

Sistem Kontrol Kecepatan Sepeda Listrik Menggunakan Metode *Self-Tuning* Parameter PI dengan Metode Logika Fuzzy

Mochamad Adityo Rachmadi, M. Aziz Muslim, dan. Erni Yudaningtyas

Abstract— A hill, which is in real circumstance can have a different incline in a continuous road, is one of the obstacle that a vehicle, in this case a bike, must overcome. That is why, a controlling action which can adapt the various loading condition is needed. A well-tuned PI controller can produce a good output in a linear load, however, it will struggle to adapt in a load that is constantly changing. A speed control system using a fuzzy logic base to self-tune PI parameters is used to fine tune PI controller gain parameters, so that the PI controller can adapt such loading condition. In this research, a no-load system performance is compared to those with load, with the setpoint of 120 rpm.

Keywords: *Electric Bicycle, Steady State Error, PI Controller, Fuzzy Logic*

Abstrak— Pada kendaraan sepeda, salah satu beban yang dihadapi adalah tanjakan, di mana sebuah tanjakan dalam keadaan sebenarnya dapat memiliki kemiringan yang berbeda-beda dalam suatu jalan. Karena itu, dibutuhkan sebuah aksi pengontrolan yang dapat beradaptasi dengan kondisi pembebanan yang berubah-ubah tersebut. Sebuah kontroler PI dengan parameter penguatan yang telah ditala sedemikian rupa dapat menghasilkan keluaran yang baik pada beban linier, namun akan kesulitan beradaptasi terhadap pembebanan yang berubah-ubah. Sistem kontrol kecepatan menggunakan metode *self-tuning* parameter PI dengan metode logika fuzzy digunakan untuk menala parameter penguatan kontroler PI, agar kontroler PI dapat beradaptasi dengan pembebanan tersebut. Pada penelitian ini, dibandingkan performansi sistem dengan kondisi tanpa beban dan kondisi berbeban dengan setpoint 120 rpm.

Kata Kunci: *Sepeda Listrik, Error Steady State, Kontroler PI, Logika Fuzzy*

I. PENDAHULUAN

SISTEM transportasi merupakan suatu hal yang harus ada. Mayoritas masyarakat Indonesia menganggap bersepeda sebagai aktivitas sampingan yang dilakukan pada hari libur, dengan tujuan olahraga dan rekreasi. Namun di beberapa negara seperti Cina, Jepang, dan Belanda, sepeda menjadi alat transportasi alternatif

untuk bepergian. Seiring dengan tingginya tingkat mobilitas masyarakat, Sepeda listrik merupakan salah satu alternatif bagi masyarakat karena tidak menghasilkan emisi gas buang. Sepeda listrik menggunakan motor arus searah tanpa sikat atau dikenal dengan *brushless DC* (BLDC) yang menghasilkan efisiensi tinggi, torsi tinggi, kecepatan tinggi dan biaya perawatan yang rendah [1].

Sistem kontrol PI adalah sistem paling umum dipakai pada aplikasi kontrol di dunia industri. Pada kontrol *Propotional-Integral* (PI) konvensional, penalaan pada kondisi tertentu tanpa memperhitungkan perubahan pembebanan yang terjadi pada *plant* dan gangguan yang muncul mendapatkan hasil kontrol yang sudah bagus dan teruji. Tetapi jika terjadi perubahan pembebanan pada *plant*, maka perlu diadakan setting ulang untuk membuat kontrol PI menghasilkan unjuk kerja yang baik [2]. Pada aplikasi kontrol kecepatan motor, kontroler PI dapat menghilangkan *steady-state error* pada *output* motor, namun berdampak pada *settling time* yang lambat dalam mencapai nilai *steady* [3].

Seiring dengan perkembangan sistem berbasis pengetahuan, penalaan parameter PI dapat ditentukan dengan menganalisis *output* suatu sistem, dan hasil analisis tersebut dibentuk dalam sejumlah aturan. Dengan mengkombinasikan sejumlah aturan, pengaturan ini menjadi sebuah kontrol logika *fuzzy* sebagai salah satu sistem berbasis pengetahuan sehingga parameter PI dapat ditala secara *real time*. Dengan penggabungan aksi kontroler PI dan kontrol logika *fuzzy* untuk menala parameter PI, diharapkan akan didapatkan suatu tanggapan yang mempunyai tingkat kestabilan yang tinggi, dan dengan nilai *settling time* yang cepat.

II. METODE PENELITIAN

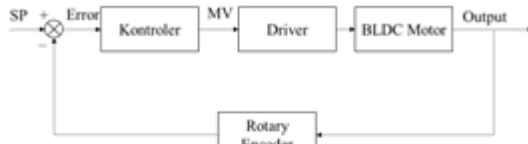
A. Diagram Blok Sistem

Diagram blok sistem kontrol kecepatan pada sepeda listrik dapat dilihat dalam Gambar 1.

