

# Kendali Posisi Motor DC Dengan Menggunakan *Pole Placement*

Yogi Ramadhan

Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia

Jalan Kaliurang Km.14,5 Sleman, Yogyakarta 55501

Telp. (0274) 895007, 895287 Faks.(0274) 89 500 7 Ext. 131

yogiramadhan@gmail.com

## ABSTRACT

Motor DC merupakan salah satu jenis penggerak yang banyak digunakan di industri. Oleh karena itu, Motor DC sangat diharapkan memiliki karakteristik yang baik serta efisiensi yang tinggi. Penelitian ini merancang pengendali *state feedback* untuk mengatur posisi Motor DC serta memperoleh parameter nilai *close loop pole* terbaik. Simulasi Motor DC dengan menggunakan *Pole Placement* dibuat dengan MATLAB. Berdasarkan hasil pengujian terhadap berbagai nilai *close loop pole* dimana saat *close loop pole* =  $[-10 + j10 \quad -10 \quad -10 - j10]$  respon sistem memiliki *overshoot* 0%, *settling time* 0,4459 detik dan *rise time* 0,2558 detik.

**Kata Kunci;** Motor DC, Kendali State Feedback, Pole Placement.

## I. PENDAHULUAN

### A. Latar Belakang

Perkembangan dan kemajuan teknologi memberikan pengaruh yang cukup besar dalam kehidupan sehari-hari. Berbagai macam perangkat elektronik terus berkembang menjadi peralatan yang canggih dan mudah digunakan. Motor DC merupakan salah satu jenis penggerak yang banyak digunakan di industri. Pemakaian motor listrik telah meliputi segala bidang mulai dari peralatan rumah tangga, robot, pesawat ruang angkasa, komputer, sarana transportasi bahkan peralatan audio video pun memerlukan motor listrik untuk beroperasi. Dewasa ini dengan semakin berkembangnya teknologi tentang motor diharapkan dapat diperoleh motor yang memiliki karakteristik yang baik serta efisiensi yang tinggi.

Permasalahan dalam menentukan posisi motor DC adalah berapa banyak energi listrik yang harus diberikan agar motor DC berputar pada posisi yang diinginkan. Hal itu dapat dicapai dengan melakukan pengaturan-pengaturan pada bagian-bagian motor sehingga didapatkan unjuk kerja yang terbaik. Berdasarkan pada uraian diatas, permasalahan dalam menentukan posisi motor DC diselesaikan dengan menggunakan pengendali *state feedback*.

## II. TINJAUAN PUSTAKA

### A. Tinjauan Pustaka

Penelitian sebelumnya berjudul "Simulasi Pengaturan Kecepatan Motor DC Berbasis PID Menggunakan LabVIEW" [1]. Penelitian ini merancang pengendali PID untuk mengatur kecepatan Motor DC. Simulasi Motor DC dengan sistem pengendali PID dan dibuat dengan LabVIEW. *Interface* simulasi pada *front panel* menampilkan grafik tegangan dan

kecepatan. Terdapat input *set point* kecepatan dalam RPM. Berdasarkan hasil percobaan terhadap berbagai parameter  $K_p, T_i$  dan  $T_d$  dimana saat  $K_p = 0,001, T_i = 0,001$  dan  $T_d = 0,001$  respon sistem memiliki *rise time* 0,348 detik, *overshoot* 0,354% dan *settling time* 0,570 detik.

