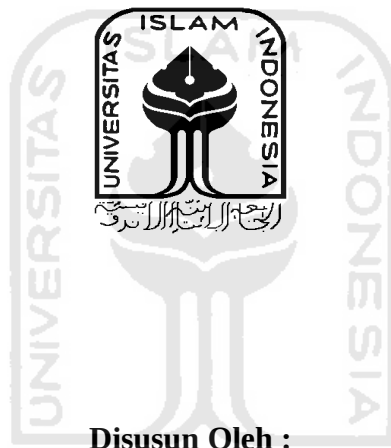


**PEMODELAN DAN PEMBUATAN PROTOTIPE SISTEM  
KENDALI POSISI DAN KECEPATAN SUDUT**

**TUGAS AKHIR**

**Diajukan Sebagai Salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana Teknik Mesin**



**Disusun Oleh :**

**Nama : Vicky Indra Pratama**

**No. Mahasiswa : 16525002**

**NIRM : 2016010498**

**JURUSAN TEKNIK MESIN  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
YOGYAKARTA**

**2020**

## PERNYATAAN KEASLIAN

Dengan ini saya menyatakan bahwa Tugas Akhir yang saya buat ini merupakan karya sendiri bukan hasil plagiarisme dari karya tulis orang lain. Semua referensi dalam Tugas Akhir ini saya cantumkan sitasi dan sumber pustakanya. Apabila saya dianggap melakukan pelanggaran hak intelektual dan Tugas Akhir yang saya buat ini tidak benar maka saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Yogyakarta, 21 September 2020



Vicky Indra Pratama



**LEMBAR PENGESAHAN DOSEN PEMBIMBING**

**PEMODELAN DAN PEMBUATAN PROTOTIPE SISTEM  
KENDALI POSISI DAN KECEPATAN SUDUT**



**Disusun Oleh :**

**Nama : Vicky Indra Pratama**

**No. Mahasiswa : 16525002**

**NIRM : 2016010498**

Yogyakarta, 4 Agustus 2020

Pembimbing I,

A handwritten signature in black ink, appearing to be 'Purtojo'.

Purtojo S.T., M.Sc

**LEMBAR PENGESAHAN DOSEN PENGUJI**

**PEMODELAN DAN PEMBUATAN PROTOTIPE SISTEM  
KENDALI POSISI DAN KECEPATAN SUDUT**

**TUGAS AKHIR**

**Disusun Oleh :**

**Nama : Vicky Indra Pratama**

**No. Mahasiswa : 16525002**

**NIRM : 2016010498**

**Tim Penguji**

Purtojo, S.T., M.Sc.

Ketua

Tanggal : 18/09/2020

Muhammad Ridlwan, S.T., M.T

Anggota I

Tanggal : 15/09/2020

Donny Suryawan, S.T., M.Eng

Anggota II

Tanggal : 16/09/2020



Mengetahui

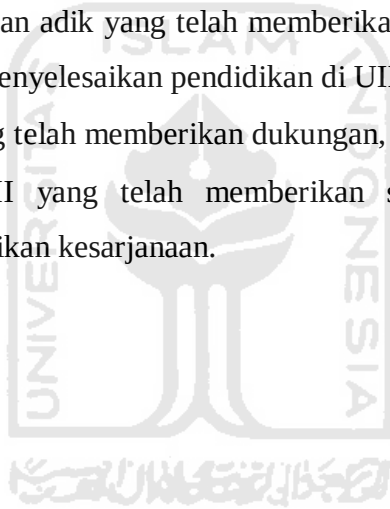
Ketua Jurusan Teknik Mesin

Dr. Eng. Risdiono, S.T., M.Eng

## HALAMAN PERSEMBAHAN

Alhamdulillah segala puji bagi Allah Subhanahu Wa Ta'ala yang telah memberikan kemudahan, kelancaran, dan kesempatan dalam mengerjakan tugas akhir yang berjudul “Pemodelan dan Pembuatan Prototipe Sistem Kendali Posisi dan Kecepatan Sudut”. Tidak lupa shalawat serta salam selalu tercurahkan pada junjungan besar Nabi Muhammad Shallallahu ‘Alaihi Wasallam, manusia yang menjadi suri tauladan yang baik dan semoga kita termasuk dalam syafaatnya, aamiin. Tugas akhir yang saya kerjakan masih jauh dari kata sempurna karena kesempurnaan hanya milik Allah Subhanahu Wa Ta'ala. Tugas akhir ini saya persembahkan untuk :

- Kedua orang tua dan adik yang telah memberikan kasih sayang, doa, dan dukungan untuk menyelesaikan pendidikan di UII.
- Teman-teman yang telah memberikan dukungan, semangat, dan motivasi.
- Teknik mesin UII yang telah memberikan saya kesempatan untuk menempuh pendidikan keserjanaan.



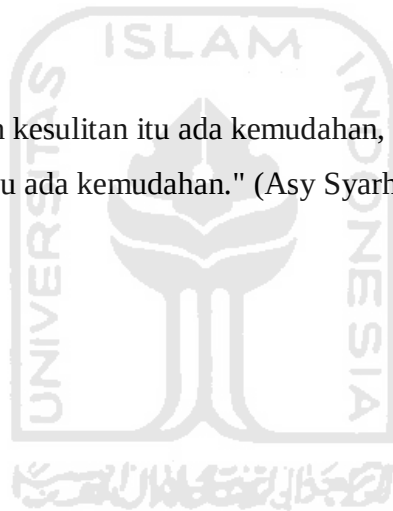
## HALAMAN MOTTO

“Witing tresna jalaran saka kulina, Witing mulya jalaran wani rekasa”

“Urip Kudu Urup”

“Ilmu pengetahuan itu bukanlah yang dihafal, melainkan yang memberi manfaat” (Imam Syafi’i)

"Sesungguhnya sesudah kesulitan itu ada kemudahan, sesungguhnya sesudah kesulitan itu ada kemudahan." (Asy Syarh ayat 5-6)



## KATA PENGANTAR

Alhamdulillah segala puji bagi Allah Subhanahu Wa Ta'ala yang telah memberikan rahmat dan hidayah-Nya sehingga laporan tugas akhir yang berjudul "Pemodelan dan Pembuatan Prototipe Sistem Kendali Posisi dan Kecepatan Sudut" dapat diselesaikan dengan baik dan tepat waktu. Tidak lupa shalawat serta salam selalu tucurahkan pada junjungan besar Nabi Muhammad Shallallahu 'Alaihi Wasallam, semoga kita termasuk dalam syafaatnya, aamiin.

Laporan tugas akhir ini disusun sebagai bentuk pertanggungjawaban setelah melakukan semua kegiatan akademik di Teknik Mesin, Fakultas Teknologi Industri, Universitas Islam Indonesia dan syarat untuk memperoleh gelar sarjana. Penulis berusaha menyusun laporan tugas akhir ini dalam bentuk sederhana, mudah dipahami, dan menarik agar pembaca tertarik membaca dan dapat dengan mudah memahami laporan tugas akhir ini. Laporan tugas akhir ini disusun berdasarkan catatan-catatan terdahulu yang terkait sebagai landasan teori dan data-data yang diperoleh dari percobaan.

Penulis mengucapkan terima kasih kepada:

1. Allah Allah Subhanahu Wa Ta'ala yang telah memberikan kemudahan dan petunjuk dalam menyelesaikan laporan tugas akhir ini.
2. Nabi Muhammad Shallallahu 'Alaihi Wasallam yang telah memberikan suri tauladan yang baik.
3. Kedua orang tua dan adik yang telah memberikan kasih sayang, doa, dan dukungan untuk menyelesaikan pendidikan di UII.
4. Bapak Purtojo S.T., M.Sc sebagai dosen pembimbing tugas akhir yang telah membagikan ilmunya dalam membantu menyelesaikan tugas akhir ini.
5. Teman-teman yang dapat mengurangi tingkat kejenuhan dalam penyelesaian tugas akhir maupun kuliah.
6. Teknik mesin UII yang telah memberikan penulis kesempatan untuk menempuh pendidikan kesarjana.
7. Pihak-pihak terkait yang tidak dapat dituliskan satu persatu.

Begitulah yang penulis harapkan dan lakukan. Penulis masih merasa banyak kekurangan dalam menyusun laporan dan mengharap kepada pembaca dan semua pihak untuk memberikan masukan. Semoga laporan ini bermanfaat bagi pembacanya.

## **Pemodelan dan Pembuatan Prototipe Sistem Kendali Posisi dan Kecepatan Sudut**

**Vicky Indra Pratama**

### **ABSTRAK**

Perkembangan teknologi menimbulkan lahirnya otomasi pada industri yang juga dipelajari di bidang akademik. Perancangan sistem kendali posisi dan kecepatan sudut adalah salah satu contohnya. Penelitian ini dilakukan dengan membuat model matematika dan prototipe sistem kendali posisi dan kecepatan sudut. Model matematika diperoleh dari perhitungan *datasheet* dan eksperimen. Prototipe dibuat menggunakan material logam dan perangkat elektronik. Perangkat keras arduino menerima input dari perangkat lunak Scilab yang akan mengendalikan motor dc. Gerakan dan posisi motor dc dimonitor oleh enkoder. Output dari enkoder kemudian dibaca oleh arduino dan respon sistem kendali ditampilkan pada perangkat lunak Scilab. Sistem kendali posisi dan kecepatan sudut dibuat berdasarkan simulasi dan implementasi. Simulasi berdasarkan perhitungan yang variabelnya diperoleh dari *datasheet* dan eksperimen. Implementasi dilakukan dengan menghubungkan objek kendali dengan perangkat lunak. Hasil yang diperoleh adalah perbandingan respon dari simulasi dan implementasi berdasarkan variabel *rise time*, *peak time*, *percent overshoot*, *settling time*, dan *error steady state*. Kesimpulan dari penelitian ini adalah pembuatan prototipe sistem kendali posisi dan kecepatan sudut berhasil terealisasi, model matematika berdasarkan *datasheet* dan eksperimen berhasil dibuat, dan sistem kendali posisi dan kecepatan sudut berhasil dibuat yang menghasilkan perbandingan parameter respon kendali berdasarkan *datasheet*, eksperimen, dan implementasi.

