quad 1.6\%$$
$$ess\ (bawah) = \frac{58.7 - 60}{60} \times 100\%$$
$$= \quad -2.1\%$$

Gambar 6 Hasil Produk Akhir

KESIMPULAN

Dari hasil pengujian dan analisa pada implementasi kontrol PI pada pengaturan kecepatan motor DC dalam proses pembentukan bonggol jagung berbasis ESP 32,penggunaan metode PI dengan penerapan metode *Ziegler Nichols* 2 diperoleh nilai Kp = 0,255 Ki = 0,135 dengan menggunakan pengujian penepanan bonggol jagung, *setpoint* kecepatan 40 RPM dan waktu pembentukan 6 menit. Hasil grafik respon sistem yaitu *delay time* (td) 1.8 s, *rise time* (tr) 4 s, *peak time* (tp) 3 s, settling time(ts) 8 s, *error steady state* atas 2,5%, *error steady state* bawah -5,5%, *overshoot* 2,5%. Hasil terbaik berdasarkan pengukuran setpoint menunjukan kecepatan optimal pengadukan 50 RPM menghasilkan bonggol jagung yang halus dan bentuk nya sesuai dengan kriteria yang dinginkan.

REFERENSI

[1] Rancang Bangun Sistem Recording Dan Kontrol Kecapatan Motor Dc Menggunakan Esp8266 Berbasis Internet Of Things

[2] Analisa motor DC (Direct Current) sebagai penggerak mobil listrik

[3] Rancang Bangun Pengereman Motor Direct Current Pada Mobil Listrik

[4] Amin, Muhammad., Ananda, Ricki., "SISTEM KENDALI JARAK JAUH ROBOT PEMADAM API DENGAN MENGGUNAKAN SENSOR FLAM DAN SENSOR MQ BERBASIS MOTOR POMPA "

[5] Rachmadi, Bagas., Pristisahida, Adelia Octora., "Optimasi Kenyamanan Suhu Ruangan Melalui Perancangan Otomatisasi Kipas Angin di Masjid Nurhidayatullah Maguwoharjo"

[6] Analysis of DC MCB Usage Characteristics for AC and DC Load Usage

[7] Rancang Bangun Pemrograman Smart Relay Zelio Untuk Pengaturan Pemberian Pakan Ikan Secara Otomatis

[8] Gultom, Togar Timoteus., S. Suhelmi., "SISTEM PEMBUMIAN PADA KAPAL PENANGKAP IKAN MENGGUNAKAN ENERGI TERBARUKAN"

[9] Prasetyo , Joyo., Tohir, Safaruddin., Purwanto, Heru,. "PENGAPLIKASIAN VARIABLE SPEED DRIVE UNTUK MENGONTROL KECEPATAN MAIN MOTOR DRIVE DC PADA ROTARI KILN PADA PT SEMEN BATURAJA (PERSERO) Tbk" Vol. 1 No. 04 (2022): JURNAL MULTIDISIPLINER KAPALAMADA

[10] Sabri, Muhammad Irvan "PROSES PENGOPERASIAN MOTOR DC 10 KW DENGAN MENGGUNAKAN DC DRIVE PARKER 590 DAN PERAWATAN MOTOR DC 10 KW DI PT. BICC BERCA CABLES".

[11] Rancang Bangun Jangkar Motor DC

[12] Kontrol kecepatan motor brushless dc menggunakan six step comutation dengan kontrol pid (propotional integral derivative)
[13] Sensorless Indirect Field-Oriented Control of Induction Motor using Intelligent PI Controller
[14] Adaptive Partial PID Controller For Speed Control of Induction Motor
[15] Self-Tuned Output Scaling Factor of Fuzzy Logic Speed Control of Induction Motor Drive
[16] PENGATURAN KECEPATAN MOTOR INDUKSI DENGAN METODE V/f (Volt/Frequence) DAN KONTROL PI-FUZZY
[18] R. Aisuwarya and Y. Hidayati, "Implementation of Ziegler-Nichols PID Tuning Method on Stabilizing Temperature of Hot-water Dispenser," 2019. https://doi.org/10.1109/QIR.2019.8898259.
[19] H. M. Shariff, M. H. F. Rahiman, R. Adnan, M. H. Marzaki, M. Tajjudin, and M. H. A. Jalil, "The PID Integrated Anti-Windup Scheme by ZieglerNichols Tuning for Small-Scale Steam Distillation Process," 2019. https://doi.org/10.1109/ICSEngT.2019.8906436.
[20] N. N. B. M. Mazlan, N. M. Thamrin, and N. A. Razak, "Comparison Between Ziegler-Nichols and AMIGO Tuning Techniques in Automated Steering Control System for Autonomous Vehicle". 2020 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), 2020, pp. 7-12. https://doi.org/10.1109/I2CACIS49202.2020.9140089.
[21] S. A. Bhatti, S. A. Malik, and A. Daraz, "Comparison of P-I and I-P Controller by Using Ziegler-Nichols Tuning Method for Speed Control of DC Motor," 2016. https://doi.org/10.1109/INTELSE.2016.7475144.
[22] C. A. Aung, Y. V. Hote, G. Pillai, and S. Jain, "PID controller design for solar tracker via modified ziegler nichols rules," in 2020 2nd International Conference on Smart Power and Internet Energy Systems, SPIES 2020, Sep. 2020, pp. 531–536. https://doi.org/10.1109/SPIES48661.2020.9243009.