

Pengendalian Kadar Keasaman (pH) Pada Pengendapan Tahu Menggunakan Kontroler PID Berbasis ATmega328

Dyah Ayu Anggreini T, Retnowati, Rahmadwati.

Abstrak— *Pengendalian kadar keasaman pada pengendapan tahu sangat diperlukan karena kadar keasaman air pengendapan pada umumnya cenderung basa, keadaan ini tidak memenuhi syarat kadar keasaman ideal tahu yaitu pH 5. Pada skripsi ini Arduino UNO yaitu sebuah board mikrokontroler yang didasarkan pada ATmega328 diaplikasikan sebagai perangkat pengendali kadar keasaman air pengendapan.*

Pengendalian dirancang agar kadar keasaman air pengendapan tahu sesuai dengan setpoint yaitu pH 5 dengan mengendalikan putaran pompa yang berisi cairan asam dan air. Hasil identifikasi menggunakan MATLAB 7.7 didapatkan fungsi alih sistem numerator sebesar $1.554 s + 0.07376$ dan denominator sebesar $s^2 + 1.688 s + 0.06762$. Proses perancangan kontroler PID menggunakan metode root locus dan didapatkan bahwa semua akar berada disebelah kiri bidang s , sehingga respon yang didapat dari semua pole stabil. Hasil perhitungan parameter PID dengan pole $s = -5.1109$ didapatkan nilai parameter PID terbaik yaitu $K_p = 3.9133$, $K_i = 10$ dan $K_d = 0.3828$.

Kata Kunci—pH, tahu, PID, Arduino UNO.

• PENDAHULUAN

Tahu merupakan salah satu makanan pokok masyarakat Indonesia yang berbahan dasar kedelai. Pengendapan merupakan salah satu proses dalam pembuatan tahu. Namun pengendapan tahu secara konvensional tidak dilengkapi pengendalian kadar keasaman yang dapat menjaga pH ideal pengendapan tahu yaitu pH 5[2].

Salah satu jenis kontroler yang banyak digunakan saat ini adalah kontroler PID (*Proportional Integral Differential*). PID adalah kontroler yang merupakan gabungan dari kontroler proposional, kontroler integral dan kontroler differensial. Gabungan dari ketiga kontroler ini diharapkan agar sistem menghasilkan respon dengan nilai error sekecil mungkin. Keuntungan menggunakan kontroler PID adalah cepat dalam mengambil sebuah keputusan. Diharapkan dengan menggunakan kontroler PID kadar keasaman dapat dikendalikan.

Berdasarkan permasalahan tersebut maka dalam skripsi ini dirancang sebuah perangkat yang mampu mengendalikan kadar keasaman pada sistem pengendapan tahu dengan cara mengendalikan kecepatan putaran pompa yang berisi cairan asam dan

air. Komponen pengendalian yang digunakan berbasis Arduino Uno.

Tujuan yang ingin dicapai dalam penelitian ini adalah terbuatnya perangkat yang mampu mengendalikan kadar keasaman pada sistem pengendapan tahu berbasis Arduino Uno.

• IDENTIFIKASI SISTEM

A. Pengendapan Tahu

Proses pengendapan merupakan proses pencampuran antara pati soy-bean panas (hasil proses *sifting*) dengan air asam cuka sintetik dingin, diaduk secara merata dan dapat menghasilkan gumpalan-gumpalan putih. Di proses inilah pengendapan tahu terjadi. Pengendapan tahu ini menggunakan cairan asam cuka yang telah disiapkan sebelumnya[1]. Suhu yang dibutuhkan pada pengendapan ini adalah $50^{\circ} C - 60^{\circ} C$. [2]. Air cuka yang telah dipilih dan dibuat harus ditempatkan pada air yang bersih dari segala macam kotoran dan sebaiknya tidak terkena sedikitpun garam. Rentang pH ideal yang dibutuhkan adalah 3,8-5,0 agar kualitas tahu yang dibuat maksimal. Kadar pH yang tidak sesuai menyebabkan rendahnya kadar protein yang dapat mengakibatkan rasa yang kurang disukai dan aroma yang tidak khas [3].

