

**DESAIN SISTEM KONTROL KECEPATAN PELONTAR
SHUTTLECOCKPADAROBOT DIVISI KONTES ROBOT ABU**

INDONESIA

SKRIPSI

TEKNIK ELEKTRO KONSTENRASI TEKNIK KONTROL

Ditujukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



MUHAMMAD SYAFEI

NIM 155060301111024

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

MALANG

2019



*Teriring Ucapat Terima Kasih kepada:
Ayahanda dan Ibunda Tercinta*





LEMBAR ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas di dalam Naskah Skripsi ini adalah asli dari pemikiran saya, tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu Perguruan Tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur jiplakan, saya bersedia Skripsi dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

Malang, 6 Januari 2020

Mahasiswa,

Muhammad Syafei
NIM.155060301111024





RINGKASAN

Muhammad Syafei, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Desember 2019, Desain Sistem Kontrol Kecepatan Pelontar *Shuttlecock* Pada Robot Divisi Kontes Robot Abu Indonesia, Dosen Pembimbing 1: Moch Rusli, Dosen Pembimbing 2: Erni Yudaningtyas

Kontes Robot ABU Indonesia adalah salah satu divisi yang diperlombakan dalam ajang Kontes Robot Indonesia yang diselenggarakan setiap tahunnya oleh Direktorat Jenderal Pendidikan Tinggi. Tema dan aturan pelombaan Kontes Robot ABU Indonesia mengacu pada *Asia-Pacific Broadcasting Union* Robocon. Tema dan aturan tersebut berbeda setiap tahunnya tergantung dengan tuan rumah penyelenggara *Asia-Pacific Broadcasting Union* Robocon. Pelontar *shuttlecock* adalah salah satu mekanisme yang digunakan pada robot pada perlombaan dengan mengacu pada perlombaan Kontes Robot ABU Indonesia tahun 2018. Pelontar digerakkan oleh motor DC dan kecepatan putar dibaca oleh sensor *rotary encoder*. Kontroler yang digunakan adalah kontroler Proporsional Integral. Penentuan parameter kontroler Kp dan Ki dilakukan dengan menggunakan metode *Root Locus* dan didapatkan nilai parameter kontroler Kp = 0.748755 dan Ki = 2.5. Setelah dilakukan pengujian pada sistem didapatkan nilai *settling time* rata-rata sebesar 56.34 milidetik pada pengujian tanpa beban, 259.13 milidetik pada pengujian berbeban, nilai *error steady state* kurang dari 2.1% dan *overshoot* kurang dari 5.1%.

Kata Kunci: Motor DC, Kontroler PI, *Root Locus*, *Throwing Mechanism*.



SUMMARY

Muhammad Syafei, Department of Electrical Engineering, Faculty of Engineering, Brawijaya University, December 2019, Speed Control of Shuttlecock Throwing on Indonesia Abu Robot Contest division, Supervisor 1:Moch Rusli, Supervisor 2:Erni Yudaningtyas,

The Indonesian aburobot contest is one of the divisions competed in the Indonesian robot contest held annually by the Direktorat Jenderal Pendidikan Tinggi. The theme and rules of the Indonesia Abu robot contest refer to the Asia Pacific Broadcasting Union Robocon. The themes and rules differ each year depending on the hosts of Asia Pacific Broadcasting Union Robocon organizers. Shuttlecockthrowing mechanism is one of the main mechanism to use at Indonesia abu robot contest event 2018. The throwing mechanism is driven by a DC motor and the speed is read by the rotary encoder sensor. The controller used is the PI controller. Kp and Ki controller parameters are determined using the Root Locus method and obtained controller parameter values $K_p = 0.748755$ and $K_i = 2.5$. After testing on the system, the average settling time is 56.34 miliseconds on load-free testing, 259.13 miliseconds on load testing, steady state error value is less than 2.1% and overshoot is less than 5.1%.

Keywords: DC Motor, PI Controller, Root Locus, Throwing Mechanism.



DAFTAR ISI

RINGKASAN	i
SUMMARY	iii
PENGANTAR	v
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	3
BAB II TINJAUAN PUSTAKA	5
2.1 <i>Shuttlecock</i>	5
2.2 Motor DC.....	5
2.3 <i>Rotary encoder</i>	8
2.4 Driver Motor H-bridge.....	9
2.5 Mikrokontroler STM32F407VGT6.....	10
2.6 <i>Pulse Width Modulation (PWM)</i>	12
2.7 Kontroler.....	12
2.6.1 Kontroler Proporsional.....	13
2.6.2 Kontroler Integral.....	13
2.6.3 Kontroler PI (Proporsional Integral).....	14
2.7 Root Locus.....	15
BAB III METODE PENELITIAN	17
3.1 Perancangan Blok Diagram Sistem.....	17
3.2 Spesifikasi Desain.....	18

3.3	Penentuan <i>Setpoint</i>	18
3.4	Karakterisasi Motor DC	20
3.5	Karakteristik Driver Motor	22
3.6	Penentuan Fungsi Alih Motor DC	24
3.7	Penentuan Parameter Kontroler PI dengan Metode <i>Root Locus</i>	27
3.8	Pembuatan Perangkat Keras	37
3.9	Pembuatan Algoritma dan Program	42
3.10	Perancangan Pengujian Sistem	42
BAB IV HASIL DAN PEMBAHASAN		45
4.1.	Pengujian Sistem Tanpa Beban	45
4.2.	Pengujian Sistem Berbeban	48
4.3.	Pengujian Sistem dengan Variasi Beban	51
4.4.	Pengujian Sifat Reversibel Sistem	55
BABV KESIMPULAN		57
DAFTAR PUSTAKA		59

DAFTAR GAMBAR

Gambar 2.1 Shuttlecock	5
Gambar 2.2 Rangkaian ekivalen motor DC magnet permanen.....	7
Gambar 2.3 Diagram blok pengontrolan motor DC magnet permanen.....	7
Gambar 2.4 Motor DC PG45	8
Gambar 2.5 Rotary encoder	9
Gambar 2.6 Driver Motor VNH3SP30	10
Gambar 2.7 Rangkaian H-Bridge.....	10
Gambar 2.8 STM32F407VGT6	11
Gambar 2.9 Duty Cycle Pada Sinyal PWM.....	12
Gambar 2.10 Diagram Blok Kontroler Proporsional	13
Gambar 2.11 Diagram Blok Kontroler Integral	14
Gambar 2.12 Diagram Blok Kontroler PI (Proporsional Integral).....	15
Gambar 2.13 Sistem kontrol loop tertutup.....	15
Gambar 3.1 Blok Diagram Sistem	17
Gambar 3.2 Lapangan lomba	19
Gambar 3.3 Pengujian motor DC.....	21
Gambar 3.4 Hasil pengujian driver motor	24
Gambar 3.5 Sinyal Masukan PRBS dan Keluaran Motor DC	25
Gambar 3.6 System Identification Toolbox	25
Gambar 3.7 Hasil Estimasi Model	26
Gambar 3.8 Model <i>continuous</i> fungsi alih motor DC.....	26
Gambar 3.9 Validasi fungsi alih motor DC	27
Gambar 3.10 Penentuan pole pada diagram <i>Root Locus</i>	29
Gambar 3.11 Keluaran sitem dengan nilai K_p 0,169341 dan K_i 1	31
Gambar 3.12 Keluaran sistem dengan nilai K_p 0,238212 dan K_i 1,5.....	31
Gambar 3.13 Keluaran sistem dengan nilai K_p 0,307082 dan K_i 2	32
Gambar 3.14 Keluaran sistem dengan nilai K_p 0,375953 dan K_i 2,5.....	32
Gambar 3.15 Keluaran sistem dengan nilai K_p 0,444823 dan K_i 3	33
Gambar 3.16 Keluaran sistem dengan nilai K_p 0,513694 dan K_i 3,5.....	34
Gambar 3.17 Keluaran sistem dengan nilai K_p 0,582564 dan K_i 4	34
Gambar 3.18 Keluaran sistem dengan nilai K_p 0,651435 dan K_i 4,5.....	35
Gambar 3.19 Keluaran sistem dengan nilai K_p 0,720305 dan K_i 5	36

Gambar 3.20 Keluaran sistem dengan nilai K_p 0,789176 dan K_i 5,5.....	36
Gambar 3.21 Skema pembuatan perangkat keras	38
Gambar 3.22 Baterai LiPo 3S	38
Gambar 3.23 Mikrokontroler STM32F407.....	38
Gambar 3.24 Sensor rotary encoder	39
Gambar 3.25 Driver motor VNH3SP30.....	39
Gambar 3.26 Motor DC planetary gearbox.....	39
Gambar 3.27 PC atau komputer	40
Gambar 3.28 Foto alat tampak depan	40
Gambar 3.29 Foto pengkabelan alat.....	41
Gambar 3.30 Flowchart Pemrograman	42
Gambar 4.1 Keluaran sistem tanpa beban dengan <i>setpoint</i> 100 RPM	45
Gambar 4.2 Keluaran sistem tanpa beban dengan <i>setpoint</i> 125 RPM	46
Gambar 4.3 Keluaran sistem tanpa beban dengan <i>setpoint</i> 150 RPM	47
Gambar 4.4 Keluaran sistem berbeban dengan <i>setpoint</i> 100 RPM.....	48
Gambar 4.5 Keluaran sistem berbeban dengan <i>setpoint</i> 125 RPM.....	49
Gambar 4.6 Keluaran sistem berbeban dengan <i>setpoint</i> 150 RPM.....	50
Gambar 4.7 Keluaran sistem dengan beban 308 gr.....	52
Gambar 4.8 Keluaran sistem dengan beban 318 gr.....	53
Gambar 4.9 Keluaran sistem dengan beban 325 gr.....	54
Gambar 4.10 Keluaran sistem dengan beban 328 gr.....	54
Gambar 4.11 Keluaran pada pengujian sifat reversibel sistem.....	56

DAFTAR TABEL

Tabel 3.1 <i>Setpoint</i>	20
Tabel 3.2 Data pengujian sensor kecepatan <i>rotary encoder</i>	20
Tabel 3.3 Data pengujian driver motor.....	23
Tabel 3.4 Nilai parameter K_p dan K_i	30
Tabel 3.5 Data Pengujian Parameter Kontroler.....	37
Tabel 4. 1 Data Hasil Pengujian Sistem Tanpa Beban dan Berbeban.....	51
Tabel 4. 2 Data Pengujian Sistem dengan Variasi Beban.....	55





BAB I PENDAHULUAN

1.1 Latar Belakang

Kontes Robot Indonesia (KRI) adalah ajang kompetisi rancang bangun dan rekayasa dalam bidang robotika yang diselenggarakan oleh Direktorat Kemahasiswaan, Direktorat Jenderal Pembelajaran dan kemahasiswaan, Kementerian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia. Kontes robot Indonesia ini dapat diikuti oleh tim mahasiswa pada Perguruan Tinggi yang tercatat di kemenristekdikti. Pada ajang KRI terdapat lima kategori robot yang dilombakan yaitu Kontes Robot Seni Tari Indonesia (KRSTI), Kontes Robot Sepak Bola Indonesia (KRSBI) humanoid dan beroda, Kontes Robot Pemadam Api Indonesia (KRPAI) dan Kontes Robot ABU Indonesia (KRAI).

Sejak Tahun 2001 dan seterusnya, Kontes Robot Indonesia (KRI) diadakan setiap tahun sekali. Dengan mengambil tema yang sesuai dengan tema yang digunakan dalam ABU Robocon agar tim dari Indonesia dapat menjadi salah satu peserta yang mewakili negara Indonesia. Tema dan aturan pada KRAI setiap tahunnya selalu berubah tergantung dengan negara tuan rumah ABU Robocon.

Pada tema ABU Robocon 2018 yang mengacu pada permainan tradisional Vietnam Lempar Bola Berkah. Game ini dimainkan dengan melemparkan *shuttlecock* melewati lingkaran yang ditempatkan diatas tiang yang berdiri secara vertikal dan harus mendarat pada titik yang ditentukan. Salah satu mekanisme melontar adalah dengan menggunakan lengan pelontar yang sumbu putarnya disambung dengan motor DC sebagai penggerak.

Motor DC adalah motor listrik yang memerlukan suplai tegangan arus searah pada kumparan medan untuk diubah menjadi energi mekanik. Motor DC yang digunakan pada penelitian kali ini adalah motor DC PG45. Motor DC PG45 adalah motor DC yang telah digabungkan dengan *rotary encoder* dan juga gearbox untuk menambahkan torsi motor tersebut. Dalam penelitian pendahuluan untuk menggunakan motor DC sebagai aktuatur dalam pelontar *shuttlecock* yang sudah dilakukan yaitu tanpa menggunakan kontroler. Didapatkan kecepatan motor kurang dari kecepatan yang diinginkan, atau kecepatan berlebih. Untuk itu dibutuhkan pengontrol kecepatan dari motor pelontar *shuttlecock*. Agar kecepatan sesuai dengan yang diinginkan.

Dibutuhkan sebuah kontroler untuk mengatur kecepatan motor DC. Terdapat beberapa macam tipe kontroler seperti kontroler proporsional (P) yang berfungsi untuk mempercepat *rise time*, kontroler integral (I) berfungsi mengurangi *error steady state*, proporsional

integral (PI) gabungan dari kontroler proporsional (P) dan integral (I), dan proporsional integral diferensial (PID). Berdasarkan karakteristik dari masing-masing kontroler maka pada penelitian ini digunakan kontroler proporsional integral (PI) adalah aksi kontrol yang memiliki keluaran cepat, dan menghilangkan *error steady state* sehingga sesuai untuk mengontrol motor DC. (R. Selvy, 2016). Dengan menggunakan kontroler PI pada kontrol kecepatan motor DC pada penelitian ini. Kontroler PI sendiri membutuhkan parameter K_p dan K_i yang harus dicari terlebih dahulu. Maka dibutuhkan sebuah metode untuk menentukan parameter kontroler PI.

Metode *Root Locus* merupakan cara perhitungan atau metode untuk mencari parameter PID dengan mencari fungsi alih suatu sistem langsung dari perhitungan spesifik dari aktuator dan sensornya yang kemudian dicari beberapa letak pole sistemnya. Metode ini memungkinkan untuk menentukan pole dari sistem secara manual sehingga bisa diganti apabila tidak sesuai dengan yang diharapkan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan sebelumnya, maka rumusan masalah dalam skripsi ini ditekankan pada:

1. Bagaimana merancang sistem kontrol kecepatan pada pelontar *shuttlecock* robot divisi KRAI
2. Bagaimana keluaran sistem setelah diberi kontroler PI.
3. Bagaimana keluaran sistem setelah diberikan nilai *setpoint* berbeda.
4. Bagaimana keluaran sistem setelah diberi gangguan.

1.3 Batasan Masalah

Demi terfokusnya objek pengkajian, maka dilakukan pembatasan masalah. Adapun batasan masalah pada skripsi ini adalah:

1. Metode penentuan parameter kontroler yang digunakan adalah *Root Locus*
2. Motor yang digunakan adalah motor DC PG45
3. Tegangan yang digunakan pada motor DC adalah 24V
4. Tegangan mikrokontroler adalah 5V
5. Alat ini hanya bekerja dengan efisien pada *duty cycle* 85% – 100%

1.4 Tujuan

Tujuan yang ingin dicapai dari skripsi ini adalah dapat merancang dan membuat sistem kontrol kecepatan motor pada sistem pelontar *shuttlecock* pada robot KRAI.

1.5 Manfaat

Penelitian skripsi ini diharapkan mampu memberikan manfaat sekurang – kurangnya antara lain:

1. Bagi Peneliti

Sebagai sarana belajar dan sebagai wadah untuk mengimplementasikan ilmu pengetahuan yang didapat selama masa perkuliahan.

2. Bagi Tim Robotika

Agar dapat digunakan sebagai acuan apabila diadakan kembali rule lomba yang menggunakan pelontar dengan prinsip kerja yang serupa.

3. Bagi Calon Peneliti

Sebagai sumber referensi dalam pembelajaran dan penelitian yang serupa sehingga dapat menambah dan memperkaya wawasan peneliti berikutnya.





BAB II

TINJAUAN PUSTAKA

2.1 *Shuttlecock*

Konsep dari tema ABU Robocon 2018 berdasarkan permainan tradisional Vietnam “NEM CÒN” atau lempar bola berkah. Game ini dimainkan dari halaman terbuka dengan bambu yang ditancapkan di tengah lapangan dengan tinggi 15m, dengan bambu berbentuk lingkaran yang diletakkan di atas tiang bambu tersebut.

Inti dari permainan ini terletak pada “CÒN” atau dalam *rule* kali ini disebut dengan *shuttlecock*. *Shuttlecock* disini dibuat dengan bentuk bulat berisikan kapas dan beras, yang melambangkan kemakmuran dan kebahagiaan. *Shuttlecock* digantung dengan tali yang dijait di tengah-tengah *shuttlecock* dan diberi hiasan berupa kain warna-warni yang dijait pada sisi *shuttlecock*. Saat dimainkan *shuttlecock* dipegang pada ujung tali dan diputar searah jarum jam beberapa kali dan dilemparkan ke arah tengah lingkaran. Jika *shuttlecock* berhasil melewati lingkaran maka pemain dianggap menang. *Shuttlecock* terbang menggambarkan naga terbang yang ikonik dengan kekuatan manusia dan semesta. Gambar dari *shuttlecock* dapat dilihat dalam Gambar 2.1



Gambar 2.1 *Shuttlecock*

2.2 **Motor DC**

Motor DC merupakan motor listrik yang sangat sering di gunakan sebagai elemen kontrol akhir dalam sistem kontrol posisi dan kecepatan. Prinsip kerja motor dc sesuai

dengan hukum Lorentz, apabila arus dialirkan melalui kumparan jangkar dari mesin DC dan kumparan medannya diberi penguatan, maka akan timbul Gaya Lorentz pada tiap sisi kumparan jangkar tersebut (Soemarwanto, 2010)

Besar gaya Lorentz yang di timbulkan motor DC adalah

$$F = B \cdot I \cdot l \text{ dyne} \quad (2-1)$$

dengan:

B = kerapatan medan magnetik (Gauss).

I = arus listrik (Ampere).

l = panjang konduktor (cm).

Akibat timbulnya gaya lorentz, maka timbul kopel yang besarnya adalah:

$$\text{Kopel: } M = F \times r \text{ dyne-cm} \quad (2-2)$$

Dimana:

r = jari jari jangkar (cm)

$$M = B \cdot I \cdot l \cdot r \text{ dyne-cm} \quad (2-3)$$

Apabila jumlah: $M = z \cdot B \cdot I \cdot l \cdot r \text{ dyne-cm}$

$$\text{Karena } B = \frac{\phi}{A}$$

dimana:

ϕ = fluksi tiap kutub (Maxwell)

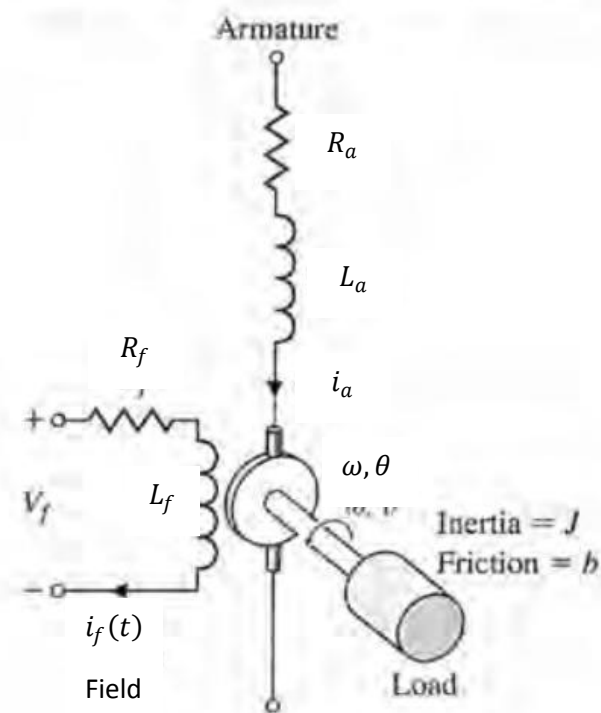
A = Luas penampang (cm^2)

z = jumlah konduktor di jangkar

$$\text{maka: } M = z \cdot \frac{\phi}{A} \cdot I \cdot l \cdot r$$

Motor *Direct Current* (DC) merupakan aktuator yang banyak digunakan dalam teknologi kontrol. Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Catu tegangan DC dari baterai menuju ke lilitan melalui sikat yang menyentuh komutator, dua segmen yang terhubung dengan dua ujung lilitan. Kumparan dalam satu lilitan disebut *armature* (jangkar). *Armature* adalah sebutan untuk komponen yang berputar di antara medan magnet.

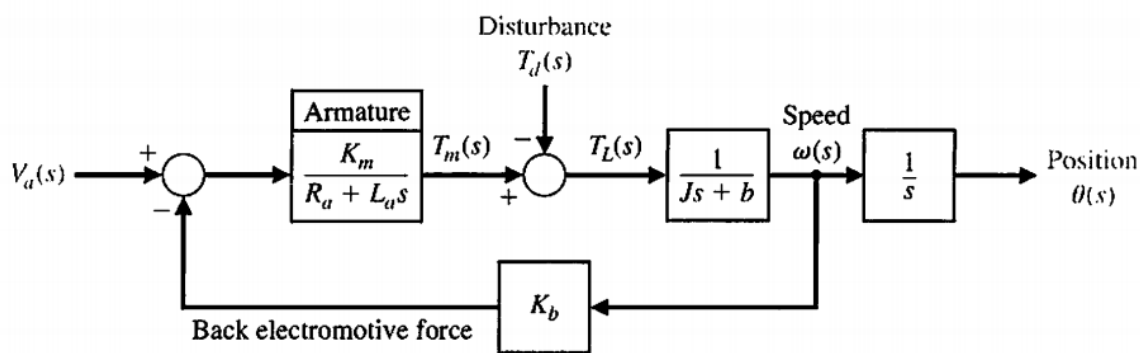
Rangkaian ekivalen motor DC magnet permanen dapat dilihat dalam Gambar 2.2.



Gambar 2.2 Rangkaian ekivalen motor DC magnet permanen

Sumber: Dorf, R. C. & Bishop, R. H., 2011

Pada motor DC magnet permanen, arus kumparan medan I_f dibuat konstan. Diagram blok pengontrolan kecepatan motor DC magnet permanen dapat dilihat dalam Gambar 2.3.



Gambar 2.3 Diagram blok pengontrolan motor DC magnet permanen

Sumber: Dorf, R. C & Bishop, R. H., 2011

Apabila *disturbance* (gangguan) $T_d(s)=0$, maka fungsi alih motor DC adalah

$$\frac{\omega(s)}{V_a(s)} = \frac{K_m}{(R_a + L_a s)(J s + b) + (K_b K_m)}$$

dengan:

K_m = konstanta motor

K_m = konstanta *back electromotive force*

K_m = resistansi *armature*

K_m = resistansi *armature*

J = inersia
 b = gesekan

Salah satu jenis motor DC magnet permanen adalah motor DC PG45 seperti terlihat dalam Gambar 2.4



Gambar 2.4 Motor DC PG45

2.3 Rotary encoder

Rotary encoder adalah alat elektromekanik yang dapat memonitor gerakan dan posisi. *Rotary encoder* umumnya menggunakan sensor optik untuk menghasilkan serial pulsa yang dapat diartikan menjadi gerakan, posisi, dan arah. Sehingga posisi sudut suatu poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh *rotary encoder* untuk diteruskan oleh rangkaian kontrol. *Rotary encoder* umumnya digunakan pada pengontrolan robot, motor driver, dsb.

Rotary encoder tersusun dari suatu piringan tipis yang memiliki lubang-lubang padabagian lingkaran piringan. *Light Emitting Diode* (LED) ditempatkan pada salah satu sisi piringan sehingga cahaya akan menuju ke piringan. Di sisi yang lain suatu photo-transistor diletakkan sehingga *photo-transistor* ini dapat mendeteksi cahaya dari LED yang berseberangan. Piringan tipis tadi dikopel dengan poros motor, atau divais berputar lainnya yang ingin kita ketahui posisinya, sehingga ketika motor berputar piringan juga akan ikut berputar. Apabila posisi piringan mengakibatkan cahaya dari LED dapat mencapai photo-transistor melalui lubang-lubang yang ada, maka photo-transistor akan mengalami saturasi dan akan menghasilkan suatu pulsa gelombang persegi. Semakin banyak deretan pulsa yang dihasilkan pada satu putaran menentukan akurasi *rotary encoder* tersebut, akibatnya semakin banyak jumlah lubang yang dapat dibuat pada piringan menentukan akurasi *rotary*

encoder tersebut. Bentuk fisik dari *rotary encoder* omron E6C2 dapat dilihat dalam Gambar

2.5.



Gambar 2.5 Rotary encoder

Sumber: www.omron.com

2.4 Driver Motor H-bridge

Driver motor DC tipe H-Bridge menggunakan Power Driver berupa transistor, transistor ini berfungsi sebagai saklar yaitu dengan menghasilkan keadaan on dan keadaan off. Keadaan off/tutup merupakan suatu keadaan dimana tidak ada arus yang mengalir.

Keadaan on/buka merupakan suatu keadaan yang mana arus bisa mengalir dengan bebas atau dengan kata lain (secara ideal) tidak ada resistivitas dan besar voltase pada saklar sama dengan nol (Rahmat. A, 2019).

Driver Motor H-Bridge MOSFET merupakan bagian yang berfungsi untuk mengendalikan motor BLDC yang mengatur arah putaran motor maupun kecepatan putar motor bergantung dari nilai tegangan yang dimasukan pada masukan driver itu sendiri.

Driver motor yang digunakan pada perancangan ini adalah driver motor VN3SP30.

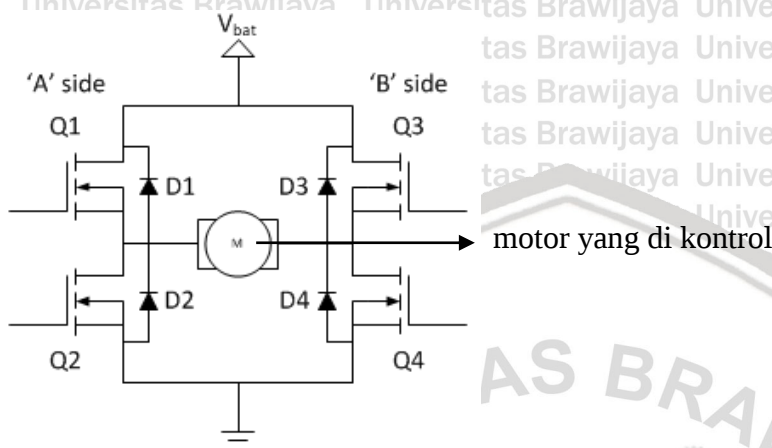
Driver motor VN3SP30 dapat dilihat dalam Gambar 2.6.



Gambar 2.6 Driver Motor VN3SP30

H-Bridge merupakan rangkaian yang berfungsi untuk mengatur arah pergerakan motor.

Diberi nama rangkaian H-Bridge karena bentuk dari rangkaian yang menyerupai huruf H seperti dalam Gambar 2.7.



