

# Desain Kontroler *Fuzzy PID Gain Scheduling* Untuk Pengaturan Kecepatan Motor DC Tanpa Sikat

Hudaibiy Hibban, Josaphat Pramudijanto, dan Nurlita Gamayanti

Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111

E-mail: jos@elect.eng.its.ac.id, nurlitagamayanti@gmail.com

**Abstrak**— Motor DC, sebagai salah satu mesin listrik yang banyak digunakan di industri, penggunaannya banyak digantikan dengan *Brushless DC Motor (BLDC)*. Penggunaan motor jenis ini hampir tidak menimbulkan polusi di jalan karena energi yang dipergunakan adalah energi listrik, sehingga selain kenyamanan karena nyaris tidak menimbulkan kebisingan motor ini juga ramah lingkungan. Di dalam tugas akhir ini, kita merancang sebuah sistem yang memberikan pembebanan pada motor BLDC menggunakan rem magnetik agar sistem menjadi tidak linear. Pada penelitian ini, digunakan kontroler *Fuzzy PID Gain Scheduling* untuk mengatur kecepatan motor BLDC. Dipilih kontroler PID karena pada kenyataannya kontrol yang paling sering dipakai adalah PID, dan *Fuzzy* digunakan sebagai pengatur nilai  $K_p$ ,  $K_i$ , dan  $K_d$ . Setelah dilakukan implementasi pada kontroler didapatkan bahwa nilai *rise time* rata-rata respon motor sebesar 2,6 detik dan nilai *settling time* rata-rata sebesar 3,6 detik. Selain itu terdapat juga overshoot yang relatif kecil sekitar 1 persen. Hasil tugas akhir ini berupa motor BLDC dapat digunakan sebagai penelitian lebih lanjut untuk adik-adik kelas di jurusan Elektro maupun jurusan lain.

**Kata Kunci**—*Brushless DC, Fuzzy PID Gain Scheduling.*

## I. PENDAHULUAN

PENGGUNAAN motor listrik dalam aplikasi sehari-hari semakin meningkat. Motor listrik banyak digunakan dalam berbagai peralatan seperti: *air conditioning, vacuum cleaner, conveyor*, lemari pendingin, dan lain sebagainya. Motor universal dan motor *direct current (DC)* banyak digunakan dalam aplikasi tersebut. Akan tetapi penggunaan motor DC konvensional menimbulkan masalah diakibatkan oleh penggunaan sikat. Penggantian sikat secara periodik untuk menjaga kinerja serta busur api adalah masalah yang umum disoroti pada motor DC. Oleh karena itu pada tugas akhir ini dipilihlah plant motor *Brushless Direct Current (BLDC)* yang merupakan alternatif pengganti motor DC. Motor ini adalah salah satu jenis motor yang popularitasnya mulai naik. [1]

Pada tugas akhir ini digunakan metode *Fuzzy PID Gain Scheduling* untuk mengendalikan kecepatan motor BLDC. Metode ini dipilih karena pada aplikasinya di dunia industri kebanyakan kontroler yang dipakai adalah kontroler PID. Dan *Fuzzy* disini berfungsi sebagai pengatur nilai  $K_p$ ,  $K_i$ , dan  $K_d$  pada PID karena pada implementasinya digunakan pembebanan yang berubah-ubah.

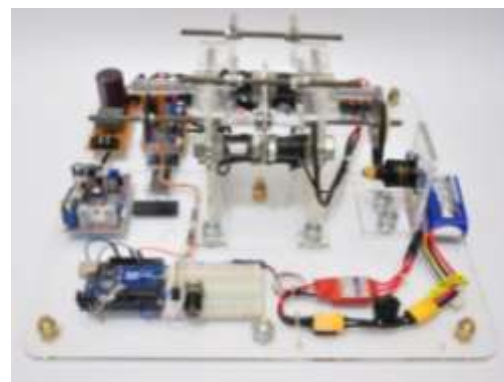
Permasalahan disini adalah pada saat keadaan nyata jalanan tidak selalu datar, banyak terdapat jalanan menanjak sehingga yang menjadi masalah pada penelitian ini adalah

bagaimana mengatur torsi motor BLDC agar dapat menanggung atau mengatasi beban yang berubah-ubah secara tidak menentu. Oleh karena itu, dibutuhkan suatu kontroler untuk melakukan pengaturan kerja dari motor listrik agar bekerja sesuai dengan kebutuhan yang diinginkan.

