

IMPLEMENTASI METODE PI (PROPORTIONAL INTEGRAL) PADA PENGATURAN KECEPATAN *CRUSHER MOTOR* DALAM PROSES EKSTRAKSI BUAH APEL

Pradana Perwira Putranto, Sungkono, ST., MT., Edi Sulistio Budi, ST., MT.

Abstrak – Sari apel adalah larutan inti yang terkandung dalam buah apel yang diekstrak kemudian dicampur dengan atau tanpa gula dan tambahan makanan lainnya yang diizinkan untuk dibuat minuman sari apel. Proses ekstraksi sari apel umumnya dilakukan dengan menghancurkan buah apel kemudian ditekan untuk mendapatkan sarinya. Perlu adanya mekanisme untuk mengekstrak sari apel dari buahnya yang dapat bekerja secara terus-menerus. Di Indonesia, skala *home industry* dalam pengolahan sari apel dengan sistem otomatisasi masih sedikit, maka dari itu perlu dibuat *miniplant* berupa ekstraktor sari apel, yang kecepatan *crusher motor* apelnya dapat dikontrol menggunakan metode PI (*Proportional Integral*). *Crusher motor* ini menggunakan *Arduino UNO* sebagai kontrolernya dan sensor *rotary encoder* sebagai sensor kecepataannya. Prinsip kerja *crusher* ini pada dasarnya menstabilkan kecepatan *crushing* terhadap berbagai macam massa apel yang diberikan. Saat nilai *setpoint* diberikan, maka kecepatan akan meningkat kemudian stabil dan sesuai dengan nilai tersebut. Massa yang diberikan berpengaruh terhadap waktu *crushing*. Setelah *crushed apple* bertekstur lembut maka selanjutnya adalah dicampur dengan air lalu proses *mixing* dijalankan. Setelah *crushed apple* tercampur rata dengan air kemudian dialirkan secara manual menuju *spinner*. *Spinner motor* dinyalakan dengan kecepatan maksimum guna memisahkan ampas apel dan sari apel. Proses ekstraksi sari apel selesai ditandai dengan berhasilnya diperoleh sari apel dari ampas yang mengering dan habisnya *crushed apple* yang tertampung pada *crusher*. Dari hasil pengujian *crusher motor*, didapatkan respon sistem yang baik dengan nilai $K_p = 0,405$ dan nilai $K_i = 5$, dengan beban maksimum massa apel adalah 2kg dan maksimum *setpoint* adalah 1.500 rpm.

Pradana Perwira Putranto adalah Mahasiswa D4 Teknik Elektronika Jurusan Teknik Elektro Politeknik Negeri Malang, email: dhanz.nyonkz@gmail.com.

Sungkono, ST., MT. adalah dosen Jurusan Teknik Elektro Politeknik Negeri Malang

Edi Sulistio Budi, ST., MT. adalah dosen Jurusan Teknik Elektro Politeknik Negeri Malang.

Kata Kunci – *Arduino UNO*, *Crusher Motor*, *PI*,

Rotary Encoder, Sari Apel.

I. PENDAHULUAN

Minuman sari buah adalah minuman ringan yang dibuat dari campuran sari buah dengan air minum dengan atau tanpa penambahan gula dan bahan tambahan makanan yang diizinkan. Sari buah pada umumnya dibuat dengan cara menghancurkan daging buah kemudian ditekan untuk memperoleh sarinya [2].

Pada tahap awal pembuatan minuman sari apel perlu adanya mekanisme untuk mengekstrak dari buah apel menjadi sari apel, untuk itu diperlukan adanya suatu alat yang mampu mengekstrak buah apel yang dapat bekerja secara stabil dan terus-menerus.

Di Indonesia, sistem otomatisasi pengolahan sari buah skala *home industry* masih sedikit. Dengan berkembangnya zaman dan kemajuan teknologi-teknologi sekarang, penulis berupaya untuk membuat alat ekstraksi buah apel yang dapat diterapkan pada skala *home industry*. Mengingat pentingnya pengkondisian proses ekstraksi pada keadaan yang stabil maka perlu dibuat *miniplant* untuk mengatur kecepatan *crusher motor* dalam proses ekstraksi buah apel menggunakan metode PI (*Proportional Integral*) dengan parameter-parameter penyusunnya yaitu K_p (*Proportional Constant*) dan K_i (*Integral Constant*). Elemen kontrol PI secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem dan menghilangkan *offset* [7].

II. TINJAUAN PUSTAKA

2.1 *Arduino UNO*

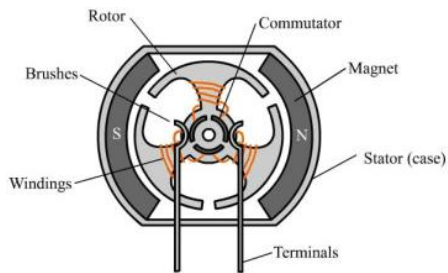
Arduino UNO adalah *board* berbasis mikrokontroler pada ATmega328. *Board* ini memiliki 14 digital input / output pin (dimana 6 pin dapat digunakan sebagai output PWM), 6 input analog, 16 MHz osilator kristal, koneksi USB, jack listrik tombol reset. Pin-pin ini berisi semua yang diperlukan untuk mendukung mikrokontroler, hanya terhubung ke komputer dengan kabel USB atau sumber tegangan bisa didapat dari adaptor AC-DC atau baterai untuk menggunakannya [8].