Gambar 2.7 Rangkaian H-Bridge

Sumber: Rahmat. A, 2019

Rangkaian ini terdiri dari dua buah MOSFET kanal P dan dua buah MOSFET kanal N. Prinsip kerja rangkaian ini adalah dengan mengatur mati-hidupnya ke empat MOSFET tersebut. Huruf M pada Gambar adalah motor DC yang akan dikontrolkan. Bagian atas rangkaian akan dihubungkan dengan sumber daya kutub positif, sedangkan bagian bawah rangkaian akan dihubungkan dengan sumber daya kutub negatif. Pada saat MOSFET A dan MOSFET D on sedangkan MOSFET B dan MOSFET C off, maka sisi kiri dari gambar motor akan terhubung dengan kutub positif dari catu daya, sedangkan sisi sebelah kanan motor akan terhubung dengan kutub negatif dari catu daya sehingga motor akan bergerak searah jarum jam. Tabel logika aksi motor ditunjukkan dalam Tabel 2.1

Tabel 2.1 Tabel Logika Aksi Motor

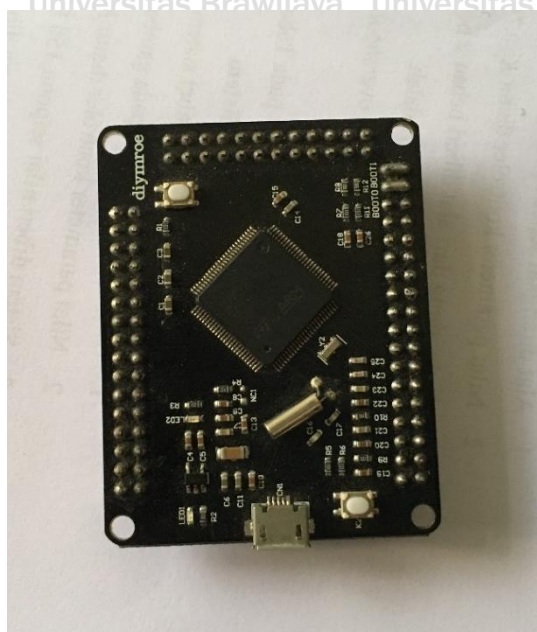
Sumber: Rahmat. A, 2019

Q1	Q2	Q3	Q4	Arah Putaran
ON	OFF	OFF	ON	Motor berputar searah jarum jam
OFF	ON	ON	OFF	Motor berputar berlawanan arah jarum jam
ON	OFF	ON	OFF	Pengereman
OFF	ON	OFF	ON	Pengereman
OFF	OFF	OFF	OFF	Bebas

2.5 Mikrokontroler STM32F407VGT6

STM32F407VGT6 didasarkan pada kinerja tinggi ARM®Cortex®-M4 32-bit RISC yang beroperasi pada frekuensi hingga 168 MHz. Inti Cortex-M4 dilengkapi dengan

akurat *Floating Point Unit* (FPU) tunggal yang mendukung semua instruksi pemrosesan data dan tipe data presisi tunggal *Advanced RISC Machine* (ARM). Ini juga mengimplementasikan satu set lengkap instruksi *Digital Signal Processing* (DSP) dan *memory Protection Unit* (MPU) yang meningkatkan keamanan aplikasi. Mikrokontroler STM32F407VGT6 dapat dilihat dalam Gambar 2.8.



Gambar 2.8 STM32F407VGT6

STM32F407VGT6 menggabungkan memori tertanam berkecepatan tinggi (memori Flash hingga 1 Mbyte, hingga 192 Kbytes *Static Random Access Memory* (SRAM)), hingga 4 Kbytes SRAM cadangan, dan rentang luas I / Os yang disempurnakan dan peripheral yang terhubung ke dua *Advanced Peripheral Bus* (APB) bus, tiga bus *Advanced High-performance Bus* (AHB) dan matriks bus multi-AHB 32-bit. Spesifikasi dari mikrokontroler STM32F407VGT6 dapat dilihat dalam Tabel 2.2

Tabel 2. 2 Spesifikasi mikrokontroler STM32F407VGT6

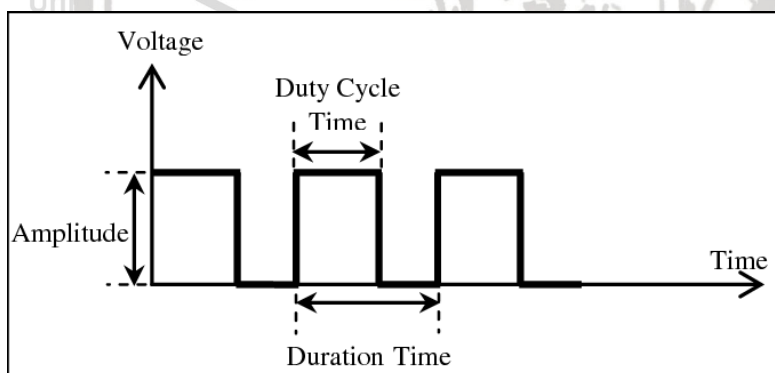
Sumber: Rahmat. A, 2019

Core	Cortex-M4F 8
Memory	1MB Flash
RAM	192KB SRAM
Debug Mode	Serial Wire Debug (SWD) & JTAG Interface
Package	LQFP100
I/O Pins	82
Timer (16-bit)	12
Advanced Control Timers	2

General Purpose Timers	10
Basic Timers	2
PWM Channels	6
ADC (12-bit)	3 (16 Channel)
12C (TWI)	3
USART	4
SPI	3 Full Duplex
DMA	2 (8Channel)
USB	1 (2,0 Full Speed)
CAN	2 (2,0 Active)
Suplly Voltage	1,8 – 36V

2.6 Pulse Width Modulation(PWM)

Pulse Width Modulation(PWM) adalah sebuah teknik modulasi dengan cara memanipulasi lebar pulsa dengan frekuensi dan amplitudo yang sama untuk mendapatkan tegangan rata-rata yang berbeda. Sinyal PWM memiliki amplitude dan frekuensi yang tetap namun memiliki lebar pulsa / *duty cycle* yang bervariasi antara 0% hingga 100%. Duty cycle merupakan perbandingan antara pulsa positif terhadap total pulsa yang dinyatakan dalam persen. Duty cycle pada sinyal PWM dapat dilihat dalam Gambar 2.9.



Gambar 2.9 Duty Cycle Pada Sinyal PWM

Sumber: Mokhtar, 2013

2.7 Kontroler

Kontroler adalah sebuah subsistem dinamis yang dimasukkan ke dalam suatu sistem untuk memanipulasi suatu persamaan matematis sebuah *plant*. Secara umum kontroler juga dikenal dengan istilah kompensator, pengendali (Ogata, 2010).

Salah satu fungsi kontroler adalah mengurangi *error* yang merupakan selisih antara nilai *setpoint* dengan keluaran *plant*. *Setpoint* adalah nilai referensi atau nilai yang diinginkan, sedangkan keluaran *plant* adalah nilai aktual yang terukur pada keluaran *plant*.

Semakin kecil nilai *error* maka kinerja sistem kontrol dinilai semakin baik.

Prinsip kerja kontroler adalah membandingkan nilai keluaran *plant* dengan nilai *setpoint*, menentukan nilai *error* dan akhirnya menghasilkan sinyal kontrol untuk meminimalkan *error* (Ogata, 2010).

2.6.1 Kontroler Proporsional

Kontroler proporsional adalah sebuah kontroler yang memiliki karakteristik mempercepat keluaran. Hubungan antara keluaran kontroler $u(t)$ dan sinyal *error* $e(t)$ ditunjukkan dalam persamaan berikut:

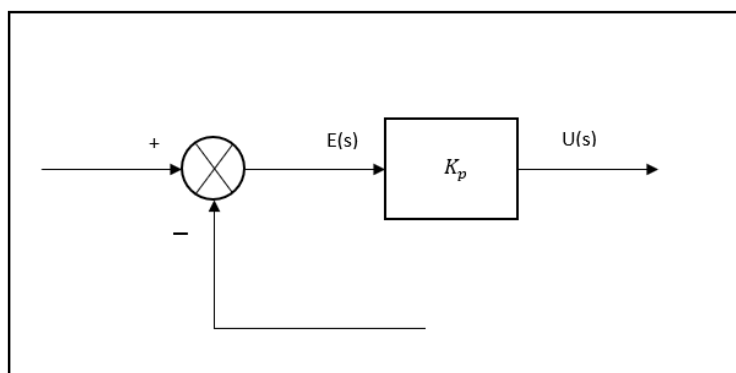
$$u(t) = K_p e(t) \quad (2-3)$$

atau, dalam fungsi alih

$$\frac{U(s)}{E(s)} = K_p$$

dimana K_p adalah penguatan

Diagram blok kontrol proporsional (P) ditunjukkan dalam Gambar 2.10.



Gambar 2.10 Diagram Blok Kontroler Proporsional

Sumber: Ogata K., 2010

Apapun wujud mekanisme yang sebenarnya dan apapun bentuk daya penggerakannya, kontroler proporsional pada dasarnya merupakan penguat dengan penguatan yang dapat diatur (Ogata K., 2010).

2.6.2 Kontroler Integral

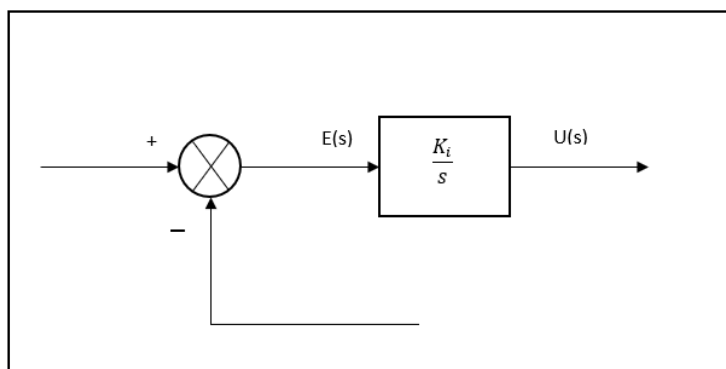
Kontroler *integral* memiliki kemampuan untuk mengurangi *error* yang diakibatkan oleh kontroler proporsional. Keluaran kontroler $u(t)$ diubah dengan laju yang sebanding dengan *error* $e(t)$. Persamaan kontroler integral (I) ditunjukkan dalam persamaan berikut (Ogata K., 2010).

$$\frac{du(t)}{dt} = K_i e(t)$$

$$u(t) = K_i \int_0^t e(t) dt$$

$$\frac{U(s)}{E(s)} = \frac{K_i}{s} \quad (2-4)$$

yang merupakan fungsi alih kontroler integral (I), dengan K_i adalah konstanta integral yang dapat diubah nilainya. Jika $e(t)$ bernilai nol, maka nilai $m(t)$ tetap konstan. Aksi kontrol integral biasa disebut dengan kontrol reset. Gambar 2.11 menunjukkan diagram blok kontroler integral (I).



Gambar 2.11 Diagram Blok Kontroler Integral
Sumber: Ogata K., 2010

2.6.3 Kontroler PI (Proporsional Integral)

Kontroler *proporsional integral* (PI) memiliki kemampuan untuk mempercepat keluaran dan mengurangi *error*. Persamaan kontroler proporsional integral (PI) adalah

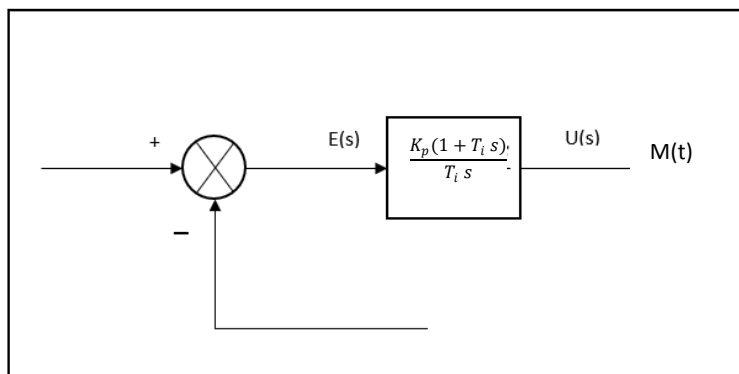
$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (2-5)$$

Adapun fungsi alihnya adalah

$$\begin{aligned} \frac{U(s)}{E(s)} &= K_p \left(1 + \frac{1}{T_i s} \right) \\ &= \frac{K_p (1 + T_i s)}{T_i s} \end{aligned}$$

Dengan K_p penguatan proporsional dan T_i disebut waktu integral, yang keduanya dapat ditentukan. Waktu integral mengatur aksi kontrol internal sedangkan perubahan nilai K_p berakibat pada bagian aksi kontrol proporsional maupun integral.

Gambar 2.12 menunjukkan diagram blok kontroler proporsional integral (PI)

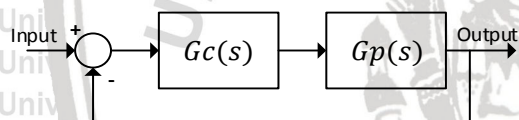


Gambar 2.12 Diagram Blok Kontroler PI (Proporsional Integral)

Sumber: Ogata, K. 2010

2.7 Root Locus

Rancangan sistem kontrol loop tertutup menggunakan *Root Locus* memungkinkan untuk mengatur sekurang-kurangnya beberapa letak *pole* sistem *loop* tertutup sehingga dapat mengatur tanggapan transien pada tingkat tertentu dan pengaruhnya terhadap tanggapan keadaan mantap (Philips, 1996). Prosedur analitis perancangan kontroler PID menggunakan metode *Root Locus* (Charles L. Philips dan Royce D. Harbour, 2010). Sistem kontrol loop tertutup dapat dilihat dalam Gambar 2.13.



Gambar 2.13 Sistem kontrol loop tertutup

Sumber: Charles. L. Philips, 2010

Untuk sistem tersebut, persamaan karakteristik diberikan pada persamaan (2-6).

$$1 + G_c(s)G_p(s) = 0 \quad (2-6)$$

Misalkan diinginkan *root locus* melalui $s = s_1$, maka hasil dari persamaan ditunjukkan pada persamaan (5)

$$G_c(s_1)G_p(s_1) = -1$$

$$G_c(s_1)|G_p(s_1)|e^{j\psi} = 1e^{j\pi} \quad (2-7)$$

Fungsi alih kontroler PID setelah ditransformasi laplace dinyatakan oleh persamaan (2-8).

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \quad (2-8)$$

Perhitungan dari persamaan (2-8) ditunjukkan pada persamaan (2-9).

$$G_c(s_1) = \frac{1}{|G_p(s_1)|} e^{j(\pi-\psi)} \quad (2-9)$$

Substitusi persamaan (2-9) kedalam persamaan (2-8) didapatkan persamaan (2-10).

$$Kd(s_1)^2 + Kp(s_1) + Ki = \frac{e^{j(\pi-\psi)}}{|Gp(s_1)|} \quad (2-10)$$

Dengan

$$s_1 = |s_1|e^{j\beta} \quad (2-11)$$

Hasil dari substusi persamaan (2-11) ke persamaan (2-10) akan didapatkan persamaan (2-12).

$$\begin{aligned} Kd|s_1|^2(\cos 2\beta + j \sin 2\beta) + Kp|s_1|(\cos \beta + j \sin \beta) + Ki \\ = \frac{|s_1|}{|Gp(s_1)|} [\cos(\beta + \pi - \psi) \\ + j \sin(\beta + \pi - \psi)] \end{aligned} \quad (2-12)$$

Menyamakan real dengan real dan imajiner dengan imajiner, didapatkan hasil pada persamaan (2-13).

$$\begin{bmatrix} |s_1|^2 \cos 2\beta & |s_1| \cos \beta \\ |s_1|^2 \sin 2\beta & |s_1| \sin \beta \end{bmatrix} \begin{bmatrix} Kd \\ Kp \end{bmatrix} = \begin{bmatrix} \frac{|s_1|}{Gp(s_1)} \cos(\beta + \pi + \psi) - Ki \\ \frac{|s_1|}{Gp(s_1)} \sin(\beta + \pi + \psi) \end{bmatrix} \quad (2-13)$$

Atau dapat ditunjukkan pada persamaan (2-14).

$$\begin{bmatrix} |s_1|^2 \cos 2\beta & |s_1| \cos \beta \\ |s_1|^2 \sin 2\beta & |s_1| \sin \beta \end{bmatrix} \begin{bmatrix} Kd \\ Kp \end{bmatrix} = \begin{bmatrix} -\frac{|s_1|}{Gp(s_1)} \cos(\psi - \beta) - Ki \\ \frac{|s_1|}{Gp(s_1)} \sin(\psi - \beta) \end{bmatrix} \quad (2-14)$$

Dari persamaan tersebut dilihat bahwa untuk perancangan kontroler PID, satu dari tiga penguatan Kp, Ki dan Kd harus ditentukan terlebih dahulu. Sedangkan untuk perancangan kontroler PI maka Kd dibuat sama dengan nol, sudut dari ψ sama dengan 0.

Maka didapatkan persamaan (2-15)

$$Kp = \frac{s_1}{|Gp(s_1)| \cos \beta} - \frac{Ki \cos \beta}{|s_1|} \quad (2-15)$$

BAB III

METODE PENELITIAN

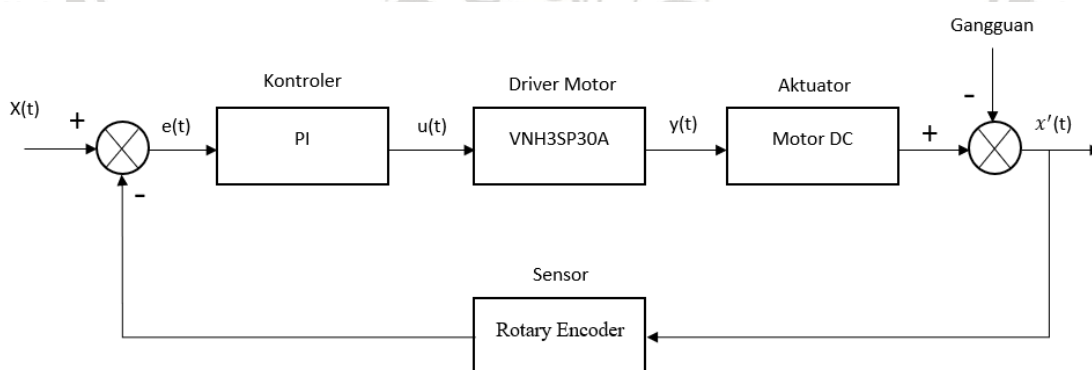
Untuk menyelesaikan rumusan masalah dan merealisasikan tujuan penelitian yang terdapat di bab pendahuluan maka diperlukan langkah-langkah untuk menyelesaikan masalah tersebut. Metode penelitian pada skripsi ini meliputi:

1. Perancangan blok diagram sistem
2. Spesifikasi desain
3. Karakterisasi setiap subsistem
4. Pembuatan perangkat keras
5. Pembuatan algoritma dan program
6. Pengujian alat
7. Pengambilan kesimpulan dan saran

Metode penelitian tersebut dijelaskan sebagai berikut

3.1 Perancangan Blok Diagram Sistem

Perancangan diagram blok sistem menjelaskan sistem secara garis besar dan diharapkan alat dapat bekerja sesuai dengan spesifikasi desain. Blok diagram sistem kontrol kecepatan dapat dilihat dalam Gambar 3.1.



Gambar 3.1 Blok Diagram Sistem

Keterangan:

1. $x(t)$ merupakan *setpoint* (nilai masukan sistem yang diinginkan pada keluaran sistem) dalam pengujian ini berupa kecepatan motor (RPM).
2. $e(t)$ adalah nilai selisih antara *setpoint* dengan *keluaran*.
3. $u(t)$ adalah sinyal terkontrol 0-5V (Volt).
4. $y(t)$ adalah tegangan terkontrol 0-24V (Volt)
5. $x'(t)$ adalah keluaran aktual motor yaitu kecepatan motor (RPM)

3.2 Spesifikasi Desain

Spesifikasi desain yang diinginkan yaitu:

1. *Settling time* kurang dari 2 detik, karena pada sistem diharapkan dengan menggunakan kontroler PI keluaran motor DC PG45 dapat mencapai kecepatan yang diinginkan dengan cepat.

2. *Error steady state* (e_{ss}) kurang dari 5%

Karena sistem yang baik memiliki *error steady state* kurang dari 5%

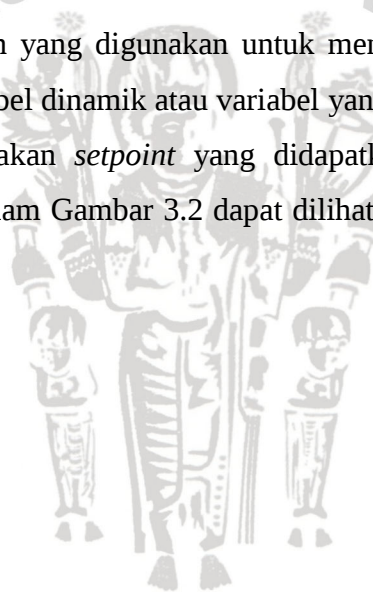
3. *Overshoot* kurang dari 5%

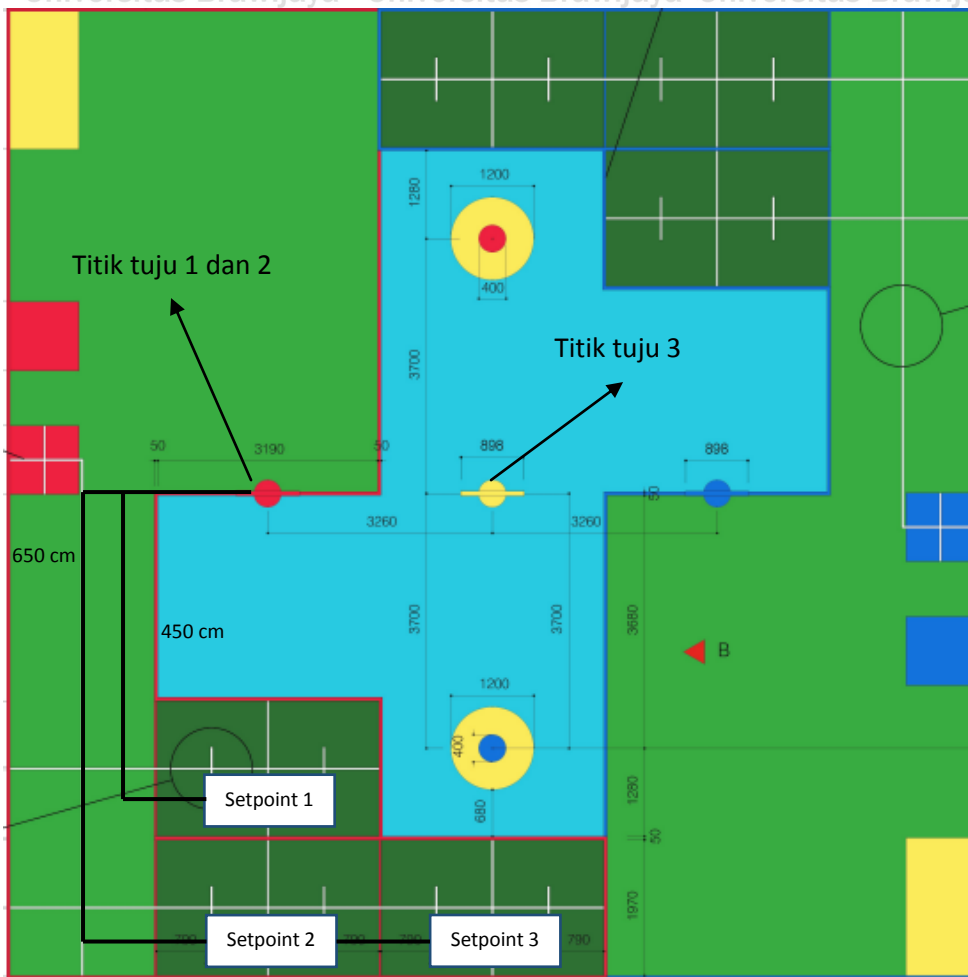
Overshoot kurang dari 5%, karena suatu sistem dapat dikatakan baik ketika *overshoot* dibawah 10%. Dengan adanya kontroler PI, sistem didesain untuk *overshoot* tidak lebih dari 4%.

3.3 Penentuan Setpoint

Setpoint adalah elemen yang digunakan untuk menyatakan nilai yang dikehendaki atau nilai referensi dari variabel dinamik atau variabel yang dikendalikan dari suatu sistem.

Dalam penelitian ini digunakan *setpoint* yang didapatkan dari perhitungan jarak dan kecepatan awal lontaran. Dalam Gambar 3.2 dapat dilihat jarak dari tempat robot pelontar dan tempat yang akan dituju.





Gambar 3.2 Lapangan lomba

Untuk mencari kecepatan lontar maka dibutuhkan rumus gerak parabola.

Kecepatan awal benda bergerak dalam sumbu x dan y saat $t = 0$

$$V_x = V \cos \theta \quad (1)$$

$$V_y = V \sin \theta \quad (2)$$

Untuk mencari jarak pada detik t digunakan rumus

$$x = V_x t \quad (3)$$

Dengan memasukkan nilai x = jarak yang diinginkan maka didapatkan waktu pada jarak x

$$t = \frac{x}{V_x} \quad (4)$$

Untuk mencari ketinggian pada detik t digunakan rumus

$$y = h + V_y t - \frac{1}{2} g t^2 \quad (5)$$

Sumber: Mikrajudin Abdullah, 2016

Dengan substitusi persamaan 4 ke persamaan 5 maka didapatkan V awal lontar lalu dikonversikan ke RPM. Dengan menggunakan perhitungan rumus diatas dapat dilihat dalam Tabel 3.1

Tabel 3.1

Setpoint	Ketinggian (m)	Jarak	Kecepatan (RPM)	Setpoint
1	2,4	450	100	
2	2,4	650	125	
3	3,4	650	150	

3.4 Karakterisasi Motor DC

a) Tujuan

Mengetahui karakteristik motor DC dengan perubahan tegangan masukan yang diberikan terhadap kecepatan putar motor DC.

b) Peralatan yang di gunakan

- Motor DC PG45
- Driver motor VNH3SP30
- Power Supply 24V
- Tachometer Digital
- Mikrokontroler STM32F407
- PC
- Software STM Studio

c) Prosedur pengujian

- 1 Pin IN1, IN2 dan PWM pada driver motor dihubungkan ke mikrokontroler STM32F407
- 2 Power pada driver motor dihubungkan dengan power supply 24V
- 3 OUTA dan OUTB pada driver motor dihubungkan dengan motor DC
- 4 Nilai PWM diubah dalam software STM Studio sesuai dengan nilai dalam Tabel 3.1
- 5 Kecepatan motor DC diukur menggunakan tachometer digital
- 6 Hasil kecepatan motor DC dicatat dalam Tabel 3.2

Setelah melakukan prosedur pengujian, didapatkan nilai kecepatan motor DC seperti yang ditunjukkan dalam Tabel 3.2.