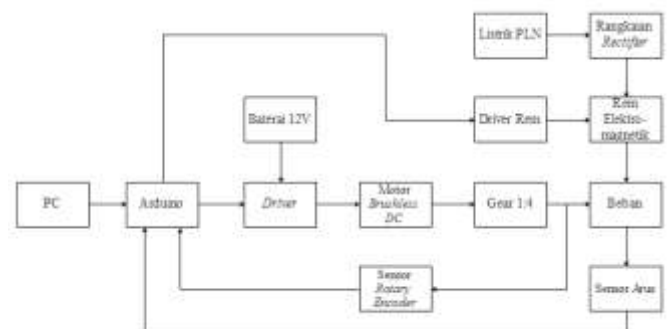
## II. PENGENALAN PLANT DAN TEORI KONTROL

### A. Plant Motor BLDC

*Plant* motor BLDC ini direncanakan dan direalisasikan oleh sebuah kelompok yang kami beri nama BLDC Team. Tim ini dibentuk karena keinginan dari kami untuk membangun sebuah *plant* baru yang dapat digunakan sebagai penelitian tugas akhir. *Plant* motor BLDC ini kemudian kami beri nama dengan motor BLDC-V1. Bentuk fisik dari BLDC-V1 ditunjukkan pada Gambar 1. *Plant* ini memiliki motor BLDC sebagai penggerak dari poros piringan yang dikopel dengan sebuah *gear*. Setelah itu terdapat rem elektromagnetik pada piringan sebagai metode pembebanan motor BLDC-V1. Diagram blok sistem ditunjukkan pada Gambar 2.



Gambar 1 Bentuk Fisik BLDC V-1.



Gambar 2 Diagram Blok Sistem.

**B. Fuzzy**

Saat ini logika *fuzzy* sudah banyak digunakan dalam berbagai bidang. Secara umum, logika *fuzzy* terdiri dari beberapa komponen, yaitu *fuzzifier*, *fuzzy rule base*, *fuzzy inference mechanisme* atau *fuzzy inference engine*, dan *defuzzifier*

*Fuzzy rule base* berisi pernyataan-pernyataan logika *fuzzy* (*fuzzy statement*), yang berbentuk pernyataan IF-THEN. *Fuzzifier* digunakan untuk memetakan nilai/harga variabel di dunia nyata kedalam himpunan *fuzzy* (*fuzzy sets*). Pemetaannya dilakukan dengan menggunakan fungsi yang disebut fungsi keanggotaan. Terdapat beberapa metode *fuzzifier*, 3 diantaranya yaitu *singleton fuzzifier*, *gaussian fuzzifier* dan *triangular fuzzifier*. *Defuzzifier* mengembalikan hasil perhitungan *fuzzy* (himpunan *fuzzy*) menjadi variabel sesuai rentangnya di dunia nyata. Sama dengan *fuzzifier*, *defuzzifier* juga menggunakan fungsi keanggotaan untuk memetakan nilai himpunan *fuzzy* menjadi variabel nyata. Terdapat beberapa metode *defuzzifier*, contohnya yaitu *Center of gravity defuzzifier*, *Center average defuzzifier*. [2]

**C. Kontroler PID**

Kontroler P, kontroler I, dan kontroler D memiliki kekurangan dan kelebihan masing – masing. Ketiga kontroler P, kontroler I, dan kontroler D dapat saling menutupi kekurangan masing-masing dengan cara menggabungkan ketiganya secara paralel. Sehingga, Kontroler PID merupakan kontroler berumpan balik yang memiliki 3 buah kontroler yaitu kontroler P (*Proportional*), kontroler I (*Integral*), dan kontroler D (*Derivative*) yang masing – masing secara keseluruhan bertujuan untuk mengatur respon dari sebuah sistem, menghilangkan *offset*, dan menghasilkan perubahan awal yang besar.