Setpoint (SP) adalah masukan berupa nilai kecepatan yang diberikan oleh pengguna. Nilai *setpoint* akan disimpan ke dalam mikrokontroler. *Error* adalah deviasi atau simpangan antara pembacaan aktual kecepatan motor dari *rotary encoder* dan nilai *setpoint*. Kedua

Mochamad Adityo Rachmadi, M. Aziz Muslim, S.T., M.T., Ph.D., Dr. Ir. Erni Yudaningtyas, M.T. are with the Electrical Engineering Department of Brawijaya University, Malang, Indonesia (corresponding author provide phone 0341-554166; email mochamad.adityo@gmail.com)

nilai tersebut akan dikalkulasi dan diolah oleh mikrokontroler.



Gambar 1. Diagram blok sistem secara umum

Nilai yang dikalkulasi adalah *manipulated variable* (MV) atau sinyal kontrol yang akan digunakan sebagai masukan pada *driver* motor. *Driver* motor ini akan menggerakkan motor *brushless* DC sesuai dengan sinyal kontrol keluaran mikrokontroler.

B. Pengujian Motor Brushless DC

Pengujian pada motor *brushless* DC dilakukan dengan tujuan untuk mengetahui karakteristik motor terhadap kenaikan tegangan *input* motor. Hasil pengujian dapat dilihat dalam Gambar 2.

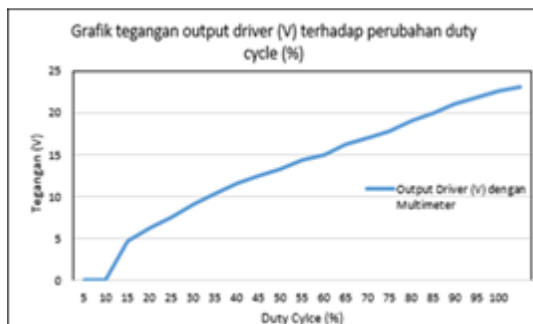


Gambar 2. Perubahan kecepatan motor terhadap input tegangan

Dari Gambar 2 dapat diketahui bahwa semakin besar input tegangan yang diberikan, maka semakin besar pula kecepatan motor tersebut.

C. Pengujian Driver Motor Tiga Fasa

Pengujian pada *driver* motor tiga fasa dilakukan dengan tujuan mengetahui karakteristik, kinerja, dan *output* rangkaian *driver* motor tiga fasa. Hasil pengujian dapat dilihat dalam Gambar 3.



Gambar 3. Perubahan tegangan output driver terhadap *duty cycle*

Dari Gambar 3 dapat diketahui bahwa semakin besar *duty cycle* yang diberikan, maka semakin besar pula tegangan efektif *output driver* tersebut.

D. Pengujian Rotary Encoder

Pengujian pada *rotary encoder* dilakukan dengan tujuan untuk mengetahui tingkat kelinieran dari *rotary encoder* dalam pembacaan putaran motor BLDC. Hasil pengujian dapat dilihat dalam Gambar 4.



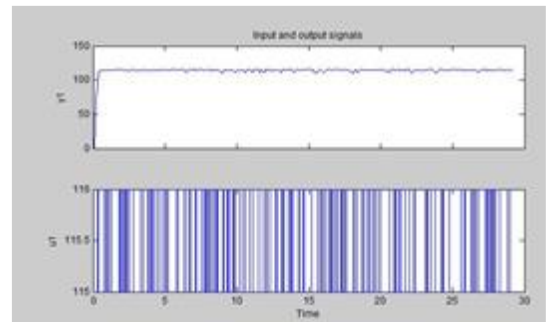
Gambar 4. Perubahan kecepatan motor terhadap *duty cycle*

Dari Gambar 4 dapat diketahui bahwa pembacaan *rotary encoder* mulai linear pada *duty cycle* 5%.

E. Penentuan Fungsi Alih Motor Brushless DC

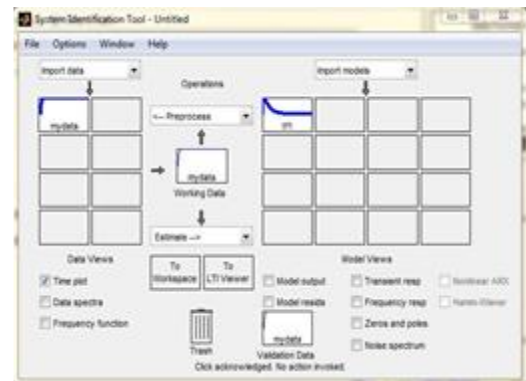
Fungsi alih dari motor *brushless* DC didapatkan melalui identifikasi dengan cara membangkitkan sinyal *Pseudo Random Binary Sequence* (PRBS), dengan langkah sebagai berikut :

1. Mencari nilai yang linier dari hasil kecepatan motor terhadap *duty cycle* PWM.
2. Memasukkan nilai batas atas dan bawah berdasarkan nilai yang linier untuk membangkitkan sinyal PRBS.
3. Sinyal PRBS yang telah dibangkitkan kemudian digunakan sebagai masukan motor BLDC.
4. Setelah didapatkan data sinyal PRBS dan data kecepatan motor BLDC (lihat Gambar 5), selanjutnya adalah melakukan identifikasi dengan menggunakan software MATLAB.