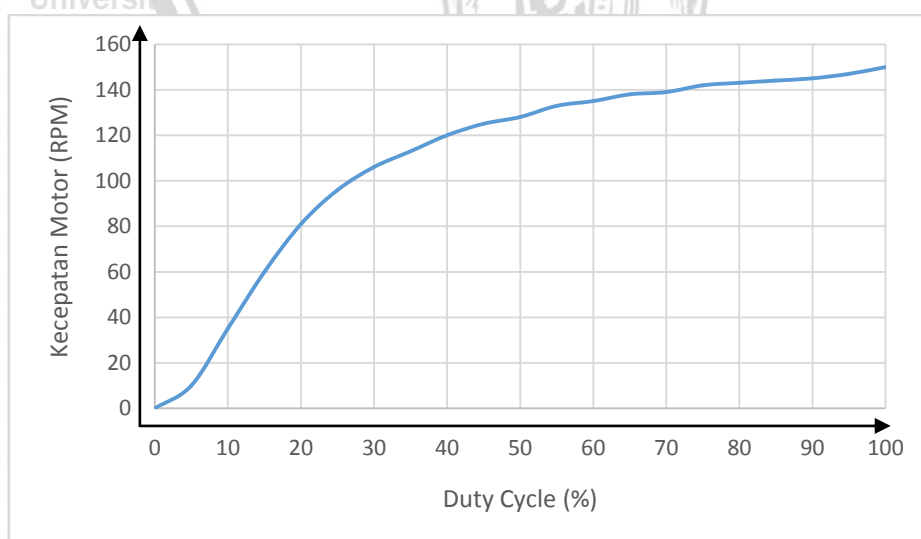
Tabel 3.2 Data pengujian sensor kecepatan rotary encoder

Duty Cycle (%)	PWM	RPM
0	0	0
5	13	10
10	25	35
15	38	60
20	51	81
25	64	96
30	77	106
35	90	113
40	102	120
45	115	125
50	127	128
55	140	133
60	153	135
65	166	138
70	178	139
75	191	142
80	204	143
85	217	144
90	229	145
95	242	147
100	255	150

Gambar pengujian motor DC ditunjukkan dalam Gambar 3.2

Tabel di atas merupakan keluaran motor DC yang diberi masukan tegangan sebesar 24V dan perubahan kecepatan dibaca menggunakan sensor *rotary encoder*. Dari Tabel diatas didapatkan kecepatan putar motor dengan tegangan maksimal 24V sebesar 150 RPM. Dari Tabel di atas agar mempermudah melihat hasil pembacaan sensor maka dibuatlah kurva kecepatan motor (RPM) terhadap *duty cycle* yang dapat dilihat dalam

Gambar 3.3



Gambar 3.3 Pengujian motor DC

Dari pengujian ini bisa dilihat bahwa kecepatan motor berubah dengan nilai yang cukup besar saat diberi masukan *duty cycle* antara 1% sampai 40%.

3.5 Karakteristik Driver Motor

a) Tujuan

Mengetahui karakteristik tegangan keluaran efektif dari driver motor VNH3SP30 terhadap masukan berupa *duty cycle* sinyal PWM yang dibangkitkan oleh mikrokontroler STM32F407.

b) Peralatan yang digunakan

- *Driver* motor VNH3SP30
- *Power Supply* 24V
- Mikrokontroler STM32F407
- Voltmeter Digital
- PC
- *Software* STMStudio

c) Prosedur Pengujian

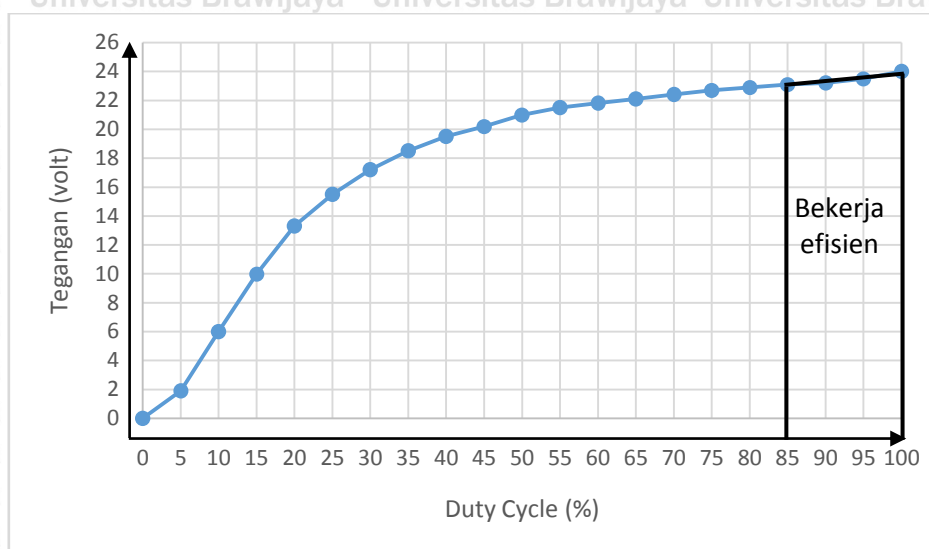
1. Pin IN1, IN2 dan PWM pada *driver* motor dihubungkan ke mikrokontroler STM32F407
2. *Power* pada *driver* motor dihubungkan dengan *power supply* 24V
3. Nilai PWM diubah pada *software* STM Studio sesuai dengan nilai dalam Tabel 3.2
4. Tegangan keluaran pada pin OUTA dan OUTB *driver* motor diukur sesuai dengan nilai PWM yang diberikan dengan voltmeter digital
5. Tegangan keluaran *driver* motor dicatat dalam Tabel 3.3

Pengujian ini dilakukan dengan memberi masukan tegangan 24V pada *driver* motor dan keluaran dari *driver* motor dimasukkan ke motor DC PG45 dan pembacaan kecepatan putar motor di baca menggunakan sensor *rotary encoder*. Data pengujian tegangan keluaran pada *driver* motor dapat dilihat dalam Tabel 3.3

Tabel 3.3 Data pengujian driver motor

Duty Cycle (%)	PWM	RPM	Tegangan (Volt)	Gain
0	0	0	0	0
5	13	10	1,9	2
10	25	35	6	3,5
15	38	60	10	4
20	51	81	13,3	4,05
25	64	96	15,5	3,84
30	77	106	17,2	3,53
35	90	113	18,5	3,23
40	102	120	19,5	3
45	115	125	20,2	2,77
50	127	128	21	2,56
55	140	133	21,5	2,42
60	153	135	21,8	2,25
65	166	138	22,1	2,12
70	178	139	22,4	1,98
75	191	142	22,7	1,8
80	204	143	22,9	1,78
85	217	144	23,1	1,69
90	229	145	23,2	1,61
95	242	147	23,5	1,54
100	255	150	24	1,5

Tabel di atas merupakan hasil dari pengujian driver VNH3SP30 dengan masukan berupa tegangan 24V. Dari data tersebut didapatkan *gain* yang merupakan hasil dari membagi kecepatan putar motor (RPM) dengan *Duty cycle*. penelitian ini bekerja dengan efisien saat *gain* driver motor memiliki nilai yang tidak jauh beda. Dalam Tabel 3.2 dapat dilihat bahwa nilai *gain* driver motor dari *duty cycle* 85 hingga 100% yang memiliki nilai *gain* yang tidak jauh berbeda, oleh karena itu pengujian kali ini menggunakan *duty cycle* antara 85 sampai dengan 100%. Dari Tabel tersebut maka digambarkan kurva tegangan terhadap *duty cycle* yang dapat dilihat dalam Gambar 3.4

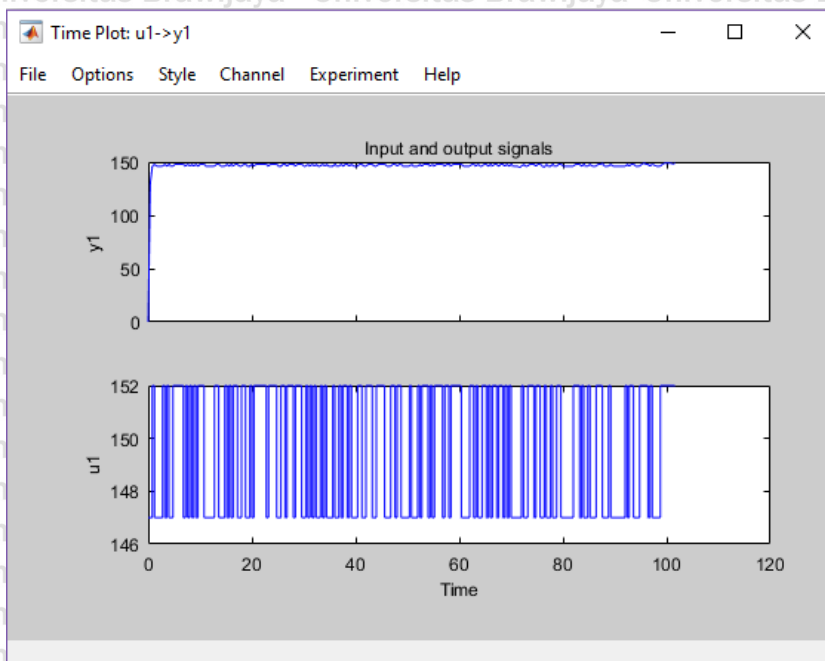


Gambar 3.4 Hasil pengujian driver motor

3.6 Penentuan Fungsi Alih Motor DC

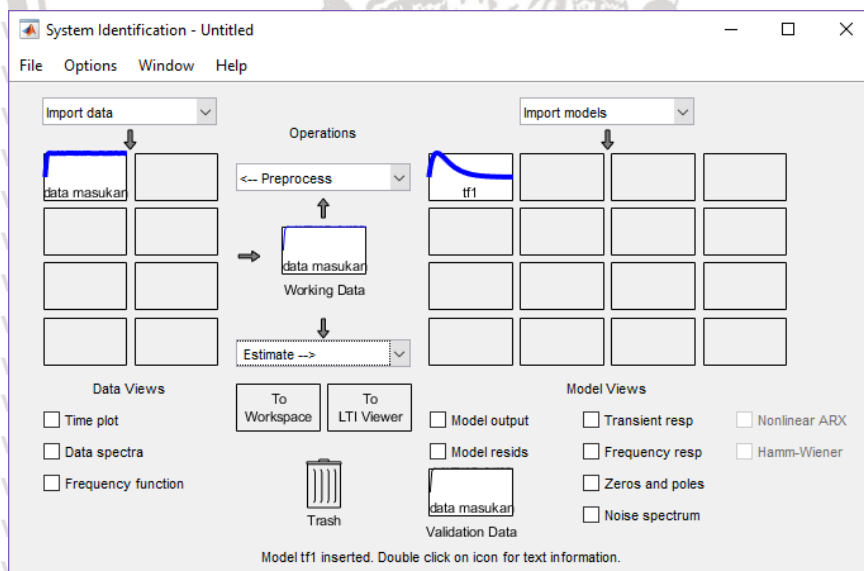
Pada pengontrolan kecepatan pelontar *shuttlecock* dibutuhkan kontroler agar mampu mengontrol kecepatan motor DC sehingga sesuai dengan spesifikasi desain yang telah ditentukan. Motor DC yang digunakan dalam penelitian ini masih belum diketahui fungsi alihnya. Fungsi alih akan digunakan untuk menentukan parameter dari kontroler PI pada kontrol kecepatan motor DC sehingga perlu dicari fungsi alih dari motor DC. Karakteristik motor DC didapatkan dengan cara memberikan masukan *unit step*. Fungsi alih akan didapatkan dari pemodelan dengan cara mencari keluaran dari motor yang diberi masukan berupa sinyal PRBS. Langkah-langkah yang dilakukan untuk menentukan fungsi alih dengan membangkitkan sinyal PRBS adalah sebagai berikut: (Rahmat. A, 2019)

1. Mencari nilai yang linier dari hasil kecepatan motor terhadap duty cycle PWM.
2. Memasukkan nilai batas atas dan batas bawah berdasarkan nilai yang linier untuk membangkitkan sinyal PRBS.
3. Sinyal PRBS yang telah dibangkitkan kemudian digunakan sebagai masukan motor DC.
4. Setelah didapatkan data sinyal PRBS dan data kecepatan motor DC, selanjutnya adalah melakukan identifikasi dengan menggunakan software matlab seperti dalam Gambar 3.5



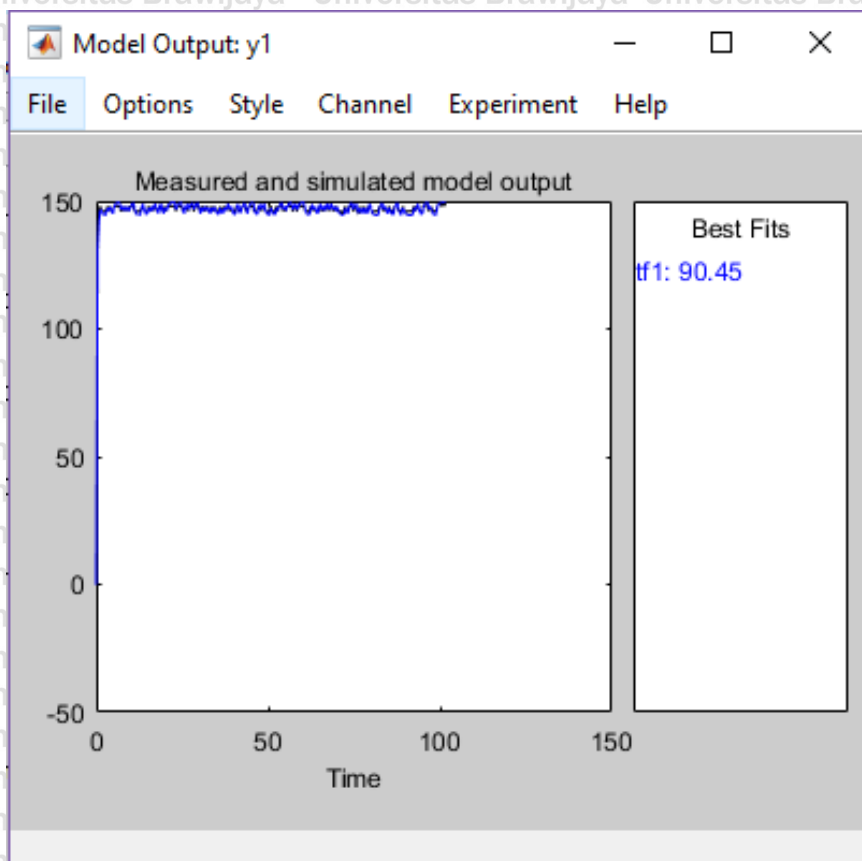
Gambar 3.5 Sinyal Masukan PRBS dan Keluaran Motor DC

6. Dengan menggunakan rlocusplot pada matlab. Data keluaran dari motor yang didapatkan dari masukan berupa sinyal PRBS dan data kecepatan motor DC. Kemudian di-import pada block *System Identification Toolbox* seperti dalam Gambar 3.6



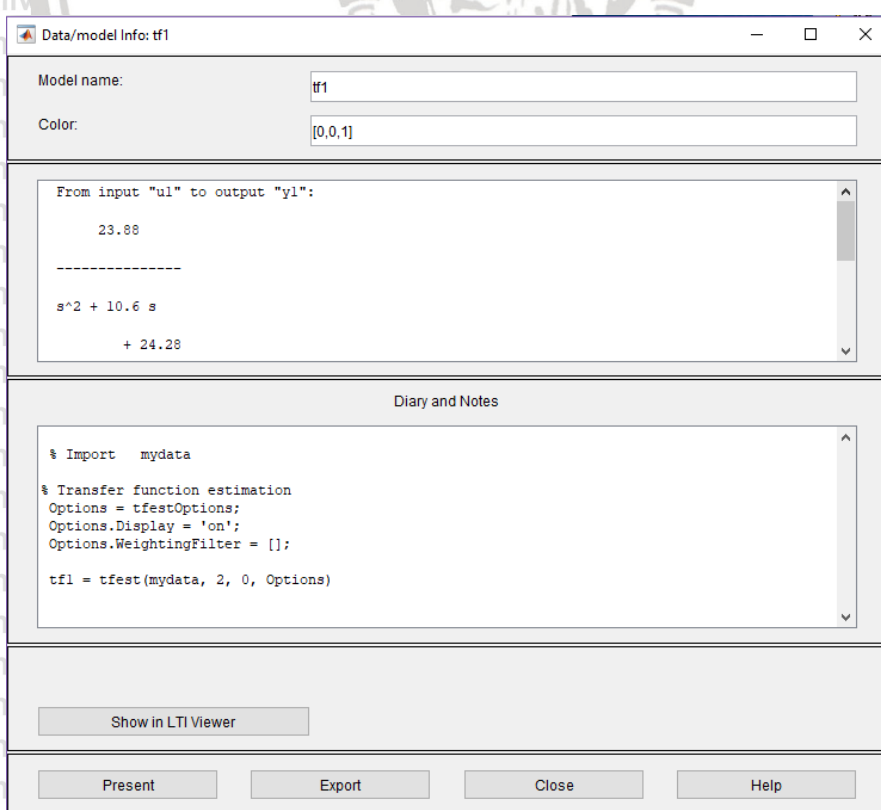
Gambar 3.6 System Identification Toolbox

7. Setelah melakukan beberapa estimasi model berdasarkan data yang telah didapatkan dari data yang telah di-import, didapatkan fungsi alih motor DC dengan *best fits* sebesar 90,45% seperti dalam Gambar 3.7



Gambar 3.7 Hasil Estimasi Model

Model *continuous* fungsi alih motor DC dapat dilihat dalam Gambar 3.8

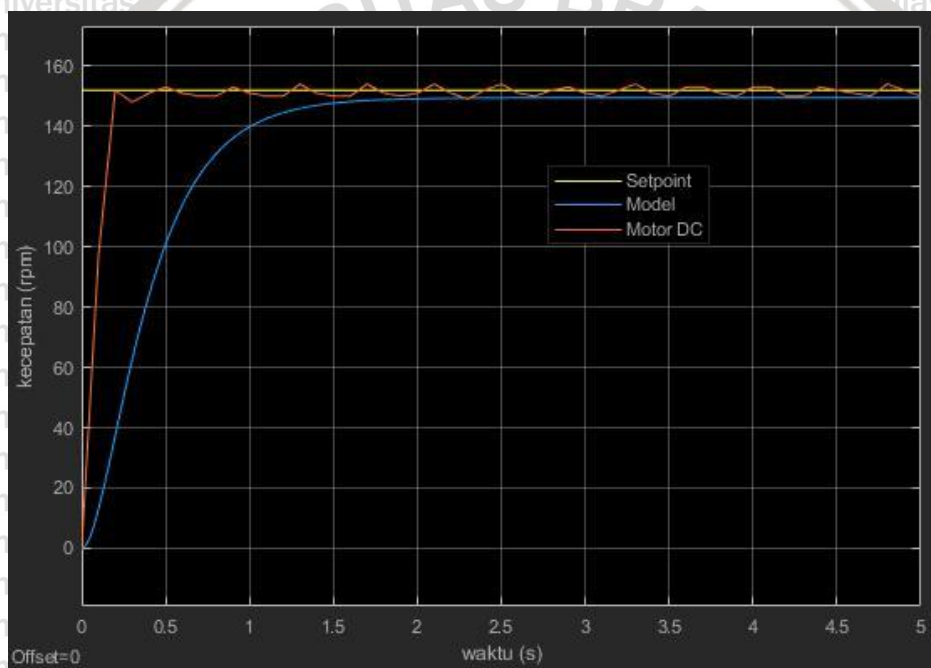


Gambar 3.8 Model *continuous* fungsi alih motor DC

8. Dari hasil identifikasi fungsi alih motor DC yang didapat adalah:

$$\frac{\theta(s)}{U(s)} = \frac{23,88}{s^2 + 10,6 s + 24,28} \quad (3-1)$$

Setelah didapatkan fungsi alih maka fungsi alih harus divalidasi karena suatu fungsi alih dapat dikatakan mewakili *plant* apabila keluaran fungsi alih yang diperoleh dari proses identifikasi hampir menyerupai keluaran *plant*. Proses validasi dilakukan dengan membandingkan nilai keluaran fungsi alih dan nilai dari kecepatan motor DC dari pembacaan sensor kecepatan yaitu *rotary encoder* sebagai proses validasi. Proses ini dilakukan dengan memberikan masukan *unit step* pwm 100% yaitu kecepatan sebesar 150 *Rotation Per Minute* (RPM). Keluaran motor DC ditunjukkan dalam Gambar 3.9



Gambar 3.9 Validasi fungsi alih motor DC

3.7 Penentuan Parameter Kontroler PI dengan Metode *Root Locus*

Metode *Root Locus* atau letak kedudukan akar digunakan untuk meneliti perilaku sistem dengan parameter sistem berubah pada lingkup tertentu. Rancangan dimaksudkan agar letak pole dan zero dari fungsi alih loop tertutup terletak pada daerah yang ditentukan.

Agar sistem stabil maka pole dan zero harus terletak pada bidang *s* disebelah kiri sumbu imajiner. Untuk memenuhi tujuan performansi sistem yang diinginkan, maka perlu ditambahkan kontroler pada sistem tersebut. Kontroler yang digunakan adalah kontroler proporsional integral (PI).

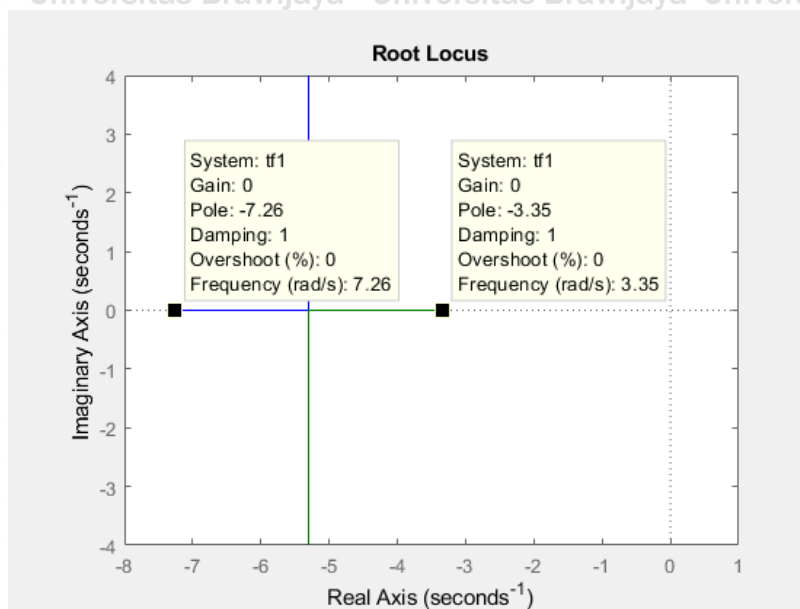
Langkah-langkah dalam penentuan parameter kontroler dengan metoder *Root Locus* adalah sebagai berikut:

1. Mencari fungsi alih sistem, pada penelitian ini digunakan identifikasi dengan membangkitkan sinyal PRBS (*Pseudo Random Binary Sequence*).
2. Setelah didapatkan fungsi alih, maka diperiksa kestabilannya dengan *Root Locus* yang ada pada Matlab. Sistem yang stabil adalah sistem dengan fungsi alih yang penyebutnya mempunyai akar (pole) pada bidang s sebelah kiri sumbu imajiner.
3. Menentukan letak pole s_1 sesuai dengan spesifikasi desain yang telah ditentukan sebelumnya.
4. Setelah itu mencari nilai parameter K_p dan K_i pada Matlab dengan mensubstitusi nilai s_1 dan nilai fungsi alih sistem dalam persamaan (3-3).
5. Untuk mendapatkan nilai parameter K_p dan K_i , maka variasikan terlebih dahulu nilai parameter K_i kemudian akan didapatkan nilai dari parameter K_p yang dilakukan pada program matlab.

Setelah didapatkan fungsi alih sistem yaitu:

$$\frac{\theta(s)}{U(s)} = \frac{23,88}{s^2 + 10,6s + 24,28} \quad (3-1)$$

Maka selanjutnya adalah mentukan letak pole loop tertutup. Penentuan letak pole pada *Root Locus* harus sesuai dengan spesifikasi desain yang telah ditentukan. Pada penelitian ini spesifikasi desain yang ditentukan adalah *settling time* kurang dari 2 detik, *overshoot* kurang dari 5% serta rasio redaman sama dengan 1. Pada penelitian ini digunakan nilai $s_1 = -7,26$ Penentuan letak pole *Root Locus* dapat dilihat dalam Gambar 3.10 (Rahmat. A, 2019)



Gambar 3.10 Penentuan pole pada diagram *Root Locus*

Setelah ditentukan letak pole yang diinginkan maka selanjutnya mensubstitusi nilai pole s_1 dan nilai fungsi alih sistem pada persamaan (3-1) dan memvariasikan nilai K_i , maka akan didapatkan nilai dari parameter PI dalam Tabel 3.4. Pencarian parameter K_p dan K_i dengan menggunakan Matlab ditunjukkan pada listing program berikut.

```
clc
s1 = -7.26;
KI = [1 1.5 2 2.5 3 3.5 4 4.5 5 5.5];
plant_num = [0 0 23.8801];
plant_den = [1 10.6035 24.2815];
slmag = abs(s1);
beta = angle(s1);
plant_al = polyval(plant_num,s1)/polyval(plant_den,s1);
plantslmag = abs(plant_al);
psi = angle(plant_al);
t = 0:1:20:300;
for k = 1:10
    KP(k) = -slmag/(plantslmag*cos(beta))-KI(k)*cos(beta)/slmag;
    Nilai_KI = KI(k);
    Gcnum = [KP KI(k)];
    Gcden = [0 1];
    hold on
end
hold off
```

Hasil pencarian parameter K_p dan K_i dari perhitungan pada program diatas ditunjukkan dalam Tabel 3.4.

Tabel 3.4 Nilai parameter K_p dan K_i

K_i	K_p
1	0,169341
1,5	0,238212
2	0,307082
2,5	0,375953
3	0,444823
3,5	0,513694
4	0,582564
4,5	0,651435
5	0,720305
5,5	0,789176

Untuk memvalidasi perhitungan matlab maka dilakukan perhitungan manual dengan memasukkan nilai $s_1 = -7,26$, $\beta = 3,1416$, $\psi = 0$, $K_i = 1$ yang disubstitusikan ke persamaan (2-15).

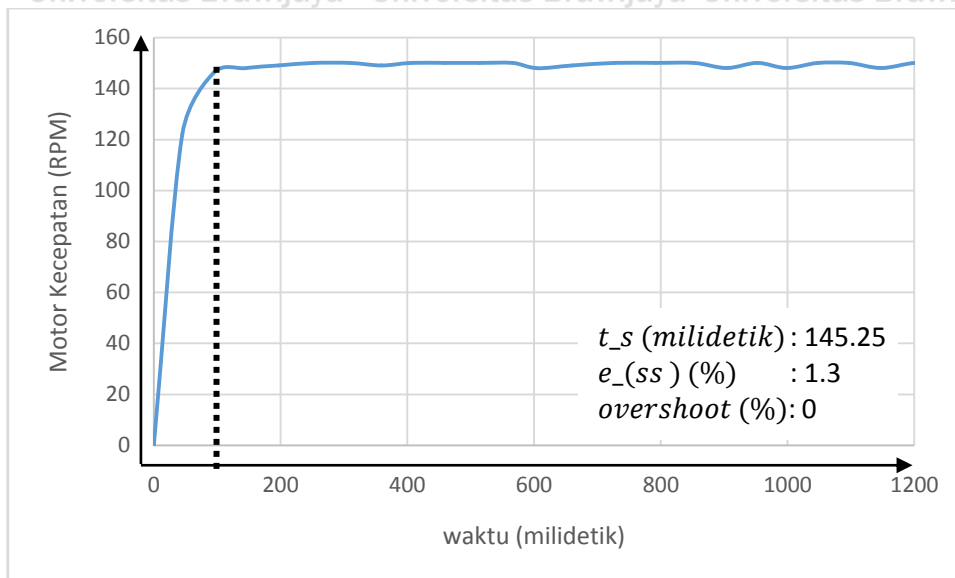
$$K_p = \frac{s_1}{|Gp(s_1)| \cos \beta} - \frac{K_i \cos \beta}{|s_1|} \quad (2-15)$$

$$K_p = \frac{-7,26}{\left(\frac{23,88}{(-7,26)^2 + 10,6(-7,26) + 24,28} \right) \cos 3,1416} - \frac{1 \cos 3,1416}{|-7,26|}$$

$$K_p = \frac{-7,26}{-229,7468} + 0,1377$$

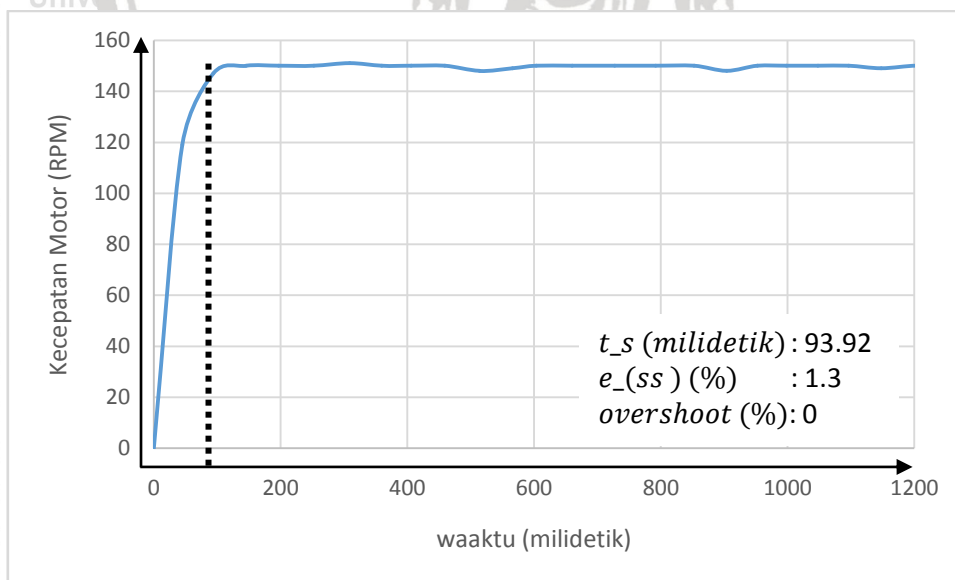
$$K_p = 0,1693$$

Setelah didapatkan nilai parameter K_p dan K_i dari hasil perhitungan kemudian dilakukan pengujian terhadap sistem untuk mencari nilai parameter yang sesuai dengan sistem. Gambar 3.11 s.d. Gambar 3.20 adalah keluaran sistem dengan nilai parameter kontrad dalam Tabel 3.4.