Penelitian selanjutnya berjudul "Aplikasi *Hardware In Loop* Pada Pengendalian Motor DC Menggunakan MATLAB" [2]. *Hardware In the Loop (HIL)* sering digunakan untuk mengidentifikasi suatu objek atau suatu sistem secara virtual yang berguna secara langsung untuk mengurangi kesalahan-kesalahan yang mengakibatkan kerusakan sistem yang akan dijalankan, memberikan perspektif baru dalam pengembangan suatu sistem dan juga sebagai sarana pembelajaran dalam bidang kendali dan instrumentasi. Penelitian yang dilakukan mengambil tema pengendalian kecepatan Motor DC dengan metode PID dengan menggabungkan Mikrokontroler Arduino Uno sebagai objek fisik dan MATLAB sebagai objek virtual untuk model Motor DC yang akan dikendalikan. Pengujian dilakukan untuk mengetahui respon sistem pada model motor DC dan tuning PID dengan cara mengirimkan instruksi program ke dalam plant model di MATLAB serta pengujian dengan variasi *set point* dengan parameter kendali PID yang telah optimal. Berdasarkan hasil yang diperoleh, maka konsep *Hardware In The Loop* telah berjalan dengan baik dan sistem telah sinkron antara MATLAB dan Arduino Uno dan juga mendapatkan parameter PID terbaik dengan  $K_p = 26, K_i = 1$  dan  $K_d = 0.30$  kemudian *rise time* dibawah 0.5 detik, *overshoot* dibawah 10% dan *steady state error* dibawah 5% untuk diterapkan pada pengendalian Motor DC. Kemudian konsep dan teori kendali PID juga dapat dijalankan dengan menggunakan pemrograman pada Mikrokontroler Arduino Uno.

Penelitian lainnya berjudul "Aplikasi Fuzzy Logic Sebagai Kontrol Posisi Motor DC" [3]. Pada artikel ini membahas kendali posisi dari suatu motor DC menggunakan Logika Fuzzy, Kendali fuzzy logic diimplementasikan pada komputer dan algoritma kendali dirancang menggunakan program LabVIEW dan melihat kondisi yang riil dari posisi kendali motor DC. Pengaturan posisi gerak motor dilakukan dengan mengatur tegangan motor dan menggunakan metode *Pulse Width Modulation (PWM)*. *Feedback* sistemnya adalah sinyal frekuensi dari tachometer dan sinyal frekuensi ini diubah menjadi tegangan untuk diinputkan pada komputer melalui ADC. Sistem *fuzzy logic* mempunyai 2 *crisp input* yaitu *error* dan perubahan *error* posisi motor dan mempunyai 1 *crisp output*

yaitu perubahan tegangan. Metode defuzzifikasi yang digunakan adalah metode *center of area*. Jumlah label dari *membership function* 7 label. Respon sistem ditampilkan dalam bentuk grafik sudut posisi motor terhadap waktu. Hasil pengujian menunjukkan *membership function* dengan bentuk segitiga dan defuzzifikasi *center of area* dapat memberikan hasil yang baik. Pada percobaan membandingkan kendali PID dengan FLC dan hasilnya memperlihatkan bahwa tanggapan motor DC dengan *Control Fuzzy Logic* (FLC) mampu memberikan performa yang baik.

Kemudian penelitian yang kini dilakukan dengan menggunakan blok *simulink* MATLAB dengan persamaan model matematis motor DC dengan *state feedback* sebagai pengendali.

### III. PERANCANGAN SISTEM

#### A. Perancangan Model Motor DC

Model motor DC yang digunakan adalah representasi *state space*. Input sistem adalah tegangan yang diberikan pada motor DC yaitu  $V$ . Sedangkan output sistem berupa posisi sudut motor DC yaitu  $\theta$ .

Apabila dianalisis dari sudut pandang mekanik motor DC berdasarkan hukum Newton II tentang rotasi yang menyatakan bahwa jumlah torsi merupakan perkalian antara beban inersia dengan percepatan sudut, maka didapatkan persamaan 3.1.

$$K_m \cdot i - K_f \frac{d\theta}{dt} = J \frac{d^2\theta}{dt^2} \quad (3.1)$$

$$\frac{d^2\theta}{dt^2} = \frac{K_m}{J} \cdot i - \frac{K_f}{J} \cdot \frac{d\theta}{dt}$$

dimana  $K_m$  merupakan konstanta *armature* motor DC,  $K_f \cdot \frac{d\theta}{dt}$  merupakan gaya gesek yang berlaku pada motor DC, dan  $J$  merupakan beban inersia motor DC.