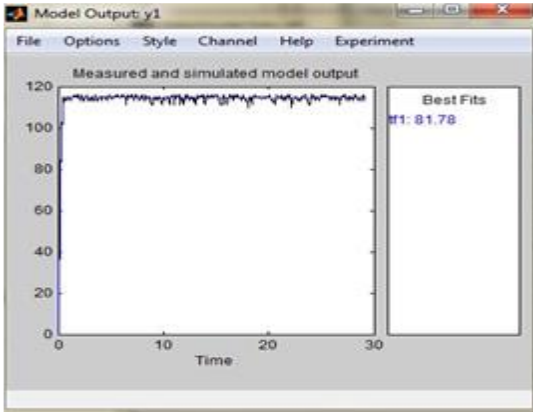
Gambar 5. Output sinyal PRBS

5. Dengan menggunakan sintaks *ident* pada command window pada MATLAB, data sinyal PRBS dan data kecepatan motor yang telah disimpan kemudian di *import* pada blok *System Identification Toolbox*. Tampilan blok *System Identification Toolbox* dapat dilihat dalam Gambar 6.



Gambar 6. Tampilan aplikasi *ident* di software MATLAB

6. Setelah melakukan beberapa estimasi model berdasarkan data yang telah di-import, didapatkan fungsi alih motor dengan *best fits* sebesar 81,78. Output motor BLDC yang telah diidentifikasi dapat dilihat dalam Gambar 7.

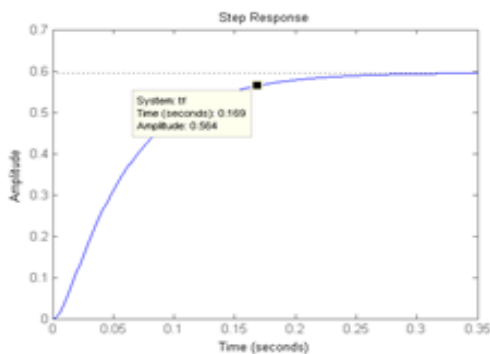


Gambar 7. Hasil estimasi model

7. Dari hasil identifikasi, fungsi alih motor yang didapat adalah

$$\frac{Y(s)}{U(s)} = \frac{1182}{s^2 + 125,3s + 1985} \quad (1)$$

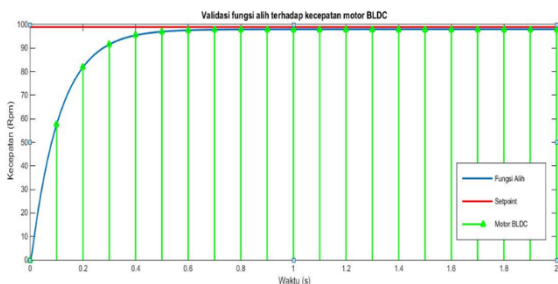
8. Dengan memberikan masukan unit step pada program MATLAB didapatkan output dengan didapatkan nilai *settling time* sebesar 0,169 detik dengan *output* maksimal berada di 0,594. Output motor BLDC yang telah diidentifikasi dapat dilihat pada Gambar 8.



Gambar 8. Output fungsi alih motor BLDC dengan masukan unit step

F. Validasi Fungsi Alih Motor Brushless DC

Validasi fungsi alih motor dilakukan dengan cara membandingkan *output* motor BLDC yang didapatkan dari identifikasi dan *output* kecepatan motor BLDC yang didapatkan dari pembacaan *rotary encoder* dengan memberikan masukan pulsa unit step. Perbandingan kedua output dapat dilihat pada gambar 9.

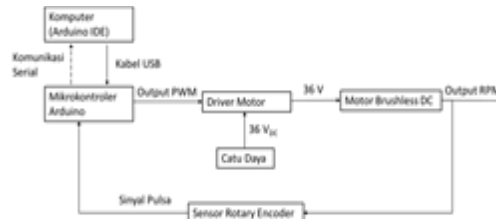


Gambar 9. Validasi fungsi alih dengan output motor BLDC

Dari Gambar 9 dapat dilihat bahwa output motor BLDC yang telah didapat dari proses identifikasi hampir menyerupai output kecepatan motor BLDC. Oleh karena itu, fungsi alih yang telah didapatkan dianggap dapat mewakili pemodelan *plant* motor BLDC.

G. Perancangan Perangkat Keras

Skema pembuatan sistem pengontrolan kecepatan sepeda listrik dapat dilihat dalam Gambar 10.

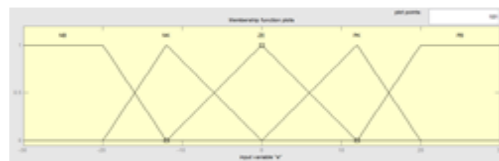


Gambar 10. Skema perangkat keras

H. Perancangan Kontrol Logika Fuzzy

Algoritma fuzzy dirancang dengan menggunakan 2 buah masukan, yaitu *error* dan *delta error* dari kecepatan. *Error* adalah selisih antara *setpoint* dan *output* yang dibaca oleh sensor *rotary encoder*. *Error* berfungsi untuk mengetahui seberapa besar deviasi yang terjadi pada sistem. *Delta error* berfungsi untuk mengetahui tingkat kesalahan *error* tersebut, yaitu seberapa besar perubahan *error* sekarang terhadap *error* sebelumnya.

