

# Penerapan Kontrol Fuzzy Logic Berbasis Matlab Pada Perangkat Mesin Cuci

Hendri D. Putra, Matias K. Kelviandy, Bintang E. Putera

Universitas Gunadarma  
Depok, Indonesia

hendri\_dp@staff.gunadarma.ac.id, matias\_kk@staff.gunadarma.ac.id, bintang\_ep@staff.gunadarma.ac.id

**Abstrak** - Pemanfaatan Fuzzy logic dalam implementasi perangkat seringkali menjadi pilihan terbaik karena dapat menjembatani bahasa manusia yang cenderung tidak presisi dengan bahasa mesin yang cenderung presisi. Dalam pemanfaatannya terhadap implementasi kendali kecepatan mesin cuci sesuai dengan beban dan tingkat kekotoran pakaian tertentu, metode ini jelas berhasil untuk membuat kecepatan putaran mesin cuci lebih optimal. Kendali tidak menitikberatkan pada angka, tetapi menekankan pada suatu nilai semesta pembicaraan. Sehingga respon kecepatan yang dihasilkan sesuai dengan toleransi dari input dan menciptakan kenaikan sinyal grafik yang dihasilkan untuk mendapatkan suatu peningkatan kecepatan yang optimal.

**Kata Kunci:** Fuzzy Logic, Mesin Cuci, Kecepatan, Kendali.

## I. PENDAHULUAN

Fuzzy logic sudah berkembang dalam konsep sistem kendali dalam segala bidang, fuzzy logic dalam arti logika dengan tingkat kebenaran tinggi, menjadi alternatif terbaik dalam mengendalikan, mengatur, menggerak, menjalankan suatu pengendalian sistem yang berfungsi untuk mempermudah manusia dalam menyelesaikan suatu masalah.

Fuzzy logic seringkali menjadi pilihan terbaik karena mempunyai beberapa kelebihan, antara lain dapat menjembatani bahasa manusia yang cenderung tidak presisi dengan bahasa mesin yang cenderung presisi, karena fuzzy logic dibangun berdasarkan pada bahasa manusia. Penggunaan fuzzy logic lebih mudah dengan adanya bahasa pemrograman Matlab, yang dapat mensimulasikan suatu sistem fuzzy, dengan hasil analisis dapat berupa sinyal grafik pada fitur scope.

Fuzzy logic dapat digunakan pada benda yang bekerja pada suatu sistem otomatis, misalnya pada mesin cuci, fuzzy logic dapat mengatur kendali kecepatan mesin cuci sesuai dengan beban dan tingkat kekotoran pakaian tertentu, jika beban

dan tingkat kekotoran pakaian bertambah besar, maka kecepatan yang dihasilkan akan semakin besar juga.

Percobaan sistem kendali fuzzy logic dalam mesin cuci pernah dilakukan dalam penelitian oleh Frans kurniadi (2004), dimana berdasarkan penelitiannya, dengan mekanisme variabel berat 0-7,5 Kg, intensitas cahaya untuk mengasumsikan tingkat kekotoran pakaian dengan nilai 0-50 lux satuan cahaya, dan kecepatan yang dikontrol dengan nilai 0-23,3 rad/sec.

Hasil dari penelitian tersebut berupa kecepatan yang stabil, dalam arti kecepatan yang dihasilkan oleh sinyal grafik pada fitur scope matlab mengalami rise time yang sama, yaitu setiap 0,02 detik pada jalannya waktu simulasi, yang menandakan bertambahnya variabel masukan yaitu beban dan kotor pakaian, tetapi dalam penggunaan fuzzy logic mekanisme ini belum optimal, karena fuzzy logic tidak selalu menitikberatkan pada angka, dan hasil yang didapat memungkinkan mendapat acuan dari nilai crisp, yaitu adanya himpunan (kecepatan) yang sama walaupun sudah mengalami kenaikan harga/nilai dari variabel input, sehingga kecepatan yang dihasilkan dapat tidak sesuai dengan fungsi keanggotaannya.

Penelitian ini dilakukan untuk mencoba mengembangkan dari penelitian yang sebelumnya, dimana hasil yang diharapkan dapat membuat kecepatan mesin cuci lebih optimal, dalam arti kecepatan yang dihasilkan bersifat fuzzy, yaitu tidak menitikberatkan pada angka, tetapi menekankan pada suatu nilai semesta pembicaraan (nilai berada di antara 0 dan 1), sehingga respon kecepatan yang dihasilkan sesuai dengan toleransi dari input, berupa beban dan tingkat kekotoran pakaian, bertambah besar nilai toleransi input, bertambah besar/panjang juga tingkat kenaikan sinyal grafik yang dihasilkan.

## II. LANDASAN TEORI

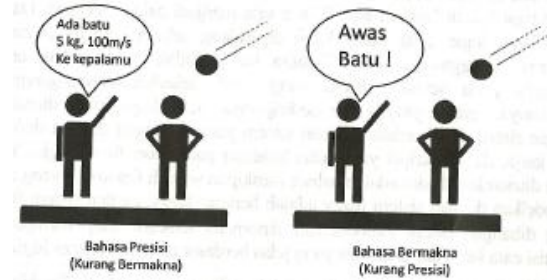
### A. Fuzzy Logic

*Fuzzy logic* di dalam istilah kamus besar diartikan sebagai *blurred* (kabur atau remang-remang), *indistinct* (tidak jelas), *confused* (membingungkan), istilah-istilah ini bukan berarti menggambarkan sistem yang tidak jelas/kabur/remang-remang, akan tetapi, istilah *fuzzy logic* bekerja sebagai sebuah sistem *fuzzy* yang digunakan pada perancangan sistem dengan cara kerja dan deskripsi yang ditujukan untuk membantu memperoleh pemecahan dari masalah dengan cepat dan efisien. Mentransfer kecerdasan manusia ke dalam sebuah komputer atau mesin lebih mudah dilakukan menggunakan *fuzzy logic*. Saat ini, implementasinya sering digunakan pada perangkat seperti mesin cuci, kamera, microwave, dan lain-lain[1].