B. Motor DC

Motor yang digunakan dalam penelitian ini merupakan motor DC magnet permanen. Untuk melakukan perubahan kecepatan dari motor ini adalah dengan mengatur tegangan yang masuk ke dalam motor



Gambar 1. Motor DC

C. Sensor pH

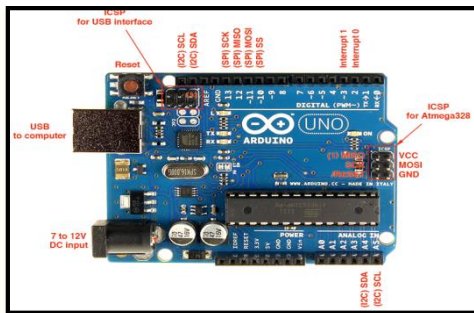
Sensor pH merupakan elektroda gelas yang terdiri dari gelembung gelas yang sensitif pH pada ujungnya, berisi larutan klorida yang diketahui pHnya dan elektroda referensi.



Gambar 2. Sensor pH

D. Arduino UNO

Arduino UNO adalah sebuah board mikrokontroler yang didasarkan pada ATmega328. Arduino UNO mempunyai 14 pin digital input dan output, 6 di antaranya dapat digunakan sebagai output PWM. 6 input analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah power jack, sebuah ICSP header, dan sebuah tombol reset. Arduino UNO memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkannya ke sebuah komputer dengan sebuah kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk memulainya.



Gambar 3. Arduino UNO

E. Pengambilan Data Input-Output

Pengujian ini tentu memerlukan sinyal uji tertentu yang akan diberikan kepada sistem fisik yang akan diidentifikasi. Agar diperoleh model yang tepat maka dalam pemilihan sinyal uji ini tidak boleh sembarangan. Syarat pemilihannya adalah suatu sinyal uji harus memiliki cakupan frekuensi yang lebar dan standard yang digunakan adalah sinyal *Pseudo Random Binary Sequences* (PRBS). [4].

Pseudo Random Binary Sequence (PRBS) adalah sinyal kotak yang termodulasi pada lebarnya dan berlangsung secara sekuensial. Sinyal ini biasanya dibangkitkan menggunakan *Linear Feedback Shift Register* (LFSR). Pada LFSR memiliki 2 parameter dasar yang menentukan sifat sekuensial yang dihasilkan, yaitu: panjang dari shift register dan susunan umpan balik. PRBS memiliki variasi panjang sekuensialnya, tergantung dari panjangnya shift register seperti ditunjukkan Tabel 1.

Panjang dari shift register menentukan periode maksimum yang dapat dihasilkan dari sekuensial PRBS dan tidak berulang yang dapat dinyatakan dengan persamaan:

$$LPRBS = 2n - 1 \quad (1)$$

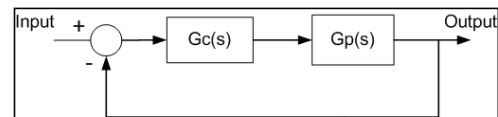
Dimana n adalah panjang dari register LFSR (jumlah bit). Panjang maksimum dari PRBS disebut *M-sequence*.

Tabel 1. Variasi Panjang Sekuensial PRBS

| Panjang Register (N) | Panjang Sekuensial L=2N-1 | Posisi Tap Umpan Balik |
|----------------------|---------------------------|------------------------|
| 2 | 3 | 1 dan 2 |
| 3 | 7 | 1 dan 3 |
| 4 | 15 | 3 dan 4 |
| 5 | 31 | 3 dan 5 |
| 6 | 63 | 5 dan 6 |
| 7 | 127 | 4 dan 7 |
| 8 | 255 | 2, 3, 4, dan 8 |
| 9 | 511 | 5 dan 9 |
| 10 | 1023 | 7 dan 10 |

F. Perancangan PID dengan Root Locus

Rancangan sistem kendali loop tertutup dengan metode *root locus* dimungkinkan untuk mengatur sekurang-kurangnya beberapa letak pole sistem loop tertutup sehingga dapat diatur tanggapan transient pada tingkat tertentu dan pengaruhnya terhadap tanggapan keadaan mantap. Prosedur analitis perancangan kontroler PID menggunakan metode *root locus* dapat dilihat dalam gambar berikut [5].