Gambar 2.1. Arduino UNO [5]

2.2 Motor DC (*Direct Current*)

Motor DC adalah motor yang digerakkan oleh energi listrik arus searah [5]. Konstruksi motor DC dapat dilihat pada gambar 2.2 di bawah ini.



Gambar 2.2. Konstruksi Motor DC [1]

Terminal motor terdiri atas 2 bagian yaitu terminal positif (+) dan terminal negatif (-) yang berfungsi sebagai penghubung sumber energi listrik searah dengan motor DC. Stator merupakan badan motor yang didalamnya terdapat magnet kutub utara dan magnet kutub selatan. *Brushes* berfungsi menyalurkan arus dari sumber energi listrik menuju kumparan (*windings*) rotor yang selanjutnya rotor dan komutator berputar. Komutator berputar karena komutator sendiri adalah bagian dari rotor yang terhubung pada *brushes*.

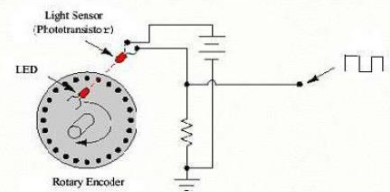
Ketika terminal motor DC diberi energi listrik searah dengan polaritas yang sama, maka motor akan berputar searah jarum jam, dan jika polaritas energi listrik dipasang terbalik pada terminal motor maka putarannya akan terbalik atau berlawanan arah jarum jam.

2.3 Rotary Encoder

Rotary encoder adalah device elektromagnetik yang dapat memonitor gerakan dan posisi [4]. *Rotary encoder* umumnya menggunakan sensor optik untuk menghasilkan serial pulsa yang dapat diartikan menjadi gerakan, posisi, dan arah.

Rotary encoder tersusun dari suatu piringan tipis yang memiliki lubang-lubang pada bagian lingkaran piringan. LED ditempatkan pada salah satu sisi piringan sehingga cahaya akan menuju ke piringan. Di sisi yang lain suatu *phototransistor* diletakkan sehingga *phototransistor* ini dapat mendeteksi cahaya dari LED yang berseberangan. Apabila posisi piringan mengakibatkan cahaya dari LED dapat mencapai *phototransistor* melalui lubang-lubang yang ada, maka *phototransistor* akan

mengalami saturasi dan akan menghasilkan suatu pulsa gelombang kotak.



Gambar 2.3. Prinsip Kerja *Rotary Encoder* [6]

2.4 LCD (*Liquid Crystal Display*)

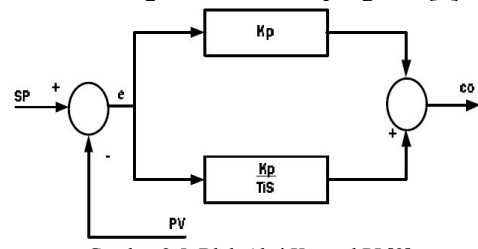


Gambar 2.4. LCD 16x2 [8]

LCD (*Liquid Crystal Display*) adalah suatu jenis media tampil yang menggunakan kristal cair sebagai penampil utama [5]. Sumber cahaya di dalam sebuah perangkat LCD adalah lampu neon berwarna putih di bagian belakang susunan kristal cair. LCD 16x2 menandakan bahwa LCD dapat menampilkan sebanyak 16 kolom dan 2 baris.

2.5 Metode PI (*Proportional Integral*)

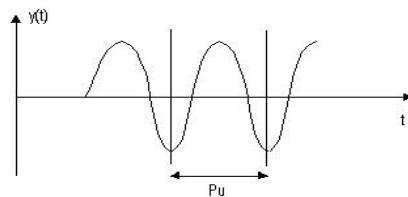
PI (*Proportional integral*) adalah sebuah logika kontroler yang terdiri atas 2 komponen penyusun yaitu P (*Proportional*) dan I (*Integral*), dimana gabungan aksi kontrol ini memiliki keunggulan dibandingkan dengan masing-masing kontrol yang menyusunnya. Keunggulan utamanya adalah diperolehnya keuntungan dari masing-masing aksi kontrol dan kekurangan dari aksi kontrol yang satu dapat diatasi dengan aksi kontrol yang lain [9].