**B. Himpunan Fuzzy**

Himpunan *fuzzy* merupakan sebuah gagasan utama yang digunakan untuk memperluas jangkauan sebuah fungsi hingga mencakupi bilangan nyata pada interval [0.1]. Nilai semesta yang disebutkan adalah sebuah nilai *fuzzy*, nilai ini tidak hanya pada angka pasti, dimana nilai 0 dapat berarti salah atau sebagai nilai paling minimum, sedangkan nilai 1 berarti benar atau nilai paling maksimum, *fuzzy* beroperasi pada nilai yang terletak diantara nilai tersebut. Himpunan *fuzzy* merupakan himpunan yang berlawanan dengan nilai *crisp*, yakni nilai yang didefinisikan oleh nilai-nilai terdapat di dalam himpunan itu, misalnya jika terdapat nilai "a" merupakan anggota dari himpunan "A", maka nilai yang berkaitan dengan nilai "a" memiliki nilai 1, sedangkan nilai yang tidak berhubungan dengan nilai "a" mempunyai nilai 0. [7]

*Fuzzy logic* pada kemampuannya untuk menekankan sebuah makna atau arti (*significance*) dapat menjadi jembatan dari bahasa mesin yang sangat presisi dengan bahasa yang digunakan oleh manusia yang memiliki presisi rendah. Menggunakan *fuzzy logic*, sebuah sistem kepakaran manusia dapat ditanam ke dalam bahasa mesin dengan mudah dan efisien. Gambar 1 merupakan fungsi dari sistem *fuzzy* sebagai sebuah ilustrasi, dimana terdapat bahasa presisi (dengan bilangan) yang digunakan dan bahasa kata-kata. Manusia cenderung kesulitan mengartikan bahasa presisi yang dibutuhkan oleh mesin, sebagai lawannya, manusia lebih mudah mengartikan istilah/variabel perkataan yang jauh lebih sederhana dan singkat. Dalam hal ini, *fuzzy* dapat diimplementasikan sebagai fungsi yang dapat menjembatani komunikasi antara mesin dengan manusia agar menjadi lebih efektif dan efisien antara mesin dan manusia[1].

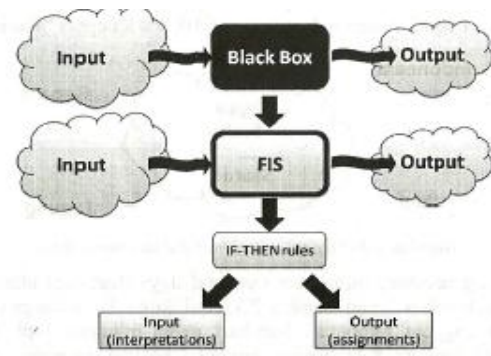


Gambar 1 Bahasa Presisi dan Bahasa Bermakna[1]

Pengendali *fuzzy* umumnya digunakan karena handal, mudah diperbaiki, dan dapat memberikan pengendalian yang sangat baik jika dibandingkan dengan teknik sejenis yang biasanya butuh usaha dan dana yang tidak kecil.

**C. Fuzzy Inference System (FIS)**

*Fuzzy logic* memiliki prinsip kerja sebagai pemetaan area *input* ke dalam area *output* dengan menerapkan aturan IF-THEN. Gambar 2 menjelaskan logika pemetaan hubungan antara *input* dengan *output*. Di tengah antara *input* dengan *output* terdapat sebuah blok yang disebut *black box*. Bagian ini bertugas melakukan pemetaan. *Fuzzy Inference System (FIS)* memiliki beberapa aturan dengan urutan yang dapat sembarang. aturan ini digunakan untuk menghasilkan sebuah kesimpulan. Oleh sebab itu, seluruh aturan harus ditentukan lebih awal jika membuat sebuah FIS yang mengimplementasikannya.



Gambar 2 Konsep Umum Pembangunan FIS [1]

FIS dapat disebut sebagai metode yang melakukan interpretasi nilai di dalam vektor *input* dan menghasilkan kesimpulan dari aturan IF-THEN yang diterapkan, kemudian FIS akan menghasilkan nilai pada vektor *output*. FIS dapat diimplementasikan ke dalam beberapa bidang, misalnya kendali otomatis, klasifikasi data, analisis keputusan, dan sistem pakar[1].

**D. Matlab (Matrix Laboratory)**

Matlab merupakan sebuah bahasa pemrograman tingkat tinggi yang arti perintah dan fungsi-fungsinya lebih mudah dipahami. Di awal penggunaannya, matlab dapat menangani bermacam-macam operasi matriks dan vektor menggunakan berbagai rutin dan pustaka LINPACK dan EISPACK. Saat ini Matlab lebih berkembang karena berbagai masukan dari banyak penggunanya. Matlab di dalam dunia akademis pada bidang perkuliahan menjadi perangkat bantu standar instruksional dan pada bidang matematika, keilmuan dan teknik menjadi perangkat bantu instruksional tingkat lanjut.