Jika dipandang dari sudut pandang elektrik motor DC berdasarkan hukum Kirchoff tentang tegangan (KVL) yang menyatakan bahwa jumlah beda potensial suatu rangkaian tertutup adalah sama dengan nol, maka didapatkan persamaan berikut 3.2.

$$v = i \cdot R + L \frac{di}{dt} + K_b \cdot \frac{d\theta}{dt} \quad (3.2)$$

$$\frac{di}{dt} = \frac{v}{L} - i \cdot \frac{R}{L} - \frac{K_b}{L} \cdot \frac{d\theta}{dt}$$

dimana  $K_b \cdot \frac{d\theta}{dt}$  merupakan gaya gerak listrik induksi pada motor DC dan  $K_b$  merupakan konstanta gaya gerak listrik motor DC.

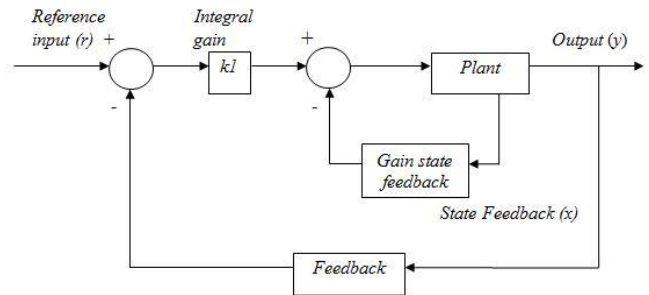
Variabel-variabel keadaan sistem yang digunakan dalam model ini adalah posisi sudut  $\theta$ , kecepatan sudut  $\omega$  yang merupakan turunan pertama dari posisi sudut  $\theta$  dan arus  $i$ . Berdasarkan pemilihan variabel-variabel tersebut dan persamaan 3.1 dan 3.2, maka model representasi *state space* motor DC dapat dituliskan dalam persamaan 3.3.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{K_f}{J} & \frac{K_m}{J} \\ 0 & -\frac{K_b}{J} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \theta \\ \omega \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} v$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \omega \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} v \quad (3.3)$$

#### B. Perancangan Pengendali State Feedback Dengan Metode Pole Placement

Secara umum, perancangan pengendali yang digunakan dalam penelitian ini digambarkan dalam gambar 3.1.



Gambar 3.1 Blok diagram sistem

Blok diagram diatas menjelaskan bahwa komponen pengendali yang digunakan terdiri dari *integral gain* dan *gain state feedback*. *Plant* yang digunakan adalah model motor DC pada persamaan 2.9. Keluaran dari *feedback* yang berupa posisi sudut  $\theta$  motor DC sebelumnya dibandingkan dengan nilai *reference input* oleh pengendali yang digunakan dan diartikan sebagai nilai *error*. Kendali *state feedback* dalam penelitian ini berfungsi untuk mengatur posisi sudut  $\theta$  *output* (radian) *plant* sesuai dengan nilai *reference* yang diinginkan.

Respon sistem pada penelitian ini dapat diamati dari beberapa parameter keluaran *software* MATLAB *Simulink* yang menjadi acuan dalam proses *tuning*. Jika respon sistem sudah sesuai dengan target penelitian, maka proses *tuning* dihentikan dan diuji dengan beberapa variasi nilai *reference*.

Berdasarkan model motor DC pada persamaan 3.3, maka

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{K_f}{J} & \frac{K_m}{J} \\ 0 & -\frac{K_b}{J} & -\frac{R}{L} \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix}$$

Untuk dapat mengaplikasikan *state feedback controller* dengan metode *pole placement*, maka matriks A dan B diatas harus berada dalam bentuk *controllable*. Untuk mengecek apakah sistem tersebut *controllable* atau tidak maka dapat dilakukan dengan mengecek *rank* matriks M yaitu:

$$M = [B : AB : A^2B]$$

Jika model yang digunakan sudah berada dalam bentuk *controllable*, maka selanjutnya mencari karakteristik persamaan dari sistem. Karakteristik persamaan model yang diinginkan dibentuk dari *close loop pole* yang diinginkan. *Close loop pole* yang diinginkan adalah  $s_1 = -10$ ,  $s_2 = -7$ , dan  $s_3 = -5$ . Respon sistem yang diinginkan adalah eksponensial dan tidak memiliki nilai *overshoot*, sehingga dipilih *close loop pole* yang berada pada sumbu *real* dan sumbu kompleks.