Kata kunci : kendali posisi dan kecepatan sudut, arduino, motor DC, enkoder, Scilab

### ***Modeling and Manufacturing of Angular Position and Angular Speed Control System Prototypes***

**Vicky Indra Pratama**

### **ABSTRACT**

*Technological developments led to the birth of automation in industry which also studied in the academic. The design of angular position and speed control systems is one example. This research was making mathematical models and prototypes of angular position and angular speed control systems. Mathematical models are obtained from datasheet and experiments. The prototype was made using metal materials and electronic devices. Arduino hardware accepts input from Scilab software which will control the dc motor. The movement and position*

*of the dc motor is monitored by encoder. The output from the encoder is then read by arduino and control system response is displayed in the Scilab software. The angularposition and angularspeed control system is based on simulation and implemetation. Simulation based on calculation that the variables are obtained from datasheets and experiments. Implementation is done by by connecting the control object with software. The result obtained are comparison of responses from simulation and implemetation based on rise time, peak time, percent overshoot, settling time, and error steady state variables. The conclusion of this research is that the prototype of the angularposition and angularspeed control system was successfully realized, a mathematical model based on the datasheet and experiment was successfully created, and theangular position and angularspeed control system was successfully created which produced a comparison of control response parameters based on the datasheet, experiment, and implemetation.*

*Keyword : angularposition and angularspeed control, arduino, DC motor, encoder, Scilab*



## DAFTAR ISI

|                                                    |                                     |
|----------------------------------------------------|-------------------------------------|
| Halaman Judul .....                                | i                                   |
| PERNYATAAN KEASLIAN .....                          | <b>Error! Bookmark not defined.</b> |
| Lembar Pengesahan Dosen Pembimbing .....           | iii                                 |
| Lembar Pengesahan Dosen Penguji .....              | iv                                  |
| Halaman Persembahan .....                          | v                                   |
| Halaman Motto .....                                | vi                                  |
| Kata Pengantar .....                               | vii                                 |
| ABSTRAK.....                                       | viii                                |
| ABSTRACT.....                                      | viii                                |
| Daftar Isi .....                                   | x                                   |
| Daftar Tabel .....                                 | xii                                 |
| Daftar Gambar .....                                | xiii                                |
| Daftar Notasi.....                                 | xvi                                 |
| Bab 1 Pendahuluan.....                             | 1                                   |
| 1.1 Latar Belakang.....                            | 1                                   |
| 1.2 Rumusan Masalah.....                           | 3                                   |
| 1.3 Batasan Masalah .....                          | 4                                   |
| 1.4 Tujuan Penelitian atau Perancangan .....       | 4                                   |
| 1.5 Manfaat Penelitian atau Perancangan .....      | 4                                   |
| 1.6 Sistematika Penulisan.....                     | 4                                   |
| Bab 2 Tinjauan Pustaka .....                       | 6                                   |
| 2.1 Kajian Pustaka .....                           | 6                                   |
| 2.2 Dasar Teori 1 .....                            | 7                                   |
| 2.2.1 Motor DC.....                                | 7                                   |
| 2.2.2 <i>Driver Motor(Motor Shield L298)</i> ..... | 7                                   |
| 2.2.3 <i>Rotary Encoder</i> .....                  | 8                                   |
| 2.2.4 Arduino.....                                 | 8                                   |
| 2.2.5 Catu Daya .....                              | 9                                   |
| 2.2.6 Perangkat Lunak Scilab.....                  | 10                                  |
| 2.2.7 Model Matematika pada Motor DC.....          | 10                                  |

|                                                                 |    |
|-----------------------------------------------------------------|----|
| 2.2.7.1 Perhitungan Fungsi Transfer dari Eksperimen .....       | 10 |
| 2.2.7.2 Perhitungan Fungsi Transfer dari <i>Datasheet</i> ..... | 19 |
| 2.2.8 Parameter pada Respon Sistem.....                         | 21 |
| 2.2.9 RL <i>Low-Pass Filter</i> .....                           | 22 |
| Bab 3 Metode Penelitian .....                                   | 25 |
| 3.1 Alur Penelitian .....                                       | 25 |
| 3.2 Peralatan dan Bahan.....                                    | 26 |
| 3.2.1 Perangkat Keras .....                                     | 26 |
| 3.2.2 Perangkat Lunak .....                                     | 30 |
| 3.3 Perancangan.....                                            | 30 |
| 3.3.1 Perancangan Perangkat Keras.....                          | 30 |
| 3.3.2 Perancangan Perangkat Lunak.....                          | 32 |
| Bab 4 Hasil dan Pembahasan.....                                 | 35 |
| 4.1 Hasil Perancangan.....                                      | 35 |
| 4.1.1 Perhitungan Fungsi Transfer .....                         | 35 |
| 4.1.2 Perhitungan RL <i>Low-Pass Filter</i> .....               | 37 |
| 4.1.3 Hasil Perancangan Perangkat Keras .....                   | 37 |
| 4.1.4 Hasil Perancangan Perangkat Lunak.....                    | 38 |
| 4.2 Hasil Pengujian .....                                       | 43 |
| 4.3 Analisis dan Pembahasan .....                               | 55 |
| Bab 5 Penutup.....                                              | 57 |
| 5.1 Kesimpulan.....                                             | 57 |
| 5.2 Saran atau Penelitian Selanjutnya .....                     | 57 |
| Daftar Pustaka.....                                             | 58 |

## DAFTAR TABEL

|                                                            |    |
|------------------------------------------------------------|----|
| Tabel 3.1 Tabel penjelasan blok Xcos Scilab .....          | 34 |
| Tabel 4.1 Perbandingan respon pengendalian kecepatan ..... | 55 |
| Tabel 4.2 Perbandingan respon pengendalian posisi .....    | 56 |



## DAFTAR GAMBAR

|                                                                         |    |
|-------------------------------------------------------------------------|----|
| Gambar 2.1 Motor DC.....                                                | 7  |
| Gambar 2.2 Driver motor L298 .....                                      | 8  |
| Gambar 2.3 Rotary encoder.....                                          | 8  |
| Gambar 2.4 Arduino.....                                                 | 9  |
| Gambar 2.5 Catu daya .....                                              | 9  |
| Gambar 2.6 Software Scilab .....                                        | 10 |
| Gambar 2.7 Motor DC.....                                                | 11 |
| Gambar 2.8 Blok diagram <i>field-controlled dc motor</i> .....          | 13 |
| Gambar 2.9 Blok diagram <i>armature-controlled dc motor</i> .....       | 15 |
| Gambar 2.10 Fungsi transfer <i>armature-controlled dc motor</i> .....   | 16 |
| Gambar 2.11 (a) Respon motor. (b) Estimasi $\tau_m$ .....               | 17 |
| Gambar 2.12 Motor DC : a. Skema, b. Blok diagram.....                   | 19 |
| Gambar 2.13 Pembebanan mekanik pada motor .....                         | 20 |
| Gambar 2.14 Gambar respon sistem .....                                  | 22 |
| Gambar 2.15 Rangkaian RL untuk membentuk <i>filter low-pass</i> . ..... | 22 |
| Gambar 3.1 Diagram alir penelitian.....                                 | 25 |
| Gambar 3.2 Maxon 2322.983-11.225-200 .....                              | 26 |
| Gambar 3.3 Motor shield L298.....                                       | 27 |
| Gambar 3.4 Arduino mega 2560.....                                       | 27 |
| Gambar 3.5 Catu daya 24V 2A.....                                        | 28 |
| Gambar 3.6 Kabel jumper .....                                           | 28 |
| Gambar 3.7 Pulley dan belt .....                                        | 28 |
| Gambar 3.8 Bracket motor DC dan <i>rotary encoder</i> .....             | 29 |
| Gambar 3.9 Lengan penunjuk.....                                         | 29 |
| Gambar 3.10 Base .....                                                  | 29 |
| Gambar 3.11 Software Scilab .....                                       | 30 |
| Gambar 3.12 Arduino IDE .....                                           | 30 |
| Gambar 3.13 Rotary servo base unit.....                                 | 31 |
| Gambar 3.14 Skema rangkaian elektrik .....                              | 31 |
| Gambar 3.15 Hasil rangkaian perangkat keras .....                       | 32 |