Kontroler PID merupakan hasil penjumlahan dari *output* kontroler P, *output* kontroler I, dan *output* kontroler D. Karakteristik kontroler PID sangat dipengaruhi oleh kontribusi besar ketiga parameter dari P, I dan D yaitu nilai  $K_p$ ,  $K_i$ , dan  $K_d$ . Penyetelan konstanta  $K_p$ ,  $K_i$ , dan  $K_d$  akan mengakibatkan reaksi dari respon yang berbeda-beda. Satu atau dua dari ketiga konstanta tersebut dapat disetel lebih menonjol dibanding yang lain. Konstanta yang menonjol itulah akan memberikan kontribusi pengaruh pada respon sistem secara keseluruhan. Parameter-parameter tersebut tidak bersifat independen, sehingga pada saat salah satu nilai konstantanya diubah, maka mungkin sistem tidak akan bereaksi seperti yang diinginkan. [3]

Pada penelitian ini kontroler PID akan dipakai sebagai kontroler utama dari sistem pengaturan kecepatan motor BLDC. Oleh karena itu pemilihan nilai  $K_p$ ,  $K_i$ , dan  $K_d$  harus sangat diperhatikan. Pemberian nilai  $K_p$ ,  $K_i$ , dan  $K_d$  sangat berpengaruh terhadap respon yang akan keluar dari motor BLDC ini.

Pengendali proporsional  $K_p$  akan memberikan efek mengurangi waktu naik tetapi tidak menghapus kesalahan keadaan tunak. Pengendali integral  $K_i$  akan memberikan efek menghapus kesalahan keadaan tunak tetapi berakibat memburuknya tanggapan transient. Pengendali derivatif  $K_d$  akan memberikan efek meningkatnya stabilitas sistem, mengurangi lewatan maksimum dan menaikkan tanggapan fungsi alih. Efek dari setiap pengendali dalam sistem *close loop* dapat dilihat pada tabel berikut ini :

Tabel 1. Pengaruh  $K_p$ ,  $K_i$ ,  $K_d$  pada Respon Sistem. [3]

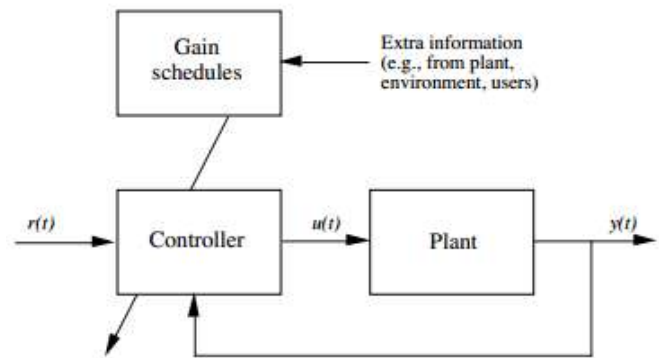
Respon Close-Loop	Rise Time	Overshoot	Setting Time	SS Error
$K_p$	Turun	Naik	Perubahan Kecil	Turun
$K_i$	Turun	Naik	Naik	Hilang
$K_d$	Perubahan Kecil	Turun	Turun	Perubahan Kecil

**D. Kontroler Fuzzy PID Gain Scheduling.**

Kontrol PID merupakan algoritma kontrol yang banyak digunakan di industri proses karena bentuknya yang sederhana dan mudah diimplementasikan. Pada kondisi operasi tertentu (seperti misalnya sering terjadi gangguan pada proses atau parameter proses yang berubah-ubah), parameter kontrol ini harus sering di-tune agar kinerjanya tetap baik. Salah satu teknik dalam sistem kontrol yang sering dilakukan untuk mengatasi permasalahan ini adalah dengan menggunakan metode *PID Gain Scheduling*, dimana parameter kontrol diubah secara otomatis jika terjadi perubahan kondisi operasi yang menyebabkan kinerja kontrol menurun.