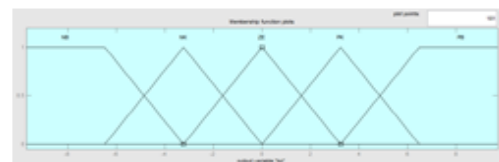
Dalam perancangan algoritma fuzzy dibuatlah sebuah fungsi keanggotaan untuk masing-masing input dan output. Setiap keanggotaan dideskripsikan dalam bahasa linguistik seperti NB (Negatif Besar), NK (Negatif Kecil), ZE (Zero/ Nol), PK (Positif Kecil), dan PB (Positif Besar). *Range* nilai digunakan negatif hingga positif agar parameter *Kp* dan *Ki* mampu fluktuasi mengikuti setpoint. Untuk fungsi keanggotaan *error* dan *delta error* ditunjukkan dalam Gambar 11 dan 12, serta fungsi keanggotaan *Kp* dan *Ki* ditunjukkan dalam Gambar 13 dan 14.



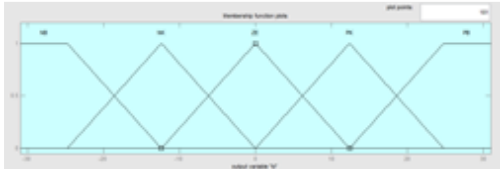
Gambar 11 Membership function input untuk error



Gambar 12 Membership function input untuk turunan error



Gambar 13 Membership function output untuk Kp



Gambar 14 Membership function output untuk Ki

Secara umum, hubungan antara *error* (e) dan *delta error* (de) terhadap nilai Kp dan Ki dijabarkan sebagai berikut:

1. Saat e relatif besar, maka kontroler akan memperbesar nilai Kp dan membuat nilai Ki sama dengan 0.
2. Saat e dan de relatif sesuai maka kontroler akan memperkecil nilai Kp agar mengurangi overshoot. Nilai Ki tidak berubah.
3. Saat e dan de nilainya sangat kecil maka kontroler akan memperbesar nilai Kp dan Ki.

Dengan acuan yang telah dijabarkan, diperoleh tabel aturan *fuzzy* untuk Kp dan Ki dalam Tabel 1 dan Tabel 2.

TABEL 1
ATURAN FUZZY UNTUK Kp

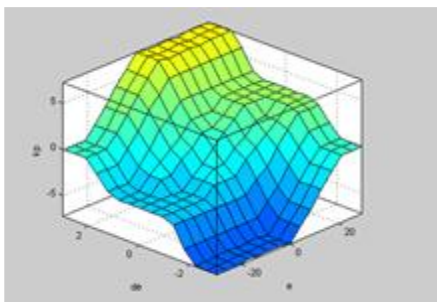
dE \ E	NB	NK	Z	PK	PB
NB	NB	NK	NK	NK	Z
NK	NB	NK	NK	Z	PK
Z	NB	NK	Z	PK	PB
PK	NK	Z	PK	PK	PB
PB	Z	PK	PK	PK	PB

TABEL 2
ATURAN FUZZY UNTUK Ki

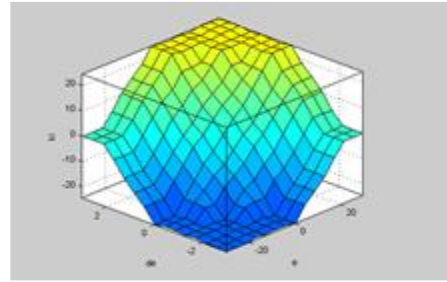
dE \ E	NB	NK	Z	PK	PB
NB	NB	NK	NK	NK	NB
NK	NB	NK	NK	Z	NB
Z	NB	NK	Z	PK	NB
PK	NK	Z	PK	PK	NK
PB	Z	PK	PK	PK	Z

Dari Tabel 1 dan Tabel 2 dapat digambarkan ruang solusi *fuzzy* untuk aturan *fuzzy* untuk Kp dalam Gambar 15, dan ruang solusi *fuzzy* untuk aturan *fuzzy* untuk Ki dalam Gambar 16.

Setelah didapatkan fungsi keanggotaan dan aturan *fuzzy*, dapat ditentukan metode untuk inferensi dan defuzzifikasi dari sistem kontrol *fuzzy* tersebut. Pada penelitian ini, metode inferensi yang digunakan adalah metode Min - Max, sedangkan untuk defuzzifikasi digunakan metode *Weighted Average*.



Gambar 15 Ruang solusi fuzzy untuk Kp



Gambar 16 Ruang solusi fuzzy untuk Ki

I. Perancangan Kontroler Proporsional Integral

Dalam kawasan waktu kontroler PI dapat dinotasikan dengan persamaan berikut:

$$c(t) = Kp(e(t) + \frac{1}{Ti} \int_0^t e(t)dt) \tag{2}$$

dengan c(t) adalah keluaran kontroler, Kp adalah gain proporsional, Ti adalah waktu konstanta Integral atau *reset time* dan e(t) adalah *error* yang terjadi. Persamaan (2) dapat diubah menjadi kawasan frekuensi dengan Transformasi Laplace sehingga menjadi persamaan berikut:

$$C(s) = Kp \left(1 + \frac{1}{Ti s} \right) E(s) \tag{3}$$

Persamaan (3) belum bisa dimasukkan kedalam mikrokontroler. Maka dari itu persamaan kontinyu tersebut harus diubah kedalam bentuk diskrit melalui Transformasi Z. Dalam Transformasi Z dibutuhkan waktu sampling (Ts). Digunakan metode *Bilinear Transform* sehingga nilai notasi s pada Laplace setara dengan

$$s = \frac{2}{Ts} \left(\frac{1-z^{-1}}{1+z^{-1}} \right) \tag{4}$$

persamaan (3) disubstitusikan ke dalam persamaan (4) menjadi

$$C(z) = \left[Kp + \frac{KpTs(1+z^{-1})}{2Ti(1-z^{-1})} \right] E(z) \tag{5}$$

Berikutnya adalah memodifikasi persamaan agar dapat disederhanakan. Kedua ruas dikalikan dengan (1-z⁻¹).

$$C(z)(1-z^{-1}) = Kp E(z)(1-z^{-1}) + \frac{Kp Ts}{2Ti} E(z)(1+z^{-1}) \tag{6}$$

kemudian persamaan (6) disusun kembali menjadi *output* kontroler

$$C(z) = C(z)(z^{-1}) + Kp (E(z) - E(z)(z^{-1})) + \frac{Kp Ts}{2Ti} (E(z) + E(z)(z^{-1})) \tag{7}$$

persamaan (7) diubah kedalam persamaan beda sehingga didapatkan persamaan (8).

$$C(k) = C(k-1) + Kp (E(k) - E(k-1)) + \frac{Kp Ts}{2Ti} (E(k) + E(k-1)) \tag{8}$$

Dimana k-1 adalah kondisi sebelumnya. Persamaan diatas lalu dimasukkan kedalam program pada mikrokontroler.

Pada kontroler PI, dibutuhkan sebuah parameter awal yang berfungsi untuk memperbaiki *settling time* dan *steady-state error* pada *output*. Untuk menentukan parameter tersebut digunakan metode *Symmetrical Optimum*. Tuning parameter dengan metode *Symmetrical Optimum* pertama dikemukakan oleh Kessler pada 1958. Metode ini memaksimalkan *phase margin* dari sistem kontrol dan mengarahkan ke fasa yang simetris dan karakteristik amplitudo [4]. Metode

tersebut dipilih karena sistem memiliki keunggulan berupa kemandapan pada parameter *phase margin*, *gain margin*, dan dinamika sistem itu sendiri [5]

Untuk menentukan parameter dengan metode *Symmetrical Optimum*, pertama – tama, persamaan fungsi alih dimodifikasi menjadi *open loop system* seperti berikut:

$$F_{ol}(s) = \underbrace{Gc}_{\text{Kontroler PI}} \left(\frac{T_i s + 1}{T_i s} \right) \underbrace{\frac{G_m}{T_m s + 1}}_{\text{Plant}} \frac{1}{T_{mn} s} \tag{9}$$

Untuk membentuk persamaan plant (9) maka fungsi alih motor perlu diubah, di mana *Gm* adalah nilai *gain* dari motor yang telah dimodifikasi dan *Tm* adalah *time constant* motor. Oleh Karena itu, persamaan (1) harus diubah terlebih dahulu ke bentuk persamaan (10).

$$F(s) = \frac{G_{cw}}{(T_1 s + 1)(T_2 s + 1)} \tag{10}$$

Mulanya bagian penyebut diakarkan menjadi seperti persamaan (11).

$$F(s) = \frac{0,596}{(\frac{1}{18,6}s + 1)(\frac{1}{106,7}s + 1)} \tag{11}$$

$$T_1 = \frac{1}{18,6} = 0,054$$

$$T_2 = \frac{1}{106,7} = 0,00937$$

$$T_2 < T_1$$

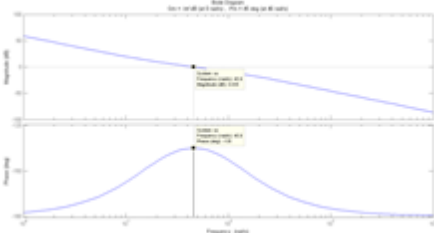
Maka nilai *Gcw* adalah 0,596 dan *Tcw* = *T*₁ = 0,054 . Nilai *Tmn* ditetapkan dengan *Tmn* = 1. Dengan menentukan nilai faktor redaman (*D*) maka dapat ditentukan nilai *Kp* dan *Ti* melalui persamaan (12).

$$K_p = \frac{1}{a} \frac{T_{mn}}{G_{cw} T_{cw}}$$

$$T_i = a^2 T_{cw} \quad a > 1$$

$$a = 2D + 1 \tag{12}$$

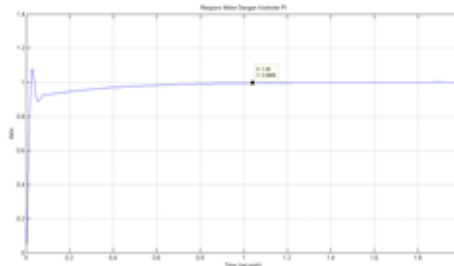
Nilai-nilai dari variabel *Tcw*, *Gcw*, *Tmn* dan *D* diolah menggunakan MATLAB dan hasilnya ditampilkan dalam *bodeplot* untuk mengetahui kestabilan sistem.