|                                                                                                              |    |
|--------------------------------------------------------------------------------------------------------------|----|
| Gambar 4.1 Hasil perancangan perangkat keras.....                                                            | 38 |
| Gambar 4.2 Skema blok pengukuran gir reduksi.....                                                            | 38 |
| Gambar 4.3 Skema blok pengukuran fungsi transfer eksperimen .....                                            | 39 |
| Gambar 4.4 Skema blok fungsi transfer orde 1 <i>open loop</i> .....                                          | 39 |
| Gambar 4.5 Skema blok fungsi transfer orde 2 <i>open loop</i> .....                                          | 40 |
| Gambar 4.6 Skema blok pengendali kecepatan close loop .....                                                  | 40 |
| Gambar 4.7 Skema blok pengendali posisi close loop.....                                                      | 41 |
| Gambar 4.8 Skema blok pengendali kecepatan pada perangkat keras.....                                         | 41 |
| Gambar 4.9 Skema blok pengendali posisi pada perangkat keras .....                                           | 42 |
| Gambar 4.10 Hasil perancangan skema blok pengukuran gir reduksi .....                                        | 43 |
| Gambar 4.11 Respon skema blok pengukuran gir reduksi.....                                                    | 44 |
| Gambar 4.12 Hasil perancangan skema blok pengukuran fungsi transfer eksperimen.....                          | 44 |
| Gambar 4.13 Respon skema blok pengukuran fungsi transfer eksperimen .....                                    | 45 |
| Gambar 4.14 Hasil perancangan skema blok fungsi transfer dari <i>datasheet</i> orde 1 <i>open loop</i> ..... | 45 |
| Gambar 4.15 Respon skema blok fungsi transfer <i>datasheet</i> orde 1 <i>open loop</i> .....                 | 46 |
| Gambar 4.16 Hasil perancangan skema blok fungsi transfer dari eksperimen orde 1 <i>open loop</i> .....       | 46 |
| Gambar 4.17 Respon skema blok fungsi transfer eksperimen orde 1 <i>open loop</i> ..                          | 47 |
| Gambar 4.18 Hasil perancangan skema blok fungsi transfer dari <i>datasheet</i> orde 2 <i>open loop</i> ..... | 47 |
| Gambar 4.19 Respon skema blok fungsi transfer <i>datasheet</i> orde 2 <i>open loop</i> .....                 | 48 |
| Gambar 4.20 Hasil skema blok fungsi transfer dari eksperimen orde 2 <i>open loop</i> .....                   | 48 |
| Gambar 4.21 Respon skema blok fungsi dari eksperimen transfer orde 2 <i>open loop</i> .....                  | 49 |
| Gambar 4.22 Hasil perancangan skema blok pengendali kecepatan <i>datasheet close loop</i> .....              | 49 |
| Gambar 4.23 Respon skema blok pengendali kecepatan <i>datasheet close loop</i> .....                         | 50 |
| Gambar 4.24 Hasil perancangan skema blok pengendali kecepatan eksperimen <i>close loop</i> .....             | 50 |

|                                                                                               |    |
|-----------------------------------------------------------------------------------------------|----|
| Gambar 4.25 Respon skema blok pengendali kecepatan eksperimen <i>close loop</i> .             | 51 |
| Gambar 4.26 Hasil perancangan skema blok pengendali posisi <i>datasheet close loop</i> .....  | 51 |
| Gambar 4.27 Respon skema blok pengendali posisi <i>datasheet close loop</i> .....             | 52 |
| Gambar 4.28 Hasil perancangan skema blok pengendali posisi eksperimen <i>close loop</i> ..... | 52 |
| Gambar 4.29 Respon skema blok pengendali posisi eksperimen <i>close loop</i> .....            | 53 |
| Gambar 4.30 Hasil skema blok pengendali kecepatan pada perangkat keras .....                  | 53 |
| Gambar 4.31 Respon skema blok pengendali kecepatan pada perangkat keras...                    | 54 |
| Gambar 4.32 Hasil perancangan skema blok pengendali posisi pada perangkat keras.....          | 54 |
| Gambar 4.33 Respon skema blok pengendali posisi pada perangkat keras .....                    | 55 |



## DAFTAR NOTASI

|          |                                              |
|----------|----------------------------------------------|
| $i_f$    | = arus medan                                 |
| $i_a$    | = arus <i>armature</i>                       |
| $T$      | = torsi                                      |
| $J$      | = momen inersia                              |
| $\theta$ | = perpindahan sudut                          |
| $f$      | = koefisien gesekan <i>bearing</i>           |
| $u$      | = tegangan input                             |
| $\tau_f$ | = <i>electrical time constant</i>            |
| $\tau_m$ | = <i>mechanical time constant</i>            |
| $v_b$    | = tegangan <i>back emf</i>                   |
| $L_a$    | = induktansi <i>armature</i>                 |
| $k_m$    | = <i>motor gain constant</i>                 |
| $w$      | = kecepatan sudut                            |
| $a$      | = tegangan listrik                           |
| $B$      | = kuat medan magnet                          |
| $l$      | = panjang konduktor                          |
| $e$      | = tegangan listrik                           |
| $v$      | = kecepatan konduktor normal ke medan magnet |
| $v_b$    | = <i>back emf</i>                            |
| $K_b$    | = <i>back emf constant</i>                   |
| $e_a$    | = tegangan <i>armature</i>                   |
| $T_m$    | = torsi motor                                |
| $K_t$    | = <i>motor torque constant</i>               |
| $D_m$    | = <i>viscous damping</i>                     |
| $R_a$    | = resistansi <i>armature</i>                 |

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan teknologi yang pesat tidak lepas dari revolusi industri yang dimulai pada abad ke-18 yang ditandai dengan adanya temuan mesin uap. Seabad kemudian penggunaan listrik, minyak, dan produksi baja menyebabkan produksi massal yang menandai lahirnya revolusi industri kedua. Pada tahun 1970, penggunaan elektronik dan otomasi industri menyebabkan pergantian revolusi industri dari kedua menjadi ketiga. Di abad ke-20 revolusi industri sudah memasuki tahap keempat yang ditandai dengan virtualisasi industri. Dewasa ini, implementasi dari virtualisasi industri sudah banyak diterapkan, seperti pada industri manufaktur.

Industri 4.0 tidak lepas dari komputasi sistem pada manufaktur. Robot mengendalikan hampir semua proses di manufaktur agar waktu produksi lebih efisien tanpa mengurangi kualitas produk. Suatu robot dikendalikan oleh sistem kendali yang dirancang untuk melakukan suatu perintah. Output perintah pergerakan robot dapat berupa posisi, kecepatan, atau percepatan. Selain di industri sistem kendali juga ada di bidang akademis untuk bahan penelitian, misalnya pada perancangan pengendalian posisi dan kecepatan sudut.

Penelitian sistem kendali pada perancangan pengendalian posisi dan kecepatan sudut berfokus untuk menggerakkan lengan penunjuk dengan output posisi dan/atau kecepatan. Kualitas perancangan pengendalian posisi dan kecepatan sudut dipengaruhi oleh perangkat keras dan lunak. Perangkat lunak mengirimkan perintah untuk menggerakkan motor DC dan enkoder akan membaca pergerakan motor DC secara *realtime*. Motor DC diharapkan mampu bergerak sesuai dengan input yang diberikan oleh kontroler dan enkoder diharapkan mampu membaca pergerakan motor DC secara akurat dan presisi. Ketelitian

enkoder dibutuhkan agar hasil simulasi dengan eksperimen aktual tidak jauh berbeda.

Perancangan pengendalian posisi dan kecepatan sudut menggunakan metode konvensional, yaitu dengan menghitung fungsi transfer. Parameter pada fungsi transfer antara lain *steady state*, *error steady state*, *overshoot*, *rise time*, dan *settling time*. Identifikasi fisik motor DC dilakukan untuk dapat menghitung fungsi transfer. Fungsi transfer yang didapatkan kemudian disimulasikan agar mengetahui responnya. Selanjutnya dilakukan perbandingan antara respon fungsi transfer dengan respon aktual dari objek kendali.

Kelebihan dari perancangan pengendalian posisi dan kecepatan sudut yang dibuat adalah biaya perancangan lebih ekonomis daripada produk yang sudah ada, penggunaan software bersifat *open-sources*, komponen-komponen *hardware* relatif mudah didapatkan di Indonesia, dan dimensi keseluruhan produk relatif kecil sehingga mudah disimpan maupun dibawa. Kekurangan dari perancangan pengendalian posisi dan kecepatan sudut yang dibuat adalah *base* menggunakan plat besi dengan ketebalan 5 mm sehingga produk relatif berat dan dimensi *base* tidak presisi sempurna karena pembuatan *base* menggunakan cara manual.

Permasalahan yang terjadi dalam proses membuat desain sistem kendali adalah adanya perusahaan dari motor DC yang tidak mencantumkan *datasheet* sehingga menyulitkan dalam merancang sistem kendali. Tidak hanya motor DC, enkoder dengan tingkat ketelitian yang rendah juga dapat menjadi masalah karena dapat mempengaruhi hasil dari eksperimen aktual dan simulasi software. Simulasi implementasi fungsi transfer pada objek kendali juga menjadi masalah karena tidak 100% sesuai dengan respon simulasi fungsi transfer. Tidak sesuainya respon tersebut terjadi pada parameter-parameter seperti terjadi *error steady state*, *percent overshoot* terlalu besar, dan *settling time* yang terlalu lama. Hal ini dapat terjadi akibat keterbatasan objek kendali.

Permasalahan pada perancangan pengendalian posisi dan kecepatan sudut tidak dapat diselesaikan secara sempurna tetapi dapat

diminimalisir. Solusi untuk mengatasi permasalahan pada motor DC adalah menggunakan motor DC yang memiliki *datasheet* untuk mempermudah perancangan pengendalian posisi dan kecepatan sudut. Penggunaan enkoder dengan tingkat ketelitian tinggi juga akan mengatasi permasalahan dan mempermudah pembacaan pergerakan motor DC. Diharapkan respon parameter-parameter fungsi transfer yang diimplementasikan pada objek kendali berada pada batas toleransi. Untuk mengatasi atau meminimalisir keterbatasan objek kendali dilakukan dengan pemilihan komponen objek kendali yang berkualitas.

Berdasarkan uraian yang sudah dibahas, perkembangan teknologi menimbulkan lahirnya otomasi pada industri. Otomasi industri juga dipelajari di bidang akademik. Pembuatan desain sistem kendali seperti perancangan pengendalian posisi dan kecepatan sudut adalah salah satu contohnya. Perancangan pengendalian posisi dan kecepatan sudut bertujuan untuk merancang sistem kendali dengan output dapat menggerakkan lengan penunjuksesuai dengan keinginan. Perancangan ini tidak lepas dari peran perangkat lunak dan keras. Perangkat lunak akan menterjemahkan input yang diberikan menjadi bahasa pemrograman untuk menggerakkan perangkat keras berupa objek kendali. Kualitas objek kendali akan menentukan respon yang dihasilkan.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah disampaikan, untuk membuat desain sistem kendali dapat dilakukan dengan menghitung fungsi transfer yang kemudian disimulasikan pada software untuk melihat responnya. Kemudian diimplementasikan pada objek kendali. Bagaimana cara membuat desain sistem kendali dengan output posisi dan kecepatan sudut? Bagaimana perbandingan respon sistem kendali posisi dan kecepatan sudut berdasarkan perhitungan dan implementasi?