Berdasarkan pada pengendali *state feedback* yang digunakan maka nilai  $K$  dapat dicari nilainya berdasarkan karakteristik persamaan model representasi *state space* motor

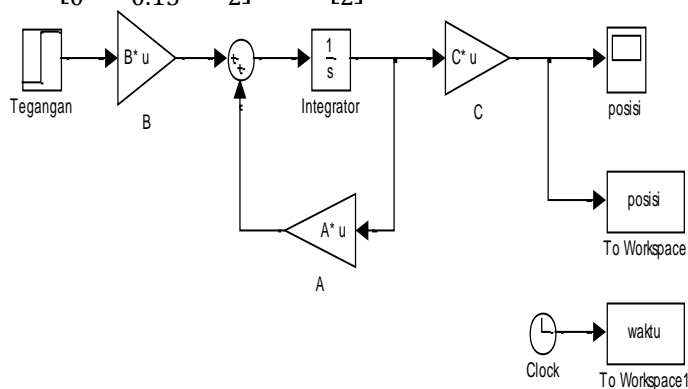
DC dan karakteristik persamaan model dari *close loop pole* yang diinginkan.

#### IV. PENGUJIAN DAN ANALISIS

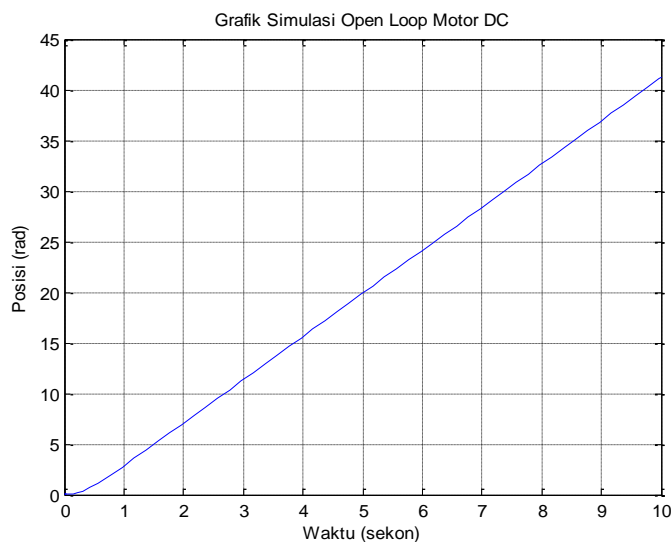
##### A. Pengujian Open Loop Motor DC

Pengujian simulasi *open loop* berfungsi untuk mengamati model motor DC apakah memiliki dinamik sama dengan motor DC yang sesungguhnya. Selain itu, pengujian simulasi *open loop* juga digunakan untuk mengamati respon motor DC terhadap *input*. Pengujian simulasi model motor DC dilakukan dengan memberikan *input* tegangan 12 volt. Pengamatan dilakukan pada posisi dan respon motor DC saat diberi *input* tegangan. Tabel parameter motor DC yang digunakan ditunjukkan pada tabel 2.1, model perancangan motor DC ditunjukkan pada gambar 4.1 dan *state* dari motor DC ditunjukkan pada persamaan (4.1)

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -10 & 7.5 \\ 0 & -0.15 & -2 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}, C = [1 \ 0 \ 0] \quad (4.1)$$



Gambar 4.1 Blok kendali *open loop* motor DC



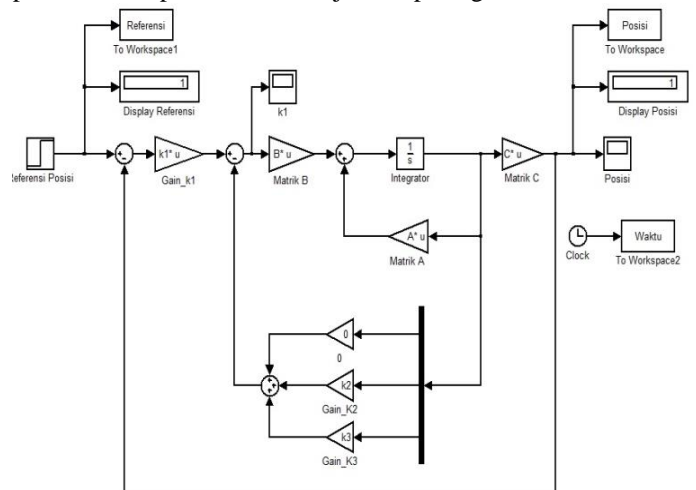
Gambar 4.2 Grafik hasil *open loop* simulasi motor DC