Dalam rangka memberikan contoh penggunaan *fuzzy logic* di bidang kontrol proses, maka selanjutnya akan dibahas salah satu metode *PID gain scheduling* dengan menggunakan *fuzzy logic*. Metode ini dinamakan *Fuzzy PID gain scheduling*.

Pada aplikasi ini, *fuzzy* berfungsi menghitung parameter kontrol PID ( $K_p$ ,  $T_i$  dan  $T_d$ ), berdasarkan perubahan parameter luar ( $\Delta E$ ). Secara umum, diagram kontrol *fuzzy PID gain scheduling* dapat digambarkan seperti berikut.



Gambar 3. Fuzzy Gain Scheduling. [4]

Gambar 3 merupakan diagram kontrol sistem pengaturan *Fuzzy Gain Scheduling*. Pada diagram tersebut bisa dilihat bahwa struktur kontrol merupakan struktur konvensional tetapi bedanya kontroler ini dapat diubah parameter kontrolnya dengan menggunakan *Gain Schedules*. *Gain schedules* ini mendapatkan *input* dari informasi ekstra dari luar. Informasi ekstra ini dapat berupa informasi perubahan pada *plant* atau informasi gangguan pada *plant* yang dapat merubah fungsi alih dari sistem.

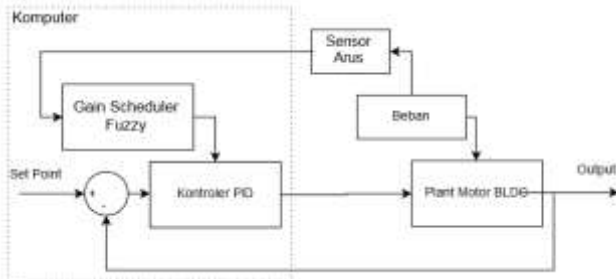
Pada penelitian ini kontroler yang digunakan merupakan kontroler PID yang sudah biasa dipakai pada dunia-dunia industri. Kontroler PID ini nantinya bisa diubah parameter kontrolnya yang berupa nilai  $K_p$ ,  $K_i$ , dan  $K_d$ . Parameter ini diubah dengan *gain schedules* yang pada tugas akhir ini berupa *fuzzy*. *Fuzzy* ini nantinya akan mendapatkan

informasi ekstra berupa informasi dari rem. Informasi rem ini berupa informasi arus yang masuk pada belitan rem magnetik.

### III. PERANCANGAN SISTEM

#### A. Gambaran Umum Sistem

Rencana dari sistem pengaturan kecepatan motor BLDC yang dibuat kali ini dapat dilihat pada diagram blok yang ada pada Gambar 4.



Gambar 4. Diagram Blok Sistem Pengaturan Kecepatan Motor BLDC.

Sistem pengaturan motor BLDC ini menggunakan kontroler PID. Tetapi parameter  $K_p$ ,  $K_i$ , dan  $K_d$  dapat berubah apabila terjadi perubahan pembebanan.

#### B. Perancangan Kontroler

##### 1) Perancang Kontroler PID

Sebelum merancang kontroler *fuzzy* harus dicari dulu kontroler PID yang sesuai untuk setiap pembebanan. Dengan merancang kontroler PID ini maka akan didapatkan nilai fungsi keanggotaan untuk *output* pada kontroler *fuzzy*.

Pada tahap ini direncanakan menggunakan metode Ziegler-Nichols untuk mencari nilai  $K_p$ ,  $K_i$ , dan  $K_d$ . Namun setelah PID hasil metode Ziegler-Nichols disimulasikan ternyata tidak menghasilkan *output* yang sesuai dengan keinginan penulis. Oleh karena itu PID yang telah dirancang disini merupakan PID hasil percobaan penulis sendiri yang telah diatur sedemikian rupa agar memiliki *output* yang sesuai dengan keinginan. Nilai konstanta ini sebenarnya bukan berdasarkan pencarian nilai secara coba-coba melainkan memiliki dasar juga. Dasar dari pemberian nilai ini memiliki konsep dasar pengetahuan secara heuristik misalnya apabila respon memiliki *error steady state* lalu apabila pada kontroler diberi konstanta  $K_i$  maka *error steady state* akan terhapus, lalu misalnya apabila terdapat *overshoot* maka nilai  $K_d$  harus diberi nilai yang lebih besar sehingga menambah transien pada respon. Sehingga setelah semua nilai  $K_p$ ,  $K_i$ ,  $K_d$  dicari didapatkanlah nilai-nilai berikut.