Gambar 4. Sistem Kendali

Untuk sistem tersebut, persamaan karakteristik diberikan oleh

$$1 + Gc(s)Gp(s) = 0 \quad (2)$$

Misalkan diinginkan lokus akar melalui $s = s_1$, maka $Gc(s_1)Gp(s_1) = -1 \quad (3)$

$$Gc(s_1)|Gp(s_1)|e^{j\psi} = 1e^{j\pi} \quad (4)$$

Fungsi alih kontroler PID setelah ditransformasi laplace dinyatakan oleh

$$Gc(s) = Kp + \frac{Ki}{s} + Kd.s \quad (5)$$

Sehingga dari persamaan diatas didapatkan

$$Gc(s_1) = \frac{1}{|Gp(s_1)|} e^{j(\pi-\psi)} \quad (6)$$

atau

$$Kd s_1^2 + Kp s_1 + Ki = \frac{e^{j(\pi-\psi)}}{|Gp(s_1)|} \quad (7)$$

Dengan

$$s_1 = |s_1|e^{j\beta} \quad (8)$$

Maka

$$Kd |s_1|^2 (\cos 2\beta + j \sin 2\beta) + Kp |s_1| (\cos \beta + j \sin \beta) + Ki = \frac{|s_1|}{|Gp(s_1)|} [\cos(\beta + \pi - \psi) + j \sin(\beta + \pi - \psi)] \quad (9)$$

Menyamakan real dengan real dan imajiner dengan imajiner, didapat

$$\begin{bmatrix} |s_1|^2 & |s_1| \cos \beta \\ |s_1|^2 & |s_1| \sin \beta \end{bmatrix} \begin{bmatrix} Kd \\ Kp \end{bmatrix} = \begin{bmatrix} \frac{|s_1|}{Gp(s_1)} \cos(\beta + \Pi + \psi) - Ki \\ \frac{|s_1|}{Gp(s_1)} \sin(\beta + \Pi + \psi) \end{bmatrix} \quad .. (10)$$

$$\begin{bmatrix} |s_1|^2 & |s_1| \cos \beta \\ |s_1|^2 & |s_1| \sin \beta \end{bmatrix} \begin{bmatrix} Kd \\ Kp \end{bmatrix} = \begin{bmatrix} -\frac{|s_1|}{Gp(s_1)} \cos(\psi - \beta) - Ki \\ \frac{|s_1|}{Gp(s_1)} \sin(\psi - \beta) \end{bmatrix} \quad \dots\dots(11)$$

Dari persamaan di atas dapat dilihat bahwa untuk perancangan kontroler PID, satu dari tiga penguatan K_p , K_i , K_d , harus ditentukan dahulu. Sedangkan untuk perancangan PI atau PD, penguatan yang sesuai pada persamaan di atas dibuat sama dengan nol.

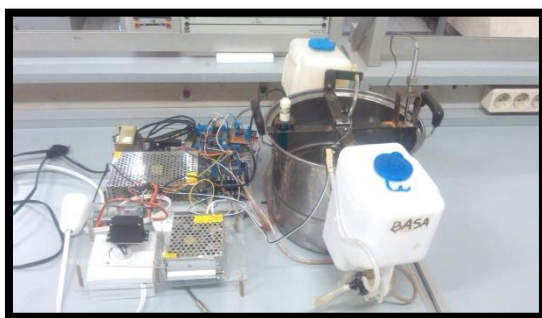
Untuk kasus s_1 adalah imajiner, persamaan di atas akan menghasilkan dua persamaan dalam K_p dan K_d serta besar K_i harus ditentukan terlebih dahulu.

• PERANCANGAN MODUL

Perancangan ini meliputi pembuatan perangkat keras dan perangkat lunak, perangkat keras meliputi perancangan alat pengendalian suhu dan perancangan rangkaian Arduino *shield* yang terdiri atas rangkaian pengondisi sinyal dan *regulator* tegangan, untuk perangkat lunak meliputi pembuatan program pada Arduino UNO 1.0.5 untuk keperluan analisis sistem yaitu dengan membangkitkan sinyal PRBS dan program kontrol PWM motor pompa dan motor servo *dimmer* kompor induksi.

A. Perancangan Alat Pengendalian Suhu

Konstruksi alat pengendalian suhu dapat dilihat pada gambar 5.