Gambar 3. 11 Keluaran sitem dengan nilai K_p 0,169341 dan K_i 1

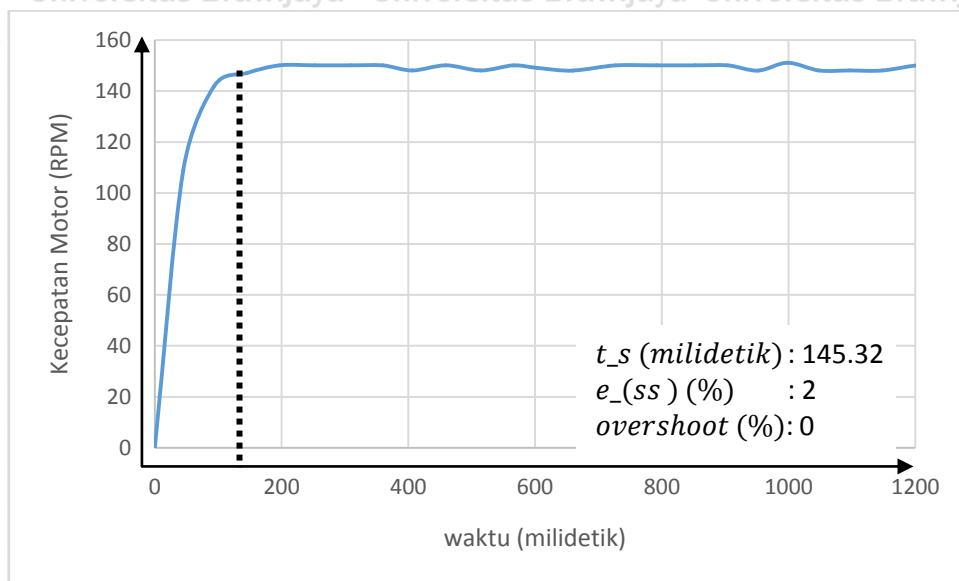
Gambar 3.11 Merupakan keluaran sistem yang di gambarkan dalam bentuk kurva kecepatan motor terhadap waktu garis putus-putus menandakan titik dimana keluaran sistem mencapai *setling time*. Dari kurva di atas didapatkan nilai *setling time* sebesar 120,25 milidetik dan *error steady state* sebesar 1,3% yang didapatkan dari $\frac{|\text{keluaran sistem} - \text{setpoint}|}{\text{keluaran sistem}}$, dan overshoot sebesar 0%. Keluaran dari sistem berubah-ubah sesaat setelah mencapai *setling time* dikarenakan *error* dari pembacaan *rotary encoder*.



Gambar 3. 12 Keluaran sistem dengan nilai K_p 0,238212 dan K_i 1,5

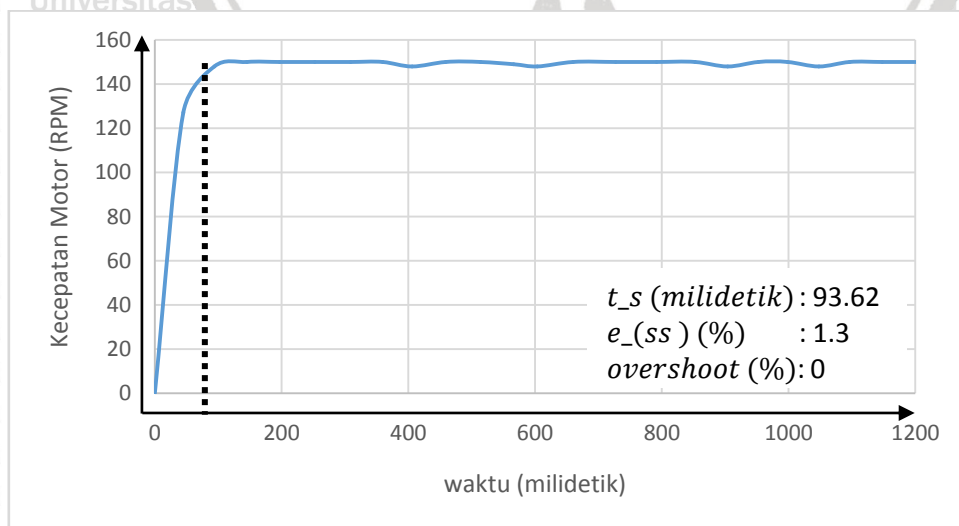
Gambar 3.12 Merupakan keluaran sistem yang di gambarkan dalam bentuk kurva kecepatan motor terhadap waktu garis putus-putus menandakan titik dimana keluaran sistem mencapai *setling time*. Dari kurva di atas didapatkan nilai *setling time* sebesar

93,92milidetik dan *error steady state* sebesar 1,3% yang didapatkan dari $\frac{|\text{keluaran sistem} - \text{setpoint}|}{\text{keluaran sistem}}$, dan overshoot sebesar 0%. Keluaran dari sistem berubah-ubah sesaat setelah mencapai *setling time* dikarenakan *error* dari pembacaan *rotary encoder*.



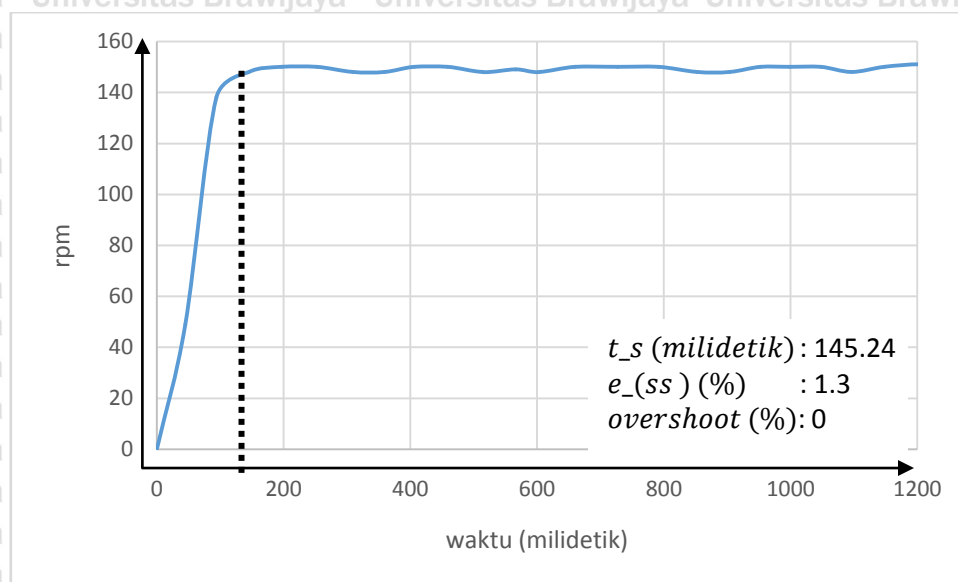
Gambar 3. 13Keluaran sistem dengan nilai K_p 0,307082 dan K_i 2

Gambar 3.13 Merupakan keluaran sistem yang di gambarkan dalam bentuk kurva kecepatan motor terhadap waktu garis putus-putus menandakan titik dimana keluaran sistem mencapai *setling time*. Dari kurva di atas didapatkan nilai *setling time* sebesar 145,32 milidetik dan *error steady state* sebesar 2% yang didapatkan dari $\frac{|\text{keluaran sistem} - \text{setpoint}|}{\text{keluaran sistem}}$, dan overshoot sebesar 0%. Keluaran dari sistem berubah-ubah sesaat setelah mencapai *setling time* dikarenakan *error* dari pembacaan *rotary encoder*.



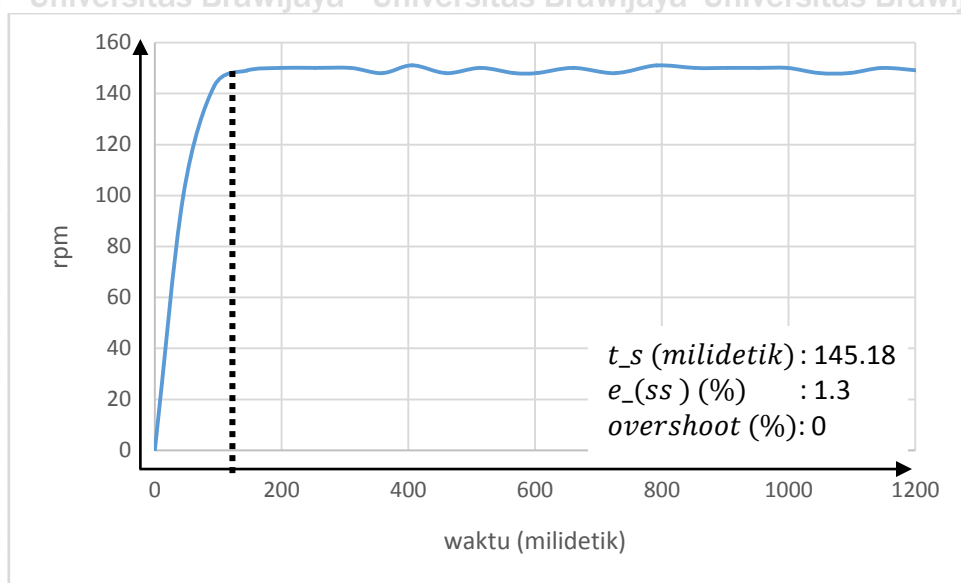
Gambar 3. 14Keluaran sistem dengan nilai K_p 0,375953 dan K_i 2,5

Gambar 3.14 Merupakan keluaran sistem yang di gambarkan dalam bentuk kurva kecepatan motor terhadap waktu garis putus-putus menandakan titik dimana keluaran sistem mencapai *setling time*. Dari kurva di atas didapatkan nilai *setling time* sebesar 93,62 milidetik dan *error steady state* sebesar 1,3% yang didapatkan dari $\frac{|\text{keluaran sistem} - \text{setpoint}|}{\text{keluaran sistem}}$, dan overshoot sebesar 0%. Keluaran dari sistem berubah-ubah sesaat setelah mencapai *setling time* dikarenakan *error* dari pembacaan *rotary encoder*.



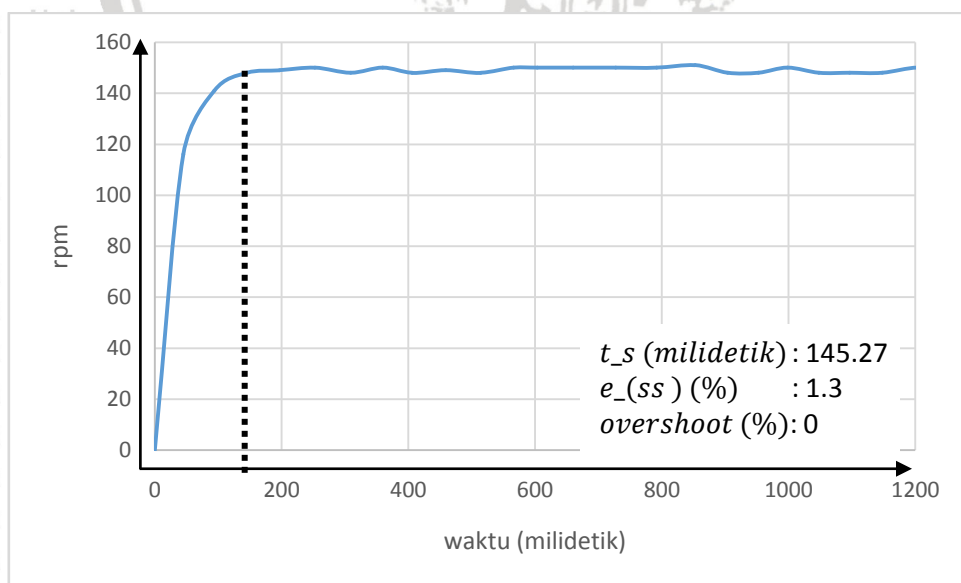
Gambar 3. 15Keluaran sistem dengan nilai K_p 0,444823 dan K_i 3

Gambar 3.15 Merupakan keluaran sistem yang di gambarkan dalam bentuk kurva kecepatan motor terhadap waktu garis putus-putus menandakan titik dimana keluaran sistem mencapai *setling time*. Dari kurva di atas didapatkan nilai *setling time* sebesar 145,24 milidetik dan *error steady state* sebesar 1,3% yang didapatkan dari $\frac{|\text{keluaran sistem} - \text{setpoint}|}{\text{keluaran sistem}}$, dan overshoot sebesar 0%. Keluaran dari sistem berubah-ubah sesaat setelah mencapai *setling time* dikarenakan *error* dari pembacaan *rotary encoder*.



Gambar 3. 16 Keluaran sistem dengan nilai K_p 0,513694 dan K_i 3,5

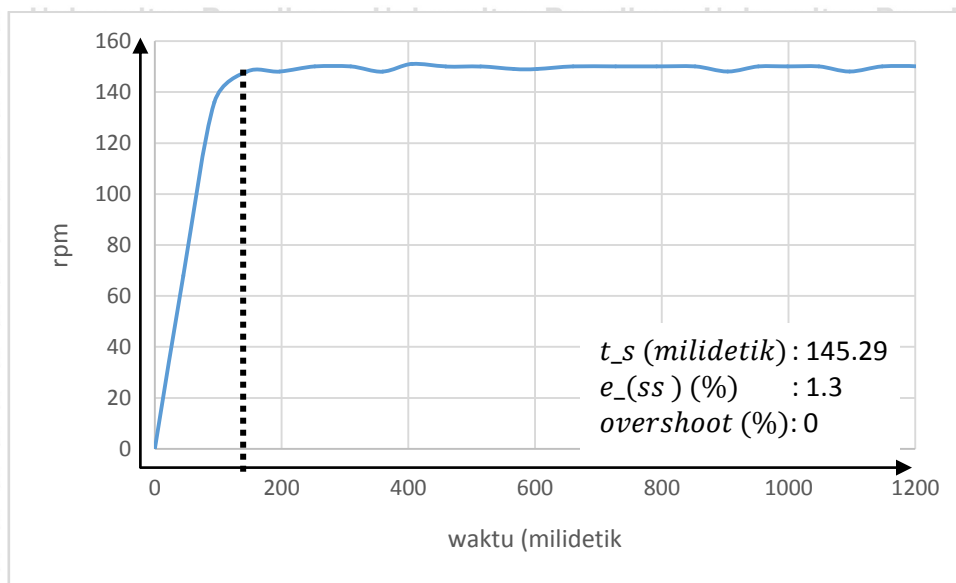
Gambar 3.16 Merupakan keluaran sistem yang di gambarkan dalam bentuk kurva kecepatan motor terhadap waktu garis putus-putus menandakan titik dimana keluaran sistem mencapai *setling time*. Dari kurva di atas didapatkan nilai *setling time* sebesar 145,32 milidetik dan *error steady state* sebesar 2% yang didapatkan dari $\frac{|\text{keluaran sistem} - \text{setpoint}|}{\text{keluaran sistem}}$, dan overshoot sebesar 0%. Keluaran dari sistem berubah-ubah sesaat setelah mencapai *setling time* dikarenakan *error* dari pembacaan *rotary encoder*.



Gambar 3. 17 Keluaran sistem dengan nilai K_p 0,582564 dan K_i 4

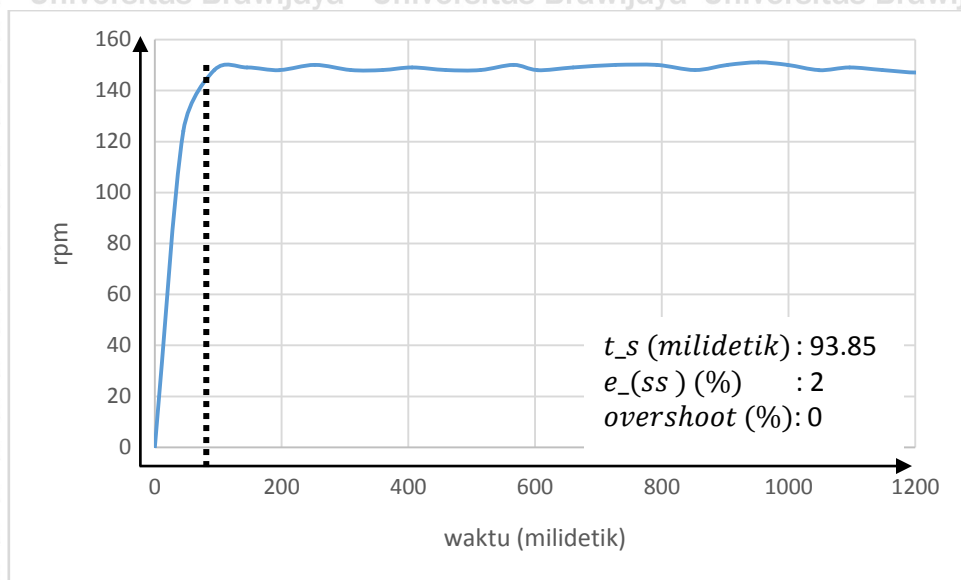
Gambar 3.17 Merupakan keluaran sistem yang di gambarkan dalam bentuk kurva kecepatan motor terhadap waktu garis putus-putus menandakan titik dimana keluaran

sistem mencapai *setling time*. Dari kurva di atas didapatkan nilai *setling time* sebesar 145,27 milidetik dan *error steady state* sebesar 1,3% yang didapatkan dari $\frac{|\text{keluaran sistem} - \text{setpoint}|}{\text{keluaran sistem}}$, dan overshoot sebesar 0%. Keluaran dari sistem berubah-ubah sesaat setelah mencapai *setling time* dikarenakan *error* dari pembacaan *rotary encoder*.



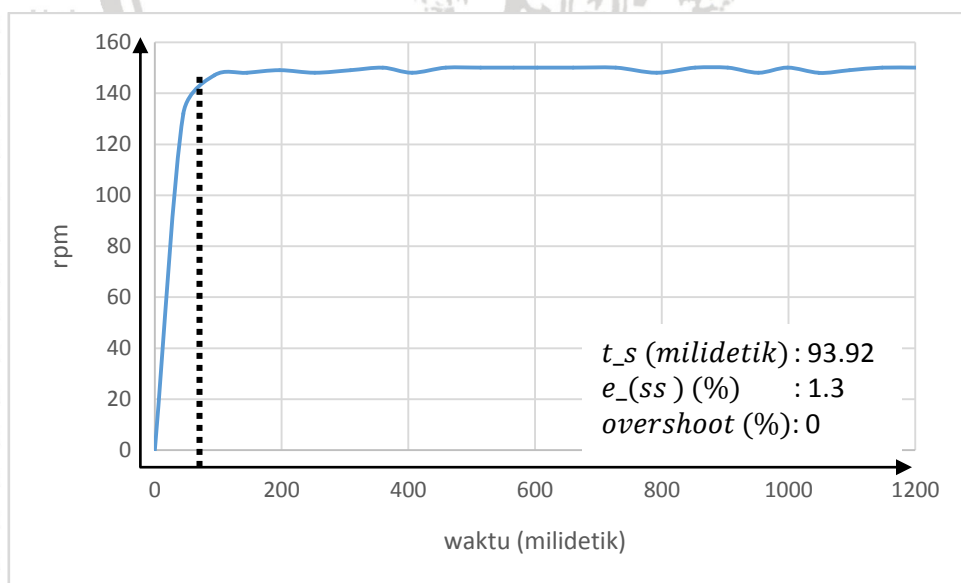
Gambar 3. 18 Keluaran sistem dengan nilai K_p 0,651435 dan K_i 4,5

Gambar 3.18 Merupakan keluaran sistem yang di gambarkan dalam bentuk kurva kecepatan motor terhadap waktu garis putus-putus menandakan titik dimana keluaran sistem mencapai *setling time*. Dari kurva di atas didapatkan nilai *setling time* sebesar 145,29 milidetik dan *error steady state* sebesar 1,3% yang didapatkan dari $\frac{|\text{keluaran sistem} - \text{setpoint}|}{\text{keluaran sistem}}$, dan overshoot sebesar 0%. Keluaran dari sistem berubah-ubah sesaat setelah mencapai *setling time* dikarenakan *error* dari pembacaan *rotary encoder*.



Gambar 3. 19 Keluaran sistem dengan nilai K_p 0,720305 dan K_i 5

Gambar 3.19 Merupakan keluaran sistem yang di gambarkan dalam bentuk kurva kecepatan motor terhadap waktu garis putus-putus menandakan titik dimana keluaran sistem mencapai *setling time*. Dari kurva di atas didapatkan nilai *setling time* sebesar 93,85 milidetik dan *error steady state* sebesar 2% yang didapatkan dari $\frac{|\text{keluaran sistem} - \text{setpoint}|}{\text{keluaran sistem}}$, dan overshoot sebesar 0%. Keluaran dari sistem berubah-ubah sesaat setelah mencapai *setling time* dikarenakan *error* dari pembacaan *rotary encoder*.



Gambar 3. 20 Keluaran sistem dengan nilai K_p 0,789176 dan K_i 5,5

Gambar 3.20 Merupakan keluaran sistem yang di gambarkan dalam bentuk kurva kecepatan motor terhadap waktu garis putus-putus menandakan titik dimana keluaran

sistem mencapai *setling time*. Dari kurva di atas didapatkan nilai *setling time* sebesar 93,32 milidetik dan *error steady state* sebesar 1,3% yang didapatkan dari $\frac{|\text{keluaran sistem} - \text{setpoint}|}{\text{keluaran sistem}}$, dan overshoot sebesar 0%. Keluaran dari sistem berubah-ubah sesaat setelah mencapai *setling time* dikarenakan *error* dari pembacaan rotary encoder.

Data pengujian parameter kontroler ditunjukkan dalam Tabel 3.5

Tabel 3.5 Data Pengujian Parameter Kontroler

K_p	K_i	t_s (milidetik)	e_{ss} (%)	overshoot (%)
0,169341	1	120,25	1,3	0
0,238212	1,5	93,92	1,3	0
0,307082	2	145,32	2	0
0,375953	2,5	93,62	1,3	0
0,444823	3	145,24	1,3	0
0,513694	3,5	145,18	1,3	0
0,582564	4	145,27	1,3	0
0,651435	4,5	145,29	1,3	0
0,720305	5	93,85	2	0
0,789176	5,5	93,92	1,3	0

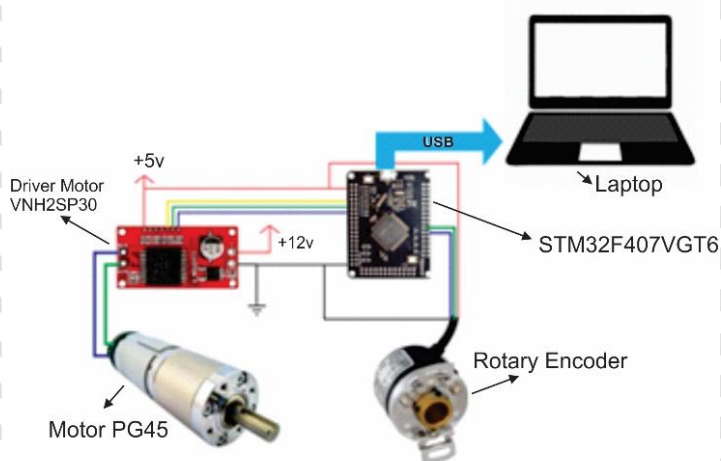
Berdasarkan data pengujian nilai parameter kontroler dalam Tabel 3.4 maka digunakan nilai K_p 0,375953 dan K_i 2,5 karena memiliki keluaran terbaik dan memiliki nilai *error steady state* (e_{ss}) yang kecil.