### 1.3 Batasan Masalah

Penelitian ini dilakukan sesuai dengan batasan-batasan permasalahan sebagai berikut :

- a. Perancangan pengendalian posisi dan kecepatan sudut hanya sampai pada pembuatan prototipe
- b. Membuat model matematika
- c. Driver motor DC menggunakan L298.
- d. MotorDCdisambungkan enkoder menggunakan belt.
- e. Jenis enkoder yang digunakan adalah *rotary encoder*.
- f. Sebagian besar rototipe menggunakan material logam dan akrilik.
- g. Simulasi menggunakan perangkat lunak *Scilab*.

### 1.4 Tujuan Penelitian atau Perancangan

Penelitian inimemiliki tujuan sebagai berikut :

- a. Membuat prototipe sistem kendali posisi dan kecepatan sudut dengan arduino sebagai perangkat komunikasi antara objek kendali dengan PC, material *base* dari logam, dan lengan penunjuk dari akrilik.
- b. Membuat model matematika dari prototipe berupa fungsi transfer.
- c. Merancang kendali posisi dan kecepatan sudut *close loop* dengan memperhatikan parameter respon kendali berdasarkan *datasheet*, eksperimen, dan implementasi.

### 1.5 Manfaat Penelitian atau Perancangan

Manfaat dari penelitian yang dilakukan penulisadalah sebagai dasar dalam perancangan sistem kendali yang nantinya diharapkan akan dilakukan penelitian dan perbaruan perangkat lunak dan keras lebih lanjut.

### 1.6 Sistematika Penulisan

Laporan tugas akhir ditulis dalam lima bab. Bab pertama adalah pendahuluan yang berisi mengapa penelitian perlu dilakukan, gambaran tentang penelitian, tujuan penelitian, dan manfaat penelitian. Bab dua

merupakan tinjauan pustaka yang berisi tentang referensi penelitian yang sudah ada dan teori-teori penelitian. Bab tiga berisi tentang bagaimana alur penelitian, alat dan bahan yang digunakan, perancangan, dan pengujian. Bab empat menjelaskan mengenai hasil dan pembahasan penelitian. Bab lima berisi kesimpulan penelitian dan saran.



## BAB 2

### TINJAUAN PUSTAKA

#### 2.1 Kajian Pustaka

Pada penelitian (Fitriansyah, Waluyo, & Syahrial, 2013) menjelaskan bagaimana mengendalikan kecepatan putaran motor DC dengan metode PID (*Proportional Integral Derivatif*). Perangkat keras yang digunakan adalah motor DC, *tachometer*, dan *silicon controlled rectifier* dengan Simulink sebagai perangkat lunak pendukung. Berdasarkan penelitian nilai yang diperoleh untuk mencapai kecepatan yang diinginkan sebesar 1750 rpm adalah  $K_p = 0,2$ ,  $K_I = 1,6$ , dan  $K_D = 0,2$ . Parameter yang dihasilkan memiliki *error steady state* 0,999% dan *settling time* 0,84 detik pada *rise time* 0,49 detik dengan nilai *peak* pada 0,99. Kecepatan awal yang dihasilkan mendekati *set point* yang diinginkan pada detik kedua dan kecepatannya tidak mengalami penurunan atau *steadys* sampai dengan detik ke-100.

Penelitian (Sailan & Kuhnert, 2013) menjelaskan tentang bagaimana mengendalikan posisi angular menggunakan kontroler PID (*Proportional Integral Derivatif*) dengan tujuan untuk mengendalikan pompa hidrolik. Motor DC dihubungkan dengan katup pompa hidrolik, katup akan membuka dan menutup pada rentang sudut tertentu. Perangkat keras yang digunakan adalah motor DC, *potentiometer*, katup pompa, pompa hidrolik, dan atmega 16 *microcontroller* sedangkan perangkat lunak yang digunakan adalah Simulink. Hasil dari penelitian tersebut adalah pengendalian posisi angular dengan kontroler PID memiliki kelebihan pada pengurangan kebisingan, tingkat osilasi yang rendah, kendali sistem yang berjalan dengan baik, dan respon sistem yang baik.

## 2.2 Dasar Teori 1

### 2.2.1 Motor DC

Motor DC (gambar 2.1) merupakan jenis motor yang menggunakan tegangan searah sebagai sumber tenaganya. Dengan memberikan beda tegangan pada kedua terminal tersebut, motor akan berputar pada satu arah, dan bila polaritas dari tegangan tersebut dibalik maka arah putaran motor akan terbalik pula. Polaritas dari tegangan yang diberikan pada dua terminal menentukan arah putaran motor sedangkan besar dari beda tegangan pada kedua terminal menentukan kecepatan motor (Chapman, 1991).

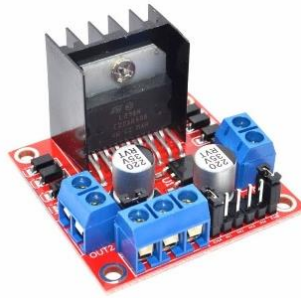


Gambar 2.1 Motor DC

### 2.2.2 Driver Motor(Motor Shield L298)

L298 (gambar 2.2) adalah komponen elektronik yang dipergunakan untuk mengontrol arah putaran motor DC. Satu buah L298 bisa dipergunakan untuk mengontrol dua buah motor DC. Selain bisa dipergunakan untuk mengontrol arah putaran motor DC, L298 ini pun bisa dipergunakan sebagai *motor driver stepper bipolar*.

IC driver L298 memiliki kemampuan menggerakkan motor DC sampai arus 2A dan tegangan maksimum 40 volt dc untuk satu kanalnya. Pin enable A dan B untuk mengendalikan jalan atau kecepatan motor, pin input 1 sampai 4 digunakan untuk mengendalikan arah putaran. Pin output pada IC L298 13 dihubungkan ke motor DC yang sebelumnya melalui dioda yang disusun secara H-bridge. Pengaturan kecepatan motor digunakan teknik PWM (*pulse width modulation*) yang diinputkan dari mikrokontroler melalui pin *Enable*. PWM untuk kecepatan rotasi yang bervariasi level highnya (Adriansyah & Hidyatama, 2013). Gambar 2.2 merupakan *driver motor* L298.



Gambar 2.2 Driver motor L298

### 2.2.3 Rotary Encoder

*Rotary encoder* (gambar 2.3) adalah perangkat elektromekanik yang dapat memonitor gerakan dan posisi. Secara umum *Rotary encoder* 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 kendali. *Rotary encoder* umumnya digunakan pada pengendalian robot, motor *driver*, dan sebagainya (Rizqiawan, 2009).

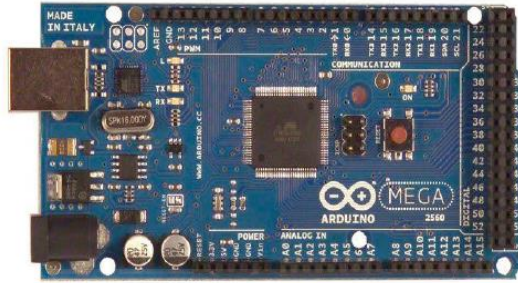


Gambar 2.3 Rotary encoder

### 2.2.4 Arduino

Arduino (gambar 2.4) adalah *platform prototyping open-source hardware* yang dapat digunakan untuk membuat proyek berbasis pemrograman. *Hardware* Arduino memiliki prosesor mikrokontroler ATMegayang dirilis oleh Atmel AVR, tetapi *software* yang digunakan memiliki bahasa pemrograman tersendiri. Arduino dirancang untuk memudahkan pengguna elektronik atau siapapun yang ingin mengembangkan peralatan elektronik interaktif berdasarkan *hardware* dan

*software* yang fleksibel dan mudah untuk digunakan. Dalam berbagai aplikasi, Arduino dapat digunakan untuk mendeteksi lingkungan dengan menerima *input* dari berbagai sensor atau tombol (sensor cahaya, suhu, inframerah, ultrasonik, jarak, tekanan, kelembaban) dan dapat mengontrol perangkat lainnya seperti mengontrol kecepatan dan arah putar motor, menyalakan LED, dan sebagainya. (Junaidi & Prabowo, 2018).



Gambar 2.4 Arduino

### 2.2.5 Catu Daya

Catu daya (gambar 2.5) adalah sebuah perangkat yang memasok listrik energi untuk satu atau lebih beban listrik. Catu daya menjadi bagian yang penting dalam elektronika yang berfungsi sebagai sumber tenaga listrik misalnya pada baterai atau accu. Pada dasarnya *power supply* ini mempunyai konstruksi rangkaian yang hampir sama yaitu terdiri dari trafo, penyearah, dan penghalus tegangan. Istilah ini paling sering diterapkan ke perangkat yang mengubah satu bentuk energi listrik yang lain, meskipun juga dapat merujuk ke perangkat yang mengkonversi bentuk energi lain (misalnya, mekanik, kimia, solar) menjadi energi listrik. Secara umum prinsip rangkaian catu daya terdiri atas komponen utama yaitu ;transformator, dioda dan kondensator (Sitohang, Mamahit, & Tulung , 2018).