Matlab terdiri dari beberapa jenis *toolbox* dimana didalamnya terdapat bermacam bidang keilmuan yang lebih khusus, seperti kecerdasan buatan (*fuzzy logic, neural network, control system, signal processing* dan *wavelet*).

Matlab memiliki bermacam kemudahan untuk dipelajari dan digunakan untuk merancang sebuah sistem kendali. Matlab akan menghasilkan keluaran yang beragam, salah satunya adalah analisis grafik dari perilaku sebuah sistem kendali dengan metode yang diterapkan[4].

#### E. Fuzzy Logic Toolbox

*Fuzzy Logic Toolbox* merupakan sekumpulan *tool* yang dapat digunakan dalam perancangan sistem *fuzzy*. Perangkat ini juga dapat digunakan untuk membuat atau mengubah FIS (*Fuzzy Inference System*) pada lingkungan kerja Matlab. *Tool* ini mudah untuk digunakan dimana pengguna dapat berkreasi dengan bebas saat melakukan rancang bangun sistem FIS. *Fuzzy Logic Toolbox* memiliki 5 jenis *Graphic user Interface* (GUI), yakni:

1. FIS Editor
2. Membership Function Editor
3. Rule Editor
4. Rule Viewer
5. Surface viewer

*FIS Editor, Membership Function Editor, dan Rule Editor* memiliki fungsi untuk membaca dan memodifikasi data yang sudah ada, sedangkan *tool Rule Viewer* dan *Surface Viewer* hanya memiliki fungsi untuk pembacaan data dimana modifikasi melalui perangkat ini tidak dapat dilakukan jika terjadi kesalahan.

Semua GUI yang telah disebutkan saling terikat dan mempengaruhi, dimana perubahan yang dilakukan pada GUI satu akan mempengaruhi GUI lain[1].

### III. METODE PENELITIAN

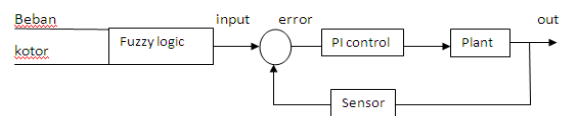
Dalam penelitian ini, metode penelitian yang digunakan adalah metode studi pustaka, analisis dan perbandingan (*comparison*). Metode studi pustaka, peneliti mengambil sumber teori pendukung yang sesuai dengan judul penelitian, baik sumber yang berupa buku, atau berupa jurna.l. Metode analisis menjabarkan bagaimana suatu *fuzzy logic* dapat bekerja untuk mengendalikan kecepatan pada mesin cuci, sehingga dapat mendapat suatu kesimpulan faktor-faktor yang mempengaruhi suatu *output*. Metode perbandingan (*comparison*) membandingkan hasil kesimpulan dari yang penulis buat, dengan hasil kesimpulan dari percobaan yang dilakukan peneliti sebelumnya.

### IV. PERANCANGAN

#### A. Perancangan Model

Perancangan model merupakan suatu rancangan awal sistem untuk membuat simulasi. Simulasi yang dirancang adalah simulasi mesin cuci yang diberikan pengendalian *fuzzy logic* untuk mengon-trol kecepatan sesuai dengan toleransi yang diberikan oleh *input*. *Fuzzy logic* pada sistem ini sebagai penentu nilai *setpoint* yaitu harga yang diinginkan bagi variabel yang dikontrol selama pengontrolan.

Sistem yang digunakan untuk simulasi ini adalah sistem tertutup, dimana adanya perbandingan dari nilai *output* sebagai sinyal umpan balik dengan nilai *input* yang masuk, seperti Gambar 3 berikut.



Gambar 3 Sistem Fuzzy Logic Sebagai *Setpoint*

Pada Gambar 3 di atas terlihat jelas bahwa beban dan kotor sebagai *input* variabel, nilai dari kedua *input* ini akan ditoleransi untuk selanjutnya diproses. Toleransi dari *input* tersebut diproses *fuzzy logic* untuk mengontrol putaran motor pada mesin cuci dan menghasilkan nilai *setpoint*. Sedangkan *PI Control* untuk memproses kesalahan (*error*) yang masuk setelah melalui elemen pengontrol yaitu *fuzzy logic*, di bagian ini juga dapat di tentukan nilai konstanta ( $k_p$  dan  $k_i$ ) yang akan terintegritas oleh nilai hasil dari proses *fuzzy logic* untuk menghasilkan suatu hasil yang optimal.

*Plant* adalah model sistem yang digunakan untuk menjalankan suatu rancangan sistem, misalnya untuk *plant* pada mesin cuci adalah perancangan desain *flux vektor* yang didalamnya berisi vektor kontrol untuk membuat suatu motor sistem.

Sensor merupakan elemen yang mengontrol sinyal umpan balik keluaran untuk di selisihkan dengan nilai *setpoint*.

*Error* adalah selisih antara nilai *setpoint* dan umpan balik masukan dari sensor.