3.8 Pembuatan Perangkat Keras

Pembuatan perangkat keras dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta pemrogramannya. Hal ini dimaksudkan agar sistem pengontrolan kecepatan motor DC dapat bekerja sesuai dengan deskripsi awal yang telah direncanakan. Pembuatan perangkat keras yang dilakukan pada penelitian ini meliputi:

a. Skema pembuatan perangkat keras

Skema pembuatan perangkat keras ditujukan untuk penempatan dalam menyambungkan antar komponen agar mengurangi *error* pemasangan. Gambar skema pembuatan perangkat keras dapat dilihat dalam Gambar 3.21



Gambar 3.21 Skema pembuatan perangkat keras

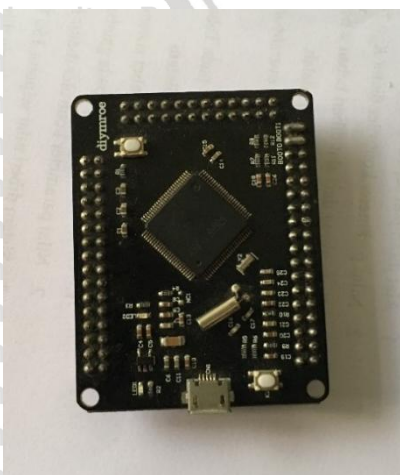
b. Penentuan modul elektronik yang digunakan

1. Baterai LiPo 3S 12V digunakan sebagai catu daya driver motor dan di seri dengan jenis yang sama agar mendapatkan tegangan 24V. bentuk fisik dari baterai LiPo 3S 12V dapat dilihat dalam Gambar 3.22



Gambar 3.22 Baterai LiPo 3S

2. Mikrokontroler STM32F407 sebagai perangkat kontroler dapat dilihat dalam Gambar 3.23



Gambar 3.23 Mikrokontroler STM32F407

3. Sensor rotary encoder sebagai pembaca posisi dapat dilihat dalam Gambar 3.24



Gambar 3.24 Sensor rotary encoder

Sumber: www.omron.com

4. Driver motor VNH3SP30 untuk menggerakkan motor dapat dilihat dalam Gambar 3.25



Gambar 3.25 Driver motor VNH3SP30

5. Motor DC planetary gearbox dapat dilihat dalam Gambar 3.26



Gambar 3.26 Motor DC planetary gearbox

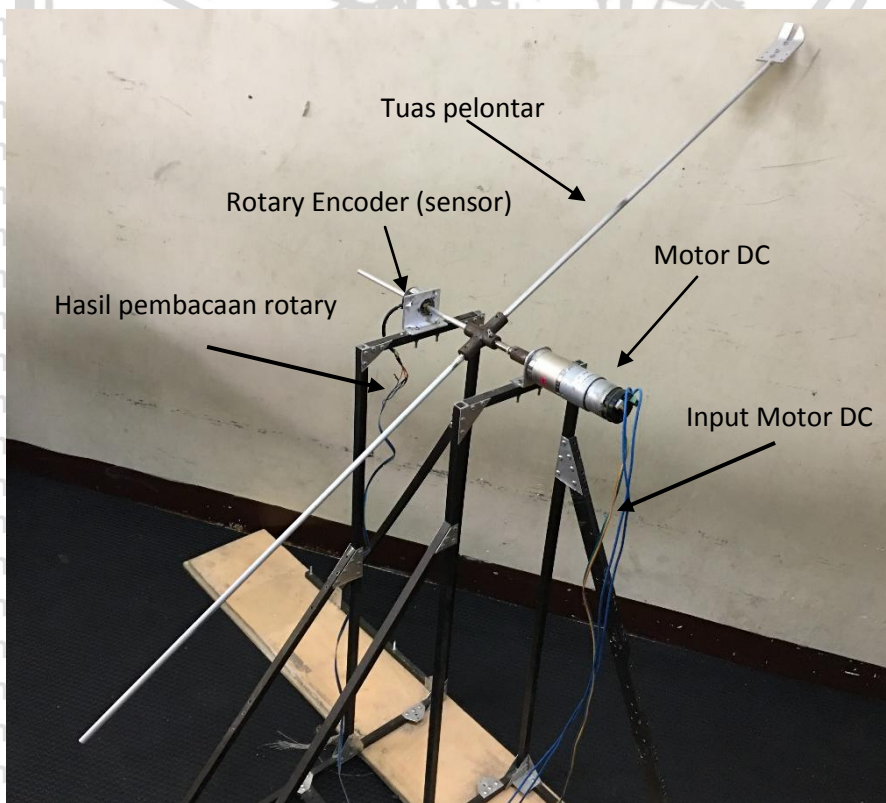
6. PC atau komputer yang sudah terinstall CoIDE, STMStudio dan Matlab dalam dilihat dalam Gambar 3.27



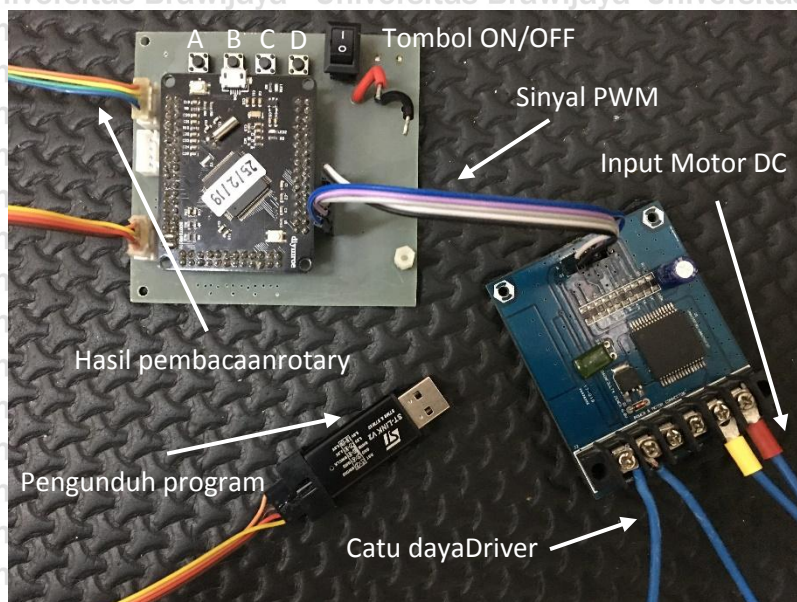
Gambar 3.27 PC atau komputer

3.7.1. Foto Alat

Bentuk fisik dari alat yang dibuat seperti ditunjukkan dalam Gambar 3.28 dan pengkabelan alat dapat dilihat dalam Gambar 3.29



Gambar 3.28 Foto alat tampak depan



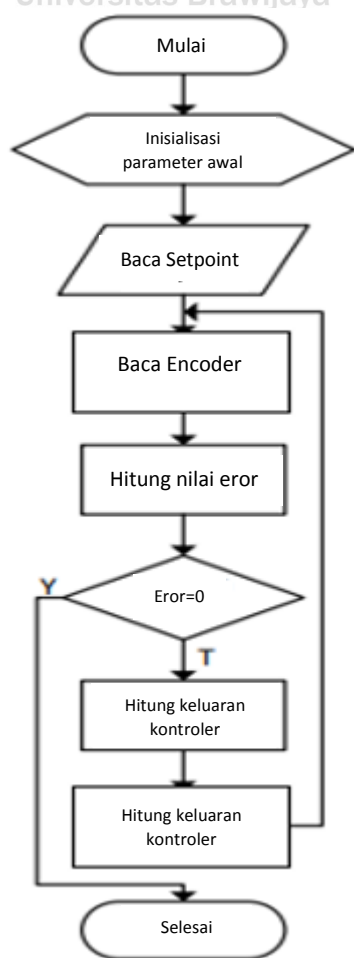
Gambar 3.29 Foto pengkabelan alat

3.7.2. Prinsip Kerja Sistem

1. LiPo 3S dua buah sebagai catu daya driver motor DC dengan tegangan 12V dan dipasang seri
2. Catu daya mikrokontroler STM32F407 diambil dari PC dengan menghubungkan pada port USB
3. Pada shaft motor DC dan shaft *Rotary encoder* disambungkan menggunakan Alumunium 8mm dan shaft berbahan besi.
4. Pembacaan kecepatan putar motor DC menggunakan *rotary encoder* yang disambungkan dengan shaft pada motor DC.
5. Menekan tombol ON pada papan elektrik.
6. Menekan tombol B sampai dengan D sesuai setpoint yang diinginkan.
7. Memasukkan kaitan shuttlecock pada ujung tuas pelontar.
8. Menekan tombol A pada papan elektrik untuk menjalankan perintah menyalakan motor.
9. Keluaran kontroler PI pada mikrokontroler berupa sinyal PWM yang dihubungkan dengan driver motor
10. Hasil keluaran sinyal PWM dari mikrokontroler digunakan untuk menggerakkan motor DC
11. Motor DC berputar sesuai dengan kecepatan atau nilai *setpoint* yang diberikan.
12. Motor DC akan diberi *setpoint*0 saat mencapai 90 derajat.

3.9 Pembuatan Algoritma dan Program

Perancangan perangkat lunak berkaitan dengan flowchart program yang akan diimplementasikan ke mikrokontroler STM32F407VGT6. Gambar 3.29 adalah flowchart sistem secara keseluruhan.



Gambar 3.30 Flowchart Pemrograman

3.10 Perancangan Pengujian Sistem

Pengujian pada penelitian ini meliputi:

a) Pengujian Sistem Tanpa Beban

Prosedur pengujian sistem tanpa beban adalah sebagai berikut:

1. Alat dirangkai seperti dalam Gambar 3.21
2. Nilai parameter kontrol diatur K_p 0,375953 dan K_i 2,5
3. *Setpoint* yang digunakan adalah 100RPM, 125 RPM, 150RPM
4. Nilai *settling time* (t_s) dan *overshoot* dari data log software STM studio dihitung dan dicatat dalam Tabel 4.1

b. Pengujian Sistem Berbeban

Prosedur pengujian sistem berbeban adalah sebagai berikut:

1. Alat dirangkai seperti dalam Gambar 3.21
2. Nilai parameter kontroler diatur K_p 0,375953 dan K_i 2,5
3. Pada ujung dari alat diberi beban sebesar 500 gr
4. *Setpoint* yang digunakan adalah 100RPM, 125 RPM, 150 RPM

1. Nilai *settling time*(t_s) dan *overshoot* dari data log *software* STMStudio dihitung
2. Hasil perhitungan dicatat dalam Tabel 4.2

c. Pengujian Sistem dengan Variasi Beban

Prosedur pengujian sistem variasi beban adalah sebagai berikut

1. Alat dirangkai seperti dalam Gambar 3.21.
2. Nilai parameter kontroler diatur K_p 0,375953 dan K_i 2,5
3. Pada ujung dari alat diberi beban sebesar 308 gr, 318 gr, 325 gr dan 328 gr.
4. *Setpoint* yang digunakan adalah 150 RPM.
5. Nilai *settling time*(t_s) dan *overshoot* dari data log *software* STMStudio dihitung.
6. Hasil perhitungan dicatat dalam Tabel 4.3.

d. Pengujian Sifat Reversibel Sistem

Prosedur pengujian sifat reversibel sistem adalah sebagai berikut:

1. Alat dirangkai seperti dalam Gambar 3.21
2. Nilai parameter kontroler diatur K_p 0,375953 dan K_i 2,5
3. Sistem diberi nilai *setpoint* 150 RPM dan ketika keluaran sudah dalam keadaan *steady state*, sistem diberikan nilai *setpoint* 0 RPM.



BAB IV

HASIL DAN PEMBAHASAN

Dalam bab ini dilakukan pengujian dan analisis sistem setiap blok maupun secara keseluruhan. Pengujian sistem dilakukan untuk mengetahui apakah alat yang telah dibuat berfungsi dengan baik dan sesuai dengan perancangan yang telah ditentukan. Pengujian setiap blok dilakukan untuk menemukan letak kesalahan dan untuk mempermudah dalam melakukan analisis pada sistem apabila tidak bekerja sesuai dengan perancangan. Pengujian pada sistem dibagi menjadi beberapa bagian, yaitu:

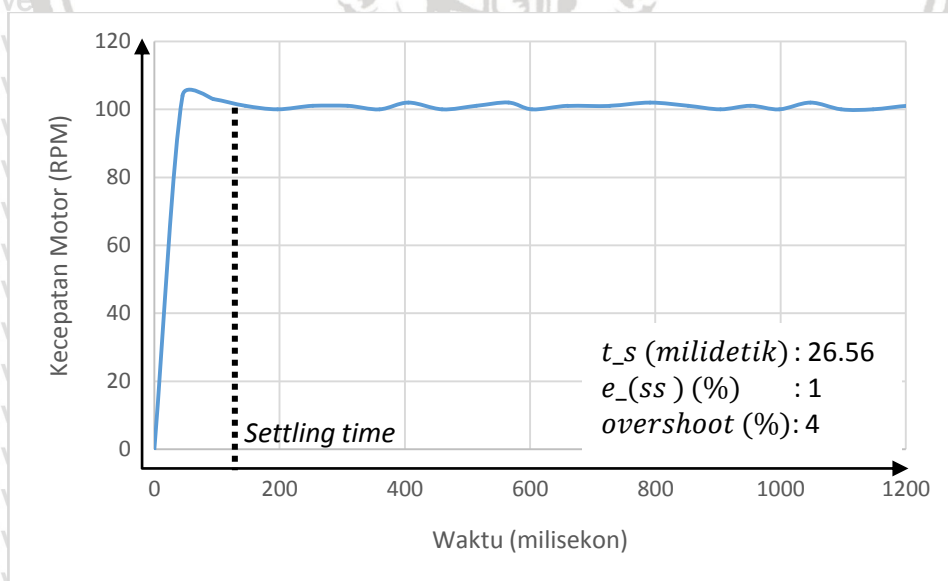
1. Pengujian sistem tanpa beban
2. Pengujian sistem berbeban
3. Pengujian sistem dengan variasi beban
4. Pengujian sifat reversibel sistem

4.1. Pengujian Sistem Tanpa Beban

Pengujian ini dilakukan untuk mengetahui performansi sistem ketika tidak diberi beban. Prosedur pada pengujian ini terdapat dalam bab 3.10.

4.1.1. Pengujian Dengan Setpoint 100RPM

Keluaran sistem pada pengujian sistem tanpa beban dengan setpoint 100RPM ditunjukkan dalam Gambar 4.1.



Gambar 4.1 Keluaran sistem tanpa beban dengan setpoint 100RPM

Gambar 4.1 Merupakan keluaran sistem yang di gambarkan dalam bentuk kurva kecepatan motor terhadap waktu garis putus-putus menandakan titik dimana

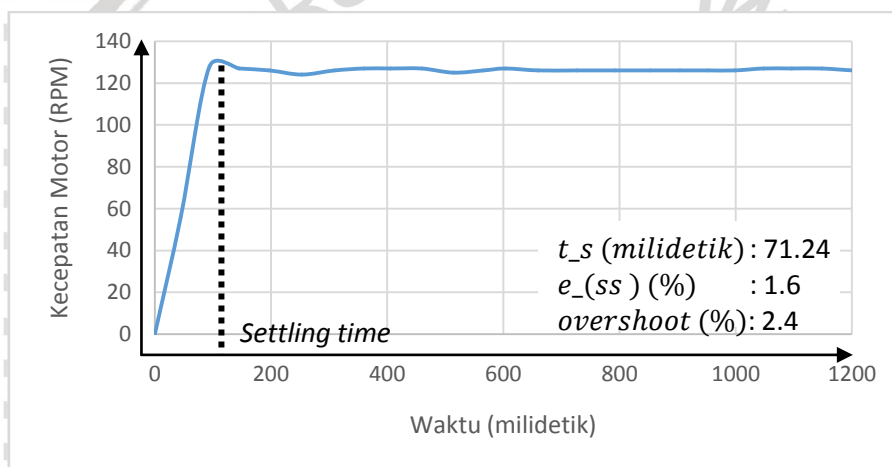
keluaran sistem mencapai *settling time*. Dari pengujian ini didapatkan nilai *settling time* sebesar 145,29milidetik, *overshoot*4% dan *error steady state* 2%. Nilai yang berubah-ubah sesudah *settling time* merupakan eror pembacaan sensor *rotary encoder*.

$$e_{ss}(\%) = \frac{|101 - 100|}{100} \times 100 \% \\ = 1 \%$$

$$\text{overshoot}(\%) = \frac{104 - 100}{100} \times 100 \% \\ = 4 \%$$

4.1.2. Pengujian dengan Setpoint125 RPM

Keluaran sistem pada pengujian sistem tanpa beban dengan *setpoint*125RPM ditunjukkan dalamGambar 4.2.



Gambar 4.2Keluaran sistem tanpa beban dengan *setpoint*125RPM

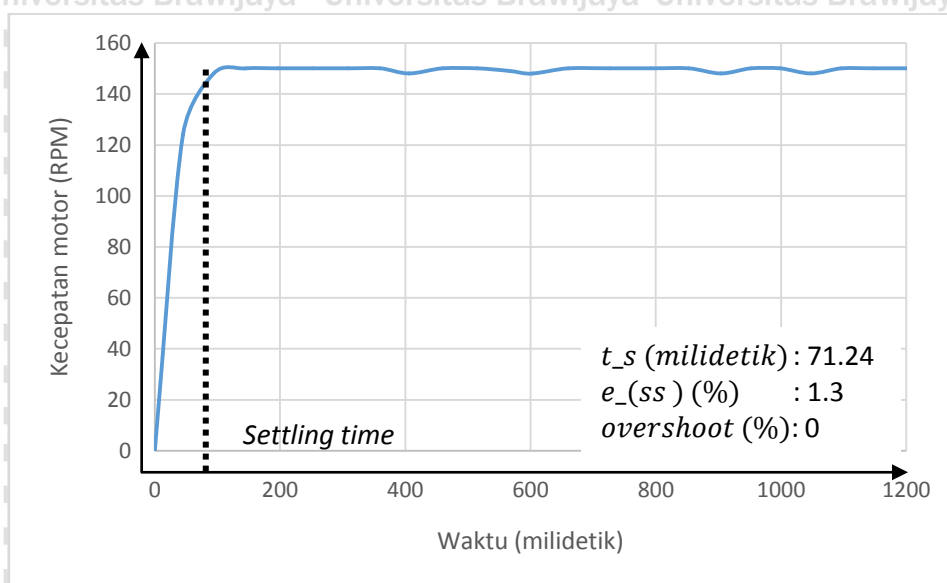
Gambar 4.2 Merupakan keluaran sistem yang di gambarkan dalam bentuk kurva kecepatan motor terhadap waktu garis putus-putus menandakan titik dimana keluaran sistem mencapai *settling time*. Dari pengujian ini didapatkan nilai *settling time* sebesar 145,87milidetik, *overshoot*2.4 % dan *error steady state*1.6%. Nilai yang berubah-ubah sesudah *settling time* merupakan eror pembacaan sensor *rotary encoder*.

$$e_{ss}(\%) = \frac{|127 - 125|}{125} \times 100 \% \\ = 1,6 \%$$

$$\text{overshoot}(\%) = \frac{128 - 125}{125} \times 100 \% \\ = 2,4 \%$$

4.1.3. Pengujian dengan Setpoint 150 rpm

Keluaran sistem pada pengujian sistem tanpa beban dengan setpoint 150 RPM ditunjukkan dalam Gambar 4.3. Nilai yang berubah-ubah sesudah *settling time* merupakan eror pembacaan sensor *rotary encoder*.



Gambar 4.3 Keluaran sistem tanpa beban dengan setpoint 150 RPM

Gambar 4.3 Merupakan keluaran sistem yang di gambarkan dalam bentuk kurva kecepatan motor terhadap waktu garis putus-putus menandakan titik dimana keluaran sistem mencapai *settling time*. Dari pengujian ini didapatkan nilai *settling time* sebesar 93,92 milidetik, *error steady state* 1.3% dan *overshoot* 0%. Nilai yang berubah-ubah sesudah *settling time* merupakan eror pembacaan sensor *rotary encoder*.

$$e_{ss} (\%) = \frac{|148 - 150|}{150} \times 100 \% \\ = 1,3 \%$$

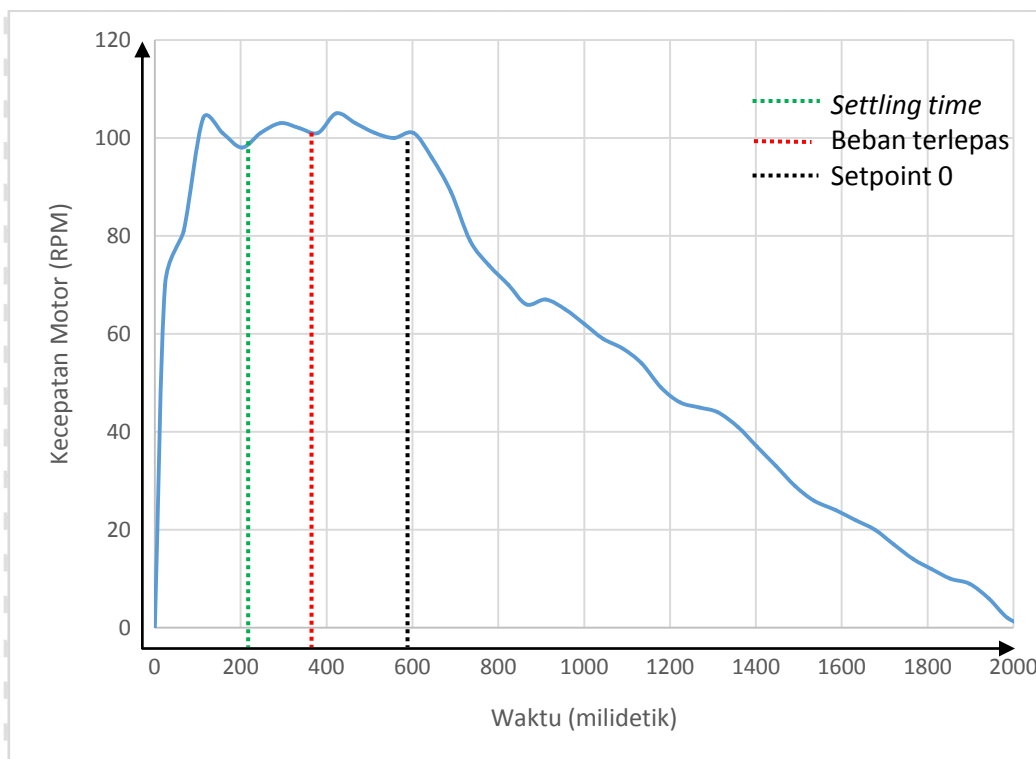
$$overshoot (\%) = \frac{150 - 150}{150} \times 100 \% \\ = 0\%$$

4.2. Pengujian Sistem Berbeban

Pengujian ini dilakukan untuk mengetahui performansi sistem ketika diberikan beban. Prosedur pada pengujian ini terdapat dalam bab 3.10.

4.2.1. Pengujian dengan Setpoint100rpm

Keluaran sistem pada pengujian sistem berbeban dengan *setpoint*100 RPM ditunjukkan dalam Gambar 4.4.



Gambar 4.4 Keluaran sistem berbeban dengan *setpoint*100RPM

Gambar 4.4 Merupakan keluaran sistem yang di gambarkan dalam bentuk kurva kecepatan motor terhadap waktu. Garis putus-putus berwarna hijau merupakan titik dimana keluaran sistem mencapai *settling time*, garis putus-putus berwarna merah merupakan titik dimana beban terlepas dari tuas, dan garis putus-putus berwarna hitam merupakan titik dimana sistem diberi *setpoint* sebesar 0.

Dari pengujian ini didapatkan nilai *settling time* sebesar 246.22 milidetik, *error steady state* 5%, *overshoot* 4% dan ketinggian 2,1m. Sistem diberi *setpoint* 0 saat tuas pelontar melewati 90°. Nilai yang berubah-ubah sesudah *settling time* merupakan eror pembacaan sensor *rotary encoder*.

$$ess (\%) = \frac{|101 - 100|}{100} \times 100 \%$$

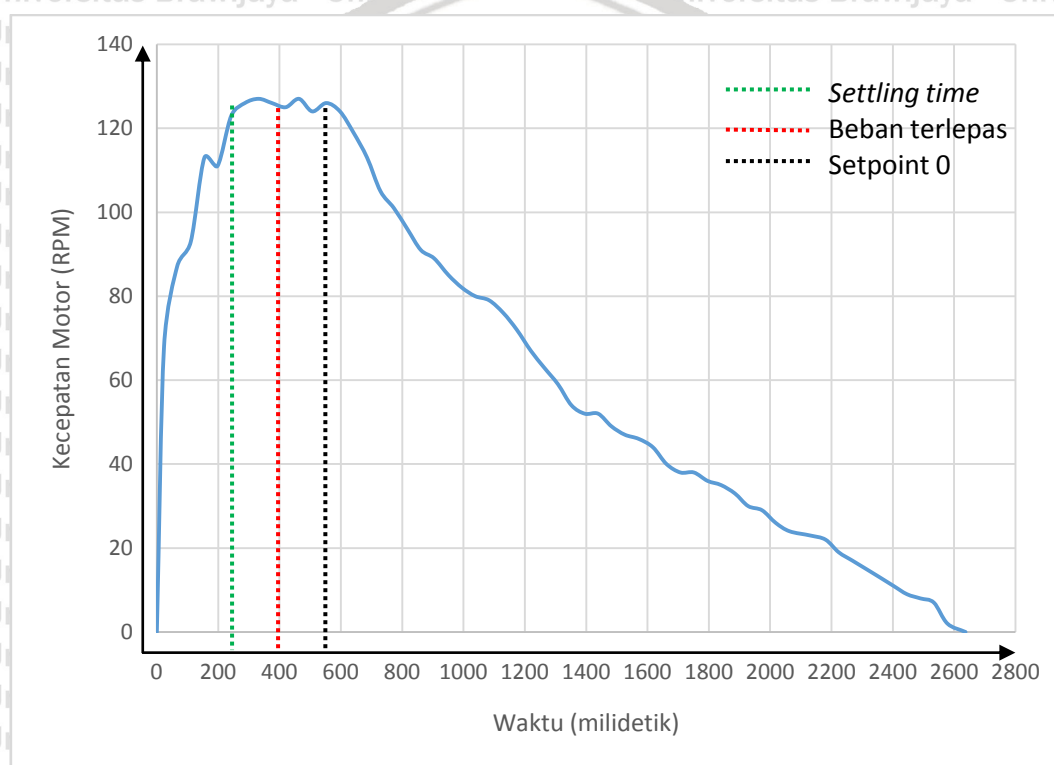
$$= 5 \%$$

$$overshoot (\%) = \frac{104 - 100}{100} \times 100 \%$$

$$= 4 \%$$

4.2.2. Pengujian dengan Setpoint125rpm

Keluaran sistem pada pengujian sistem berbeban dengan *setpoint*125RPM ditunjukkan dalam Gambar 4.5.



Gambar 4.5 Keluaran sistem berbeban dengan *setpoint*125RPM

Gambar 4.5 Merupakan keluaran sistem yang di gambarkan dalam bentuk kurva kecepatan motor terhadap waktu. Garis putus-putus berwarna hijau merupakan titik dimana keluaran sistem mencapai *settling time*, garis putus-putus berwarna merah merupakan titik dimana beban terlepas dari tuas, dan garis putus-putus berwarna hitam merupakan titik dimana sistem diberi *setpoint* sebesar 0.

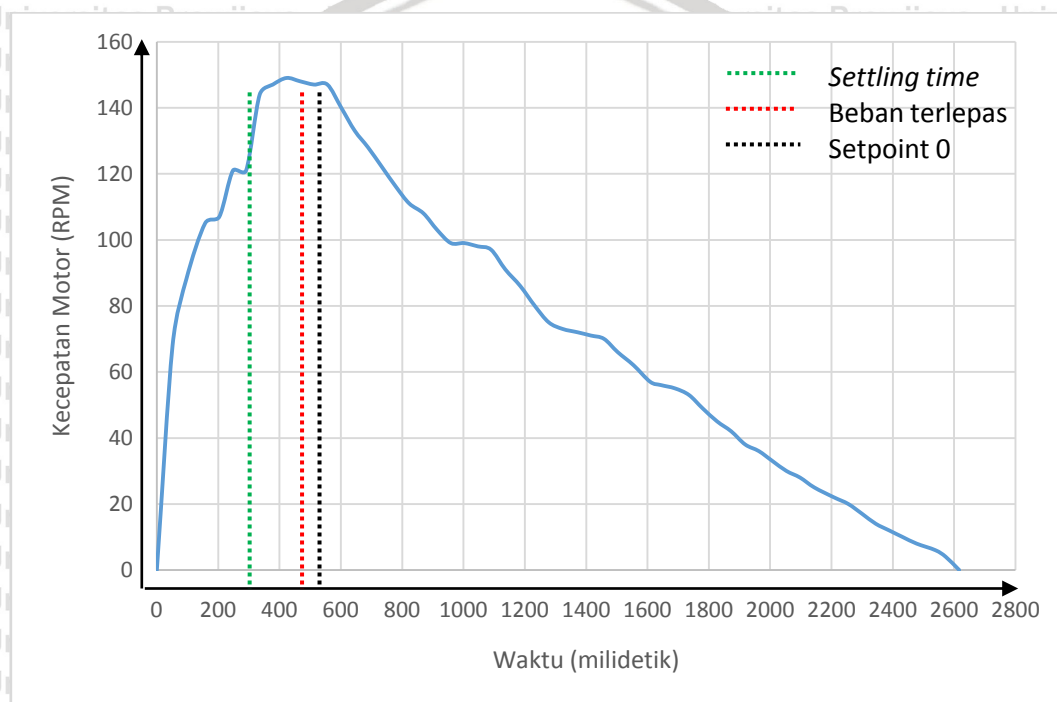
Dari pengujian ini didapatkan nilai *settling time* sebesar 287,16 detik, *error steady state* 1.6%, *overshoot* 1.6% dan ketinggian 2,7m. Sistem diberi *setpoint* 0 saat tuas pelontar melewati 90°. Nilai yang berubah-ubah sesudah *settling time* merupakan error pembacaan sensor *rotary encoder*.

$$\begin{aligned} ess (\%) &= \frac{|127 - 125|}{125} \times 100 \% \\ &= 1,6 \% \end{aligned}$$

$$\begin{aligned} overshoot (\%) &= \frac{127 - 125}{125} \times 100 \% \\ &= 1,6\% \end{aligned}$$

4.2.3. Pengujian dengan Setpoint150rpm

Keluaran sistem pada pengujian sistem berbeban dengan *setpoint*150RPM ditunjukkan dalam Gambar 4.6.