Gambar 2.5. Blok Aksi Kontrol PI [9]

2.5.1 Metode Osilasi PI *Ziegler-Nichols*

Ziegler-Nichols menemukan metode ini tahun 1942. Metode ini digunakan sebagai dasar pada rangkaian dengan *loop* tertutup (*closed loop*). Dan untuk menjalankan metode ini, parameter integral di-*set* tak berhingga (~) dan parameter derivatif di-*set* nol (0). Kemudian parameter proporsional di-*set* bertahap mulai dari nol (0) hingga mencapai suatu nilai yang membuat reaksi sistem tersebut berosilasi. Reaksi sistem harus berosilasi dengan *magnitude* tetap (*sustained oscillation*) [3].



Gambar 2.6. Kurva Respon Sustained Oscillation

Nilai penguatan proporsional pada sistem mencapai *sustained oscillation* disebut *ultimate gain* (K_u), dan periode dari *sustained oscillation* disebut *ultimate period* (P_u).

Setelah nilai K_u dan P_u ditemukan, maka untuk *setting* parameter tertampil pada tabel 2.1 di bawah ini, dengan asumsi K_{cr} sebagai K_u dan P_{cr} sebagai P_u .

Tabel 2.1. *Setting* Parameter Kontrol dengan Metode Osilasi

Controller	K_p	T_i	T_d
P	$0,5.K_{cr}$	~	0
PI	$0,45.K_{cr}$	$P_{cr}/1,2$	0
PID	$0,6.K_{cr}$	$P_{cr}/2$	$0,125 P_{cr}$

III. METODE PENELITIAN

3.1. Spesifikasi Alat :

- a. Dimensi Alat
 Panjang = 100 cm
 Lebar = 60 cm
 Tinggi = 210 cm
- b. Bahan
 Crusher dan Spinner = Stainless Steel
 Rangka Alat = Besi
- c. Warna
 Crusher dan Spinner = Metallic Silver
 Rangka Alat = Biru
- d. Kecepatan Putar
 Crusher (terkontrol) = maksimal 1500 rpm
 Spinner (max speed) = 1400 rpm
- e. Penggunaan Listrik
 Crusher = 180 VDC (max rpm)
 6 Ampere (max torque)
 Spinner = 220 VAC, 50 Hz

3.2. Prinsip Kerja Alat

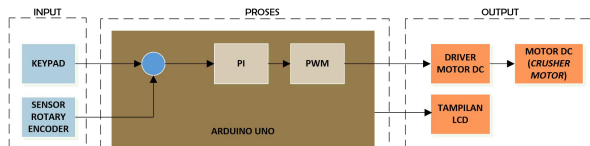
Prinsip kerja alat ini yaitu pertama kali memasukkan *setpoint* kecepatan *crusher motor* yang dikehendaki pada keypad (maksimal *setpoint* adalah 1500 dalam rpm). Setelah itu proses *crushing* buah apel dimulai dengan berputarnya *crusher motor* untuk menghancurkan. Sebelum apel dimasukkan ke proses *crushing*, apel dipotong menjadi 4 bagian sama besar, dan ditampung di wadah penampung apel pada penutup *crusher* apel. Penutup *crusher* dilengkapi dengan *hopper*, untuk membuka atau menutup *inlet crusher*. Setelah apel dihancurkan,

sensor *rotary encoder* yang terpasang pada *crusher motor* men-sensing kecepatan *crusher motor*, ketika beban (apel) kecil atau sedikit maka putaran akan semakin cepat, dan disini *rotary encoder* men-sensing kenaikan kecepatan *crusher motor* lalu akan disalurkan menuju *Arduino UNO*, setelah itu *Arduino UNO* melakukan pengaturan kecepatan otomatis melalui driver motor DC menuju *crusher motor* (untuk mengurangi kecepatannya) agar sesuai dengan *setpoint* kecepatan yang diberikan. Namun pengurangan kecepatan yang dilakukan oleh *Arduino UNO* membuat kecepatan *crusher motor* turun di bawah *setpoint*, saat ini pula sensor *rotary encoder* men-sensing pula penurunan kecepatan *crusher motor*, lalu disalurkan lagi menuju *Arduino UNO*, kemudian *Arduino UNO* melakukan pengaturan kecepatan otomatis melalui driver motor DC menuju *crusher motor* (untuk meningkatkan kecepatannya) agar sesuai dengan *setpoint* kecepatan yang diberikan, dan sistem berjalan secara kontinyu seperti itu setiap saat. Sistem otomatis tersebut juga berlaku saat terjadi beban (apel) besar atau banyak. Sensor *rotary encoder* men-sensing penurunan kecepatan *crusher motor* lalu akan disalurkan menuju *Arduino UNO*, setelah itu *Arduino UNO* melakukan pengaturan kecepatan otomatis melalui driver motor DC menuju *crusher motor* (untuk meningkatkan kecepatannya) agar sesuai dengan *setpoint* kecepatan yang diberikan. Namun peningkatan kecepatan yang dilakukan oleh *Arduino UNO* membuat kecepatan *crusher motor* naik di atas *setpoint*, saat ini pula sensor *rotary encoder* men-sensing pula kenaikan kecepatan *crusher motor*, lalu disalurkan lagi menuju *Arduino UNO*, kemudian *Arduino UNO* melakukan pengaturan kecepatan otomatis melalui driver motor DC menuju *crusher motor* (untuk menurunkan kecepatannya) agar sesuai dengan *setpoint* kecepatan yang diberikan. Kecepatan putar proses *crushing* apel tertampil pada LCD pada panel box alat.