Gambar 2.5 Catu daya

### 2.2.6 Perangkat Lunak Scilab

Scilab (gambar 2.6) adalah *software free* dan *open source* untuk komputasi numerik. *Software* ini menyediakan lingkungan untuk perhitungan yang dibutuhkan bidangnya sains dan keteknikan. Scilab dirilis sebagai *open source* dibawah lisensi CeCILL, dan dapat didownload secara gratis. Scilab tersedia pada sistem operasi GNU/Linux, Mac OS X dan Windows. Scilab juga memiliki dukungan *enterprise* bagi perusahaan-perusahaan yang membutuhkan dukungan. Scilab menyediakan ratusan fungsi matematis. *Software* ini memiliki bahasa pemrograman tingkat tinggi dan dapat mengakses struktur data, plot 2D, plot 3D dan fungsi grafis (Aulia, 2013).



Gambar 2.6 *Software* Scilab

### 2.2.7 Model Matematika pada Motor DC

Model matematika berupa fungsi transfer. Menurut (Ogata, 2010) fungsi transfer suatu sistem adalah model matematika karena merupakan metode operasional untuk mengekspresikan persamaan diferensial yang menghubungkan variabel output ke variabel input. Jika fungsi transfer suatu sistem diketahui, output atau respon dapat dipelajari untuk berbagai bentuk input dengan pandangan untuk memahami sifat sistem. Jika fungsi transfer tidak diketahui maka dapat diperoleh secara eksperimen dengan input yang diketahui dan mempelajari output sistem.

#### 2.2.7.1 Perhitungan Fungsi Transfer dari Eksperimen

Berdasarkan (Chen, 1995) motor sangat diperlukan dalam banyak sistem kendali. Motor digunakan untuk memutar antena, teleskop, dan kemudi kapal; untuk membuka dan menutup katup; untuk menggerakkan kaset dalam perekam, dan rol di pabrik baja. Terdapat beberapa tipe motor: DC, AC, stepper, dan

hidrolik. Medan magnet motor DC dapat tereksitasi oleh rangkaian yang terhubung secara seri dengan rangkaian *armature* yang dapat juga tereksitasi oleh rangkaian medan yang tidak tergantung pada dengan rangkaian *armature* atau magnet permanen. Motor DC digunakan untuk mengendalikan sistem yang disebut juga dengan motor servo yang memiliki ciri-ciri rasio torsi inersia rotor besar, ukuran kecil, dan karakteristik linier yang baik. Tentu saja harganya lebih mahal dari motor DC biasa.

### Field-Controlled DC Motor

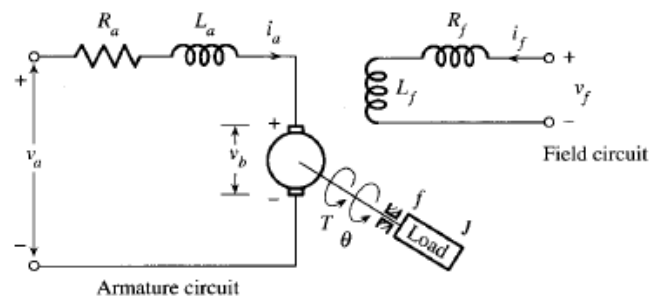
Sebagian besar motor DC digunakan pada sistem kendali yang dimodelkan dalam persamaan (2.1). Terdapat dua rangkaian yang disebut dengan rangkaian medan dan rangkaian *armature*. Arus medan  $i_f$ ,  $i_a$  arus *armature*, dan torsi  $T$  yang dihasilkan oleh motor dituliskan dalam persamaan berikut

$$T(t) = k i_a(t) i_f(t) \quad (2.1)$$

dimana  $k$  adalah konstanta. Torsi yang dihasilkan digunakan untuk mengerakkan beban melalui poros yang diasumsikan kaku. Analisis disederhanakan dengan hanya mempertimbangkan gesekan antara poros dengan *bearing*. Total momen inersia beban, poros, dan rotor motor  $J$ ; perpindahan sudut beban  $\theta$ ; dan koefisien gesekan *bearing*  $f$  sehingga dapat ditulis persamaannya seperti berikut

$$T(t) = J \frac{d^2 \theta(t)}{dt^2} + f \frac{d\theta(t)}{dt} \quad (2.2)$$

Persamaan tersebut merupakan hubungan antara torsi motor dan perpindahan beban angular



Gambar 2.7 Motor DC

Jika arus *armature*  $i_a$  tetap konstan dan tegangan input  $u(t)$  diaplikasikan pada rangkaian medan, motor disebut *field-controlled dc motor*.

Pada *field-controlled dc motor* arus *armature*  $i_a(t)$  adalah konstan. Oleh karena itu (2.1) dapat direduksi menjadi

$$T(t) = k i_a(t) i_f(t) = k_f(t) i_f(t) \quad (2.3)$$

dimana  $k_f = i_a$ , dari rangkaian medan diperoleh

$$L_f \frac{di_f(t)}{dt} + R_f i_f(t) = v_t = u(t) \quad (2.4)$$

Transformasi Laplace digunakan pada persamaan (2.3) dan (2.4)<sup>1</sup>, dengan asumsi kondisi awal nol,

$$\begin{aligned} T(s) &= k_f I_f(s) \\ L_f s I_f(s) + R_f I_f(s) &= U(s) \\ I_f(s) &= \frac{1}{L_f s + R_f} U(s) \\ T(s) &= \frac{k_f}{L_f s + R_f} U(s) \end{aligned}$$

Jadi, jika torsi yang dihasilkan dianggap sebagai output motor kemudian fungsi transfer dari *field-controlled dc motor* adalah

$$G_m(s) = \frac{T(s)}{U(s)} = \frac{k_f}{L_f s + R_f} \quad (2.5)$$

Fungsi transfer tetap sama, apapun yang membebani gerakan motor. Perhitungan fungsi transfer dengan beban dituliskan berikut ini, dengan menggunakan persamaan Laplace ke persamaan (2.2) dan asumsi kondisi awal nol

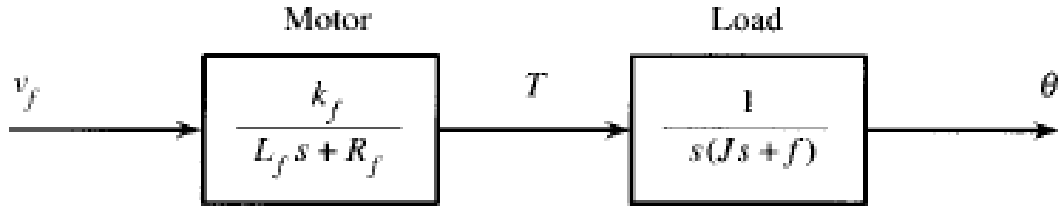
$$T(s) = J s^2 \theta(s) + f s \theta(s)$$

---

<sup>1</sup> Huruf kapital digunakan untuk menunjukkan transformasi Laplace dari huruf yang sesuai. Dalam kasus  $T(t)$  dan  $T(s)$  digunakan argumen untuk membedakannya

$$G_l(s) = \frac{\theta(s)}{T(s)} = \frac{1}{s(Js + f)}$$

Persamaan diatas adalah fungsi transfer beban jika mempertimbangkan input torsi motor dan output beban perpindahan sudut. Jadi, motor dan beban dalam gambar dapat dimodelkan seperti gambar 2.8.



Gambar 2.8 Blok diagram *field-controlled dc motor*

Terdapat dua blok pada gambar: blok sebelah kiri merepresentasikan motor dan blok sebelah kanan beban. Penambahan atau pengurangan beban tidak mempengaruhi fungsi transfer.

Fungsi transfer *field controlled dc motor* dengan beban dituliskan dalam persamaan berikut

$$\begin{aligned} G(s) = G_m(s)G_l(s) &= \frac{\theta(s)}{T(s)} = \frac{k_f}{s(L_f s + R_f)(Js + f)} = \frac{k_f/R_f f}{s(\tau_f s + 1)(\tau_m s + 1)} \\ &= \frac{k_m}{(\tau_f s + 1)s(\tau_m s + 1)} \end{aligned} \quad (2.6)$$

dimana  $\tau_f = \frac{L_f}{R_f}$ ,  $\tau_m = \frac{J}{f}$ , dan  $k_m = \frac{k_f}{R_f f}$ . Konstanta  $\tau_f$  hanya tergantung pada rangkaian elektrik yang dapat disebut dengan *electrical time constant* dan konstanta  $\tau_m$  hanya tergantung pada beban yang dapat disebut dengan *mechanical time constant*. Jika *electrical time constant*  $\tau_f$  lebih kecil dari *mechanical time constant*  $\tau_m$ , seperti yang sering terjadi dalam praktiknya,  $G(s)$  dapat dirumuskan

$$G(s) = \frac{k_m}{s(\tau_m s + 1)} \quad (2.7)$$

*Field controlled dc motor* digunakan untuk menggerakkan beban, namun tidak ada masalah pada persamaan (2.2). Bagaimana ini dapat terjadi? Perbedaan beban akan menyebabkan perbedaan  $i_a$ ; bahkan untuk beban yang sama,  $i_a$  tidak akan menjadi konstanta. Oleh karena itu, permasalahan pembebanan dieliminasi dengan menjaga konstanta  $i_a$ . Dalam praktiknya menjaga konstanta  $i_a$  sangatlah sulit, jadi *field controlled dc motor* jarang digunakan.