Gambar 4.6Keluaran sistem berbeban dengan *setpoint*150RPM

Gambar 4.6 Merupakan keluaran sistem yang di gambarkan dalam bentuk kurva kecepatan motor terhadap waktu. Garis putus-putus berwarna hijau merupakan titik dimana keluaran sistem mencapai *setling time*, garis putus-putus berwarna merah merupakan titik dimana beban terlepas dari tuas, dan garis putus-putus berwarna hitam merupakan titik dimana sistem diberi *setpoint* sebesar 0.

Dari pengujian ini didapatkan nilai *settling time* sebesar 378,22milidetik, *error steady state* 2%,*overshoot* 0% dan ketinggian 3,8m. Sistem diberi *setpoint* 0 saat tuas pelontar melewati 90°. Nilai yang berubah-ubah sesudah *settling time* merupakan eror pembacaan sensor *rotary encoder*.

$$\text{ess} (\%) = \frac{|147 - 150|}{150} \times 100 \%$$

$$= 2\%$$

$$\text{overshoot} (\%) = \frac{150 - 150}{150} \times 100 \%$$

$$= 0 \%$$

Data hasil pengujian sistem tanpa beban dan berbeban ditunjukkan dalam Tabel 4.1.

Tabel 4. 1 Data Hasil Pengujian Sistem Tanpa Beban dan Berbeban

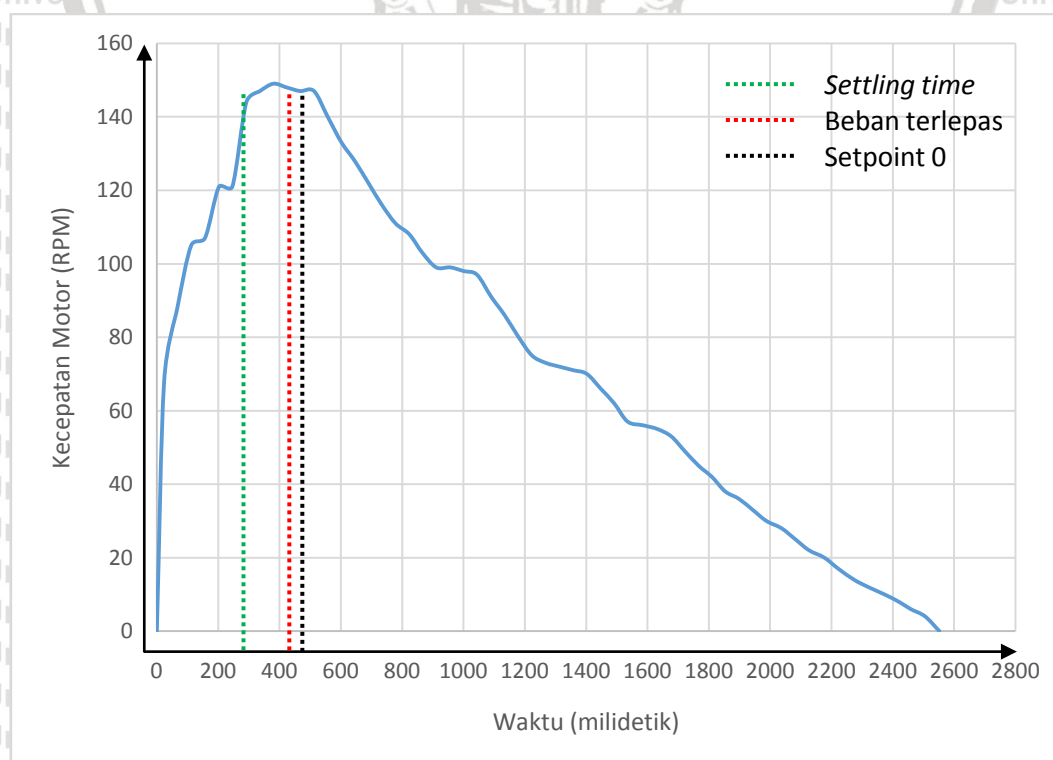
Setpoint(rpm)	Settling Time (ms)		Overshoot (%)		Error Steady State		Ketinggian (m)
	Tanpa Beban	Berbeban	Tanpa Beban	Berbeban	Tanpa Beban	Berbeban	
100	145,29	246,22	4	4	2	1	2,1
125	145,87	242,79	2,4	1,6	1,6	1,6	2,7
150	93,92	378,22	0	0	1,3	2	3,8

4.3. Pengujian Sistem dengan Variasi Beban

Pengujian ini dilakukan untuk mengetahui pengaruh variasi beban terhadap performansi sistem. Prosedur pengujian ini terdapat dalam bab 3.10.

4.3.1. Pengujian dengan Beban 308 gr

Keluaran sistem pada pengujian variasi beban dengan beban 308 gr ditunjukkan dalam Gambar 4.7.



Gambar 4.7 Keluaran sistem dengan beban 308 gr

Gambar 4.7 Merupakan keluaran sistem yang di gambarkan dalam bentuk kurva kecepatan motor terhadap waktu. Garis putus-putus berwarna hijau merupakan titik dimana keluaran sistem mencapai *settling time*, garis putus-putus berwarna merah merupakan titik dimana beban terlepas dari tuas, dan garis putus-putus berwarna hitam merupakan titik dimana sistem diberi *setpoint* sebesar 0.

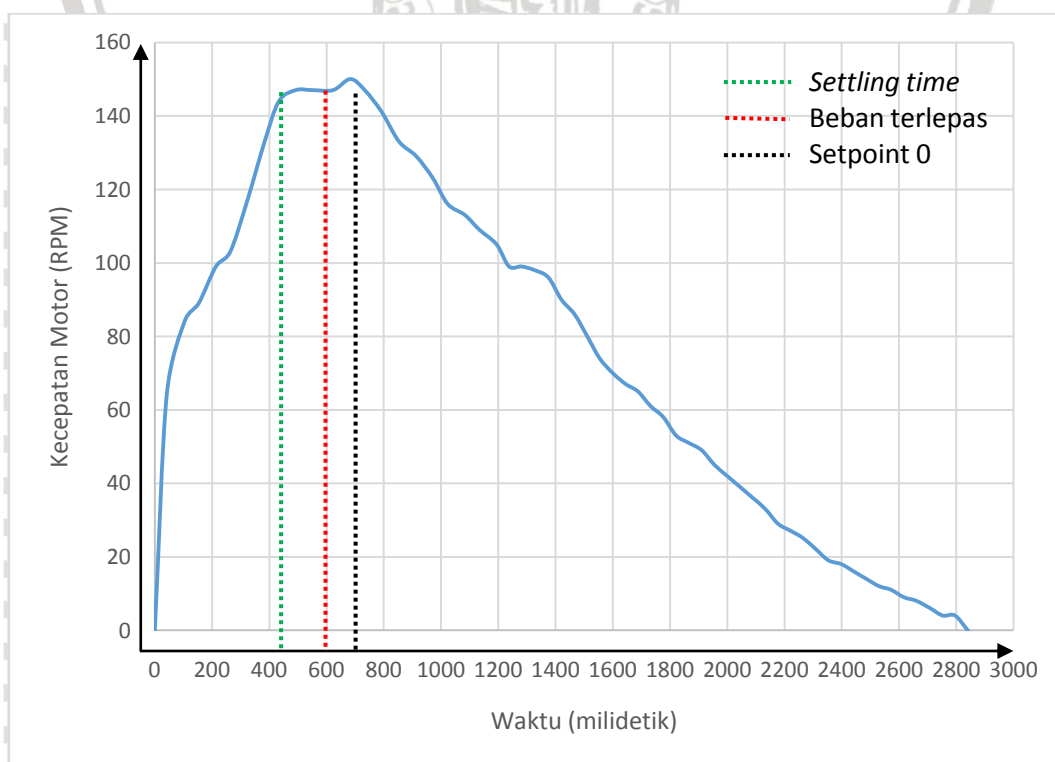
Dari pengujian ini didapatkan nilai *settling time* sebesar 378,22milidetik, *error steady state* 2%,*overshoot* 0% dan ketinggian 3,8m. Sistem diberi *setpoint* 0 saat tuas pelontar melewati 90°.

$$\text{ess} (\%) = \frac{|147 - 150|}{150} \times 100 \% \\ = 2\%$$

$$\text{overshoot} (\%) = \frac{38 - 38}{38} \times 100 \% \\ = 0 \%$$

4.3.2. Pengujian dengan Beban 318 gr

Keluaran sistem pada pengujian variasi beban dengan beban 318 gr ditunjukkan dalam Gambar 4.8.



Gambar 4.8 Keluaran sistem dengan beban 318 gr

Gambar 4.8 Merupakan keluaran sistem yang di gambarkan dalam bentuk kurva kecepatan motor terhadap waktu. Garis putus-putus berwarna hijau merupakan titik dimana keluaran sistem mencapai *settling time*, garis putus-putus berwarna merah merupakan titik dimana beban terlepas dari tuas, dan garis putus-putus berwarna hitam merupakan titik dimana sistem diberi *setpoint* sebesar 0.

Dari pengujian ini didapatkan nilai *settling time* sebesar 492,95milidetik, *error steady state* 2%,*overshoot* 0% dan ketinggian 3,7m. Sistem diberi *setpoint* 0 saat tuas pelontar melewati 90°.

$$\text{ess} (\%) = \frac{|147 - 150|}{150} \times 100 \%$$

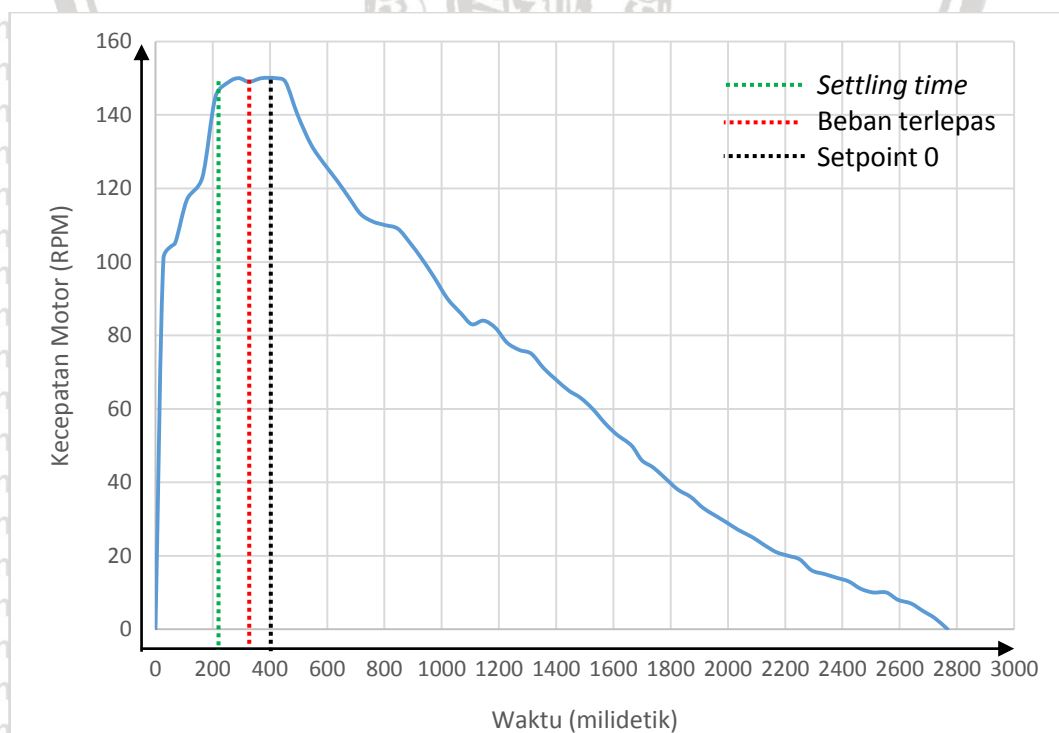
$$= 2\%$$

$$\text{overshoot} (\%) = \frac{150 - 150}{150} \times 100 \%$$

$$= 0 \%$$

4.3.3. Pengujian dengan Beban 325 gr

Keluaran sistem pada pengujian variasi beban dengan beban 325 gr ditunjukkan dalam Gambar 4.9.



Gambar 4.9 Keluaran sistem dengan beban 325 gr

Gambar 4.9 Merupakan keluaran sistem yang di gambarkan dalam bentuk kurva kecepatan motor terhadap waktu. Garis putus-putus berwarna hijau merupakan titik dimana keluaran sistem mencapai *setling time*, garis putus-putus berwarna merah merupakan titik dimana beban terlepas dari tuas, dan garis putus-putus berwarna hitam merupakan titik dimana sistem diberi *setpoint* sebesar 0.

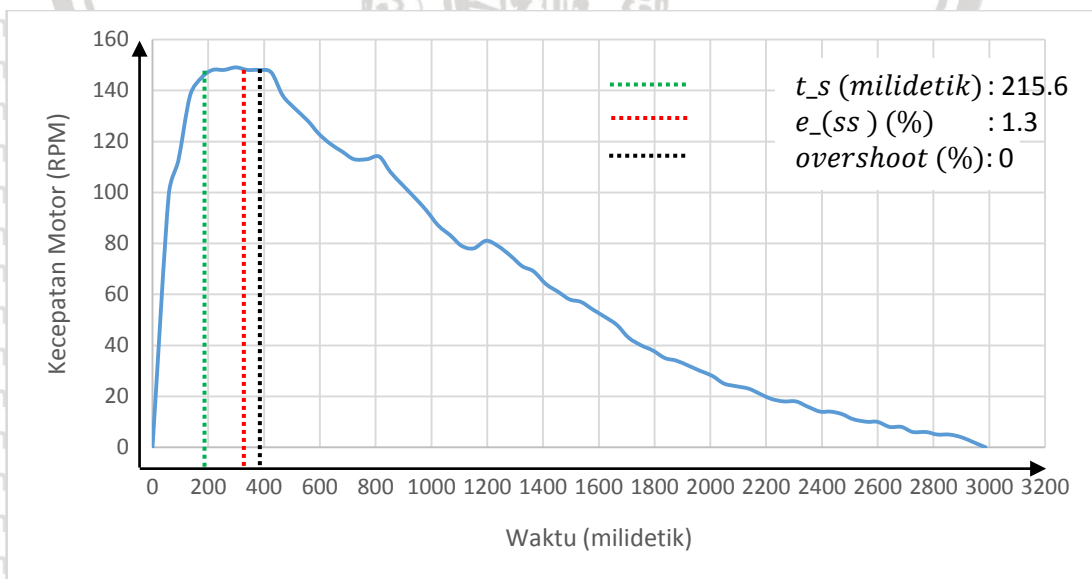
Dari pengujian inididapatkan nilai *settling time* sebesar 256,97 detik *error steady state* 0.6%, *overshoot* 0% dan ketinggian 3,5m. Sistem diberi *setpoint* 0 saat tuas pelontar melewati 90°.

$$ess (\%) = \frac{|149 - 150|}{150} \times 100 \% \\ = 0,6\%$$

$$overshoot (\%) = \frac{150 - 150}{150} \times 100 \% \\ = 0 \%$$

4.3.4. Pengujian dengan Beban 328 gr

Keluaran sistem pada pengujian variasi beban dengan beban 328 gr ditunjukkan dalam Gambar 4.10.



Gambar 4.10 Keluaran sistem dengan beban 328 gr

Gambar 4.10 Merupakan keluaran sistem yang di gambarkan dalam bentuk kurva kecepatan motor terhadap waktu. Garis putus-putus berwarna hijau merupakan

titik dimana keluaran sistem mencapai *setling time*, garis putus-putus berwarna merah merupakan titik dimana beban terlepas dari tuas, dan garis putus-putus berwarna hitam merupakan titik dimana sistem diberi *setpoint* sebesar 0.

Dari pengujian inididapatkan nilai *settling time* sebesar 215,6 milidetik *error steady state* 1,3%, *overshoot* 0% dan ketinggian 3,5m. Sistem diberi *setpoint* 0 saat tuas pelontar melewati 90°.

$$\text{ess} (\%) = \frac{|148 - 150|}{150} \times 100 \%$$

$$= 1,3\%$$

$$\text{overshoot} (\%) = \frac{149 - 149}{149} \times 100 \%$$

$$= 0 \%$$

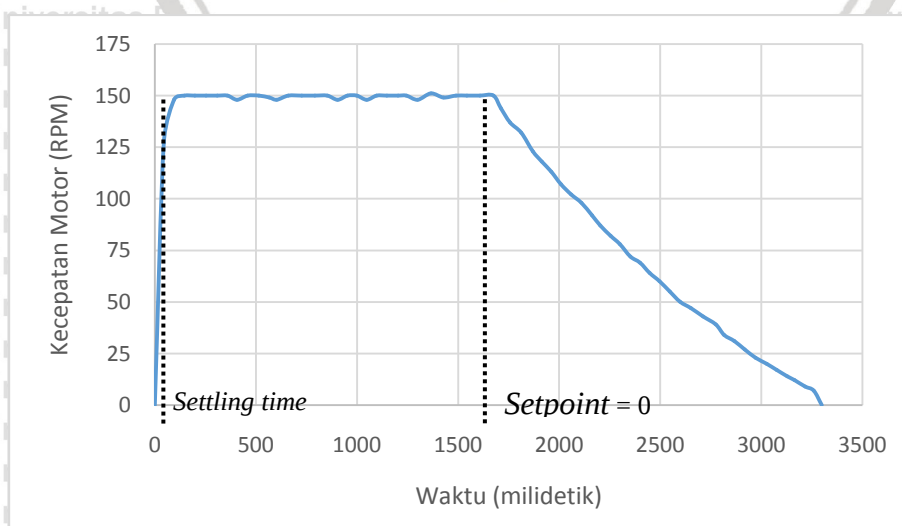
Data hasil pengujian sistem dengan variasi beban ditunjukkan dalam Tabel 4.2

Tabel 4. 2Data Pengujian Sistem dengan Variasi Beban

Setpoint (RPM)	Berat Beban (gr)	Settling Time (ms)	ErrorSteady State (%)	Overshoot (%)	Ketinggian (m)
150	308	378,22	2	0	3,8
	318	492,95	2	0	3,7
	325	256,97	0,6	0	3,5
	328	215,6	1,3	0	3,5

4.4. Pengujian Sifat Reversibel Sistem

Pengujian ini dilakukan untuk mengetahui apakah sistem bersifat reversibel. Prosedur pengujian ini terdapat dalam bab 3.10. Keluaran sistem ditunjukkan dalam Gambar 4.11.



Gambar 4.11 Keluaran pada pengujian sifat reversibel sistem

Gambar 4.11 Merupakan keluaran sistem yang di gambarkan dalam bentuk kurva kecepatan motor terhadap waktu dengan garis putus-putus pertama merupakan titik dimana keluaran sistem mencapai *settling time* dan garis putus-putus kedua merupakan titik dimana sistem di beri *setpoint* sebesar 0.

Dari pengujian ini diketahui bahwa ketika sistem sudah mencapai *setpoint* 150 RPM dan kemudian sistem diberikan *setpoint* 0 RPM, sistem dapat kembali menuju *setpoint* yang telah diberikan. Nilai yang berubah-ubah sesudah *settling time* merupakan eror pembacaan sensor *rotary encode*



BABV

KESIMPULAN

Berdasarkan pengujian dan analisis yang telah dilakukan pada penelitian ini didapatkan beberapa kesimpulan sebagai berikut:

1. Fungsi alih sistem keseluruhan adalah $\frac{\theta(s)}{U(s)} = \frac{23.88}{s^2 + 10.6s + 24.28}$
2. Nilai parameter kontroler PI yang didapatkan menggunakan metode *root locus* dengan nilai $poles_1 = -7,26$ adalah K_p 0,375953 dan K_i 2,5
3. Pada pengujian sistem dengan nilai *setpoint* berbeda-beda didapatkan nilai *settling time* (t_s) rata-rata sebesar 128,36 detik
4. Nilai *setpoint* mempengaruhi performansi sistem
5. Pada pengujian sistem dengan beban berbeda-beda didapatkan nilai *settling time* (t_s) rata-rata sebesar 289,103 detik
6. *settling time* (t_s) akan bertambah saat diberikan berupa beban
7. Sistem bekerja cukup baik sesuai dengan spesifikasi desain yang telah ditentukan



DAFTAR PUSTAKA

- Amalia, Z. (2016). *DESAIN KONTROLER STRUKTUR OUTPUT FEEDBACK CONTROL DENGAN POLE PLACEMENT PADA PENGONTROLAN TEGANGAN OUTPUT GENERATOR DC*. Malang: Universitas Brawijaya, Skripsi tidak dipublikasikan.
- Ardiansyah, R. (2019). *KONTROL POSISI MOTOR DC MENGGUNAKAN KONTROLER PI PADA LINEAR SLIDE AKTUATOR ROBOT DIVISI KRAI*. Malang: Universitas Brawijaya, Skripsi tidak dipublikasikan.
- Asiswantara. (2017). *Penentuan Gain Kontroler PI Digital Sebagai Kontroler Camera Stabilizer dengan Menggunakan Metode Root Locus*. Malang: universitas Brawijaya: Skripsi. Tidak Dipublikasikan.
- Charles L. Philips, R. D., & Widodo, a. b. (1998). *Sistem kontrol : dasar-dasar (feedback control system 3e)*. Kota Bandar Lampung: Prenhallindo.
- Fahmizal. (2011, December 4). *Driver Motor DC pada Robot Beroda dengan Konfigurasi H-BRIDGE MOSFET*. Retrieved from fahmizaleeits.wordpress.com: <https://fahmizaleeits.wordpress.com/2011/12/04/driver-motor-dc-pada-robot-beroda-dengan-konfigurasi-h-bridge-mosfet/>
- Ogata, K. (1995). *Teknik Kontrol Automatik (sistem pengaturan) (jilid 1 ed)*. Jakarta: Erlangga.
- Prinsip Kerja Motor DC*. (2019, February 7). Retrieved from elektronika-dasar.web.id: <http://elektronika-dasar.web.id/prinsip-kerja-motor-dc/>
- Rizqiawan, A. (2009, june 12). *Sekilas Rotary Encoder*. Retrieved from konversi.wordpress.com: <https://konversi.wordpress.com/2009/06/12/sekilas-rotary-encoder/>
- Rotary Encoder*. (2019, April 5). Retrieved from wikipedia.org: https://en.wikipedia.org/wiki/Rotary_encoder
- Rusli, M. (2015). *Dasar Teknik Kontrol*. Malang: UB press.
- Yana, A. (2014). *Metode Root Locus Untuk Mencari Parameter PID pada Pengendalian Posisi Stamping Rod Berbasis Pneumatic Menggunakan Arduino Uno*. Malang : Universitas Brawijaya: Skripsi. Tidak Dipublikasikan.





LAMPIRAN 1 DATA PENGUJIAN





Data Pengujian Sistem Tanpa Beban dengan Setpoint 100 rpm

Waktu (milidetik)	Keluaran (rpm)		
0	0	1543.66	100
44.69	104	1607.92	102
93.92	103	1674.18	103
145.29	101	1710.67	97
196.93	100	1756.29	92
252.67	101	1812.19	87
308.8	101	1868.92	82
359.55	100	1914.04	77
405.93	102	1960.55	71
459.17	100	2008.16	66
514.17	101	2059.91	62
565.92	102	2109.94	58
603.17	100	2162.06	54
660.04	101	2203.43	49
727.16	101	2254.05	45
791.54	102	2300.16	40
852.57	101	2351.92	36
903.92	100	2400.42	32
952.17	101	2446.59	29
998.3	100	2496.19	25
1047.92	102	2546.32	22
1096.56	100	2598.29	18
1147.92	100	2652.17	16
1203.17	101	2711.41	12
1243.55	101	2775.17	10
1300.18	101	2816.44	6
1365.19	101	2867.42	4
1427.54	102	2917.91	0
1488.67	100	2971.16	0
		3026.17	0

Data Pengujian Sistem Tanpa Beban dengan Setpoint 125 rpm

Waktu (milidetik)	Keluaran (rpm)		
0	0	1674.18	127
44.69	56	1710.67	127
93.92	128	1756.29	123
145.29	127	1812.19	115
196.93	126	1868.92	111
252.67	124	1914.04	105
308.8	126	1960.55	100
359.55	127	2008.16	95
405.93	127	2059.91	88
459.17	127	2109.94	84
514.17	125	2162.06	80
565.92	126	2203.43	75
603.17	127	2254.05	70
660.04	126	2300.16	67
727.16	126	2351.92	63
791.54	126	2400.42	57
852.57	126	2446.59	53
903.92	126	2496.19	50
952.17	126	2546.32	45
998.3	126	2598.29	42
1047.92	127	2652.17	38
1096.56	127	2711.41	33
1147.92	127	2775.17	29
1203.17	126	2816.44	26
1243.55	126	2867.42	24
1300.18	126	2917.91	20
1365.19	124	2971.16	17
1427.54	127	3026.17	13
1488.67	126	3075.93	11
1543.66	127	3127.67	7
1607.92	127	3164.92	5
		3216.54	0

Data Pengujian Sistem Tanpa Beban dengan Setpoint 150 rpm

Waktu (milidetik)	Keluaran (rpm)		
0	0	1674.18	150
44.69	124	1710.67	144
93.92	148	1756.29	132
145.29	150	1812.19	123
196.93	150	1868.92	118
252.67	150	1914.04	113
308.8	150	1960.55	107
359.55	150	2008.16	102
405.93	148	2059.91	98
459.17	150	2109.94	92
514.17	150	2162.06	87
565.92	149	2203.43	82
603.17	148	2254.05	78
660.04	150	2300.16	72
727.16	150	2351.92	69
791.54	150	2400.42	64
852.57	150	2446.59	60
903.92	148	2496.19	55
952.17	150	2546.32	50
998.3	150	2598.29	47
1047.92	148	2652.17	43
1096.56	150	2711.41	39
1147.92	150	2775.17	34
1203.17	150	2816.44	31
1243.55	150	2867.42	27
1300.18	148	2917.91	23
1365.19	151	2971.16	20
1427.54	149	3026.17	17
1488.67	150	3075.93	14
1543.66	150	3127.67	12
1607.92	150	3164.92	9
		3216.54	7

Data Pengujian Sistem Berbeban dengan Setpoint 100 rpm

Waktu (milidetik)	Keluaran (rpm)		
		1358.18	41
0	0	1401.78	37
23.79	70	1446.87	33
66.92	81	1490.79	29
112.11	104	1534.91	26
157.03	101	1586.02	24
202.92	98	1631.54	22
246.22	101	1676.86	20
291.98	103	1720.73	17
336.11	102	1766.23	14
380.34	101	1809.1	12
422.73	105	1853.04	10
466.98	103	1897.47	9
512.23	101	1942.36	6
556.98	100	1986.61	2
601.98	101	2036.16	0
644.98	96	2081.17	0
689.48	89	2125.92	0
733.73	79	2175.66	0
777.82	74	2221.16	0
823.17	70	2272.42	0
865.36	66	2318.67	0
910.73	67	2371.55	0
956.48	65	2418.42	0
1000.36	62	2457.16	0
1043.73	59	2504.29	0
1089.92	57	2550.79	0
1133.73	54	2598.66	0
1179.34	49	2651.92	0
1223.11	46	2697.42	0
1268.03	45	2743.16	0
1311.34	44	2787.16	0