Hasil pengujian *open loop* motor DC diberikan pada gambar 4.2 dari hasil pengujian *open loop* menunjukkan bahwa saat motor DC diberikan tegangan 12 volt selama 10

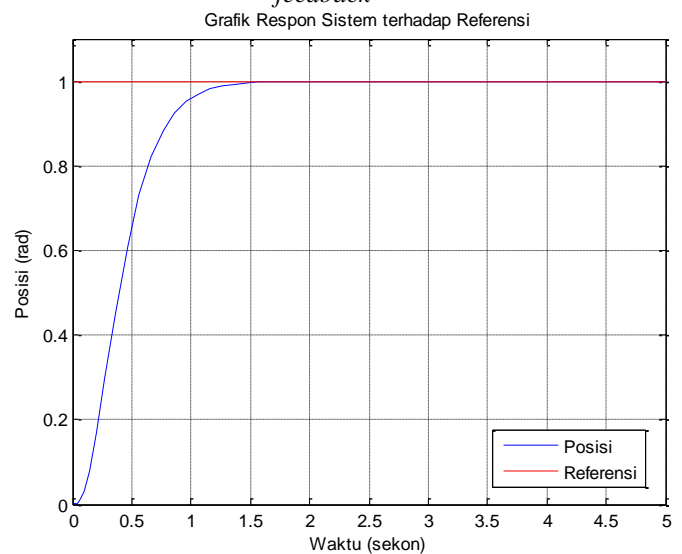
detik, motor DC dapat berputar dan mencapai posisi tertentu. Motor DC akan terus berputar terus menerus selama waktu yang diberikan. Semakin lama waktu yang diberikan untuk berputar, motor DC akan terus berputar dan menempuh posisi tertentu. Pengujian ini menunjukkan bahwa model motor DC dapat berjalan dengan semestinya sesuai yang ada di lapangan.

##### B. Pengujian Pengendali

Pengujian simulasi yang dilakukan adalah dengan menerapkan pengendali *state feedback* pada model motor DC. Input simulasi adalah sinyal referensi dan *output* simulasi adalah posisi motor DC. Input simulasi adalah unit *step* dengan nilai 0. Letak *close loop pole* yang dipilih adalah  $s = -10$ ,  $s = -7$  dan  $s = -5$ . Berdasarkan perhitungan dengan nilai *close loop pole* yang diinginkan, untuk menentukan nilai *gain state feedback* ( $K$ ) dengan menggunakan fungsi *Acker* pada MATLAB didapatkan hasil adalah  $K = [23.3333 \ 2.1833 \ 4]$ . Perancangan model simulasi ditunjukkan pada gambar 4.3 dan grafik hasil respon posisi terhadap referensi ditunjukkan pada gambar 4.4.



Gambar 4.3. Implementasi motor DC dengan pengendali *state feedback*



Gambar 4.4. Grafik hasil implementasi motor DC dengan pengendali *state feedback*

Hasil pengujian pada gambar 4.4 menunjukkan bahwa *output* posisi sistem dapat mencapai nilai referensi. Pengendali *state feedback* dapat mengendalikan posisi motor DC sehingga motor DC dapat mengikuti nilai referensi. Hasil pengujian juga menunjukkan bahwa peforma sistem tidak memiliki *overshoot* dan *rise time* dibawah 2 detik setelah nilai referensi diberikan.