Gambar 17. Hasil *Bodeplot* dengan metode *Symmetrical Optimum*

Pada Gambar 17 dapat diketahui dari *bodeplot* tersebut bahwa sistem stabil, dengan *frequency crossover* pada margin terletak pada fasa maksimumnya.

Dengan nilai *D* = 0,707 didapatkan nilai *Kp* = 12,938 dan *Ti* = 0,3133. Sehingga nilai *Kp* = 12,938 dan *Ki* = 41,298. Nilai *Kp* dan *Ki* tersebut disimulasikan pada *toolbox* Simulink untuk mengetahui *output* sistem tanpa beban dengan kontroler PI. Jika masukan berupa sinyal unit *step* maka didapatkan grafik yang dapat dilihat dalam Gambar 18, dengan besarnya nilai *settling time* setelah diberi kontroler adalah 1,04 s dan mampu mencapai *setpoint*.



Gambar 18. Simulasi output sistem tanpa beban dengan kontroler PI

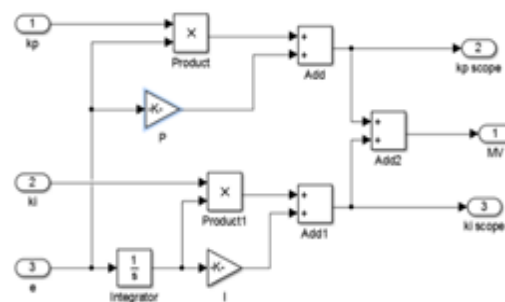
J. Perancangan Kontroler self-tuning Parameter PI

Setelah melakukan perancangan kontrol logika *fuzzy* dan kontroler PI, maka kedua sistem tersebut akan digabungkan. Secara umum, sistem kontrol akan bekerja dengan kontrol logika *fuzzy* berfungsi untuk menala parameter PI secara *real time*, dan kontroler PI berfungsi untuk mengolah sinyal error, dan memberikan sinyal kontrol. Diagram blok sistem secara keseluruhan dapat dilihat dalam Gambar 19.



Gambar 19. Diagram blok perancangan kontroler self-tuning parameter PI

Agar kontroler PI dapat bekerja secara optimal, maka diperlukan perancangan kontroler PI yang bobot parameter PI-nya dapat diubah, dan dapat bersinkronisasi dengan output parameter PI hasil penalaan kontrol logika Fuzzy. Diagram blok rancangan subsistem kontroler PI tersebut dapat dilihat dalam Gambar 20.



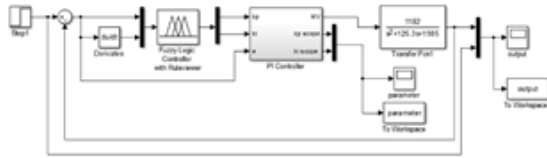
Gambar 20. Rancangan blok subsistem kontroler PI

III. PENGUJIAN SISTEM

A. Simulasi Sistem Tanpa Beban

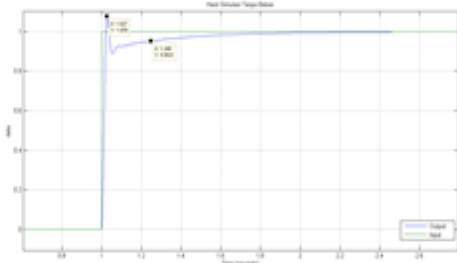
Simulasi dilakukan dengan memberikan setpoint berupa *unit step* yang berfungsi sebagai sinyal tes untuk mengetahui performansi sistem. Kemudian, setpoint dimasukkan ke dalam blok penjumlahan (*summing*) dan dikurangkan terhadap *output* sistem, untuk mendapatkan error.

Error (selisih antara setpoint dan umpan balik) yang telah didapat kemudian dimasukkan kedalam blok fuzzy. Error dibagi menjadi dua di mana salah satu masukan diberi blok fungsi diferensial, yang berfungsi untuk mengubah error menjadi delta error (selisih antara error saat ini dan error sebelumnya). Diagram blok simulasi pada SIMULINK dapat dilihat dalam Gambar 21.



Gambar 21. Skema keseluruhan blok simulasi dengan MATLAB SIMULINK

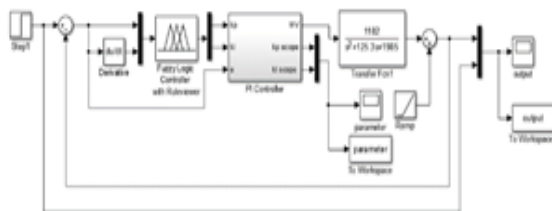
Hasil simulasi dari output sistem tanpa beban ditunjukkan dalam Gambar 22, dan diketahui bahwa sistem stabil, memiliki settling time (t_s) sebesar 1,248 s, memiliki maximum overshoot (M_p) sebesar 7,8%, dan tidak memiliki error steady-state.