Data Pengujian Sistem Berbeban dengan Setpoint 125 rpm

Waktu (milidetik)	Keluaran (rpm)		
0	0	1352.06	54
23.48	68	1395.53	52
66.01	87	1439.16	52
110.78	93	1482.17	49
155.1	113	1525.84	47
197.78	111	1571.47	46
242.79	123	1617.73	44
287.16	126	1661.97	40
332.04	127	1707.22	38
375.16	126	1751.28	38
420.19	125	1795.78	36
463.48	127	1839.91	35
506.78	124	1885.03	33
552.36	126	1928.16	30
597.16	124	1973.47	29
640.91	119	2017.97	26
685.53	113	2062.99	24
729.16	105	2130.12	23
772.78	101	2180.46	22
816.86	96	2223.29	19
860.8	91	2268.16	17
904.16	89	2313.28	15
951.11	85	2357.91	13
995.16	82	2401.83	11
1037.91	80	2446.16	9
1083.73	79	2490.23	8
1129.73	76	2533.78	7
1173.73	72	2578.03	2
1219.41	67	2637.09	0
1262.78	63	2673.63	0
1308.16	59	2715.97	0
		2762.79	0

Data Pengujian Sistem Berbeban dengan Setpoint 150 rpm

Waktu (milidetik)	Keluaran (rpm)		
0	0	1415.16	71
51.34	68	1457.71	70
95.96	88	1502.28	66
157.5	105	1554.35	62
203.84	107	1608.37	57
248.28	121	1643.46	56
291.15	121	1689.32	55
335.48	144	1734.84	53
378.22	147	1778.85	49
422.85	149	1826.58	45
466.48	148	1872.47	42
512.03	147	1918.21	38
555.91	147	1962.71	36
601.2	140	2006.97	33
645.28	133	2052.46	30
688.34	128	2096.71	28
734.18	122	2143.46	25
779.88	116	2207.47	22
822.71	111	2254.2	20
869.22	108	2299.45	17
913.72	103	2343.92	14
958.49	99	2387.28	12
1002.21	99	2431.23	10
1046.97	98	2476.9	8
1089.59	97	2538.21	6
1136.22	91	2571.9	4
1184.96	86	2615.72	0
1232.1	80	2660.65	0
1277.59	75	2704.9	0
1322.85	73	2749.4	0
1371.22	72	2793.64	0
		2836.9	0



Data Pengujian Variasi Beban dengan Beban 308 gr

Waktu (milidetik)	Keluaran (rpm)		
0	0	1415.16	71
51.34	68	1457.71	70
95.96	88	1502.28	66
157.5	105	1554.35	62
203.84	107	1608.37	57
248.28	121	1643.46	56
291.15	121	1689.32	55
335.48	144	1734.84	53
378.22	147	1778.85	49
422.85	149	1826.58	45
466.48	148	1872.47	42
512.03	147	1918.21	38
555.91	147	1962.71	36
601.2	140	2006.97	33
645.28	133	2052.46	30
688.34	128	2096.71	28
734.18	122	2143.46	25
779.88	116	2207.47	22
822.71	111	2254.2	20
869.22	108	2299.45	17
913.72	103	2343.92	14
958.49	99	2387.28	12
1002.21	99	2431.23	10
1046.97	98	2476.9	8
1089.59	97	2538.21	6
1136.22	91	2571.9	4
1184.96	86	2615.72	0
1232.1	80	2660.65	0
1277.59	75	2704.9	0
1322.85	73	2749.4	0
1371.22	72	2793.64	0
		2836.9	0

Data Pengujian Variasi Beban dengan Beban 318 gr

Waktu (milidetik)	Keluaran (rpm)		
0	0	1644.35	67
41.53	64	1687.3	65
103.71	84	1731.6	61
153.54	89	1776.09	58
213.29	99	1820.74	53
263.47	103	1864.6	51
323.22	117	1909.97	49
383.61	133	1954.7	45
433.6	144	1998.72	42
492.95	147	2044.05	39
548.31	147	2088.85	36
623.1	147	2132.28	33
682.97	150	2176.47	29
733.05	147	2222.22	27
792.85	141	2265.66	25
852.8	133	2309.65	22
913.05	129	2353.29	19
971.44	123	2397.46	18
1022.97	116	2441.4	16
1083.1	113	2484.54	14
1133.49	109	2529.18	12
1193.66	105	2571.78	11
1237.3	99	2615.79	9
1281.22	99	2659.91	8
1326.29	98	2708.03	6
1374.67	96	2751.73	4
1419.92	90	2794.78	4
1465.93	86	2839.31	0
1510.28	80	2884.03	0
1553.72	74	2926.78	0
1599.29	70	2972.41	0
		3015.91	0

Data Pengujian Variasi Beban dengan Beban 325 gr

Waktu (milidetik)	Keluaran (rpm)		
0	0	1312.84	75
27.97	101	1356.72	71
67.98	105	1398.33	68
110.23	117	1442.98	65
163.84	123	1484.61	63
209.73	145	1526.85	60
256.97	149	1571.24	56
291.59	150	1611.33	53
326.96	149	1662.99	50
370.21	150	1698.84	46
419.84	150	1739.21	44
453.97	149	1782.22	41
496.58	140	1825.1	38
542.97	132	1869.85	36
586.21	127	1913.12	33
635.35	122	1954.22	31
680.09	117	1996.35	29
716.98	113	2036.84	27
758.1	111	2086.22	25
801.96	110	2124.86	23
847.09	109	2168.35	21
890.84	105	2209.22	20
928.97	101	2250.71	19
971.96	96	2292.58	16
1019.84	90	2337.85	15
1066.99	86	2379.6	14
1105.24	83	2421.85	13
1146.6	84	2463.6	11
1187.86	82	2506.87	10
1228.23	78	2552.99	10
1271.6	76	2593.85	8
		2639.63	7



Data Pengujian Variasi Beban dengan Beban 928 gr

Waktu (milidetik)	Keluaran (rpm)		
0	0	1325.97	71
58.22	99	1366.61	69
91.99	112	1410.86	64
134.9	138	1454.1	61
176.23	145	1495.6	58
215.6	148	1536.71	57
256.22	148	1579.33	54
298.84	149	1624.24	51
340.59	148	1665.1	48
381.85	148	1706.97	43
425.99	147	1751.85	40
466.84	138	1793.58	38
509.35	133	1837.1	35
557.34	128	1877.99	34
595.97	123	1922.1	32
637.85	119	1963.22	30
679.97	116	2007.38	28
722.11	113	2048.98	25
769.23	113	2090.85	24
813.23	114	2137.61	23
852.86	108	2177.48	21
895.6	103	2218.22	19
939.61	98	2262.6	18
981.22	93	2307.11	18
1024.6	87	2348	16
1069.25	83	2393.63	14
1108.58	79	2432.11	14
1151.6	78	2472.6	13
1195.83	81	2512.72	11
1238.11	79	2562.86	10
1285.14	75	2599.1	10
		2643.1	8

LAMPIRAN 2 KONFIGURASI PIN DAN SKEMATIK



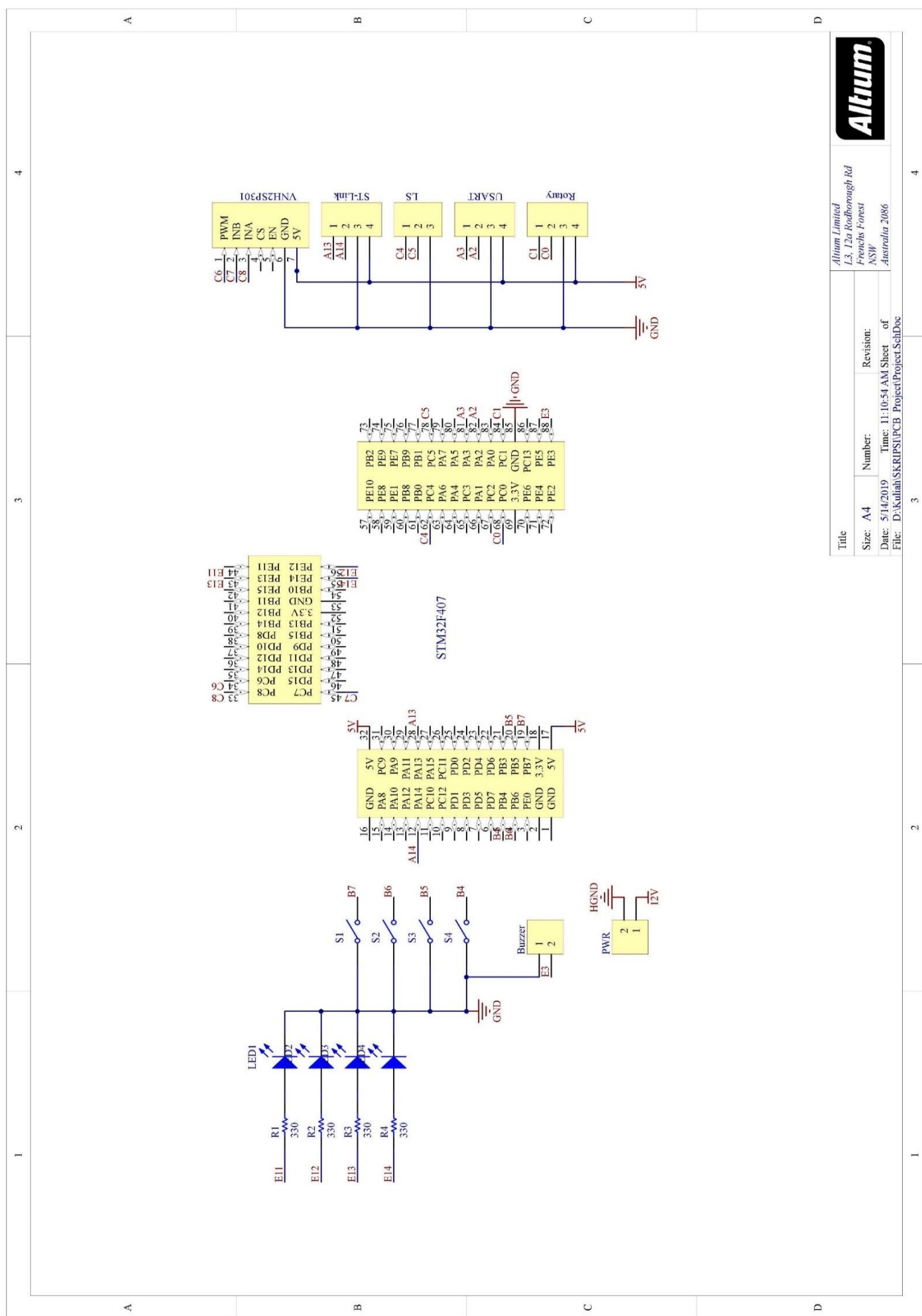


DIR2	PWM									LED3	LED1
C8	C6	D14	D12	D10	D8	B12	B11	E15	E13	E11	
	D15	D13	D11	D9	B15	B13	GND	B10	E14	E12	
DIR1									LED4	LED2	

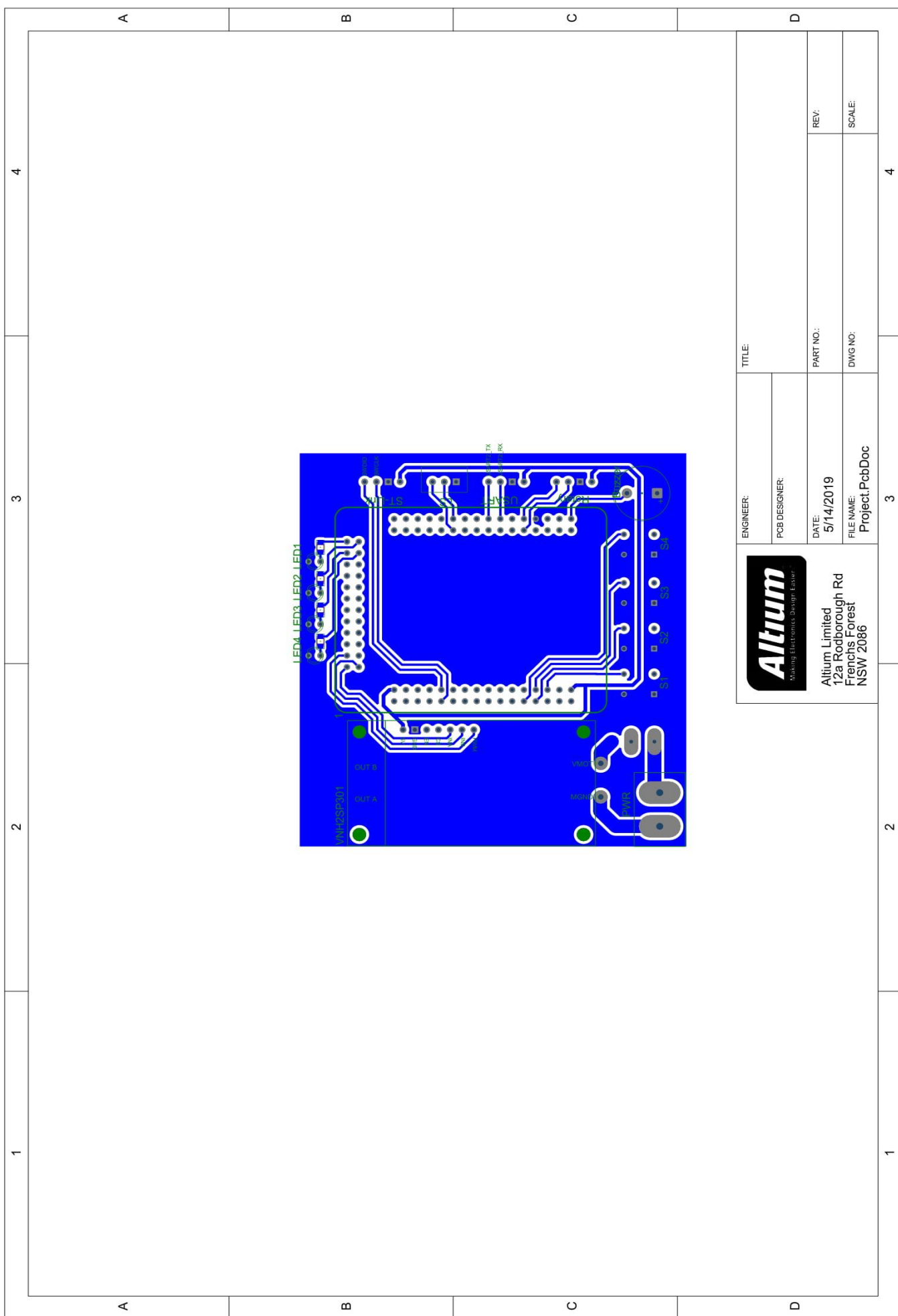
		GND	VCC		
		A8	C9		
		A10	A9		
		A12	A11		
	SWCLK	A14	A13	SWDIO	
		C10	A15		
		C12	C11		
		D1	D0		
		D3	D2		
		D5	D4		
		D7	D6		
PB4		B4	B3		
PB2		B6	B5	PB3	
		E0	B7	PB1	
		GND	3.3V		
		GND	5V		

		E10	B2		
		E8	E9		
		E1	E7		
		B8	B9		
		B0	B1		
	LS1	C4	C5		LS2
		A6	A7		
		A4	A5		
		C3	A3		
		A1	A2	USART2_RX	
	USART2_TX	C2	A0		
		C0	C1		
Rotary_CH1	EXTI1	3.3V	GND	Rotary_CH2	
		E6	C13		
		E4	E5		
		E2	E3		













LAMPIRAN 3 LISTING PROGRAM UTAMA





```

#include "bismillah.h"
int nilaiDC;
void vTaskRUN1 (void *pvParameters) {
    for (;;) {
    }
}

void vTaskRUN2 (void *pvParameters) {
    for (;;) {
        /* Motor Set & Control */
        /* SkripsiPei */
        // V1.Kp = 0.3; V1.Ki = 0.1;
        V1.set = nilaiDC;
        // PID_Kecepatan1();
        eksekusi_motor(mtrl, V1.arah, V1.set);
        /* Motor Set & Control */
        TIM3->CCR1 = (( V1.set * (TIM_3.period - 1)) / 100); //motor5
    }
}

void vTaskRUN3 ( void * pvParameters ){
    TickType_t xLastWakeTime;
    const TickType_t xPeriod = pdMS_TO_TICKS( 20 );
    xLastWakeTime = xTaskGetTickCount();

    for (;;) {
        vTaskDelayUntil( &xLastWakeTime, xPeriod );

        prt[0] = speed[0]*(6000/rt_1.ppr);
        speed[0]=0;
        if (!prt[0]) rt_1.Rpm=0;
    }
}

int waktuu, ttd, cobaa, tmpp, tandaa=0;
void vTaskRUN4 ( void * pvParameters ){
    TickType_t xLastWakeTime;
    const TickType_t xPeriod = pdMS_TO_TICKS( 100 );
    xLastWakeTime = xTaskGetTickCount();

    for (;;) {
        vTaskDelayUntil( &xLastWakeTime, xPeriod );

        if (ttd)
            waktuu++;
        V1.arah = maju;

        if (waktuu==40)
        {
            waktuu = 0;
        }
        elseif (waktuu==20)
        {
        }
    }
}

```

```

if (V1.SP==150) {
    cobaa++;
    tandaa = 1;
}
V1.SP = 0;
elseif (waktuu==5 && !tandaa)
{
    tmpp += 25;
    V1.SP = tmpp;
}
if (tandaa) {
    switch (cobaa) {
        case 1 : V1.Kp = 0.1400;
                 V1.Ki = 1; tandaa = 0; tmpp = 0;
        break;
        case 2 : V1.Kp = 0.2089;
                 V1.Ki = 1.5; tandaa = 0; tmpp = 0;
        break;
        case 3 : V1.Kp = 0.2778;
                 V1.Ki = 2; tandaa = 0; tmpp = 0;
        break;
        case 4 : V1.Kp = 0.3466;
                 V1.Ki = 2.5; tandaa = 0; tmpp = 0;
        break;
        case 5 : V1.Kp = 0.4155;
                 V1.Ki = 3; tandaa = 0; tmpp = 0;
        break;
        case 6 : V1.Kp = 0.4844;
                 V1.Ki = 3.5; tandaa = 0; tmpp = 0;
        break;
        case 7 : V1.Kp = 0.5533;
                 V1.Ki = 4; tandaa = 0; tmpp = 0;
        break;
        case 8 : V1.Kp = 0.6221;
                 V1.Ki = 4.5; tandaa = 0; tmpp = 0;
        break;
        case 9 : V1.Kp = 0.6910;
                 V1.Ki = 5.; tandaa = 0; tmpp = 0;
        break;
        case 10: V1.Kp = 0.7599;
                 V1.Ki = 5.5; tandaa = 0; tmpp = 0;
        break;
    }
}
// cobaa = 4;
// if (rt.1.encoder2>=rt.1.ppr) {
//     V1.SP = 0;
//     V1.set = 0;
// }
}
void TaskRUN5 ( void * pvParameters ) {
    TickType_t xLastWakeTime;

```

```

const TickType_t xPeriod = pdMS_TO_TICKS( 75 );
xLastWakeTime = xTaskGetTickCount();

for(;;){
    vTaskDelayUntil( &xLastWakeTime, xPeriod );
}

int main(void){
    SystemInit();
    inisTIM2();
    inisTIM3();
    External_Interrupt();
    init_IO(RCC_AHB1Periph_GPIOC, P_7|P_8, GPIO_Mode_OUT,
    GPIO_OType_PP, GPIO_PuPd_NOPULL ); //DIR Motor

    xTaskCreate( vTaskRUN1, ( signedchar * ) "RUN1",
    configMINIMAL_STACK_SIZE, NULL, 1, NULL);
    xTaskCreate( vTaskRUN2, ( signedchar * ) "RUN2",
    configMINIMAL_STACK_SIZE, NULL, 1, NULL);
    xTaskCreate( vTaskRUN3, ( signedchar * ) "RUN3",
    configMINIMAL_STACK_SIZE, NULL, 1, NULL);
    xTaskCreate( vTaskRUN4, ( signedchar * ) "RUN4",
    configMINIMAL_STACK_SIZE, NULL, 1, NULL);
    xTaskCreate( vTaskRUN5, ( signedchar * ) "RUN5",
    configMINIMAL_STACK_SIZE, NULL, 1, NULL);

    vTaskStartScheduler();
    while (1)
    {
    }
}

//External Interrupt section
void EXTI0_IRQHandler() {
    if (EXTI_GetITStatus(EXTI_Line0)) {
        speed[0]++;
        // if(GPIO_ReadInputDataBit(GPIOC, GPIO_Pin_1)){
        //     rt_1.encoder--;
        //     rt_1.encoder2--;
        // }
        // else{
        //     rt_1.encoder++;
        //     rt_1.encoder2++;
        // }

        if(GPIO_ReadInputDataBit(GPIOC, GPIO_Pin_1)){
            rt_1.encoder--;
            rt_1.encoder2--;
        }
        else{
            rt_1.encoder++;
            rt_1.encoder2++;
        }

        if(rt_1.encoder>rt_1.ppr) rt_1.encoder=1;
    }
}

```

```
    else if (rt_1.encoder < 0) rt_1.encoder = rt_1.ppr - 1;
    rt_1.sudut = (rt_1.encoder * 360) / rt_1.ppr;
    if (rt_1.CaptureNumber == 0)
    {
        rt_1.EXTI_ReadValue1 = TIM2->CNT;
        rt_1.CaptureNumber = 1;
    }
    else if (rt_1.CaptureNumber == 1)
    {
        rt_1.EXTI_ReadValue2 = TIM2->CNT;
        // Hitung Data Periode
        if (rt_1.EXTI_ReadValue2 > rt_1.EXTI_ReadValue1) rt_1.Capture =
        (rt_1.EXTI_ReadValue2 - rt_1.EXTI_ReadValue1);
        else if (rt_1.EXTI_ReadValue2 <
        rt_1.EXTI_ReadValue1) rt_1.Capture = ((TIM2 ->ARR - rt_1.EXTI_ReadValue1)
        + rt_1.EXTI_ReadValue2);
        else rt_1.Capture = 0;
        // Hitung Frekuensi
        rt_1.Freq = TIM_2.period / rt_1.Capture;
        rt_1.Rpm = (rt_1.Freq * 60) / rt_1.ppr;
        rt_1.CaptureNumber = 0;
    }
    EXTI_ClearITPendingBit(EXTI_Line0);
}
```



LAMPIRAN 4

DATASHEET







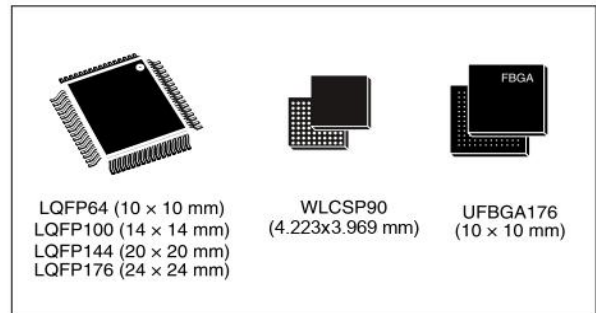
STM32F405xx STM32F407xx

ARM Cortex-M4 32b MCU+FPU, 210DMIPS, up to 1MB Flash/192+4KB RAM, USB OTG HS/FS, Ethernet, 17 TIMs, 3 ADCs, 15 comm. interfaces & camera

Datasheet - production data

Features

- Core: ARM® 32-bit Cortex®-M4 CPU with FPU, Adaptive real-time accelerator (ART Accelerator™) allowing 0-wait state execution from Flash memory, frequency up to 168 MHz, memory protection unit, 210 DMIPS/1.25 DMIPS/MHz (Dhystone 2.1), and DSP instructions
- Memories
 - Up to 1 Mbyte of Flash memory
 - Up to 192+4 Kbytes of SRAM including 64-Kbyte of CCM (core coupled memory) data RAM
 - Flexible static memory controller supporting Compact Flash, SRAM, PSRAM, NOR and NAND memories
- LCD parallel interface, 8080/6800 modes
- Clock, reset and supply management
 - 1.8 V to 3.6 V application supply and I/Os
 - POR, PDR, PVD and BOR
 - 4-to-26 MHz crystal oscillator
 - Internal 16 MHz factory-trimmed RC (1% accuracy)
 - 32 kHz oscillator for RTC with calibration
 - Internal 32 kHz RC with calibration
- Low-power operation
 - Sleep, Stop and Standby modes
 - V_{BAT} supply for RTC, 20×32 bit backup registers + optional 4 KB backup SRAM
- 3×12-bit, 2.4 MSPS A/D converters: up to 24 channels and 7.2 MSPS in triple interleaved mode
- 2×12-bit D/A converters
- General-purpose DMA: 16-stream DMA controller with FIFOs and burst support
- Up to 17 timers: up to twelve 16-bit and two 32-bit timers up to 168 MHz, each with up to 4 IC/OC/PWM or pulse counter and quadrature (incremental) encoder input
- Debug mode
 - Serial wire debug (SWD) & JTAG interfaces
 - Cortex-M4 Embedded Trace Macrocell™
- Up to 140 I/O ports with interrupt capability
 - Up to 136 fast I/Os up to 84 MHz
 - Up to 138 5 V-tolerant I/Os
- Up to 15 communication interfaces
 - Up to 3 × I²C interfaces (SMBus/PMBus)
 - Up to 4 USARTs/2 UARTs (10.5 Mbit/s, ISO 7816 interface, LIN, IrDA, modem control)
 - Up to 3 SPIs (42 Mbits/s), 2 with muxed full-duplex I²S to achieve audio class accuracy via internal audio PLL or external clock
 - 2 × CAN interfaces (2.0B Active)
 - SDIO interface
- Advanced connectivity
 - USB 2.0 full-speed device/host/OTG controller with on-chip PHY
 - USB 2.0 high-speed/full-speed device/host/OTG controller with dedicated DMA, on-chip full-speed PHY and ULPI
 - 10/100 Ethernet MAC with dedicated DMA: supports IEEE 1588v2 hardware, MII/RMII



STM32F405xx, STM32F407xx

- 8- to 14-bit parallel camera interface up to 54 Mbytes/s
- True random number generator
- CRC calculation unit
- 96-bit unique ID
- RTC: subsecond accuracy, hardware calendar

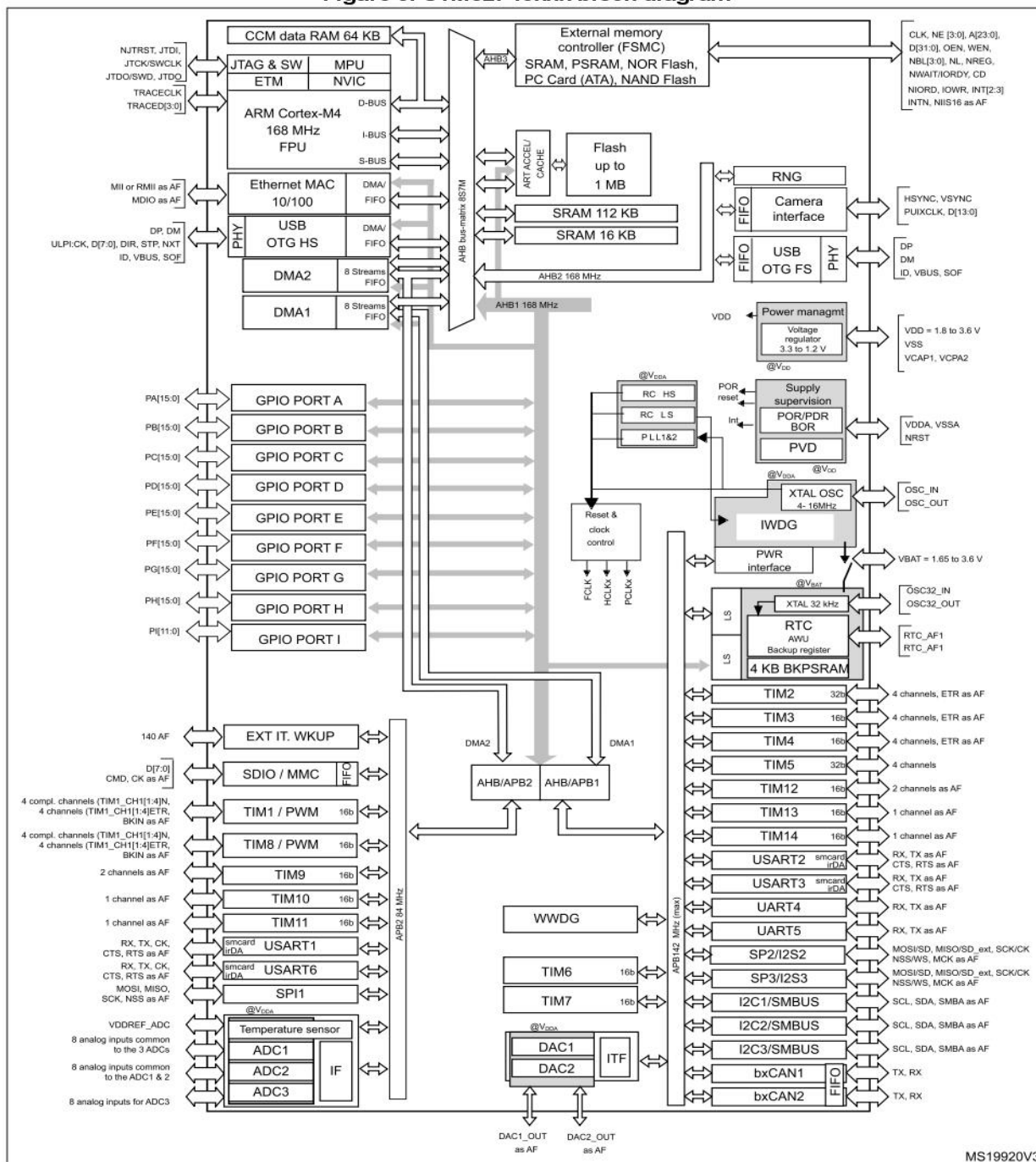
Table 1. Device summary

Reference	Part number
STM32F405xx	STM32F405RG, STM32F405VG, STM32F405ZG, STM32F405OG, STM32F405OE
STM32F407xx	STM32F407VG, STM32F407IG, STM32F407ZG, STM32F407VE, STM32F407ZE, STM32F407IE



2.2 Device overview

Figure 5. STM32F40xxx block diagram



1. The camera interface and ethernet are available only on STM32F407xx devices.

E40 Series

Diameter ø40mm Shaft type/Hollow type/Built-in type Incremental Rotary Encoder

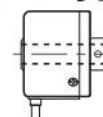
■ Features

- 12-24VDC power supply of line driver output(Line-up)
- Easy installation at narrow space
- Small moment of inertia
- Power supply : 5VDC, 12-24VDC ±5%
- Various output types

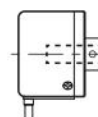
⚠ Please read "Caution for your safety" in operation manual before using.