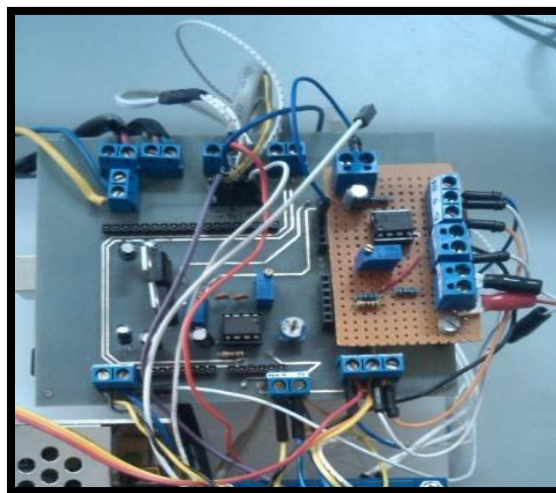
Gambar 5. Skema konstruksi Alat Pengendalian Suhu

B. Perancangan Arduino *Shield*

Shield adalah istilah untuk modul tambahan pada Arduino, karena cara menggunakan modul tambahan pada

Hasil pengujian menunjukkan bahwa RPS PT100 setelah dilakukan kalibrasi dapat membaca perubahan suhu dengan baik dengan error rata-rata 0,11.

Arduino adalah dengan cara menumpuk di bagian atas Arduino, maka diberi istilah *shield* (tameng/perisai). Arduino *shield* yang digunakan pada penelitian ini berisi rangkaian pengondisi sinyal dan *regulator* tegangan.



Gambar 6. Arduino *Shield*

C. Perancangan Perangkat Lunak

Pada penelitian ini pemrograman keseluruhan sistem menggunakan bahasa pemrograman C++ dengan *software* Arduino 1.0.5

• PENGUJIAN DAN ANALISIS DATA

Pengujian ini meliputi pengujian perangkat keras dalam hal ini pengujian sensor pH, pengujian sensor PT100, pengujian rangkaian pengondisi sinyal, pengujian driver dan hubungan PWM dengan kecepatan motor pompa dan motor servo, diteruskan dengan pengambilan data input-output dengan penggunaan sinyal PRBS, kemudian data tersebut dianalisa dengan MATLAB 7.7 dengan fasilitas ident yang tersedia. Pengujian di atas dilakukan dengan tujuan memperoleh karakteristik dari tiap alat. Setelah didapatkan karakteristiknya, kemudian dapat dilakukan penyesuaian pada sistem.

A. Pengujian Sensor pH

Pengujian ini bertujuan untuk mengetahui besar tegangan keluaran sensor pH. Pengujian dilakukan dengan larutan pH buffer 1-10.

Tabel 2. Hasil Nilai Tegangan Keluaran Sensor pH

| Larutan pH Buffer | Vout Sensor (mV) |
|-------------------|------------------|
| 1 | 134.96 |
| 2 | 119.33 |
| 4 | 105.69 |
| 5 | 101.26 |
| 6 | 37.44 |
| 7 | -10.90 |
| 8 | -37.85 |
| 9 | -45.51 |
| 10 | -127.47 |

Hasil pengujian menunjukkan bahwa sensor pH mampu mendeteksi perbedaan pH pada tiap larutan.

B. Pengujian Sensor PT100

Pengujian ini bertujuan untuk mengetahui kinerja rangkaian pengondisi sinyal sensor PT100 terhadap perubahan suhu dengan melihat hasil pembacaan suhu melalui serial monitor Arduino Uno.

Tabel 3. Hasil Pengujian Sensor PT100

| Suhu (°C) | Pengujian (Ω) | Perhitungan (Ω) | Error (%) |
|------------------|---------------|-----------------|--------------|
| 25 | 110,6 | 109,625 | 0,89% |
| 30 | 112,6 | 111,55 | 0,94% |
| 35 | 114,2 | 113,475 | 0,64% |
| 40 | 115,2 | 115,4 | 0,17% |
| 45 | 117,8 | 117,325 | 0,40% |
| 50 | 119,9 | 119,25 | 0,55% |
| 55 | 122 | 121,175 | 0,68% |
| 60 | 123,9 | 123,1 | 0,65% |
| 65 | 125,9 | 125,025 | 0,70% |
| 70 | 126,1 | 126,95 | 0,75% |
| 75 | 127,9 | 128,875 | 1,03% |
| 80 | 130,2 | 130,8 | 0,46% |
| 85 | 132,2 | 132,725 | 0,40% |
| 90 | 134,2 | 134,65 | 0,33% |
| Rata-Rata | | | 0,61% |

Hasil pengujian menunjukkan bahwa sensor PT100 memiliki kemampuan yang baik dalam melakukan pembacaan perubahan suhu.