Setelah apel hancur merata (dengan waktu tertentu tergantung *setpoint* kecepatan atau tergantung pada beban apel yang dimasukkan), dicampur dengan air sesuai dengan perbandingan 1:1 (disini setiap 1kg apel yg dihancurkan diberi 1 liter air minum) lalu *crusher* dinyalakan lagi selama 30 detik dengan *setpoint* yang sama guna mengencerkan atau mencampurkan *crushed apple* dengan air minum. Air minum dicampurkan pada akhir proses setelah *crushing* apel, karena jika dicampurkan pada awal proses nantinya akan mempengaruhi pengaturan kecepatan *crusher motor*, dan proses *mixing* ini tidak termasuk dalam proses pengaturan kecepatan.

Setelah *crushed apple* (dengan bermacam-macam massa) tercampur rata dengan air minum, disalurkan secara manual ke dalam *spinner* yang memiliki kecepatan putar 1400 rpm melalui *manual valve*. Jumlah maksimum yang masuk ke tabung

separator spinner harus 1/3 bagian. Jika lebih, yang dikhawatirkan adalah nanti sari apel (bahkan beserta ampas apel) akan terciprat atau tersebar keluar *spinner*. Kemudian saklar ON pada *spinner* dinyalakan. Dengan gaya sentrifugal, *separator spinner* akan memisahkan ampas buah apel dengan sari apel, yang kemudian sari apel akan jatuh pada wadah penampung di dalam *spinner* yang selanjutnya mengalir menuju pasteurisasi minuman sari apel.



Gambar 3.1. Diagram Blok Sistem *Crushing* Apel

3.3 Perancangan dan Pembuatan Mekanik



Gambar 3.2 Apple Extractor



Gambar 3.3 Tabung *Crusher*

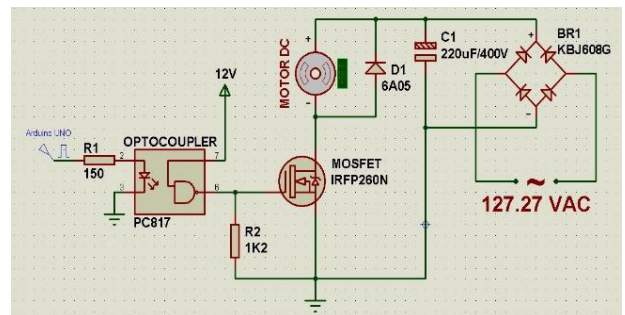


Gambar 3.4. Tabung *Spinner*

Apple Extractor terdiri dari *Crusher* dan *Spinner*. Volume *crusher* adalah 10 liter dan volume *spinner* adalah 20 liter.

3.4 Perancangan dan Pembuatan Elektrik

3.4.1 Rangkaian Driver Motor DC



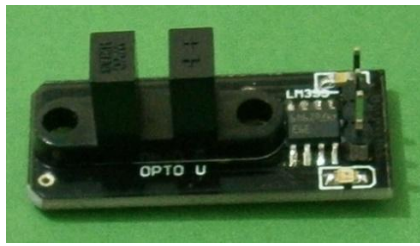
Gambar 3.5. Rangkaian Driver Motor DC

Untuk menggerakkan *crusher motor* diperlukan tegangan maksimum 180 VDC dan arus maksimum 6 Ampere. Disini 180 VDC diperoleh dari penyearahan tegangan AC sebesar 127.27 VAC dari *autotrafo* mode *stepdown* oleh kiprok KBJ608G.

$$\begin{aligned}
 VDC &= VAC * \sqrt{2} \\
 &= 127,27 * \sqrt{2} \\
 &= 179,99 \\
 VDC &= 180 V
 \end{aligned}$$

Pada rangkaian driver motor DC tersebut menggunakan PC817, yang berfungsi untuk memisahkan *Arduino UNO* sebagai rangkaian kontrol dengan MOSFET IRFP260N, supaya *Arduino UNO* tidak mengalami kerusakan.