**B. Desain Flux Vektor**

Beberapa vektor *control* diagram yang digunakan untuk desain Flux, yaitu :

- *Blok Speed Controller*
- *Current calculation Diagram*
- *Transformation Diagram*
- *Flux Calculation diagram*
- *Teta Calculation Diagram*

**1) Blok Speed Controller**

Matlab menyediakan suatu fungsi blok dengan menggunakan transformasi z, blok ini berfungsi untuk menentukan fungsi transfer pada sistem yang didapat dari *Te* (Torsi elektromagnetik) yang merupakan fungsi dari arus stator dan arus rotor, sehingga dapat dihasilkan.

$$e = W^* - W_m \tag{1}$$

$$Te = K_p e + K_i \int e dt \tag{2}$$

atau

$$\text{kontrol PI} = K_p + K_i/z \tag{3}$$

Dengan z sebagai transformasi bilinear

$$\left( \frac{T(z+1)}{2(z-1)} \right) \tag{4}$$

Dari persamaan diatas dihasilkan fungsi transfer :

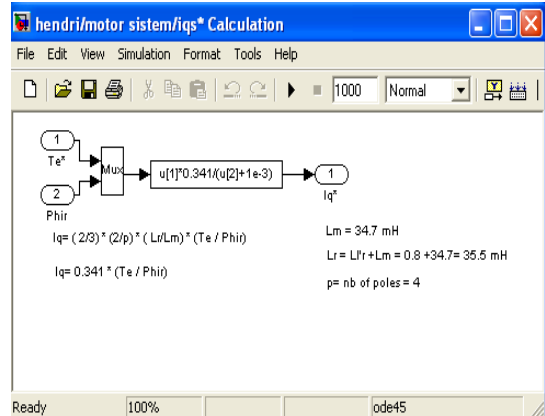
$$G(z) = Te = K_p * e + K_i \left( \frac{T(z+1)}{2(z-1)} \right) * e \tag{5}$$

Diasumsikan nilai  $K_p = 75$  dan  $K_i = 50$ , sehingga

$$G(z) = 75e + 50e \left( \frac{T(z+1)}{2(z-1)} \right) \tag{6}$$

**2) Current calculation Diagram**

Blok ini terdiri dari dua bagian yaitu  $I_d^*$  sebagai *fluks motor controlled* dan  $I_q^*$  berfungsi sebagai pengontrolan *torque* motor dengan nilai bergantung pada  $Phir^*$  dan  $Te^*$ .  $Te^*$  (torsi elektromagnetik) didapat dari blok *speed controller* dimana terdapat *PI control*, dengan perbandingan  $K_P : K_I$  yaitu  $75 : 50$ , dan *torque* motor  $300N.m$ . sedangkan nilai  $Phir^*$  tergantung pada slip motor induksi, yang ditentukan sebesar  $0.96$ . Hasil dari blok ini berupa arus yang dipakai untuk DQ-ABC.



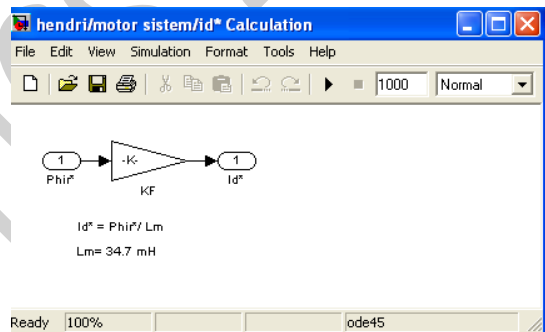
**Gambar 4**  $I_q^*$  Calculation Diagram

Rumus  $I_q^*$  Calculation Diagram :

$$I_q = (2/3) * (2/p) * (Lr/Lm) * (Te / Phir) \tag{7}$$

$$I_q = 0,341 * (Te / Phir)$$

$L_r = L_{r'} + L_m = 0,8 + 34,7 = 35,5$  mH  
dengan  $L_m = 34,7$  mH



**Gambar 5**  $I_d^*$  Calculation Diagram

Rumus  $I_d^*$  Calculation Diagram :

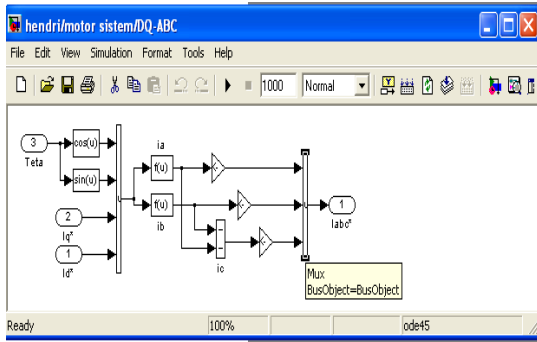
$$I_d^* = Phir^* / L_m \tag{8}$$

dengan  $L_m = 34,7$  mH

**3) Transformation Diagram**

Blok ini terdiri dari dua blok utama, yaitu Blok ABC-DQ dan Blok DQ-ABC. Blok DQ-ABC mengubah *current Direct-Quadratic references* menjadi *current references*  $I_a^*$ ,  $I_b^*$ ,  $I_c^*$ , yang menjadi *input current regulator*. Sebaliknya, Blok ABC-DQ mengubah arus *Current*  $I_a^*$ ,  $I_b^*$ ,  $I_c^*$  yang terukur di stator, menjadi *current DQ*.

Kedua Transformation Diagram merupakan satu kesatuan fungsi dan tidak bisa berdiri sendiri karena saling bergantung satu dengan yang lain.