E40S Series



E40H Series



E40HB Series

■ Ordering information

E40	H	8	5000	3	N	24	
Series	Shaft type	Hollow type	Pulse/1 Revolution	Output phase	Output	Power supply	Cable
S: Shaft type	External	Inner	Series	2: A, B	T: Totem pole output	5 : 5VDC ±5% 24: 12-24VDC ±5%	No mark: Cable type C: Connector cable type(※)
H: Hollow type	6: ø6mm	8: ø8mm		3: A, B, Z	N: NPN open collector output		
HB: Hollow built-in type	8: ø8mm	10: ø10mm		4: A, $\bar{A}$, B, $\bar{B}$	V: Voltage output		
		12: ø12mm		6: A, $\bar{A}$, B, $\bar{B}$, Z, $\bar{Z}$	L: Line driver output		

※Standard: E40S6-[PULSE]-3-N-24, E40H8-[PULSE]-3-N-24 ※Standard: A, B, Z
E40HB8-[PULSE]-3-N-24 ※Cable length : 250mm

■ Specifications

Item		Diameter ø40mm shaft/hollow shaft/hollow built-in type of incremental rotary encoder		
Resolution(P/R)* ¹		*1, *2, *5, 10, *12, 15, 20, 23, 25, 30, 35, 40, 45, 50, 60, 75, 100, 120, 150, 192, 200, 240, 250, 256, 300, 360, 400, 500, 512, 600, 800, 1000, 1024, 1200, 1500, 1800, 2000, 2048, 2500, 3000, 3600, 5000 (Not indicated resolution is customizable.)		
Electrical specification	Output phase		A, B, Z phase(Line driver A, $\bar{A}$, B, $\bar{B}$, Z, $\bar{Z}$ phase)	
	Phase difference of output		Phase difference between A and B : $\frac{T}{4} \pm \frac{T}{8}$ (T=1cycle of A phase)	
	Control output	Totem pole output	• Low - Load current:Max. 30mA, Residual voltage : Max. 0.4VDC • High - Load current:Max. 10mA, Output voltage(Power voltage 5VDC): Min. (Power voltage-2.0)VDC Output voltage(Power voltage 12-24VDC):Min. (Power voltage-3.0)VDC	
		NPN open collector output	Load current : Max. 30mA, Residual voltage : Max. 0.4VDC	
		Voltage output	Load current : Max. 10mA, Residual voltage : Max. 0.4VDC	
		Line driver output	• Low - Load current : Max. 20mA, Residual voltage : Max. 0.5VDC • High - Load current : -20mA, Output voltage(Power voltage 5VDC): Min. 2.5VDC, Output voltage(Power voltage 12-24VDC): Min. (Power voltage-3.0)VDC	
	Response time (Rise/Fall)	Totem pole output	Max. 1μs	• Measuring condition - Cable length : 2m, I sink = 20mA
		NPN open collector output		
		Voltage output		
		Line driver output		
	Max. Response frequency		300kHz	
	Power supply		• 5VDC ±5%(Ripple P-P : Max. 5%) • 12-24VDC ±5%(Ripple P-P : Max. 5%)	
	Current consumption		Max. 80mA(disconnection of the load), Line driver output : Max. 50mA(disconnection of the load)	
	Insulation resistance		Min. 100MΩ(at 500VDC megger between all terminals and case)	
	Dielectric strength		750VAC 50/60Hz for 1 minute(Between all terminals and case)	
Connection		Cable type, 250mm connector cable type		
Mechanical specification	Starting torque		Shaft type : Max. 40gf·cm(0.004N·m), Hole type : Max. 50gf·cm(0.005N·m)	
	Moment of inertia		Max. 40g·cm ² (4×10 ⁻⁶ kg·m ²)	
	Shaft loading		Radial : Max. 2kgf, Thrust : Max. 1kgf	
	Max. allowable revolution ^{※2}		5000rpm	
Vibration		1.5mm amplitude or 300m/s ² at frequency of 10 to 55Hz(for 1 min.) in each of X, Y, Z directions for 2 hours		
Shock		Approx. Max. 50G		
Environment	Ambient temperature	-10 to 70°C, storage : -25 to 85°C		
	Ambient humidity	35 to 85%RH, storage : 35 to 90%RH		
Protection		IP50(IEC standard) ※Option type is available for IP64 (IEC standard).		
Cable		ø5, 5-wire, Length : 2m, Shield cable(Line driver output : ø5, 8-wire) (AWG24, Core diameter: 0.08, Number of cores: 40, Insulator out diameter: ø1)		
Accessory		• Shaft type : ø6mm coupling standard, ø8mm coupling(Sold separately) • Hole type : Bracket		
Approval		CE (Except line driver output)		
Unit weight		Approx. 160g		

※1: * pulse is only for A, B phase(Line Driver output is for A, $\bar{A}$, B, $\bar{B}$ phase)

※2: Make sure that. Max response revolution should be lower than or equal to max. allowable revolution when selecting the resolution.

$$[\text{Max. response revolution(rpm)}] = \frac{\text{Max. response frequency}}{\text{Resolution}} \times 60 \text{ sec}$$

※Environment resistance is rated at no freezing or condensation.

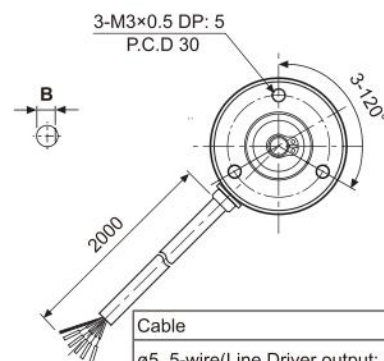
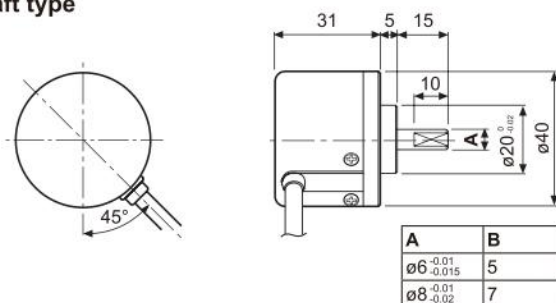
E40 Series

■ Dimensions

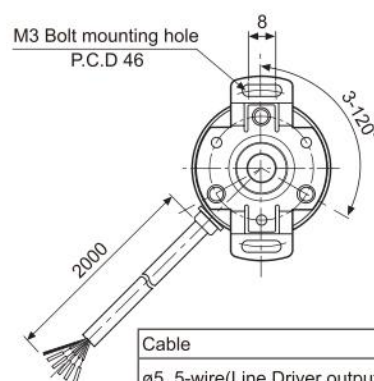
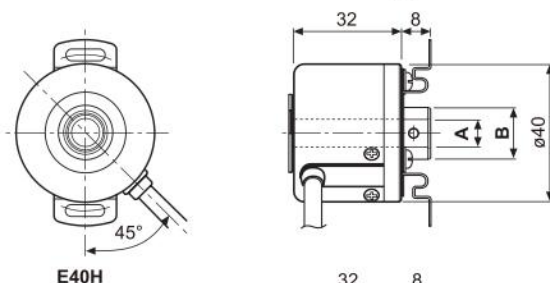
◎ Cable type

● Shaft type

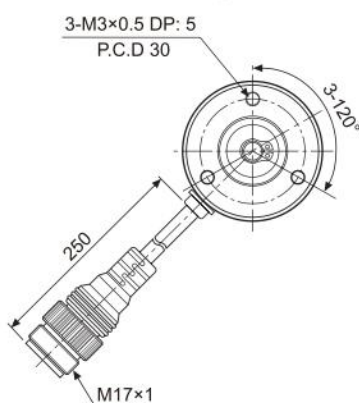
(unit: mm)



● Hollow shaft / Hollow shaft built-in type

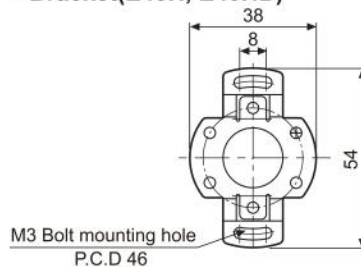


◎ Connector cable type



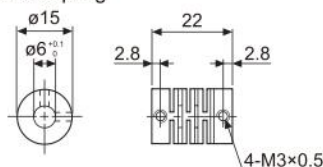
※Connector cable is customizable and refer to the G-10 for specifications.

● Bracket(E40H, E40HB)



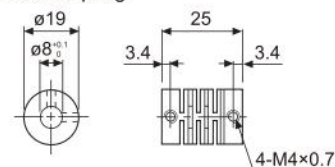
● Coupling(E40S)

● ø6 Coupling



- Parallel misalignment : Max. 0.25mm
- Angular misalignment: Max. 5°
- End-play: Max. 0.2mm

● ø8 Coupling



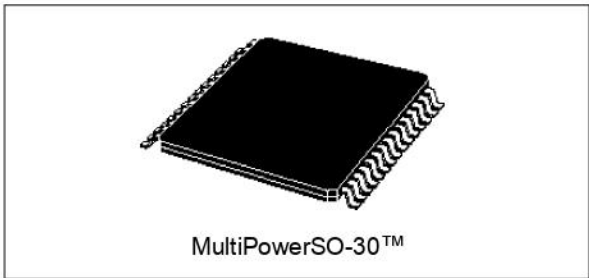
- ※For parallel misalignment, angular misalignment, end-play terms, refer to the F-78 page.
- ※For flexible coupling(ERB Series) information, refer to the F-71 page.



VNH2SP30-E

Automotive fully integrated H-bridge motor driver

Datasheet - production data



Features

Type	$R_{DS(on)}$	I_{out}	V_{CCmax}
VNH2SP30-E	19 mΩ max (per leg)	30 A	41 V

- AEC-Q100 qualified
- 5 V logic level compatible inputs
- Undervoltage and overvoltage shutdown
- Overvoltage clamp
- Thermal shutdown
- Cross-conduction protection
- Linear current limiter
- Very low standby power consumption
- PWM operation up to 20 kHz
- Protection against loss of ground and loss of V_{CC}
- Current sense output proportional to motor current
- Package: ECOPACK®



Description

The VNH2SP30-E is a full bridge motor driver intended for a wide range of automotive applications. The device incorporates a dual monolithic high side driver and two low side switches. The high side driver switch is designed

using STMicroelectronics well known and proven proprietary VIPower™ M0 technology which permits efficient integration on the same die of a true power MOSFET with intelligent signal/protection circuitry.

The low side switches are vertical MOSFETs manufactured using STMicroelectronics proprietary EHD (STripFET™) process. The three die are assembled in a MultiPowerSO-30 package on electrically isolated leadframes. This package, specifically designed for the harsh automotive environments, offers improved thermal performance thanks to exposed die pads. Moreover, its fully symmetrical mechanical design allows superior manufacturability at board level. The input signals IN_A and IN_B can directly interface with the microcontroller to select the motor direction and brake condition. The $DIAG_A/EN_A$ or $DIAG_B/EN_B$, when connected to an external pull-up resistor, enable one leg of the bridge. They also provide a feedback digital diagnostic signal. The normal operating condition is explained in the truth table. The motor current can be monitored with the CS pin by delivering a current proportional to its value. The speed of the motor can be controlled in all possible conditions by the PWM up to 20 kHz. In all cases, a low level state on the PWM pin will turn off both the LS_A and LS_B switches. When PWM rises to a high level, LS_A or LS_B turn on again depending on the input pin state.

Table 1. Device summary

Package	Order code
	Tape and reel
MultiPowerSO-30	VNH2SP30TR-E

1 **Block diagram and pin description**

Figure 1. Block diagram

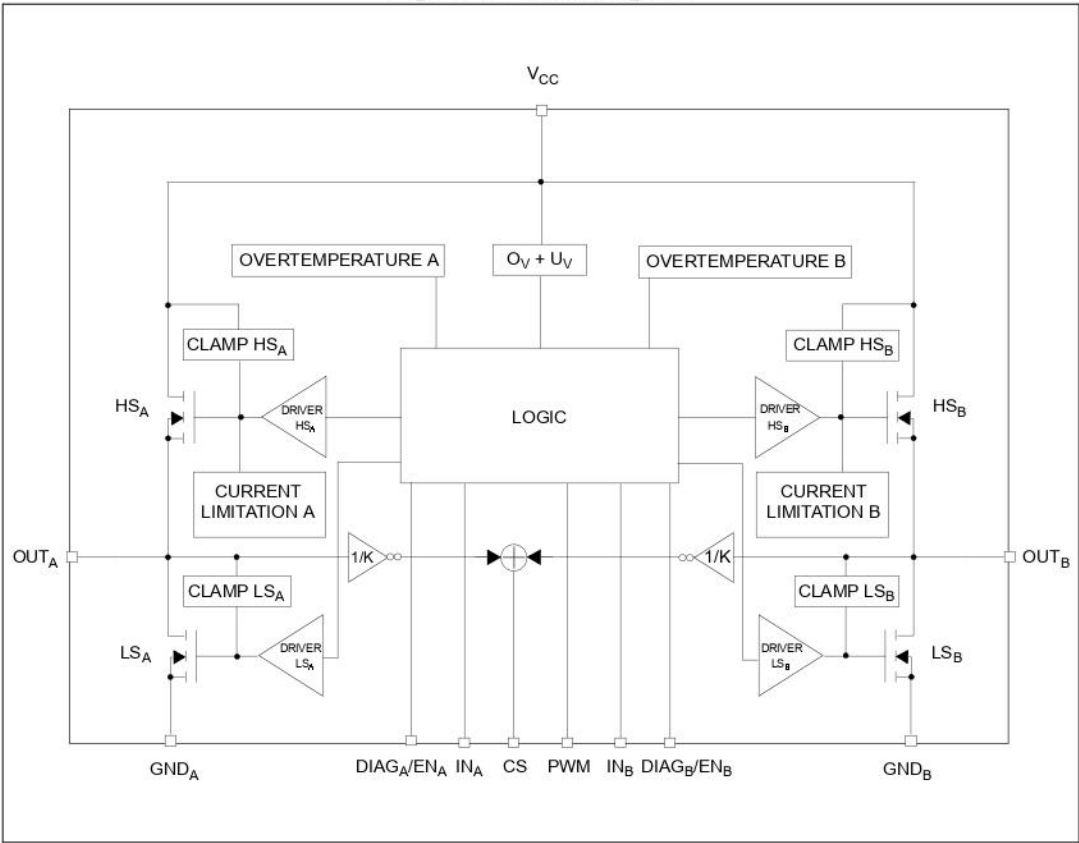


Table 2. Block description

Name	Description
Logic control	Allows the turn-on and the turn-off of the high side and the low side switches according to the truth table
Overvoltage + undervoltage	Shuts down the device outside the range [5.5V..16V] for the battery voltage
High side and low side clamp voltage	Protects the high side and the low side switches from the high voltage on the battery line in all configurations for the motor
High side and low side driver	Drives the gate of the concerned switch to allow a proper $R_{DS(on)}$ for the leg of the bridge
Linear current limiter	Limits the motor current by reducing the high side switch gate-source voltage when short-circuit to ground occurs
Overtemperature protection	In case of short-circuit with the increase of the junction's temperature, shuts down the concerned high side to prevent its degradation and to protect the die
Fault detection	Signals an abnormal behavior of the switches in the half-bridge A or B by pulling low the concerned $EN_x/DIAG_x$ pin

Block diagram and pin description

VNH2SP30-E

Figure 2. Configuration diagram (top view)

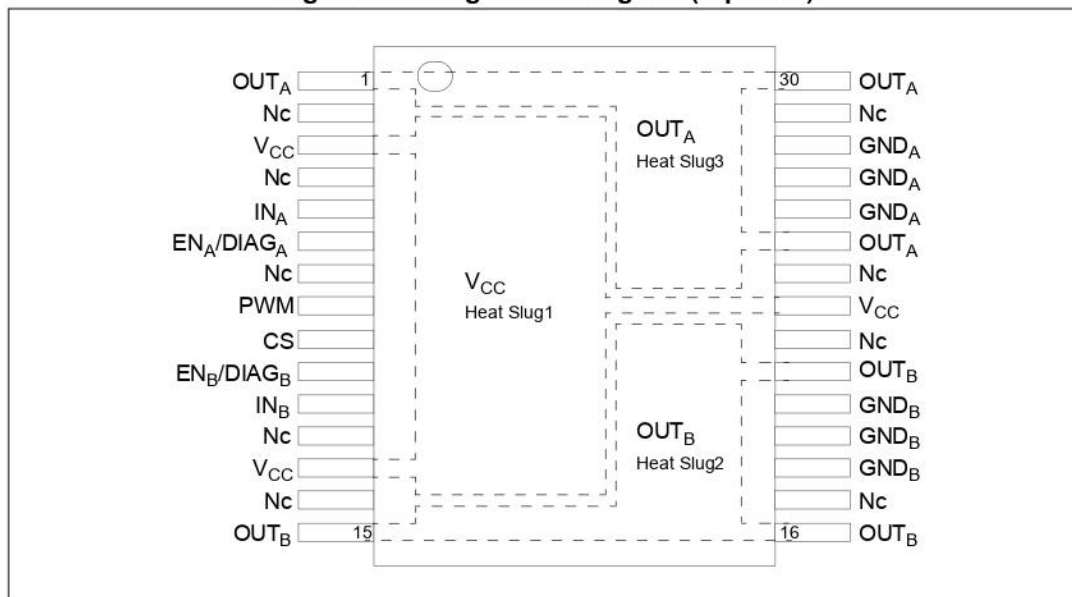


Table 3. Pin definitions and functions

Pin no.	Symbol	Function
1, 25, 30	OUT _A , Heat Slug3	Source of high side switch A / Drain of low side switch A
2, 4, 7, 12, 14, 17, 22, 24, 29	NC	Not connected
3, 13, 23	V _{CC} , Heat Slug1	Drain of high side switches and power supply voltage
6	EN _A /DIAG _A	Status of high side and low side switches A; open drain output
5	IN _A	Clockwise input
8	PWM	PWM input
9	CS	Output of current sense
11	IN _B	Counter clockwise input
10	EN _B /DIAG _B	Status of high side and low side switches B; open drain output
15, 16, 21	OUT _B , Heat Slug2	Source of high side switch B / Drain of low side switch B
26, 27, 28	GND _A	Source of low side switch A ⁽¹⁾
18, 19, 20	GND _B	Source of low side switch B ⁽¹⁾

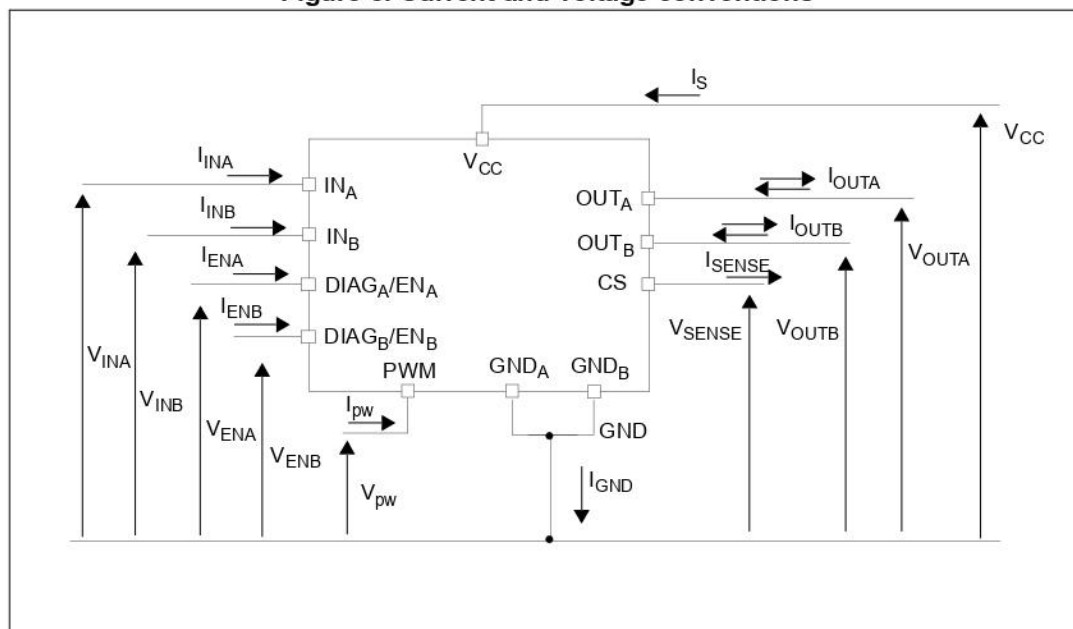
1. GND_A and GND_B must be externally connected together.

Table 4. Pin functions description

Name	Description
V _{CC}	Battery connection
GND _A , GND _B	Power grounds; must always be externally connected together
OUT _A , OUT _B	Power connections to the motor
IN _A , IN _B	Voltage controlled input pins with hysteresis, CMOS compatible. These two pins control the state of the bridge in normal operation according to the truth table (brake to V _{CC} , brake to GND, clockwise and counterclockwise).
PWM	Voltage controlled input pin with hysteresis, CMOS compatible. Gates of low side FETs are modulated by the PWM signal during their ON phase allowing speed control of the motor.
EN _A /DIAG _A , EN _B /DIAG _B	Open drain bidirectional logic pins. These pins must be connected to an external pull up resistor. When externally pulled low, they disable half-bridge A or B. In case of fault detection (thermal shutdown of a high side FET or excessive ON state voltage drop across a low side FET), these pins are pulled low by the device (see truth table in fault condition).
CS	Analog current sense output. This output sources a current proportional to the motor current. The information can be read back as an analog voltage across an external resistor.

2 Electrical specifications

Figure 3. Current and voltage conventions



2.1 Absolute maximum ratings

Table 5. Absolute maximum ratings

Symbol	Parameter	Value	Unit
V_{CC}	Supply voltage	+41	V
I_{max}	Maximum output current (continuous)	30	A
I_R	Reverse output current (continuous)	-30	
I_{IN}	Input current (IN_A and IN_B pins)	± 10	mA
I_{EN}	Enable input current ($DIAG_A/EN_A$ and $DIAG_B/EN_B$ pins)	± 10	
I_{pw}	PWM input current	± 10	
V_{CS}	Current sense maximum voltage	-3/+15	V
V_{ESD}	Electrostatic discharge ($R = 1.5k\Omega$, $C = 100pF$)		
	– CS pin	2	kV
	– logic pins	4	kV
	– output pins: OUT_A , OUT_B , V_{CC}	5	kV
T_j	Junction operating temperature	Internally limited	$^{\circ}C$
T_c	Case operating temperature	-40 to 150	
T_{STG}	Storage temperature	-55 to 150	

2.2 Electrical characteristics

V_{CC} = 9V up to 16 V; -40°C < T_J < 150°C, unless otherwise specified.

Table 6. Power section

Symbol	Parameter	Test conditions	Min	Typ	Max	Unit
V _{CC}	Operating supply voltage		5.5		16	V
I _S	Supply current	Off state with all Fault Cleared & EN _x =0 I _{NA} = I _{NB} = PWM = 0; T _J = 25°C; V _{CC} = 13V I _{NA} = I _{NB} = PWM = 0 Off state: I _{NA} = I _{NB} = PWM = 0		12	30 60	μA μA mA
		On state: I _{NA} or I _{NB} = 5V, no PWM			10	mA
R _{ONHS}	Static high side resistance	I _{OUT} = 15A; T _J = 25°C			14	mΩ
		I _{OUT} = 15A; T _J = -40 to 150°C			28	
R _{ONLS}	Static low side resistance	I _{OUT} = 15A; T _J = 25°C			5	
		I _{OUT} = 15A; T _J = -40 to 150°C			10	
V _f	High side free-wheeling diode forward voltage	I _f = 15A		0.8	1.1	V
I _{L(off)}	High side off state output current (per channel)	T _J = 25°C; V _{OUTX} = EN _X = 0V; V _{CC} = 13V			3	μA
		T _J = 125°C; V _{OUTX} = EN _X = 0V; V _{CC} = 13V			5	
I _{RM}	Dynamic cross-conduction current	I _{OUT} = 15A (see Figure 7)		0.7		A

Table 7. Logic inputs (I_{NA}, I_{NB}, EN_A, EN_B)

Symbol	Parameter	Test conditions	Min	Typ	Max	Unit
V _{IL}	Input low level voltage	Normal operation (DIAG _X /EN _X pin acts as an input pin)			1.25	V
V _{IH}	Input high level voltage		3.25			
V _{IHYST}	Input hysteresis voltage		0.5			
V _{ICL}	Input clamp voltage	I _{IN} = 1mA	5.5	6.3	7.5	
		I _{IN} = -1mA	-1.0	-0.7	-0.3	
I _{INL}	Input low current	V _{IN} = 1.25V	1			μA
I _{INH}	Input high current	V _{IN} = 3.25V			10	
V _{DIAG}	Enable output low level voltage	Fault operation (DIAG _X /EN _X pin acts as an output pin); I _{EN} = 1mA			0.4	V