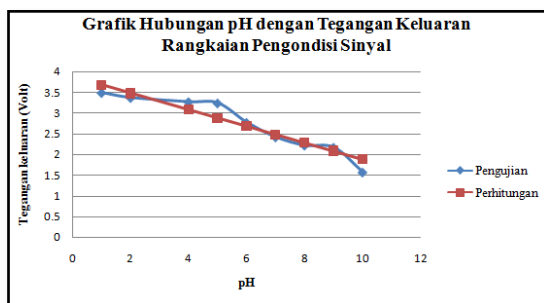
C. Pengujian Rangkaian Pengondisi Sinyal (RPS) Sensor pH

Pengujian ini bertujuan untuk memperoleh resolusi pH yang diinginkan bisa ditanggapi oleh ADC mikrokontroler yang digunakan yaitu pada kisaran 0 - 5 Volt.

Tabel 4. Hasil Nilai Tegangan Keluaran Sensor pH

| pH | Pengujian | Perhitungan |
|----|-----------|-------------|
| 1 | 3.49 | 3.68 |
| 2 | 3.37 | 3.48 |
| 4 | 3.27 | 3.08 |
| 5 | 2.24 | 2.88 |
| 6 | 2.77 | 2.68 |
| 7 | 2.42 | 2.48 |
| 8 | 2.22 | 2.28 |
| 9 | 2.17 | 2.08 |
| 10 | 1.57 | 1.88 |

Hasil pengujian RPS dapat dilihat pada gambar 7.



Gambar 7. Grafik Pengujian Rangkaian Pengondisi Sinyal

D. Pengujian Rangkaian Pengondisi Sinyal (RPS) PT100

Pengujian ini bertujuan untuk mengetahui kinerja rangkaian pengondisi sinyal sensor PT100 terhadap perubahan suhu dengan melihat hasil pembacaan suhu melalui Serial Monitor Arduino Uno.

Tabel 5. Hasil Pengujian RPS PT100

| Suhu Pembacaan Termometer (°C) | Suhu Pembacaan Serial Monitor (°C) | Error (°C) |
|--------------------------------|------------------------------------|-------------|
| 30.00 | 34.55 | 4.55 |
| 35.00 | 39.70 | 4.70 |
| 40.00 | 44.67 | 4.67 |
| 45.00 | 49.78 | 4.78 |
| 50.00 | 55.05 | 5.05 |
| 55.00 | 59.89 | 4.89 |
| 60.00 | 64.93 | 4.93 |
| 65.00 | 70.08 | 5.08 |
| 70.00 | 75.02 | 5.02 |
| 75.00 | 80.39 | 5.39 |
| 80.00 | 84.98 | 4.98 |
| Rata-rata | | 4.91 |

Karena error cukup besar, dilakukan kompensasi error dengan metode regresi linear. Sedingga didapatkan fungsi regresi linear pada persamaan 3.

$$f = -4.20889 + 0.988250178x \quad (3)$$

Hasil pengujian setelah kalibrasi dengan menggunakan persamaan 3 dapat dilihat pada tabel 3.

Tabel 6. Hasil Pengujian RPS PT100 setelah kalibrasi

| Suhu Pembacaan Termometer (°C) | Suhu Pembacaan Serial Monitor (°C) | Error (°C) |
|--------------------------------|------------------------------------|-------------|
| 30.00 | 29.79 | 0.21 |
| 35.00 | 34.425 | 0.58 |
| 40.00 | 39.755 | 0.25 |
| 45.00 | 44.72 | 0.28 |
| 50.00 | 49.55 | 0.45 |
| 55.00 | 54.34 | 0.66 |
| 60.00 | 59.37 | 0.63 |
| 65.00 | 64.385 | 0.62 |
| 70.00 | 69.45 | 0.55 |
| 75.00 | 74.55 | 0.45 |
| 80.00 | 79.43 | 0.57 |
| Rata-rata | | 0.48 |

Hasil pengujian menunjukkan bahwa RPS PT100 setelah dilakukan kalibrasi dapat membaca perubahan suhu dengan baik dengan error rata-rata 0,48.