Gambar 22. Hasil simulasi output sistem tanpa beban

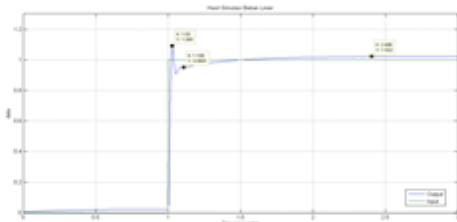
B. Simulasi Sistem Berbeban Linier

Gambar 23 adalah blok simulasi secara keseluruhan, setelah ditambahi beban linier berupa fungsi bidang miring (ramp) dengan nilai 0,577.



Gambar 23. Skema blok simulasi berbeban linier dengan MATLAB SIMULINK

Hasil simulasi dari output sistem tanpa beban ditunjukkan dalam Gambar 24, dan diketahui bahwa sistem stabil, memiliki settling time (t_s) sebesar 1,109 s, memiliki maximum overshoot (M_p) sebesar 9,3%, dan memiliki error steady-state sebesar 2,2%.

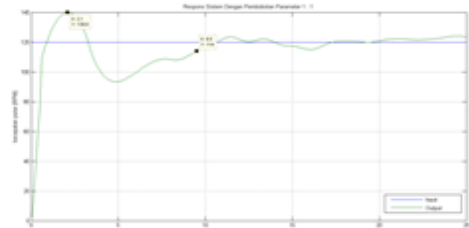


Gambar 24. Hasil simulasi output sistem berbeban linier

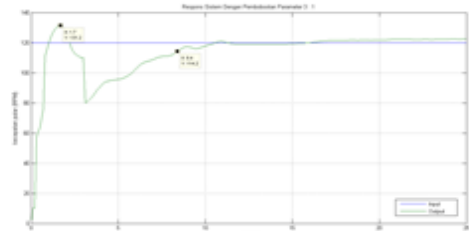
C. Pengujian Perbandingan Bobot Penalaan Parameter Sistem

Pengujian dilakukan dengan tujuan untuk mencari perbandingan parameter hasil penalaan dari kontrol logika fuzzy dan parameter tetap (kontroler PI), dengan harapan kerja sistem menjadi optimal.

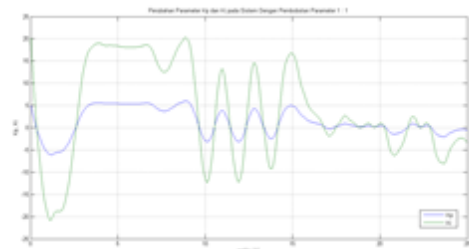
Penentuan perbandingan dilakukan dengan proses trial and error, dengan mencoba berbagai nilai perbandingan ke dalam sistem yang dijalankan tanpa beban dengan setpoint sebesar 120 rpm. Perbandingan yang digunakan dalam pengujian ini adalah 1 : 1 dan 3 : 1, dengan ketentuan parameter hasil penalaan logika fuzzy dan parameter tetap. Gambar 25 dan 26 merupakan output sistem dengan pembobotan parameter 1 : 1 dan 3 : 1, serta Gambar 27 dan 28 merupakan perubahan parameter Kp dan Ki yang didapatkan dari pengujian.



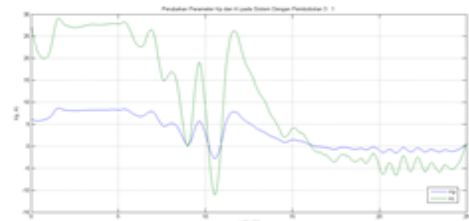
Gambar 25. Output sistem dengan pembobotan parameter 1 : 1



Gambar 26. Output sistem dengan pembobotan parameter 3 : 1



Gambar 27. Perubahan parameter Kp dan Ki pada sistem dengan perbandingan bobot 1 : 1

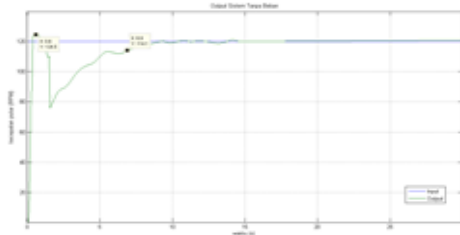


Gambar 28. Perubahan parameter Kp dan Ki pada sistem dengan perbandingan bobot 3 : 1

Karena nilai maximum overshoot (M_p) terletak di dalam batas toleransi sebesar 10%, maka perbandingan parameter 3 : 1 tersebut digunakan dalam implementasi sistem secara keseluruhan.

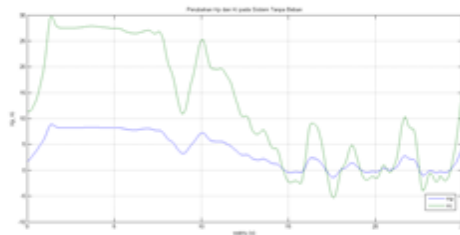
D. Pengujian Pada Alat

Pengujian ini dilakukan 2 (dua) kali, yaitu pada saat sistem berjalan tanpa beban, dan pada saat sistem berjalan dengan beban, dengan *setpoint* sebesar 120 rpm. Hasil pengujian ditunjukkan pada Gambar 29, 30, 31, dan 32.