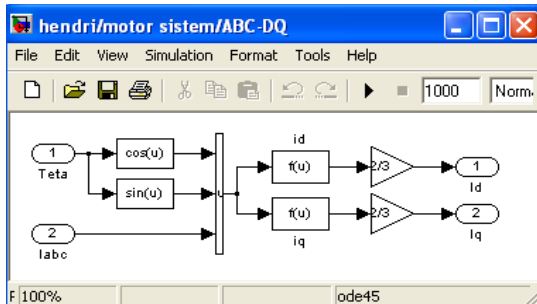
Gambar 6 DQ-ABC

Transformasi tegangan dq-to-abc:

$$\begin{bmatrix} V_{qs} \\ V_{ds} \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 2 \cos \theta & \cos \theta + \sqrt{3} \sin \theta \\ 2 \sin \theta & \cos \theta + \sqrt{3} \cos \theta \end{bmatrix} \begin{bmatrix} V_{abs} \\ V_{bcs} \end{bmatrix}$$

Tranformasi arus dq-to-abc :

$$\begin{bmatrix} i_{as} \\ i_{bs} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ \frac{-\cos \theta + \sqrt{3} \sin \theta}{2} & \frac{-\sqrt{3} \cos \theta - \sin \theta}{2} \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \end{bmatrix}$$



Gambar 7 ABC-DQ

Transformasi tegangan abc-to-dq :

$$\begin{bmatrix} V'_{qr} \\ V'_{dr} \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 2 \cos \theta \cos \beta + \sqrt{3} \sin \beta \\ 2 \sin \beta \sin \beta - \sqrt{3} \cos \beta \end{bmatrix} \begin{bmatrix} V'_{abr} \\ V'_{bcr} \end{bmatrix}$$

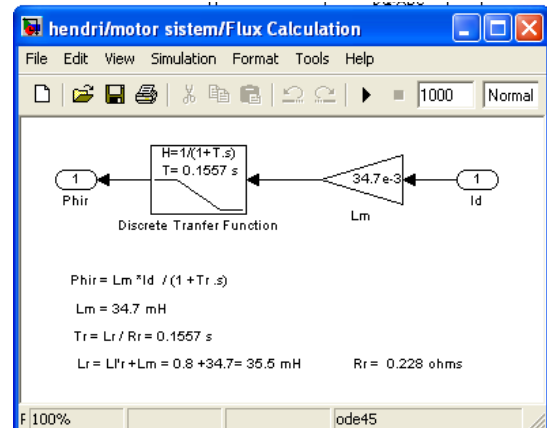
Transformasi arus abc-to-dq :

$$\begin{bmatrix} i'_{ar} \\ i'_{br} \end{bmatrix} = \begin{bmatrix} \cos \beta & \sin \beta \\ \frac{-\cos \beta + \sqrt{3} \sin \beta}{2} & \frac{-\sqrt{3} \cos \beta - \sin \beta}{2} \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \end{bmatrix}$$

#### 4) Flux Calculation Diagram

Blok flux calculation merupakan bagian yang berfungsi untuk menghitung besaran flux kemudian hasil dari bagian ini adalah Phir yang dipakai untuk menghitung iq\* calculation setiap saat. Discrete transfer function merupakan bagian yang paling penting dalam blok ini karena terdapat

rumus untuk menghitung besaran flux. Transfer function yang muncul karena perubahan yang terjadi setiap saat adalah orde satu dengan periode 0.1557s (dari perhitungan Lr/Rr) dipakai untuk mengintegrasikan perkalian id dan Lm menjadi Phir.



Gambar 8 Flux Calculation

Rumus Flux Calculation Diagram:

$$Phir = Lm * i_d / (1 + T_r * s) \tag{9}$$

dengan Lm=34,7 mH

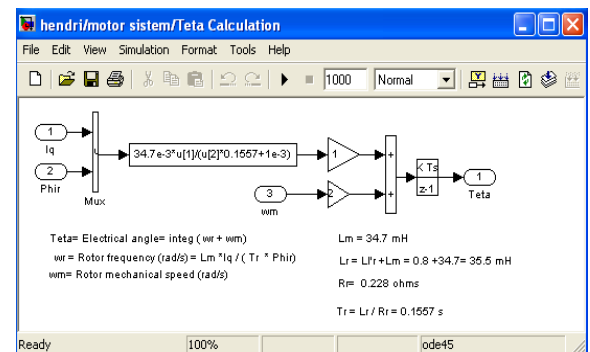
Tr = Lr/Rr = 0,1557 s

dengan Rr=0,228 ohms

Lr=Lr+Lm=0,8+34,7=35,5mH

#### 5) Teta Calculation Diagram

Blok ini berfungsi untuk menghitung nilai sudut antara rotor dan stator dalam fungsi sudut radian. Pada bagian ini diperlukan parameter Iq\* dan Phir dan rotor mechanical speed (wm).



Gambar 9 Teta Calculation

Rumus Teta Calculation

$$Teta = Electrical\ angle = \text{Integ}(wr + wm) \tag{10}$$

wr = rotor frequency (rad/sec)

$$= Lm * i_q / (Tr * Phir) \text{ dan}$$

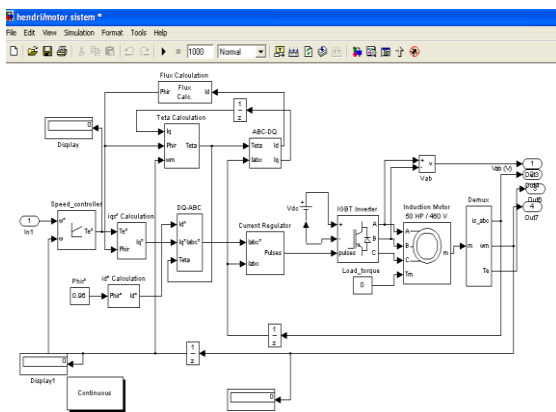
wm = rotor mechanical speed (rad/sec)

$L_r = L_l * r + L_m = 0,8 + 34,7 = 35,5 \text{ mH}$  dan  
 $T_r = L_r / R_r = 0,1557 \text{ s}$  dengan  $R_r = 0,228 \text{ ohms}$

$$\theta_e = \int (\omega_m + \omega_{sl}) dt \text{ Transformasi ke } z = \frac{T}{z-1}$$