E. Pengujian Driver Motor Pompa

Pengujian ini dilakukan untuk membandingkan nilai masukan *duty cycle* yang diberikan dengan respon *duty cycle* pada *driver* EMS H-Bridge 30 A.

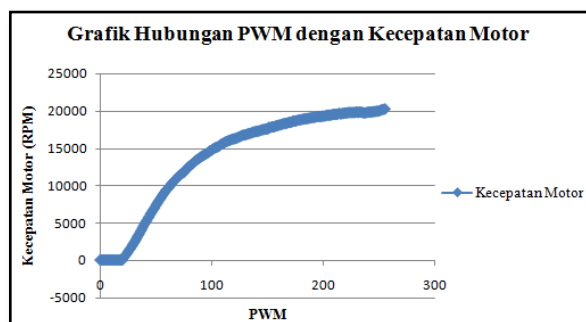
Tabel 7. Hasil Nilai Pengujian Driver EMS H-Bridge 30 A

| Duty Cycle Input (PWM MK) | Duty Cycle Output Driver | Error |
|---------------------------|--------------------------|-------|
| 10.2 | 10.5 | 0.3 |
| 19.9 | 20.2 | 0.3 |
| 29.7 | 30.1 | 0.4 |
| 39.5 | 39.7 | 0.2 |
| 49.2 | 49.5 | 0.3 |
| 59 | 59.3 | 0.3 |
| 68.8 | 69 | 0.2 |
| 78.5 | 78.8 | 0.3 |
| 98.1 | 98.4 | 0.3 |
| Error rata-rata | | 0.26 |

Berdasarkan tabel 4 ditunjukkan bahwa terdapat rata-rata error sebesar 0.26, sehingga dapat disimpulkan *driver* dapat bekerja dengan baik.

F. Pengujian Kecepatan Motor

Pengujian kecepatan motor dilakukan dengan memberikan nilai PWM sebesar 0 – 255, kemudian diperoleh respon kecepatan motor seperti gambar 8:

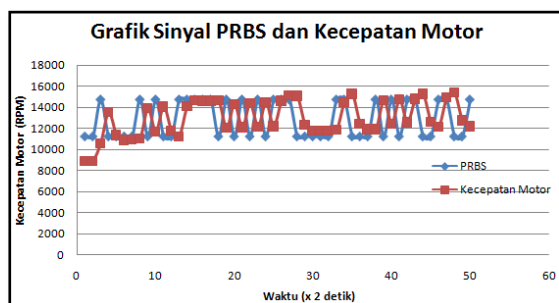


Gambar 8. Grafik Hubungan PWM dengan Kecepatan Motor

Dari gambar 8 dapat dilihat motor mulai berputar di kisaran PWM 20 - 25, kecepatan motor mengalami perubahan yang besar dalam kisaran PWM 25 – 100 sedangkan pada PWM 100 - 255 perubahan kecepatan tidak begitu besar.

G. Pengambilan data Input-Output

Pengambilan data input-output dengan cara dijalkannya program yang membangkitkan sinyal PRBS dari Arduino UNO.



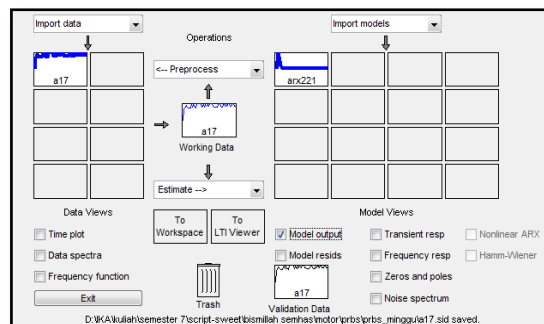
Gambar 9. Grafik Sinyal PRBS dan Kecepatan Motor

Berdasarkan grafik Gambar 9 diamati bahwa data *high* yang ditunjukkan pada titik 65 mempunyai kecepatan motor sebesar 14760 dan data *low* pada titik 35 mempunyai kecepatan motor sebesar 11220, motor