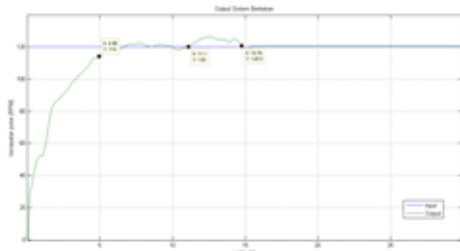


Gambar 29. Output sistem tanpa beban

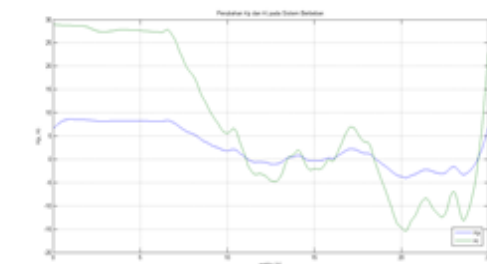
Dalam gambar 29, dapat diketahui bahwa sistem stabil, memiliki *settling time* (t_s) sebesar 6,9 s, memiliki *maximum overshoot* (M_p) sebesar 9,3%, dan tidak memiliki *error steady-state*. Dalam gambar 30, dapat diketahui bahwa hasil penalaan K_p dan K_i berubah relatif terhadap waktu, yaitu dengan menuju pada nilai positif maksimal pada awal sistem berjalan, kemudian menurun ke nilai negatif untuk K_p dan nilai positif kecil untuk K_i ketika hampir mencapai *settling-time*, dan stabil setelah sistem mencapai keadaan *steady state*, pada waktu 15 s.



Gambar 30. Perubahan parameter K_p dan K_i pada sistem tanpa beban



Gambar 31. Output sistem berbeban



Gambar 32. Perubahan parameter K_p dan K_i pada sistem berbeban

Dalam Gambar 31, dapat diketahui bahwa sistem stabil, memiliki *settling time* (t_s) sebesar 4,9 s, tidak memiliki *maximum overshoot* (M_p) dan tidak memiliki

error steady-state. Dalam gambar 32, dapat diketahui bahwa hasil penalaan K_p dan K_i berubah relatif terhadap waktu, yaitu dengan menuju pada nilai positif maksimal pada awal sistem berjalan, kemudian menurun ke nilai negatif ketika sistem telah mencapai *settling-time*, dan stabil setelah sistem mencapai keadaan *steady state*, pada waktu 11 s, dan kembali menurun pada waktu 18 s dikarenakan adanya gangguan.

IV. PENUTUP

A. Simpulan

1. Berdasarkan data *output* sistem yang diperoleh dari pengujian dengan menggunakan sinyal *Pseudo-Random Binary Sequence* (PRBS), didapatkan nilai fungsi alih

$$F(s) = \frac{1182}{s^2 + 125,3s + 1985}$$

dengan nilai *best-fit* sebesar 81,78.

2. Berdasarkan metode penalaan *Symmetrical Optimum* didapatkan batas penalaan parameter kontroler PI dengan nilai K_p di antar -12,938 sampai dengan 12,938 dan nilai K_i di antara -41,298 sampai dengan 41,298. Berdasarkan pengujian perbandingan bobot, didapatkan perbandingan untuk optimasi kerja sistem sebesar 3 : 1 untuk parameter hasil penalaan Kontrol Logika Fuzzy terhadap parameter kontroler PI.

3. Berdasarkan hasil implementasi tanpa beban, didapatkan *output* sistem yang memiliki *settling time* (t_s) sebesar 6,9 s, memiliki *maximum overshoot* (M_p) sebesar 9,3%, dan tidak memiliki *error steady-state*. Berdasarkan hasil implementasi berbeban, didapatkan *output* sistem yang memiliki *settling time* (t_s) sebesar 4,9 s, tidak memiliki *maximum overshoot* (M_p), dan tidak memiliki *error steady-state*.

B. Saran

Saran yang dapat diberikan untuk penelitian selanjutnya adalah dengan menggunakan penelitian yang telah dilakukan untuk diimplementasikan ke dalam kendaraan elektrik yang lebih besar dengan motor brushless DC sebagai aktuatur sistem, memperhitungkan koefisien gesek sebagai parameter pembebanan sistem, dan menggunakan rangkaian elektronik yang lebih baik untuk efisiensi dan performansi sistem yang lebih baik.

REFERENSI

- [1] Hamdi, I. T. 2015. Rancang Bangun Three Phase Six Step PWM Inverter sebagai Pedal Assisted System (PAS) Sepeda Listrik. *Skripsi*. Tidak dipublikasikan. Malang: Universitas Brawijaya.
- [2] Kristiyono, R. 2015. Sistem Kendali Kecepatan Motor BLDC Menggunakan Algoritma Hybrid PID Fuzzy. University Research Colloquium 2015.
- [3] Shyam, A. 2013. *A Comparative Study on the Speed Response of BLDC Motor Using Conventional PI Controller, Anti-windup PI Controller and Fuzzy Controller*, 2013 International Conference on Control Communication and Computing (ICCC).
- [4] Putra, D. A. 2015. Penerapan Kontroler Self Tuning Parameter PI dengan Metode Logika Fuzzy Pada Mobile Robot. *Skripsi*. Tidak dipublikasikan. Malang: Universitas Brawijaya.
- [5] Barbosa, A. L. S., Júnior, G. A., Barros P. R. 2014. *Symmetrical Optimum Based PI Control Redesign*. 2014 IEEE Conference on Control Applications (CCA).