Tabel 2. Nilai  $K_p$ ,  $K_i$ ,  $K_d$  di Setiap Pembebanan.

Pembebanan	$K_p$	$K_i$	$K_d$
Tanpa Beban	0,165	0,049	0,000
Ringan	0,160	0,050	0,001
Sedang	0,155	0,055	0,005
Berat	0,140	0,060	0,010
Sangat Berat	0,120	0,070	0,020

##### 2) Perancangan Fuzzy

Pada tahap perancangan kontroler *fuzzy* terdapat beberapa tahap lagi yaitu penentuan fungsi keanggotaan, penentuan bentuk fungsi keanggotaan, Menentukan fungsi keanggotaan pada *output*.

Kontroler *fuzzy* pada sistem pengaturan motor BLDC V-1 ini berfungsi sebagai pengatur nilai  $K_p$ ,  $K_i$ ,  $K_d$  pada kontroler PID. Sehingga pada sistem ini nilai  $K_p$ ,  $K_i$ ,  $K_d$  akan menjadi fungsi keanggotaan pada *output* pada *fuzzy*. Pada *output fuzzy* akan dibutuhkan 3 *output* yaitu *output* yang memberikan nilai  $K_p$ ,  $K_i$ , dan  $K_d$ . Sehingga pada fungsi keanggotaan *input* terdapat 3 fungsi keanggotaan yang masing-masing mewakili  $K_p$ ,  $K_i$ , dan  $K_d$ . Sehingga pada *fuzzy* ini terdapat 3 kontroler *fuzzy* yang masing-masing memiliki satu fungsi keanggotaan *input* dan satu fungsi keanggotaan *output*.

Karena tujuan akhir dari kontroler *fuzzy* hanya mengeluarkan nilai  $K_p$ ,  $K_i$ , dan  $K_d$  yang sesuai saat pembebanan kondisi tertentu maka *rule base* pada ada 5 pernyataan di setiap *gain*  $K_p$ ,  $K_i$ , dan  $K_d$ . Misalkan saja pembebanan direpresentasikan dengan variabel X dan *output* direpresentasikan dengan variabel Y dan anggap saja nilai  $K_p$  pada saat tanpa beban bernilai 0,165 dan saat beban ringan bernilai 0,16 dan seterusnya. Maka *rule base* yang dibuat pada kontroler *fuzzy* akan terlihat seperti berikut. Angka pada X merupakan angka hasil pembacaan Arduino pada *output* sensor arus yang dipasang seri pada rem magnetik

- If X = 12 Then Y = 0,165
- If X = 190 Then Y = 0,160
- If X = 416 Then Y = 0,155
- If X = 624 Then Y = 0,140
- If X = 852 Then Y = 0,120

Ini merupakan *rule base* pada pencarian nilai  $K_p$ , Bisa dilihat bahwa pada nilai  $K_p$  semakin besar pembebanan maka semakin kecil nilai  $K_p$  yang diberikan. Setelah *rule base* pada  $K_p$  telah ditentukan maka yang selanjutnya adalah membuat *rule base* pada nilai  $K_i$ . Berikut ini merupakan *rule base* pada pencarian nilai  $K_i$ .