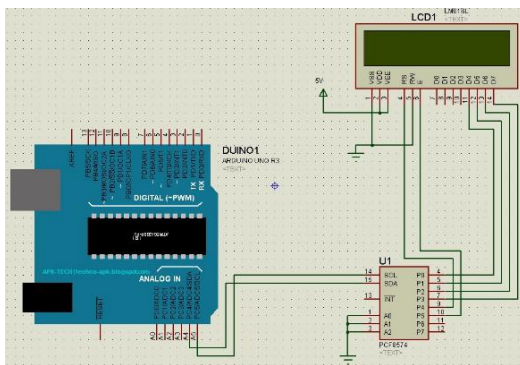
3.4.2 Rangkaian Modul Sensor *Rotary Encoder*



Gambar 3.6. Bentuk Fisik Modul Sensor Rotary Encoder

Sensor *rotary encoder* digunakan untuk *men-sensing* kecepatan *crusher motor* dengan mengeluarkan tegangan variabel yang nantinya disalurkan menuju *Arduino UNO*. Tegangan variabel keluaran *rotary encoder* dipengaruhi oleh kecepatan *crusher motor* berputar, maka semakin cepat *crusher motor* berputar, maka semakin besar tegangan yang dikeluarkan menuju *Arduino UNO*.

3.4.3 Rangkaian LCD-I2C



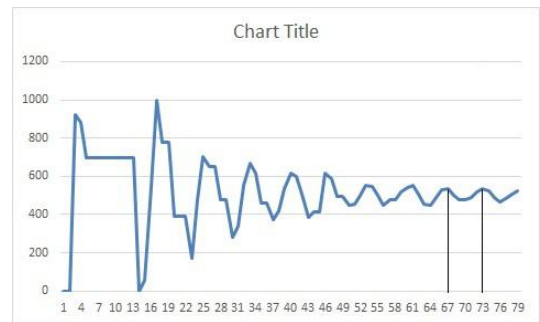
Gambar 3.7. Rangkaian LCD-I2C ke *Arduino UNO*

Fungsi rangkaian modul I2C (*Inter Integrated Circuit*) disini adalah menghemat penggunaan pin pada *Arduino UNO*. 16 pin pada LCD dihubungkan dengan 16 pin I2C selanjutnya *output pin* dari I2C yang terhubung pada pin *Arduino UNO* hanya berjumlah 4 buah. Berikut tabel wiring LCD-I2C ke *Arduino UNO*.

Tabel 3.1. Penjelasan *Wiring* LCD-I2C ke *Arduino UNO*

Pin I2C	Pin <i>Arduino UNO</i>
GND	GND
VCC	+5V
SDA (Serial Data)	A4 (Analog Input 4)
SCL (Serial Clock)	A5 (Analog input 5)

3.5 Perancangan dan Pembuatan *Software*



Gambar 3.8. Respon Osilasi *Crusher Motor*

Gambar 3.8 diatas merupakan hasil grafik dengan pemberian konstanta K_p dan K_i masing-masing 0.9 dan 0 serta *setpoint* kecepatan diberi 500 rpm tanpa diberi beban. Dimana konstanta K_p adalah nantinya berubah menjadi K_u apabila telah ditemukan respon sistem berosilasi.

Dari gambar di atas dapat ditemukan osilasi yang sama yaitu pada titik ke-67 dan ke-73, maka P_u dapat diperoleh dengan cara:

$$\begin{aligned}
 P_u &= t_2 - t_1 \\
 &= 73 - 67 \\
 P_u &= 6
 \end{aligned}$$

Dengan diperolehnya nilai K_u dan P_u , nilai K_p dan K_i dapat diperoleh melalui tabel *Setting* Parameter Kontrol dengan Metode Osilasi pada bab sebelumnya, dan diperoleh data:

$$\begin{aligned}
 K_p &= 0,45 \times K_u \\
 &= 0,45 \times 0,9 \\
 K_p &= 0,405
 \end{aligned}$$

$$\begin{aligned}
 K_i &= P_u / 1,2 \\
 &= 6 / 1,2 \\
 K_i &= 5
 \end{aligned}$$

IV. HASIL DAN PEMBAHASAN

4.1 Pengujian Rangkaian Driver Motor DC

Pengujian rangkaian driver motor DC ini dilakukan dengan mengubah-ubah nilai PWM dari *Arduino UNO* dari 0 hingga 255, dan motor DC diberi tegangan 180 VDC. Berikut akan ditampilkan perubahan tegangan pada motor DC berdasarkan perubahan nilai PWM dari *Arduino UNO*.

Tabel 4.1. Tabel Hasil Pengujian Driver Motor DC

PWM	Tegangan Motor (VDC)
0	0

25	27,47
50	41,26
75	52,54
100	70,98
125	84,29
150	95,11
175	105,89
200	120,05
225	126,27
255	132,68

4.2 Pengujian Sensor *Rotary Encoder*

Pengujian sensor *rotary encoder* dilakukan dengan cara menemukan *error* sensor, diperoleh dari perbandingan hasil *sensing* sensor terhadap kecepatan *crusher motor* yang tertampil pada LCD 16x2 dengan alat ukur kecepatan tachometer. Persentase *error* dapat ditemukan dengan rumus:

$$\% \text{ Error} = \frac{\text{rpm sensor} - \text{rpm tachometer}}{\text{rpm tachometer}} \times 100\%$$