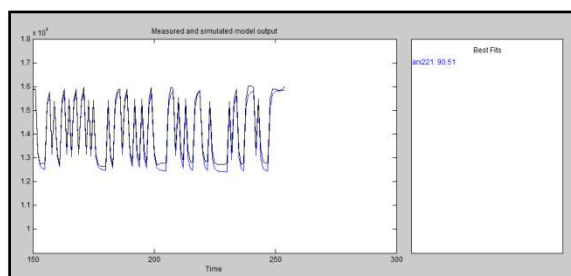
dapat bekerja dengan baik dan memberikan respon sesuai dengan pengontrolan yang diinginkan.

H. Identifikasi Menggunakan MATLAB 7.0

Identifikasi sistem bertujuan didapatkannya fungsi alih dari keseluruhan sistem, pengujian dilakukan menggunakan software MATLAB 7.7 dengan fasilitas ident yang dimilikinya. Data identifikasi yang digunakan, PRBS sebagai input sedangkan kecepatan motor sebagai output. Struktur model yang digunakan adalah *Auto Regresive with Exogenous input* (ARX) dengan estimasi parameter 2 2 1.



Gambar 9. Identifikasi Menggunakan Ident MATLAB



Gambar 10. Best Fit Karakteristik Motor

Dari hasil diatas didapatkan *best fit* terbaik yaitu 90.51 % dengan fungsi alih diskrit

$$A(q) = 1 - 1.152 q^{-1} + 0.1849 q^{-2} \dots\dots\dots (12)$$

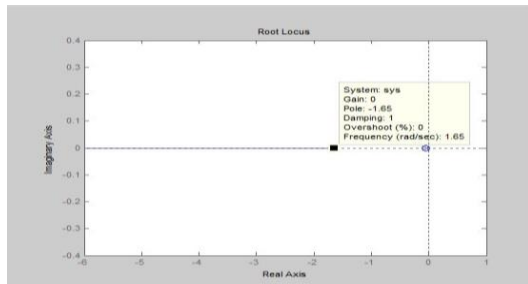
$$B(q) = 0.7648 q^{-1} - 0.7294 q^{-2} \dots\dots\dots (13)$$

Dari data diatas bisa didapatkan fungsi alih dalam bentuk s nya:

$$F(s) = \frac{1.554 s + 0.07376}{s^2 + 1.688 s + 0.06762} \dots\dots\dots (14)$$

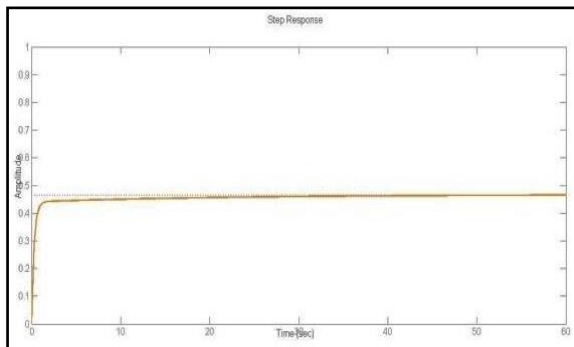
I. Penentuan Parameter PID

Untuk mendapatkan parameter PID yang diinginkan terlebih dahulu ditentukan pole yang diinginkan berdasarkan grafik root locus dari sistem. Dari grafik root locus dapat dilihat bahwa semua akar berada pada sisi kiri bidang s, dapat disimpulkan bahwa sistem stabil dalam nilai manapun, dalam penelitian ini dipilih pole $s = -5.1109$.



Gambar 11. Root Locus Fungsi Alih sistem dan Pemilihan Pole

selanjutnya dilakukan perhitungan untuk menentukan parameter PID sesuai dengan kriteria yang diinginkan dalam hal ini menggunakan metode root locus yang diimplementasikan dalam program MATLAB. Sesuai dengan persamaan 4.