6) Simulasi Dengan Menggunakan Simulink Matlab

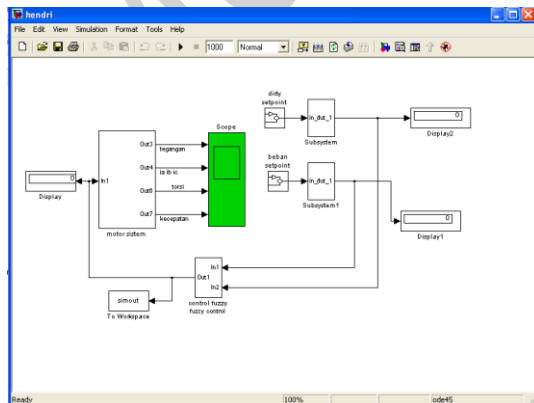
Setelah mendefinisikan desain flux vektor tersebut jika di rancang simulasi dalam simulink matlab maka akan dihasilkan blok motor.



Gambar 10 Blok Motor Sistem

Gambar 10 adalah blok motor sistem yang di dalam nya terdapat desain *plant* yang berfungsi untuk mengintegrasikan *fuzzy logic control* dengan pengontrolan PI (*proportional integrated*) yang terdapat pada blok *speed controller* untuk menghasilkan suatu grafik sinyal yang optimal.

Jika secara keseluruhan sistem simulasi mesin cuci dengan menggunakan *fuzzy logic control* telah selesai didefinisikan, maka tampilan pada Simulink Matlab sebagai berikut:



Gambar 11 Simulasi Mesin Cuci Pada Simulink Matlab

Gambar 11 merupakan simulasi dari perancangan untuk mengatur kecepatan pada mesin cuci dengan menggunakan *fuzzy logic control*. Icon terdiri dari *variable* kotor dan beban, untuk menghasilkan nilai *setpoint* maka toleransi dari *input* tersebut diproses dalam *fuzzy control* yang didalamnya terdapat FIS editor yang telah dirancang, keluaran dari *fuzzy control* masuk ke dalam motor sistem.

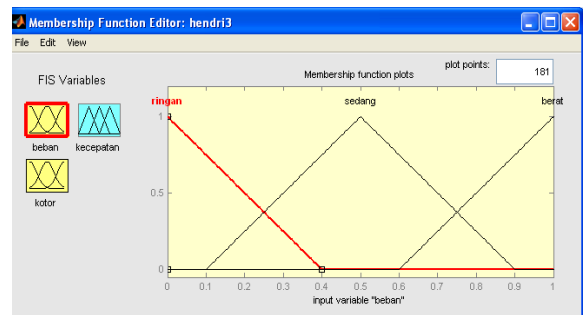
Hasil akhir dari simulasi ini berupa grafik sinyal yang dihasilkan motor sistem yaitu Tegangan,  $I_a$   $I_b$   $I_c$ , Torsi, dan kecepatan yang merupakan hasil dari proses *fuzzy logic control*.

Pada Simulink terdapat *display icon* digunakan untuk menampilkan *subsystem* dari setiap *input*, pada saat simulasi dijalankan *display icon* ini akan menunjukkan angka-angka sebagai tanda *input* telah berjalan.

V. HASIL dan PEMBAHASAN

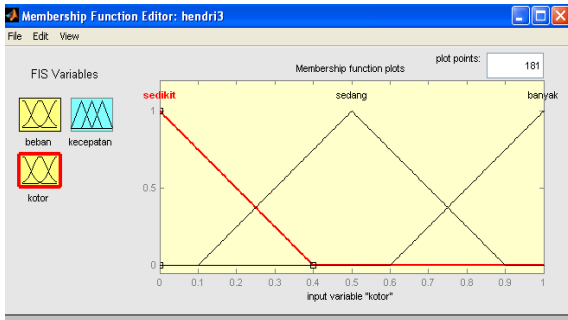
A. Hasil yang Diharapkan

Setelah meneliti hasil yang didapat dari penelitian sebelumnya, penulis mencoba mengoptimalkan hasil yang diharapkan, yaitu kecepatan yang didapat sesuai dengan fungsi keanggotaan, dengan menggunakan mekanisme nilai semesta pembicaraan, yaitu nilai tidak dititikberatkan pada angka, tetapi berdasar pada nilai diantara nilai 0 dan 1, nilai 0 berarti *input* yang minimum yaitu dimulai dari tidak ada beban dan tidak ada tingkat kekotoran, sedangkan nilai satu nilai maksimal, yaitu beban dan tingkat kekotoran maksimal sesuai dengan standar mesin cuci. Nilai semesta diberikan pada semua elemen baik *input* maupun *output*, dengan nilai fungsi keanggotaan sebagai berikut:



Gambar 12 Representasi Variabel Beban

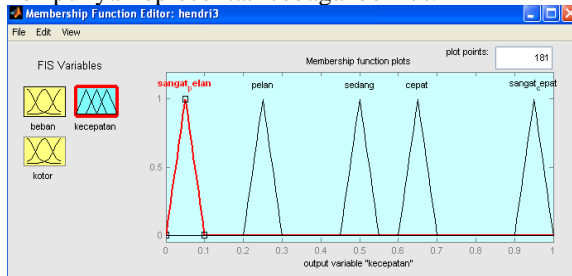
Pada Gambar 12 diterangkan representasi dari *input* berupa beban yang berkisar antara 0 dan 1, terdapat tiga fungsi keanggotaan, yaitu pada himpunan ringan (0-0,4), sedang (0,1-0,9), dan berat (0,6-1).