Tabel 4.2. Tabel Hasil Pengujian Sensor *Rotary Encoder*

PWM	Sensor(rpm)	Tachometer (rpm)	Error (%)
0	0	0	0
25	397	399	0,5012
50	697	704	0,9943
75	1.016	1.002	1,3972
100	1.333	1.303	2,3024
125	1.578	1.575	0,1905
150	1.875	1.847	1,5159
175	2.142	2.065	3,7288
200	2.222	2.300	3,3913
225	2.500	2.450	2,0408
255	2.857	2.862	0,1747

4.3 Pengujian Rangkaian LCD 16x2

Pengujian rangkaian LCD dilakukan dengan menghubungkan pin-pin pada LCD ke I2C, kemudia 4 pin keluaran I2C dihubungkan ke *Arduino UNO*. Berikut adalah tampilan karakter pada LCD yang sudah terprogram. Program pada *Arduino UNO* menampilkan kalimat “DHANZ 2016”.

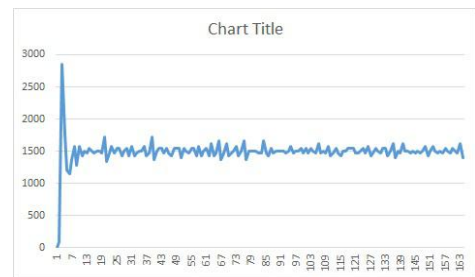


Gambar 4.1. Pengujian LCD 16x2

4.4 Pengujian Metode Osilasi *PI Ziegler-Nichols* pada *Crusher Motor*

Pengujian metode osilasi *PI Ziegler-Nichols* pada *crusher motor* dilakukan dengan memasukkan nilai K_p dan K_i yang telah diperoleh pada perancangan *software* di bab sebelumnya, yaitu $K_p = 0,405$ dan $K_i = 5$, serta *setpoint* 1500 rpm tanpa beban, berikut perbandingan respon yang dihasilkan dengan kontrol dengan tanpa kontrol.

1. $K_p=0,405$, $K_i=5$, 1500 rpm tanpa beban

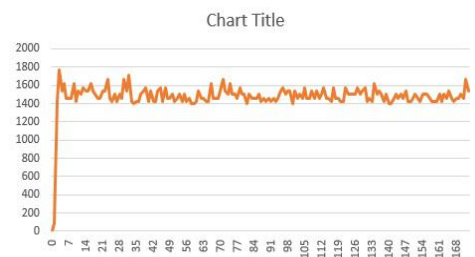


Gambar 4.2. Grafik Respon Dengan Kontrol Tanpa Beban

Diperoleh data:

- Rise time* (tr) = 145,67 ms
- Percent Overshoot* (PO) = 90,4667 %
- Settling time* (ts) = 1000 ms
- Error steady state* (ess) = 0 %

2. Tanpa kontrol, 1500 rpm tanpa beban



Gambar 4.3. Grafik Respon Tanpa Kontrol Tanpa Beban

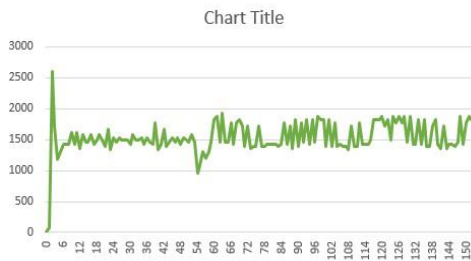
Diperoleh data:

- Rise time* (tr) = 194,21 ms
- Percent Overshoot* (PO) = 17,60 %
- Settling time* (ts) = -
- Error steady state* (ess) = -

Dari 2 grafik di atas, dapat diketahui bahwa sistem dengan kontrol dan tanpa kontrol pada saat tanpa beban memiliki keunggulan dan kelemahan masing-masing. Dari faktor *rise time* lebih baik dengan kontrol karena lebih cepat. Tapi pada faktor *percent overshoot*, lebih baik tanpa kontrol karena respon sistem tidak mengalami *overshoot* yang terlalu tinggi.

Parameter *settling time* dan *error steady state* pada sistem tanpa kontrol tidak dapat diidentifikasi karena sistem tidak dikontrol (menggunakan kecepatan maksimum 1500 rpm).

3. $K_p=0,405$, $K_i=5$, 1500 rpm, beban apel 1kg

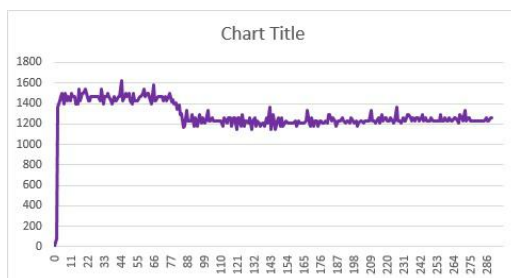


Gambar 4.4. Grafik Respon Dengan Kontrol Dengan Beban Apel Anna 1kg

Diperoleh data: (setelah pembebanan)
Rise time (tr) = -
Percent Overshoot (PO) = -
Settling time (ts) = 700 ms
Error steady state (ess) = 6,19 %

Dari hasil respon sistem dengan pembebanan berupa apel 1kg dengan kontrol, dapat diketahui bahwa sistem dapat kembali menuju *steady state* dalam waktu 700 ms, dengan *errorsteady state* antara sebelum dan setelah pembebanan sebesar 6,19%. Tapi setelah pembebanan, tidak diketahui secara pasti faktor *rise time* dan *percent overshoot* karena pembebanan dilakukan setelah sistem stabil saat tanpa beban.