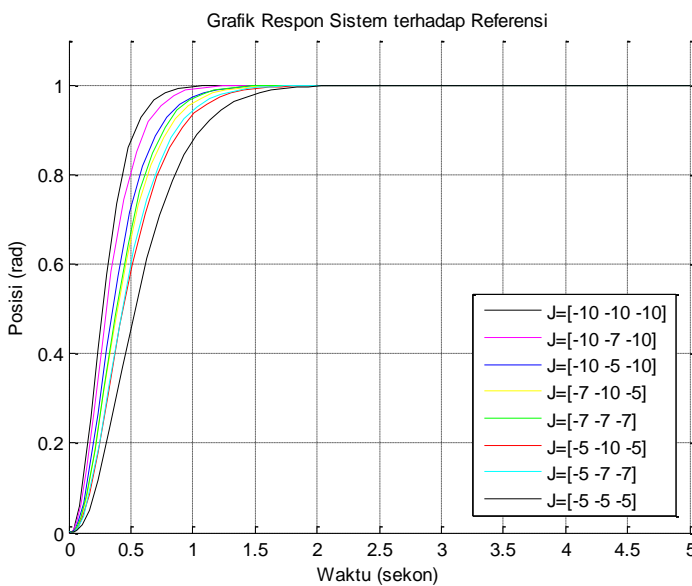
C. Pengujian Nilai Pole

1) Pengujian Pole Real

Pengujian ini berfungsi untuk melihat respon yang dihasilkan nilai *pole* pada sumbu *real*. Berikut adalah hasil pengujian ditunjukkan pada tabel 4.1 dan gambar 4.5.

Pole	Gain			Karakteristik Respon	
	$k_1$	$k_2$	$k_3$	Overshoot (%)	Settling Time
$[-10 \ -10 \ -10]$	66,6667	6,5167	8	0	0,7588
$[-10 \ -7 \ -10]$	46,6667	4,5167	6,5	0	0,8774
$[-10 \ -5 \ -10]$	33,3333	3,1833	5,5	0	1,0617
$[-7 \ -10 \ -5]$	23,3333	2,1833	4	0	1,1489
$[-7 \ -7 \ -7]$	22,8667	2,3167	3,5	0	1,0741
$[-5 \ -10 \ -5]$	16,6667	1,5167	3	0	1,2975
$[-5 \ -7 \ -7]$	16,3333	1,7833	2,5	0	1,2324
$[-5 \ -5 \ -5]$	8,3333	1,5167	0,5	0	1,5070

Tabel 4.1 Hasil pengujian sistem kendali *close loop pole real*



Gambar 4.5. Nilai step respon dengan *close loop pole (J)* yang seluruhnya *real*

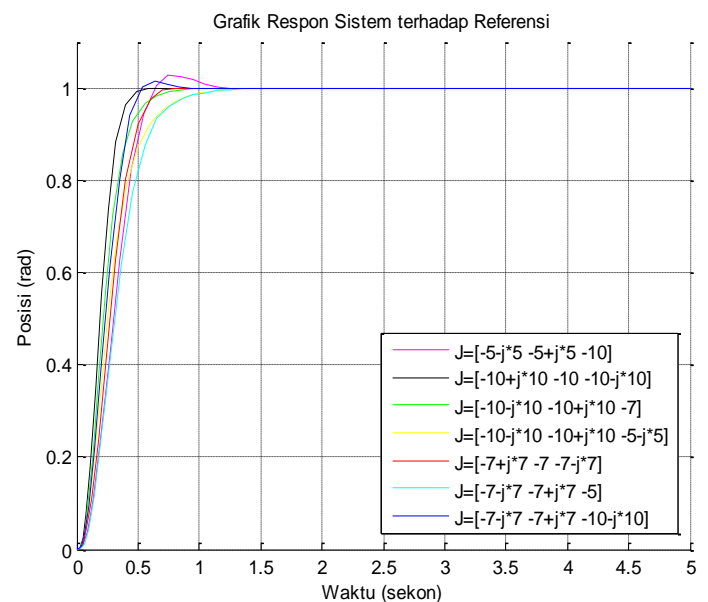
Hasil pengamatan pada tabel 4.1 dan gambar 4.5 menunjukkan bahwa, letak *pole real* negatif membuat respon dari sistem tidak memiliki *overshoot*. Respon sistem juga menunjukkan bahwa semakin negatif nilai *pole* yang diberikan, maka semakin cepat *rise time* sistem.

2) Pengujian Pole Real dan Pole Kompleks

Pengujian ini berfungsi untuk melihat respon yang dihasilkan oleh nilai *pole* pada sumbu *real* dan nilai *pole* kompleks. Berikut adalah hasil pengujian ditunjukkan pada tabel 4.2 dan gambar 4.6.