Gambar 13 Representasi Variabel Kotor

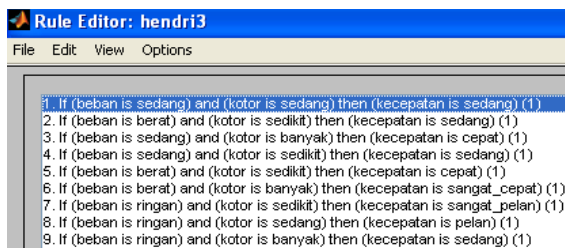
Pada Gambar 13 diterangkan representasi dari *input* berupa kotor yang berkisar antara 0 dan 1, terdapat tiga fungsi keanggotaan, yaitu pada himpunan sedikit (0-0.4), sedang (0.1-0.9), dan banyak (0.6-1).

Sedangkan pada kecepatan sebagai *output* mempunyai representasi sebagai berikut:



Gambar 14 Representasi Variabel Kecepatan

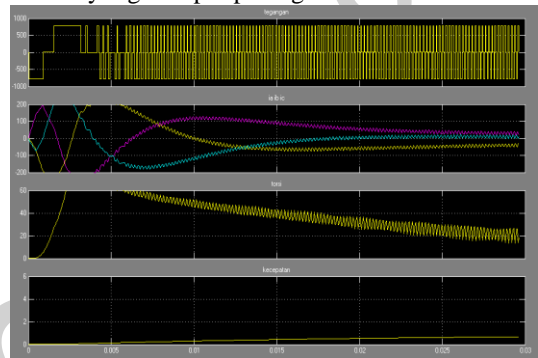
Gambar 14 merupakan representasi dari variabel yang dikontrol yaitu kecepatan, dimana untuk mendapatkan suatu hasil yang optimal diberikan beberapa suatu fungsi keanggotaan, agar respon yang dikeluarkan pada kecepatan sesuai dengan fungsi keanggotaan. Fungsi keanggotaan terdiri dari, sangat pelan (0 – 0.1), pelan (0.2 - 0.3), sedang (0.45 - 0.55), cepat (0.6 – 0.7) dan sangat cepat (0.9 – 1). Disini nilai 0,8 tidak termasuk salah satu fungsi keanggotaan, untuk membuktikan bahwa nilai semesta pembicaraan (nilai di antara 0 dan 1) tidak menitikberatkan pada kefokusannya, sehingga walaupun nilai tersebut tidak termasuk nilai suatu fungsi keanggotaan, tanggapan pada sinyal kecepatan akan tetap sama. Adapun dengan *rule* sebagai berikut:



Gambar 15 Representasi Rule Editor

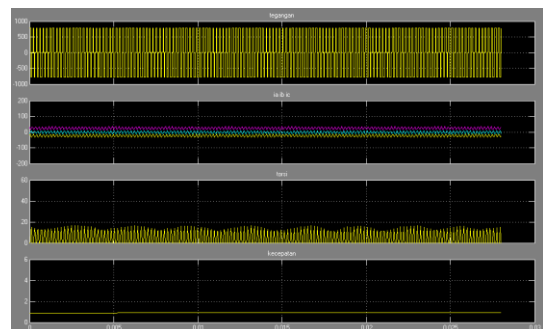
Pada Gambar 15 merupakan IF-THEN *rule* yang digunakan sebagai acuan untuk menentukan kecepatan sesuai kondisi yang ada.

Setelah mengidentifikasi *input* dan *output* beserta *rule*, disini peneliti mengintegritaskan kendali *fuzzy logic* dengan pengontrolan PI (*proportional integrated*), dimana nilai konstanta diperbesar dari percobaan sebelumnya yaitu menjadi  $K_p = 75$  dan  $K_i = 50$ , karena dengan menambah nilai konstanta, dapat memperbaiki *respon time*, walaupun akan terjadi lonjakan (*overshoot*) yang besar juga. Hasil respon dari simulasi yang didapat pada gambar berikut:



Gambar 16 Tampilan Grafik Pada Scope

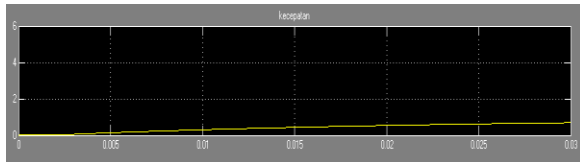
Dari Gambar 16 disimpulkan sistem dapat menghasilkan respon tanggapan berupa sinyal grafik yang terdiri dari tegangan, arus (ia ib ic), torsi, dan juga kecepatan. Grafik tegangan, arus (ia ib ic), dan torsi menunjukkan hasil dari desain flux control dengan pengontrolan PI (*proportional integrated*) dengan konstanta  $K_p = 75$  dan  $K_i = 50$ , ini memberi pengaruh untuk memberikan keadaan grafik yang halus (tidak patah-patah) saat mengalami kondisi transient pada awal jalannya simulasi, konstanta ini mengalami *overshoot* yang tinggi, karena tingginya nilai konstanta pada pengontrolan PI (*Proportional Integrated*), namun dapat mengatasi kondisi *respon transient* secara cepat dan efisien, sehingga grafik yang dihasilkan bersifat optimal, yaitu grafik dapat berjalan sesuai dengan kondisi sistem, grafik optimal dapat dilihat sebagai berikut:



Gambar 17 Tampilan Grafik Optimum Pada Scope

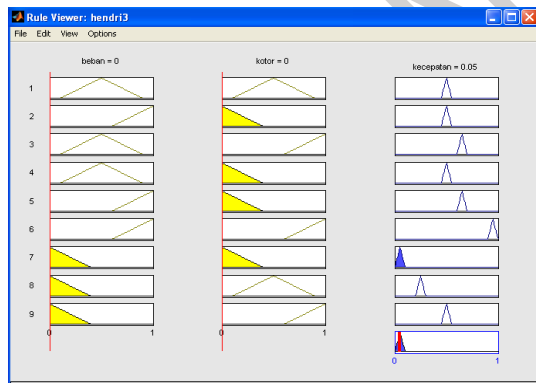
Gambar 17 merupakan grafik tampilan optimum, dimana sinyal grafik yang dihasilkan berbentuk lurus (tidak berubah).