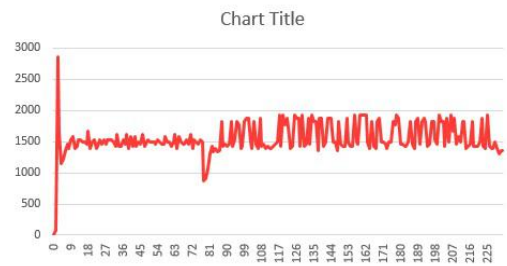
4. Tanpa kontrol, 1500 rpm, beban apel 1kg



Gambar 4.5. Grafik Respon Tanpa Kontrol Dengan Beban Apel Anna 1kg

Dari hasil respon sistem dengan pembebanan berupa apel 1kg tanpa kontrol, dapat diketahui bahwa kecepatan *crusher motor* menurun dan tidak dapat kembali menuju *setpoint* yang diinginkan, maka didapatkan suatu analisis bahwa sistem lebih baik menggunakan kontrol agar sistem kembali menuju *setpoint* setelah dilakukan pembebanan.

5. $K_p=0,405$, $K_i=5$, 1500 rpm, beban apel 2kg

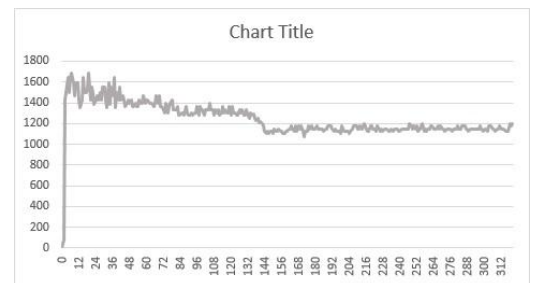


Gambar 4.6. Grafik Respon Dengan Kontrol Dengan Beban Apel Anna 2kg

Diperoleh data: (setelah pembebanan)
Rise time (tr) = -
Percent Overshoot (PO) = -
Settling time (ts) = 1000 ms
Error steady state (ess) = 7,5 %

Dari hasil respon sistem dengan pembebanan berupa apel 2kg dengan kontrol, dapat diketahui bahwa sistem dapat kembali menuju *steady state* dalam waktu 1000 ms, dengan *errorsteady state* antara sebelum dan setelah pembebanan sebesar 7,5%. Tapi setelah pembebanan, tidak diketahui secara pasti faktor *rise time* dan *percent overshoot* karena pembebanan dilakukan setelah sistem stabil saat tanpa beban.

6. Tanpa kontrol, 1500 rpm, beban apel 2kg



Gambar 4.7. Grafik Respon Tanpa Kontrol Dengan Beban Apel Anna 2kg

Dari hasil respon sistem dengan pembebanan berupa apel 2kg tanpa kontrol, dapat diketahui bahwa kecepatan *crusher motor* menurun dan tidak dapat kembali menuju *setpoint* yang diinginkan, maka didapatkan suatu analisis bahwa sistem lebih baik menggunakan kontrol agar sistem kembali menuju *setpoint* setelah dilakukan pembebanan.

Tabel 4.3. Tabel Hasil Percobaan *Crushing* Apel Dengan Metode PI

Tabel di atas merupakan hasil dari percobaan *crushing* apel. Dengan jumlah apel, *setpoint*, dan waktu yang berbeda dihasilkan *crushed apple* dengan kelembutan tekstur yang sama.



Gambar 4.8. *Crushed Apple* di Dalam *Crusher*

Setelah *crushed apple* dibuat, dicampur terlebih dahulu dengan air minum kemudian dilakukan proses *mixing* selama 30 detik, kemudian disalurkan menuju *spinner*.



Gambar 4.9. Campuran *Crushed Apple* dengan Air Minum disalurkan menuju *Spinner*

Setelah *spinner* terisi 1/3 bagian, selanjutnya adalah melakukan proses *spinning*, dan sari apel dihasilkan. Proses ini dilakukan berulang-ulang secara manual hingga ampas apel yang mongering tertinggal pada tabung *separator spinner* dan *crushed apple* dalam *crusher* habis.