### **Armature-Controlled DC Motor**

Berdasarkan motor DC pada gambar 2.7, jika arus medan  $i_f(t)$  tetap konstan atau rangkaian medan diubah menjadi medan magnet permanen dan jika tegangan input diterapkan ke rangkaian *armature* motor disebut *armature controlled dc motor*. Fungsi transfer dikembangkan dan jika  $i_f(t)$  adalah konstanta maka menjadi

$$T(t) = k_t i_a(t) \quad (2.8)$$

dimana  $k_t = k_i f(t)$  adalah konstanta. Ketika motor menggerakkan beban, *back electromotive force (back emf)* tegangan  $v_b$  akan dikembangkan di rangkaian *armature* untuk menahan tegangan yang diberikan. Tegangan  $v_b$  berbanding lurus dengan kecepatan sudut poros motor:

$$v_b(t) = k_b \frac{d\theta(t)}{dt} \quad (2.9)$$

Jadi rangkaian *armature* pada gambar 2.7 dituliskan dalam persamaan

$$R_a i_a(t) + L_a \frac{di_a(t)}{dt} + v_b(t) = v_a(t) = u(t) \quad (3.10)$$

atau

$$R_a i_a(t) + L_a \frac{di_a(t)}{dt} + k_b \frac{d\theta(t)}{dt} = u(t) \quad (3.11)$$

Persamaan fungsi transfer dikembangkan dari (2.2), (2.8), dan (2.11). Substitusi (2.8) ke (2.2) dan aplikasi transformasi Laplace ke (2.2) dan (2.11) dengan asumsi kondisi awal nol menghasilkan persamaan

$$k_t I_a(s) = Js^2 \theta(s) + fs \theta(s) \quad (2.12)$$

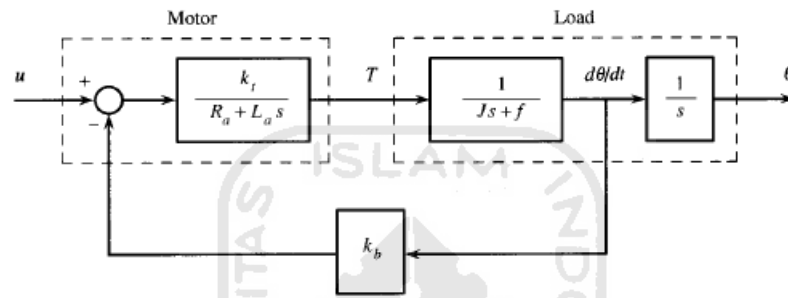
$$R_a I_a(s) + L_a s I_a(s) + k_b \theta(s) = U(s) \quad (2.13)$$

Eliminasi  $I_a$  dari kedua persamaan menghasilkan

$$G(s) = \frac{\theta(s)}{U(s)} = \frac{k_t}{s[(Js + f)(R_a + L_a s) + k_t k_b]} \quad (2.14)$$

Fungsi transfer tersebut dari  $v_a = u$  ke  $\theta$  dari *armature-controlled dc motor*.

Persamaan (2.8), (2.9), (3.10), dan (2.12) untuk *armature-controlled dc motor* dapat digambarkan ke dalam diagram blok seperti berikut



Gambar 2.9 Blok diagram *armature-controlled dc motor*

Meskipun dapat digambarkan blok untuk motor dan blok untuk beban, dua blok tidak terpisah seperti pada *field-controlled dc motor*. Sinyal dari dalam beban adalah umpan balik ke input motor. Pada dasarnya hal ini menangani masalah pembebanan karena baban, lebih mudah untuk menggabungkan motor dan beban ke dalam satu blok dengan fungsi transfer.

Pada penggunaannya, induktansi *armature*  $L_a$  sering diatur menjadi nol<sup>2</sup>.

Pada kasus ini, persamaan (2.14) direduksi ke

$$G(s) = \frac{\theta(s)}{U(s)} = \frac{k_t}{s(JR_a s + k_t k_b + f R_a)} = \frac{k_m}{s(\tau_m s + 1)} \quad (2.15)$$

dimana

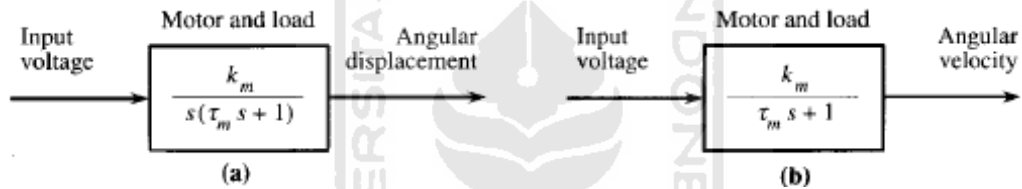
<sup>2</sup> Dalam keteknikan, sering dikatakan bahwa *electrical time constant* lebih kecil dari *mechanical time constant* sehingga  $L_a = 0$ . Jika (2.14) ditulis  $G(s) = k_m / s(as + 1)(bs + 1)$  maka  $a$  dan  $b$  disebut *time constant*. Tidak seperti (2.6),  $a$  dan  $b$  tergantung pada kedua bagian elektrik dan mekanik motor. Oleh karena itu, *electrical* dan *mechanical time constant* di *armature-controlled dc motor* tidak lagi didefinisikan di *field-controlled dc motor*.

$$k_m = \frac{k_t}{k_t k_b + f R_a} = \text{motor gain constant} \quad (2.16)$$

dan

$$\tau_m = \frac{J R_a}{k_t k_b + f R_a} = \text{motor time constant} \quad (2.17)$$

*Time constant* tergantung pada beban serta *armature circuit*. Hal ini kontras dengan persamaan (2.7) dimana  $\tau_m$  hanya tergantung pada beban. *Electrical* dan *mechanical time constants* terdefiniskan pada (2.6) dan (2.15) tetapi hal ini tidak tepat di (2.14) dan (2.15). Pada (2.14) dan (2.15) sering disebut fungsi transfer motor, meskipun kedua hal tersebut sebenarnya fungsi transfer motor dan beban. *Input* dan *output* dari fungsi transfer (2.15) adalah tegangan elektrik yang diterapkan dan posisi sudut poros motor seperti pada gambar 2.10(a).



Gambar 2.10 Fungsi transfer *armature-controlled dc motor*

Kecepatan sudut poros motor  $w$  maka diperoleh  $w(t) = d\theta/dt$  dan

$$W(s) = s\Theta(s) = s \cdot \frac{k_m}{s(\tau_m s + 1)} \cdot U(s) = \frac{k_m}{\tau_m s + 1} \cdot U(s)$$

Fungsi transfer motor dari tegangan yang diterapkan  $u$  untuk kecepatan sudut  $w$  adalah

$$\frac{W(s)}{U(s)} = \frac{k_m}{\tau_m s + 1} \quad (2.18)$$

Pada gambar 2.10(b) persamaan (2.18) berbeda dari (2.15) oleh ketiadaan satu pole pada  $s = 0$ . Jadi, fungsi transfer motor dapat (2.15) atau (2.18), tergantung pada apa yang dianggap sebagai output. Oleh karena itu, penting untuk menentukan *input* dan *output* menggunakan fungsi transfer. Perbedaan spesifikasi menyebabkan fungsi transfer yang berbeda untuk sistem yang sama.

## Pengukuran Fungsi Transfer Motor

Fungsi transfer motor yang menggerakkan beban disebutkan pada (2.15). Untuk menggunakan persamaan tersebut maka (2.16) dan (2.17) digunakan untuk menghitung  $k_m$  dan  $\tau_m$  yang jika dibalik membutuhkan  $k_t, k_b, R_a, L_a, J$  dan  $f$ . Meskipun empat konstanta pertama mungkin disediakan oleh manufaktur motor, konstanta lain yang dibutuhkan adalah  $J$  dan  $f$ . Momen inersia  $J$  beban tidak mudah dihitung jika mempunyai bentuk reguler dan  $f$  tidak dapat diperoleh secara analitis. Oleh karena itu, untuk memperoleh  $J$  dan  $f$  dilakukan pengukuran. Pengukuran juga dapat digunakan untuk mengetahui fungsi transfer secara langsung. Berikut ini cara mengukur fungsi transfer motor.

Fungsi transfer motor dengan beban dituliskan pada (2.15). Tegangan listrik dan kecepatan angular digunakan untuk pengukuran, kemudian fungsi transfer dikurangi menjadi (2.18). Jika tegangan listrik  $a$  kecepatan dapat dirumuskan

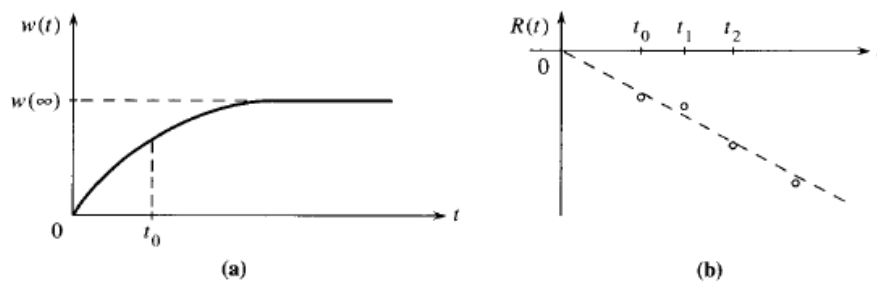
$$W(s) = \frac{k_m}{\tau_m s + 1} \cdot \frac{a}{s} = \frac{k_m a}{s} - \frac{k_m a \tau_m}{\tau_m s + 1} \quad (2.19)$$

Invers transformasi Laplace persamaan tersebut adalah

$$w(t) = k_m a - k_m a e^{-t/\tau_m}$$

untuk  $t \geq 0$ , karena  $\tau_m$  adalah positif, hubungan  $e^{-t/\tau_m}$  mendekati nol seperti  $t$  mendekati tak terbatas dan respon  $w(t)$  adalah seperti pada gambar 2.11(a). Jadi, dapat dituliskan

$$w(\infty) = k_m a \quad (2.20)$$



Gambar 2.11 (a) Respon motor. (b) Estimasi  $\tau_m$

Hal ini disebut kecepatan akhir atau *steady-state speed*. Jika tegangan listrik diketahui besarnya  $a$  dan ukuran kecepatan akhir  $w(\infty)$ , kemudian *motor gain constant* dapat ditulis kembali (2.19) seperti

$$w(t) = k_m a (1 - e^{-t/\tau_m})$$

yang berarti dengan menggunakan (2.20),

$$e^{-t/\tau_m} = 1 - \frac{w(t)}{k_m a} = 1 - \frac{w(t)}{w(\infty)}$$

$$-\frac{t}{\tau_m} = \ln \left( 1 - \frac{w(t)}{w(\infty)} \right) = R(t) \quad (2.21)$$

Jika kecepatan pada  $t$  berapapun, katakanlah  $t = t_0$  kemudian dari (2.21) diperoleh

$$\tau_m = \frac{-t_0}{\ln \left( 1 - \frac{w(t)}{w(\infty)} \right)} \quad (2.22)$$

Jadi, *time motor constant* dapat diperoleh dari kecepatan akhir dan satu tambahan pengukuran pada  $t$  berapapun.