Pole	Gain			Karakteristik Respon	
	$k_1$	$k_2$	$k_3$	Overshoot (%)	Settling Time
$[-5 - j5 \ -5 + j5 \ -10]$	33,3333	3,1833	3	2,6202	0,9157
$[-10 + j10 \ -10 \ -10 - j10]$	133,3333	13,1833	8	0	0,4459
$[-10 - j10 \ -10 + j10 \ -7]$	93,3333	11,1833	6,5	0	0,6413
$[-10 - j10 \ -10 + j10 \ -5 - j5]$	66,6667	9,85	5,5	0	0,8821
$[-7 + j7 \ -7 \ -7 - j7]$	45,7333	5,5833	3,5	0	0,6229
$[-7 - j7 \ -7 + j7 \ -5]$	32,6667	5,05	2,5	0	0,8944
$[-7 - j7 \ -7 + j7 \ -10 - j10]$	65,3333	6,3833	5	1,4462	0,5013

Tabel 4.2 Hasil pengujian sistem kendali *close loop pole real* dan kompleks



Gambar 4.6. Nilai step respon dengan *close loop pole (J)* *real* dan kompleks

Hasil pengamatan tabel 4.2 dan gambar 4.6 menunjukkan bahwa, letak nilai *pole real* dan *pole kompleks* berpengaruh pada *settling time* sistem. Hasil pengamatan juga menunjukkan bahwa jika terdapat satu *pole* dominan (*pole* = -10) maka respon sistem memiliki *overshoot*.

## V. PENUTUP

### A. Kesimpulan

Dari hasil penelitian pengendali posisi Motor DC dengan *state feedback* dan metode *pole placement* diperoleh beberapa kesimpulan sebagai berikut:

1. Letak *close loop pole* mempengaruhi kestabilan dan performa sistem.
2. Letak *close loop pole* pada sumbu *real* memberikan efek sistem tidak memiliki *overshoot*.
3. Semakin ke kiri (negatif) letak *close loop pole* dari sumbu imajiner, maka *settling time* dan *rise time* semakin cepat.
4. Berdasarkan pengujian posisi *close loop pole* terdapat hasil terbaik dari keseluruhan dengan posisi *close loop pole* =  $[-10 + j10 \quad -10 \quad -10 - j10]$  diperoleh hasil yaitu nilai *overshoot* 0 %, *settling time* 0,4459 detik dan *rise time* 0,2558 detik.

### B. Saran

Untuk saran yang dapat diberikan oleh penulis terkait penelitian tentang motor DC dan pengendali *state feedback* yaitu:

1. Penentuan nilai *pole* yang diharapkan masih menggunakan penentuan secara acak (*pole* stabil), oleh karena itu dapat diterapkan suatu metode yang dapat menentukan nilai *pole* tanpa proses acak.

## DAFTAR PUSTAKA

- [1] M. Fahrezki, "Simulasi Pengaturan Kecepatan Motor DC Berbasis PID Menggunakan LabVIEW," Universitas Islam Indonesia, Yogyakarta, 2016.
- [2] A. Aziz, "Aplikasi *Hardware In Loop* Pada Pengendali Motor DC Menggunakan MATLAB," Universitas Islam Indonesia, Yogyakarta, 2015.
- [3] T. Winarto, "Aplikasi *Fuzzy Logic* Sebagai Kontrol Posisi Motor DC," Universitas Brawijaya, Malang, 2009.
- [4] M. A. Laughton and D. F. Warne, *Electrical Engineer's Reference Book* 16th Edition, New South Wales: Newnes, 2003.
- [5] W. Bolton, *Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering*, 3rd edition, New York: Pearson Education, 2004.
- [6] Laboratorium Kendali & Automasi Industri, Buku Petunjuk Praktikum Dasar Sistem Kendali, Yogyakarta: Teknik Elektro Universitas Islam Indonesia, 2013.
- [7] K. Ogata, *Teknik Kontrol Automatik*, Jilid 2, Edisi Kedua, Jakarta: Erlangga, 1997.
- [8] K. Ogata, *Modern Control Engineering* Fifth Edition, New York: Prentice Hall, 2010.