Pada kecepatan yang dihasilkan, didapatkan suatu grafik yang bersifat fuzzy, dimana *rise time* tidak didapatkan dengan waktu yang tetap seperti yang didapatkan hasil dari penelitian sebelumnya yang mempunyai nilai *rise time* tetap, untuk lebih jelasnya dapat dilihat pada grafik berikut:



Gambar 18 Tampilan Grafik Kecepatan

Grafik yang dihasilkan pada Gambar 18 menunjukkan bahwa *rise time* atau kenaikan grafik berupa tangga ini tidak mempunyai nilai tetap, tidak seperti percobaan sebelumnya yang mempunyai waktu tetap dalam memunculkan *rise time*, pada grafik ini semakin bertambahnya kemunculan *rise time*, maka semakin bertambah juga waktu yang dihasilkan setiap waktunya jalan simulasi, ini menandakan bahwa setiap munculnya *rise time* maka bertambah juga nilai *input* yang di terima berupa beban dan tingkat kekotoran, sehingga kecepatan yang dihasilkan akan sesuai dengan fungsi keanggotaannya, seperti dalam *rule viewer* berikut:



Gambar 19 Tampilan Rule Viewer

Gambar 19 menunjukkan *output* yang berupa kecepatan sesuai dengan nilai fungsi keanggotaan, ketika *input* diberi nilai minimal atau nol, kecepatan yang dihasilkan bernilai bersifat minimal juga (mendekati nol).

## VI. KESIMPULAN

Dari penelitian yang telah dilakukan, dapat disimpulkan bahwa:

Kecepatan putaran yang dihasilkan dari *fuzzy logic* ditentukan oleh *input*, dimana *input* yang diaplikasikan adalah beban dan kotor pakaian. Dalam simulink matlab *output* yang berupa kecepatan menghasilkan grafik, dimana grafik dimulai dari tanpa beban dan kotoran (nilai *input* minimum), sampai dengan tingkat beban dan kotor pakaian tertinggi (nilai *input* maximum).

Tanggapan respon pada kecepatan yang dihasilkan bersifat optimal, karena *rise time* atau kenaikan grafik berupa tangga ini tidak mempunyai nilai tetap atau bervariasi, semakin bertambahnya kemunculan *rise time*, maka semakin bertambah juga waktu kenaikan *rise time* yang dihasilkan setiap detiknya, ini menandakan bahwa setiap munculnya *rise time* maka bertambah juga nilai *input* yang di terima berupa beban dan tingkat kekotoran.

Nilai semesta pembicaraan, yaitu nilai yang berada di antara 0 dan 1 dapat memberikan respon pada kecepatan yang lebih optimal, karena tidak mentitik-beratkan pada suatu titik angka, sehingga grafik yang dihasilkan sesuai dengan fungsi keanggotaan *fuzzy* dari *input* maupun *output*.

Pemanfaatan *Fuzzy Logic* seringkali menjadi pilihan terbaik. Walau sudah cukup banyak sistem kontrol yang sudah bekerja dengan baik tanpa *fuzzy logic*, metode ini sangat tergantung pada kebiasaan pemakaian seorang pengguna, jika sering menggunakan metode ini bahwa akan mengemukakan *fuzzy logic* adalah sebuah metode yang solid dan efisien untuk memecahkan masalah pemetaan yang memprioritaskan kepresisian

## REFERENSI

- [1]. Agus Naba, *Belajar cepat Fuzzy Logic Menggunakan Matlab*, C.V Andi Offset, Yogyakarta, 2009.
- [2]. Felix Pasila Limboto dan Bayu Krisnawan, "Desain Flux Vector Control Inverter 3 Fasa Pada Motor Induksi 1.5 HP Menggunakan Power Blockset"
- [3]. Frans Kumiadi, "Simulasi *Fuzzy Logic* Control Pada Mesin Cuci Menggunakan Simulink Matlab"
- [4]. Gunaidi Abdia A way, *Matlab Programming*, Informatika, Bandung, 2010.
- [5]. Heru Supriyono dan Rochmad Rosyidi, "Implementasi Pengendalian Kecepatan Motor DC Dengan Pengendalian *Fuzzy*-PID Menggunakan RealTime Windows Target Matlab 6.5", *Jurnal Penelitian Sains & Teknologi*, Vol. 9, 2008, pp. 165-176.
- [6]. Rudy S. Wahyudi, "Estimasi sistem orde dua dengan menggunakan Perangkat Lunak Matlab", *Jurnal Teknik Elektro Trisakti*, vol.1, Nomor.1. Agustus 2010, pp. 17-24.
- [7]. Sri Kusumadewi, *Analisis & Desain Sistem Fuzzy Menggunakan Tool Box Matlab*, Graha Ilmu, Yogyakarta, 2002.