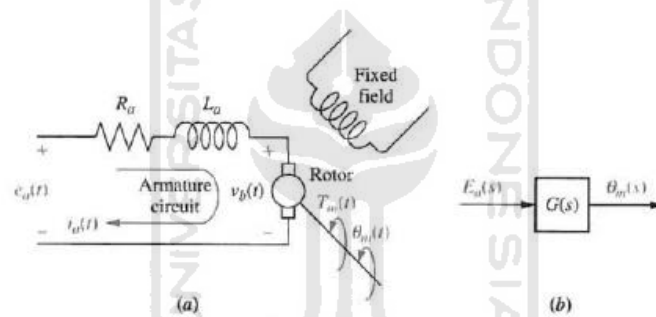
Pada (2.22) selain kecepatan akhir hanya digunakan satu datum untuk mengukur  $\tau_m$ , untuk  $\tau_m$  yang lebih akurat dapat diperoleh dari data yang banyak. Jika rencana  $R(t)$  pada (2.21) untuk bilangan  $t$  seperti gambar 2.11(b) dan menggambar dari titik awal garis lurus melewati titik-titik ini. Kemudian kemiringan garis lurus sama dengan  $1/\tau_m$ . Hal ini lebih akurat untuk memperoleh *motor time constant*. Metode ini juga dapat memeriksa apakah fungsi transfer yang disederhanakan pada (2.18) dan (2.15) dapat digunakan atau tidak. Jika semua titik pada rencana sangat dekat dengan garis lurus, maka (2.18) dan (2.15) cukup memadai. Jika tidak fungsi transfer motor yang lebih akurat seperti yang ada pada (2.14) harus digunakan untuk mendiskripsikan motor.

Masalah dalam menentukan  $\tau_m$  dan  $k_m$  pada (2.18) dari pengukuran disebut *parameter estimation*. Pada masalah ini, bentuk fungsi transfer

diasumsikan diketahui tetapi parameter tersebut tidak diketahui. Kemudian menentukan parameter dari pengukuran. Hal ini adalah kasus khusus dari masalah identifikasi dimana baik bentuk maupun parameternya diasumsikan tidak diketahui.

### 2.2.7.2 Perhitungan Fungsi Transfer dari *Datasheet*

Berdasarkan (Nise, 2011) pada gambar 2.12(a) medan magnet dihasilkan oleh magnet permanen yang tidak bergerak atau elektromagnet yang tidak berubah disebut *fixed field*. Sirkuit berputar disebut *armature*, melalui arus  $i_a(t)$  yang mengalir melewati medan magnet di sudut kanan dan menghasilkan gaya  $F = Bli_a(t)$  dimana  $B$  adalah kuat medan magnet dan  $l$  adalah panjang konduktor. Torsi yang dihasilkan memutar rotor, bagian yang berputar pada motor.



Gambar 2.12 Motor DC : a. Skema, b. Blok diagram

Terdapat fenomena lain yang terjadi di motor. Konduktor bergerak pada sudut kanan medan magnet yang menghasilkan tegangan di terminal konduktor sama dengan  $e = Blv$  dimana  $e$  adalah tegangan dan  $v$  adalah kecepatan konduktor normal ke medan magnet, karena *armature* pembawa arus berputar dalam medan magnet, tegangannya sebanding dengan kecepatan. Jadi,

$$v_b(t) = K_b \frac{d\theta_m(t)}{dt} \quad (2.23)$$

dimana *back electromotive force (back emf)* adalah  $v_b$ ;  $K_b$  adalah konstanta proporsionalitas yang disebut *back emf constant*; dan  $d\theta_m(t)/dt$  adalah kecepatan angular motor, dengan menggunakan transformasi Laplace diperoleh

$$V_b(s) = K_b s \theta_m(s) \quad (2.24)$$

Hubungan antara arus *armature*  $i_a(t)$ , penerapan tegangan *armature*  $e_a(t)$  dan *back emf*  $v_b(t)$  dapat ditemukan dengan menulis persamaan loop disekitar *armature circuit* pengubah Laplace :

$$R_a I_a(s) + L_a s I_a(s) + V_b(s) = E_a(s) \quad (2.25)$$

Torsi dihasilkan oleh motor yang sebanding dengan arus *armature*. Jadi,

$$T_m(s) = K_t I_a(s) \quad (2.26)$$

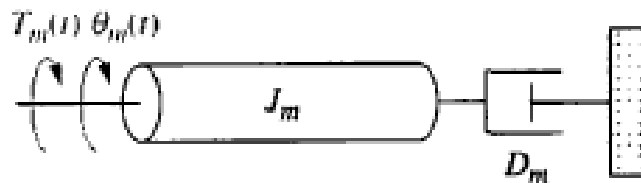
dimana  $T_m$  adalah torsi yang dihasilkan oleh motor, dan  $K_t$  adalah konstanta proporsionalitas yang disebut *motor torque constant*, yang tergantung pada motor dan karakteristik medan magnet. Dalam satu set unit yang konsisten nilai  $K_t$  sama dengan nilai  $K_b$ . Penulisan ulang persamaan (2.26)

$$I_a(s) = \frac{1}{K_t} T_m(s) \quad (2.27)$$

Untuk mencari fungsi transfer motor, persamaan (2.24) dan (2.27) disubstitusikan ke (2.25), menghasilkan

$$\frac{(R_a + L_a s) T_m(s)}{K_t} + K_b s \theta_m(s) = E_a(s) \quad (2.28)$$

$T_m$  dicari pada syarat  $\theta_m(s)$  jika memisahkan variabel input dan output dan memperoleh fungsi transfer  $\theta_m(s)/E_a(s)$ .



Gambar 2.13 Pembebanan mekanik pada motor

Gambar 2.13 memperlihatkan pembebanan mekanik pada motor.  $J_m$  adalah inersia yang setara pada *armature* dan termasuk kedua inersia *armature* dan beban inersia yang tercermin ke *armature*.  $D_m$  adalah *viscous damping* yang setara pada *armature* dan termasuk kedua *armature viscous damping* dan beban *viscous damping* yang tercermin ke *armature*. Dari gambar 2.13,

$$T_m(s) = (J_ms^2 + D_ms)\theta_m(s) \quad (2.29)$$

Substitusi ke persamaan (2.29) ke persamaan (2.28) menghasilkan

$$\frac{(R_a + L_as) = (J_ms^2 + D_ms)\theta_m(s)}{K_t} + K_bs\theta_m(s) = E_a(s) \quad (2.30)$$

Jika induktansi *armature*  $L_a$  lebih kecil dibandingkan dengan resistansi *armature*  $R_a$  yang biasanya digunakan untuk motor DC, persamaan (2.30) menjadi

$$\left[ \frac{R_a}{K_t} (J_ms + D_m) + K_b \right] s\theta_m(s) = E_a(s) \quad (2.31)$$

Setelah disederhanakan, fungsi transfer yang diinginkan  $\theta_m(s)/E_a(s)$  menjadi

$$\frac{\theta_m(s)}{E_a(s)} = \frac{K_t/(R_aJ_m)}{s \left[ s + \frac{1}{J_m} \left( D_m + \frac{K_tK_b}{R_a} \right) \right]} \quad (2.32)^3$$

## 2.2.8 Parameter pada Respon Sistem

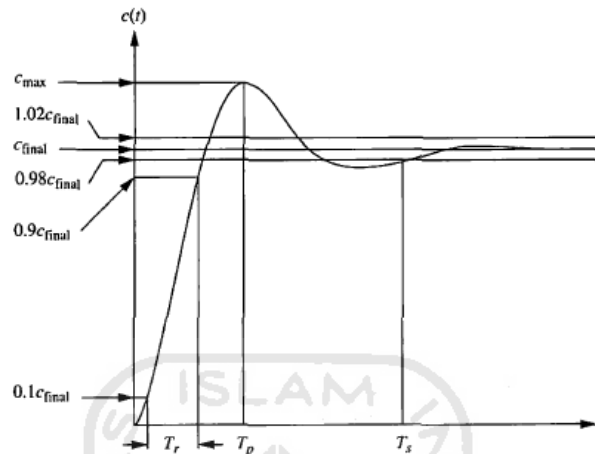
Parameter yang terkait dengan respon sistem adalah sebagai berikut:

- Rise time* ( $T_r$ ), waktu yang dibutuhkan respon dari 0.1 sampai 0.9 nilai akhir.
- Peak time* ( $T_p$ ), waktu yang dibutuhkan untuk mencapai *peak* pertama atau *peak maximum*.
- Percent overshoot* (%OS), jumlah gelombang yang melakukan *overshoot* terhadap *steady state* atau nilai *peak time* dan dinyatakan dalam presentase terhadap nilai *steady state*.