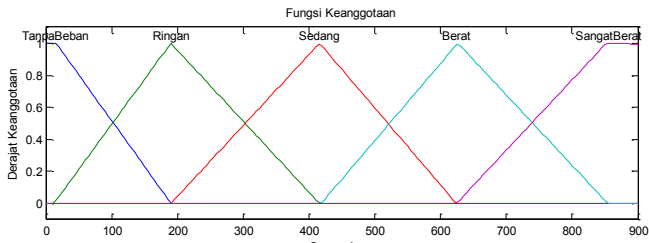
- If X = 12 Then Y = 0,049
- If X = 190 Then Y = 0,050
- If X = 416 Then Y = 0,055
- If X = 624 Then Y = 0,060
- If X = 852 Then Y = 0,070

Dari *rule base* untuk pencarian nilai  $K_i$  di setiap pembebanan ini dapat dilihat bahwa semakin besar nilai pembebanan maka semakin besar juga nilai  $K_i$  yang digunakan. Dan setelah mendapatkan *rule base* pada  $K_i$  maka yang dilakukan selanjutnya adalah mendapatkan *rule base* untuk nilai  $K_d$ . Berikut ini merupakan *rule base* untuk nilai  $K_d$ .

- If X = 16 Then Y = 0,000
- If X = 190 Then Y = 0,001
- If X = 416 Then Y = 0,005
- If X = 624 Then Y = 0,010
- If X = 852 Then Y = 0,020

Pada *rule base* untuk penentuan nilai  $K_i$  di setiap pembebanan ini dapat dilihat bahwa semakin besar pembebanan maka semakin besar nilai  $K_i$ .

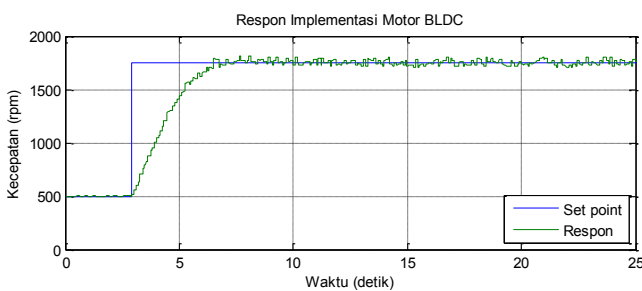
Gambar 5 merupakan fungsi keanggotaan dari *fuzzy* yang telah dibuat. Gambar 5 merupakan fungsi keanggotaan *input* dari *fuzzy*.



Gambar 5. Fungsi Keanggotaan Input.

IV. PENGUJIAN DAN ANALISA

Pada bagian pengujian dilakukan implementasi kontroler pada *plant real* motor BLDC. Pengujian ini dilakukan dengan memberikan *set point* kecepatan 500 rpm yang selanjutnya dinaikkan lagi *set point* menjadi 1750 rpm. Setelah itu respon dari motor dapat dianalisa dan dicari nilai dari *rise time*, *settling time*, *overshoot*, dan *error steady state*.



Gambar 6. Respon Step Implementasi pada Kondisi Tanpa Beban.

Pada respon motor BLDC V-1 tanpa beban tidak terdapat *error steady state*. Namun berbeda dengan simulasi, pada implementasi terdapat *overshoot* karena nilai maksimum terdapat diatas nilai *steady state*. Dan dibawah ini merupakan hasil perhitungan dan pengamatan *rise time* ( $T_r$ ), *settling time* ( $T_s$ ), dan *Percentage Overshoot* (%Mp) dari respon sistem. Pada kali ini analisa dilakukan pada saat kecepatan motor berubah dari 500 rpm menjadi 1750 rpm.

$$T_s = 6,53 - 2,93$$

$$T_s = 3,6 \text{ detik}$$

$$T_r = 5,83 - 3,15$$

$$T_r = 2,68 \text{ detik}$$

$$\%Mp = (1766 - 1750)/1250$$

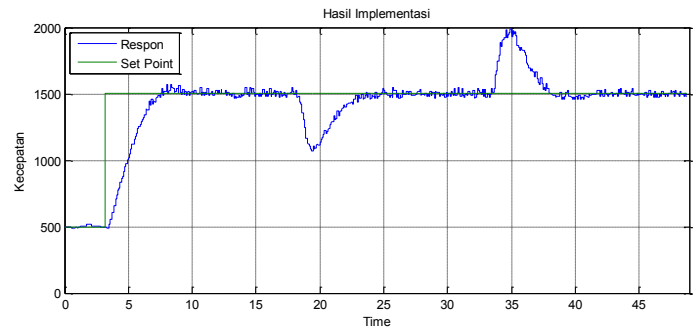
$$\%Mp = 0,0128 = 1,28 \%$$

Setelah semua data telah diambil, berikut hasil seluruh data yang telah didapatkan dari mulai kondisi tanpa beban hingga saat kondisi beban sangat berat dalam bentuk tabel.