Gambar 13. Grafik Respon sistem tanpa PID

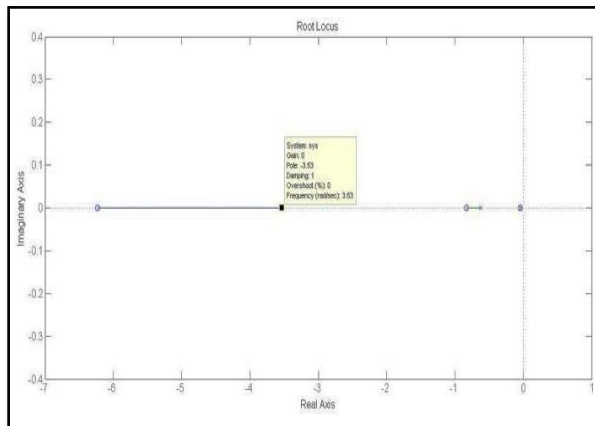
Dari gambar 13 dapat diketahui bahwa respon sistem tanpa menggunakan PID tidak dapat mencapai setpoint yang diinginkan. Dengan digunakannya parameter PID hasil tuning didapatkan respon yang lebih cepat dari pada respon tanpa menggunakan PID. Dari 4 jenis parameter PID yang didapat dipilih nilai PID yang memiliki respon terbaik yaitu :

$$K_P = 3.9133$$

$$K_I = 10$$

$$K_D = 0.3828$$

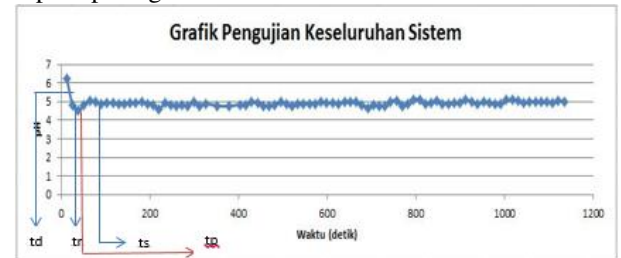
Gambar root locus dari fungsi alih plant ditambah PID adalah sebagai berikut:



Gambar 14. Root Locus Sistem Keseluruhan

J. Pengujian Sistem keseluruhan

Pengujian sistem secara keseluruhan ini dilakukan untuk mengetahui kinerja perangkat keras dan perangkat lunak serta mengetahui respon keseluruhan sistem dengan PID. Implementasi nilai parameter PID yang telah dihitung yaitu $K_P=2.0164$, $K_I=3.5$ dan $K_D=1.2903$ ke dalam rangkaian keseluruhan sistem dengan setpoint pH 6. Dari proses implementasi tersebut dihasilkan respon seperti pada gambar 15.



Gambar 15. Grafik Respon Sistem Keseluruhan

V. KESIMPULAN DAN PROSPEK

- Perancangan sistem pengaturan pemberian asam cuka pada proses pengendapan tahu menggunakan kontroler PID berdasarkan metode root locus.
- Nilai parameter yang digunakan untuk kontroler PID adalah $K_p = 3.9133$, $K_i = 10$ dan $K_d = 0.3828$.
- Setelah diimplementasikan, sistem dapat mempertahankan pH air pengendapan dengan waktu atau *settling time* (t_s) selama 91 detik dan *recovery time* 220 detik saat diberi gangguan 150 mL larutan asam.

Sebaiknya menggunakan tangki dengan kapasitas yang lebih besar agar jumlah produksi semakin meningkat. Disarankan menggunakan sensor pH jenis berbeda dengan kualitas yang lebih baik. Selain itu perlu diperhatikan lagi untuk metode pengadukan larutan asam dan air dalam pengendapan tahu ini.

DAFTAR PUSTAKA

- [1] Moh. Kafadi, Nor. 1994. *Memproduksi Tahu Secara Praktis*. Lumajang. Karya Anda.
- [2] Jalip, Ikna Suyatna. 2011. *Penuntun Praktikum Biokimia Laboratorium Kimia*. Universitas Nasional Jakarta.
- [3] R. Sarjono, Purbowantiningrum. 2006. *Profil Kandungan dan Tekstur Tahu Akibat Penambahan Fitat pada Proses Pembuatan Tahu*. Universitas Diponegoro
- [4] Landau, Ioan dan Gianluca Zito. 2006. *Digital Control Systems Design, Identification and Implementation*. Germany: Springer-Verlag London Limited
- [5] Philip, C. L. & Harbor, R. D. 1996. *Feedback Control System*. Prentice Hall. New Jersey.