Gambar 4.10. Proses *Spinning*

Jumlah Apel (kg)	Setpoint (rpm)	Waktu (Terkontrol)	Waktu (Tanpa Kontrol)
1	500	8 menit 6 detik	10 menit 26 detik
	1000	5 menit 19 detik	7 menit 24 detik
	1500	3 menit 47 detik	6 menit 2 detik
2	500	10 menit 4 detik	13 menit 31 detik
	1000	7 menit 31 detik	10 menit 45 detik
	1500	5 menit 50 detik	9 menit 9 detik



Gambar 4.11. Sari Apel Diperoleh dari *Spinner*



Gambar 4.12. Ampas Apel Tertinggal

V. PENUTUP

5.1 Kesimpulan

1. Dengan mengimplementasikan metode PI (*Proportional Integral*) dengan parameter $K_p = 0,405$ dan $K_i = 5$ pada *crusher motor*, metode PI mampu mempercepat sistem menuju *setpoint* yang ditentukan. Diperoleh *rise time* (t_r) 48,54 ms lebih cepat dibandingkan dengan tanpa kontrol saat tanpa ada beban, tetapi mengakibatkan peningkatan *overshoot* sistem. *Percent Overshoot* (PO) yang ditimbulkan pada *starting* awal tanpa beban sebesar 90,4667 %.
2. Jumlah beban apel anna yang diproses pada *crusher motor* dengan metode PI mempengaruhi *error steady state* dan *settling time* sistem. Kenaikan apel anna sebesar 1kg mengakibatkan kenaikan *error steady state* (ess) sistem sebesar 1,21 % dan memakan waktu 300 ms lebih

lama untuk *settling time* (ts) sistem tersebut.

3. Pada proses *crushing* apel, jumlah beban dan *setpoint* yang diberikan sangat berpengaruh terhadap waktu *crushing*. Pada *setpoint* yang sama, jumlah beban berbanding lurus dengan waktu *crushing* dan pada beban yang sama, *setpoint* yang diberikan berbanding terbalik dengan waktu *crushing*-nya. Serta diperoleh waktu *crushing* apel lebih cepat menggunakan metode PI dibandingkan tanpa kontrol, dengan selisih waktu antara 2 hingga 3 menit lebih cepat tergantung beban apel.

5.2 Saran

1. Dari segi mekanik, sebaiknya perlu dipertimbangkan lagi segi ergonomi *crusher* pada *apple extractor* ini agar dapat dijangkau oleh khalayak umum, seperti menambahkan tangga untuk memasukkan apel ke dalam *crusher*. Dan juga perlu adanya penutup pada *spinner* agar sari apel dan ampas apel tidak terciprat keluar *spinner* saat proses *spinning* dilakukan.
2. Dari segi elektronik, perlu adanya pengaturan kecepatan secara otomatis pada *spinner motor* terhadap beban *crushed apple* yang akan diputar.
3. Dari segi *software*, perlu adanya metode kontrol lain guna sebagai perbandingan apakah metode baru yang digunakan lebih baik atau kurang baik dibandingkan metode PI (*Proportional Integral*) yang digunakan saat ini.

DAFTAR PUSTAKA

- [1]Andrianto, M. (2014). "*Kendali Kecepatan Motor Direct Current (DC) Menggunakan Proportional Integral Derivative (PID) Controller Terhadap Beban*". Jurusan Teknik Elektro, Politeknik Negeri Malang.
- [2]Azlina, Maya. (2013). "*Pembuatan Alat Ukur Kecepatan Angin Dan Penunjuk Arah Angin Berbasis Mikrokontroler AT-MEGA 8535*". Saintia Fisika.
- [3]Gunterus, Frans. (1994). *Falsafah Dasar: Sistem Pengendalian Proses*. Elex Media Komputindo: Jakarta.
- [4]Lukman, Hakim. (2012). "*Implementasi Perhitungan Posisi Robot Dengan FPGA Menggunakan Rotary Encoder*". Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Noverber.
- [5]Meisyaroh, Ika Dewi. (2014). "*Pengontrolan Suhu Steamer Kerupuk Bawang Menggunakan Logika Fuzzy Mamdani*". Politeknik Negeri Malang.
- [6] Sa'adah, Lailufary Ichda Noor., Estiasih, Teti. (2015). "*Karakterisasi Minuman Sari Apel Produksi Skala Mikro dan Kecil di Kota Batu: Kajian Pustaka*". Universitas Brawijaya Malang.
- [7]Sadli, M. (2014). "*Disain Kontroler PI dengan Decoupling Pada Sistem Kendali Level Coupled Tank*". Jurusan Teknik Elektro, Fakultas Teknik, Universitas Malikussaleh, Lhokseumawe, Aceh Utara.
- [8]Utari, Diah M. (2010). "*Kandungan Asam lemak, Zink, Dan Copper Pada Tempe, Bagaimana Potensinya Untuk Mencegah Penyakit Degeneratif?*". Departemen Gizi Kesmas Fakultas Kesehatan Masyarakat, Universitas Indonesia
- [9]Wahyudi. (2008). "*Tuning Parameter Kontrol Proportional Integral Menggunakan Sugeno Fuzzy Inference System*". Transmisi, Jurusan Teknik Elektro, Jilid 10 Nomor 2, hlm 97-102.