Tabel 3. Data Karakteristik Respon Hasil Implementasi.

Pembebanan	Settling Time (detik)	Rise Time (detik)	Overshoot (%)	Error Steady State
Tanpa Beban	3,6	2,8	1,28	0
Ringan	3,21	2,62	2,8	0
Sedang	3,13	2,44	1,2	0
Berat	3,69	2,67	0,56	0
Sangat Berat	3,77	2,88	0,24	0

Dan gambar berikut ini merupakan gambar hasil implementasi kontroler *Fuzzy PID Gain Scheduling* yang pada *input*-nya diberi *set point* 500 rpm untuk *start up* dan lalu diberi nilai 1500 rpm. Setelah itu nilai rem pun dimainkan dan berikut respon hasil *running* dari *plant* motor BLDC.



Gambar 7. Hasil Implementasi.

Pada Gambar 7 merupakan hasil implementasi kontroler dengan beban yang berubah-ubah. Hal ini dilakukan dengan cara menjalankan motor pada kecepatan tertentu lalu beban pada rem dinaikkan hingga ke tingkat beban sangat berat yaitu pembacaan sekitar 85% dari rem penuh. Setelah itu rem kembali dihilangkan kembali sehingga motor kembali ke *plant* semula.

Setelah dilakukan hal tersebut dapat dilihat hasil respon dari kecepatan motor setelah respon diubah-ubah. Pada grafik kecepatan vs waktu di atas bisa dilihat respon kontroler saat motor diberikan beban pada detik ke 17. Kecepatan pada motor langsung turun sampai sekitar 1100 rpm, lalu kontroler pun langsung memberikan respon untuk mengembalikan kecepatan motor agar sesuai dengan *set point*. Begitu juga ketika rem dikembalikan pada posisi tanpa beban, nilai kecepatan motor naik ketika beban dilepas setelah itu kontroler akan mengembalikan kecepatan motor agar berada kembali pada keadaan *set point*.

V. KESIMPULAN

Hasil respon yang didapatkan saat implementasi memiliki *settling time* sekitar 3,6 detik. Dan untuk nilai *rise time* saat implementasi membutuhkan waktu sekitar 2,6 detik.

UCAPAN TERIMAKASIH

Terima kasih kepada rekan-rekan tim BLDC diantaranya saudara Fachrul Arifin, Habib Ibnu Hasan, Beny Setiyadi Hidayat, dan M. Ammar Huwaidi yang telah membantu dalam menyelesaikan tugas akhir ini.

DAFTAR PUSTAKA

[1] Mosavi, Mohamad; Reza Abdoreza Rahmati dan Alireza Khoshaadat, "Design of Efficient Adaptive Neuro Fuzzy Controller Based on Supervisory Learning Capable for Peed and Torque Control of BLDC Motor," *PRZGLAD ELEKTROTECHNICZHY (Electrical Review)*, Iran, 2012.

[2] Sihana, Faridah, and Izzad Abidy, "Identifikasi Model Sistem Hidraulik Kendali Tekanan pada Suplai Bahan Bakar Turbin Gas," *TEKNOFISIKA, Vol. 2 No. 1 Edisi Januari 2013*, Jogjakarta, 2013.

[3] \_\_\_\_\_, "Kontroler PID", <URL: [http://labkontrol.blogspot.com/2012/09/kontroller-pid\\_30.html](http://labkontrol.blogspot.com/2012/09/kontroller-pid_30.html)>, September, 2012.

[4] Passino, Kevin M., Stephen Yurkovich, "Fuzzy Control", Addison Wesley Longman, Inc, Menlo Park, California, 1998.