---

<sup>3</sup>Satuan untuk konstanta elektrik adalah  $K_t = Nm/A$  (Newton meter / ampere) dan  $K_b = Vs/rad$  (volt second / radian)

- d. *Settling time* ( $T_s$ ), waktu yang dibutuhkan untuk osilasi teredam (*damped*) transien untuk tetap berada  $\pm 2\%$  pada nilai *steady state*.
- e. *Steady state error* (sse), perbedaan input dan output setelah output memasuki kondisi *steady state*.
- f. *Steady state*, kondisi output pada rentang  $\pm 2\%$  dari input.

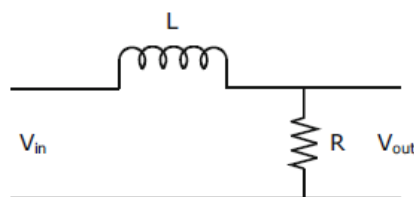


Gambar 2.14 Gambar respon sistem

Perhatikan bahwa definisi *settling time* dan *rise time* pada dasarnya sama dengan definisi untuk respon orde pertama. Semua definisi juga berlaku untuk sistem lebih dari orde 2, meskipun parameter tersebut tidak ditemukan kecuali jika respon dari orde yang lebih tinggi dapat diperkirakan sebagai orde 2.

### 2.2.9 RL Low-Pass Filter

Menurut (Izadian, 2019) untuk mengirimkan sinyal frekuensi rendah ke output filter, induktor dapat digunakan untuk menjembatani input ke port output. Parameter elemen rangkaian diatur untuk mendapatkan frekuensi *cutoff* yang diinginkan di luar frekuensi *cutoff*, sinyal frekuensi yang lebih tinggi menghasilkan impedansi tinggi dan secara otomatis terfilter. Gambar 2.15 adalah rangkaian RL untuk membentuk *filter low-pass*.



Gambar 2.15 Rangkaian RL untuk membentuk *filter low-pass*.

Induktor melewati frekuensi rendah dan jika terhubung antara sumber dan output, mereka memberikan jalur melewati frekuensi rendah ke output, karenanya membentuk *low pass-filter*. Kombinasi nilai resistansi dan induktansi menentukan frekuensi *cutoff*.

### **Mengidentifikasi frekuensi *cutoff* dari rasio *output* terhadap *input*.**

Pada frekuensi *cutoff*, amplitudo fungsi transfer menjadi 70.7% dari maksimal atau  $\frac{1}{\sqrt{2}}H_{max}$ .  $H(j\omega)$  adalah fungsi transfer filter atau sederhananya rasio dari tegangan output terhadap input tegangan.

$$|H(j\omega)|_{\omega=\omega_c} = \frac{1}{\sqrt{2}}H_{max} \quad (2.33)$$

Output maksimal yang dapat dicapai 1. Oleh karena itu,

$$|H(j\omega)|_{\omega=\omega_c} = \frac{V_o(j\omega)}{V_{in}(j\omega)} = \left| \frac{\frac{R}{L}}{j\omega + \frac{R}{L}} \right| = \frac{1}{\sqrt{2}} \quad (2.34)$$

$$\frac{\frac{R}{L}}{\sqrt{\omega_c^2 + \left(\frac{R}{L}\right)^2}} = \frac{1}{\sqrt{2}}$$

untuk mencari  $\omega_c$ ,

$$\frac{\left(\frac{R}{L}\right)^2}{\omega_c^2 + \left(\frac{R}{L}\right)^2} = \frac{1}{2}$$

$$\omega_c = \pm \frac{R}{L}$$

frekuensi positif dapat diterima. Oleh karena itu,

$$\omega_c = \frac{R}{L}$$

Frekuensi *cutoff* filter orde pertama adalah kebalikan dari waktu konstan. Pada rangkaian RL, waktu konstan adalah  $\tau = \frac{L}{R}$ , oleh karena itu frekuensi *cutoff* adalah

$\omega_c = \frac{1}{\tau}$  atau  $\omega_c = \frac{1}{R} = \frac{R}{L}$ . Fungsi transfer RL *low-pass filter* dapat dituliskan sebagai berikut,

$$\frac{V_0}{V_{in}} = \frac{R}{R + sL} = \frac{\frac{R}{L}}{s + \frac{R}{L}} \quad (2.35)$$

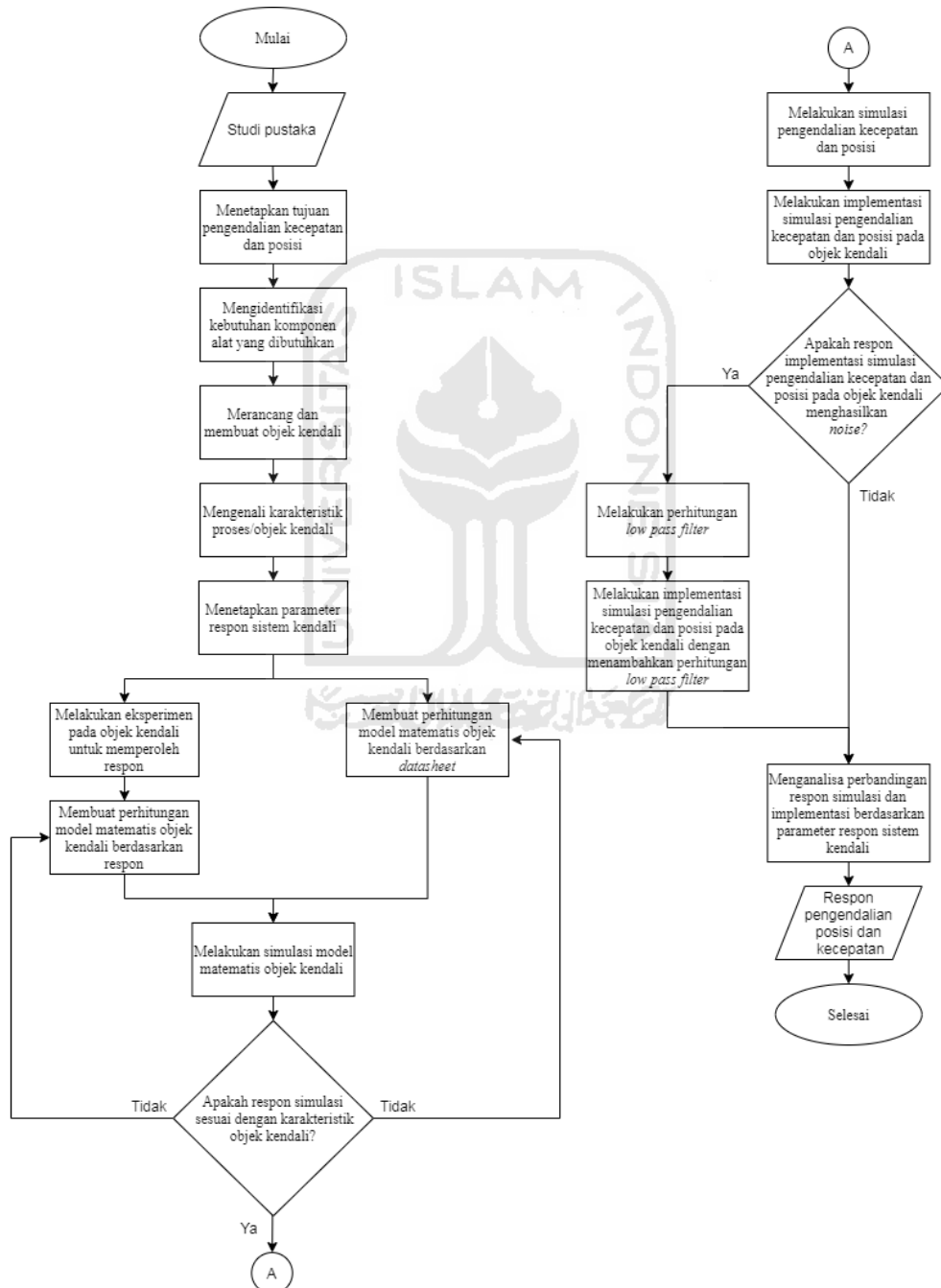


## BAB 3

### METODE PENELITIAN

#### 3.1 Alur Penelitian

Tugas akhir ini dilakukan sesuai dengan tahapan-tahapan yang dapat dilihat pada gambar 3.1.



Gambar 3.1 Diagram alir penelitian

Penelitian dilakukan berdasarkan respon simulasi dan implementasi pada sistem riil. Simulasi berdasarkan model matematis objek kendali dari perhitungan yang variabelnya diperoleh dari *datasheet* dan dari perhitungan berdasarkan eksperimen yang variabelnya diperoleh dari respon objek kendali. Implementasi pada sistem riil dilakukan dengan menghubungkan perangkat keras dan lunak yang akan dilihat respon sistemnya.

## 3.2 Peralatan dan Bahan

Tugas akhir dibuat dengan menggunakan perangkat keras dan lunak.

### 3.2.1 Perangkat Keras

Perangkat keras yang digunakan adalah sebagai berikut :

a. Motor DC

Motor DC(gambar 3.2) yang digunakan adalah Maxon 2322.983-11.225-200 (*datasheet* berada di lamiran 1) dilengkapi dengan gir reduksi sehingga output putaran poros yang dihasilkan lebih rendah dari putaran poros tanpa gir reduksi. Hal ini dapat mempermudah dalam pengendalian kecepatan dan posisi karena tidak membutuhkan putaran dengan rpm tinggi.



Gambar 3.2 Maxon 2322.983-11.225-200

b. *Motor Driver*(*Motor Shield L298*)

*Motor shield* L298(gambar 3.3) digunakan untuk mengendalikan arah dan kecepatan putaran motor DC secara presisi. Di pasaran modul ini mudah ditemukan dan praktis karena pin I/O sudah tersusun rapi.



Gambar 3.3 Motor shield L298

c. Rotary encoder

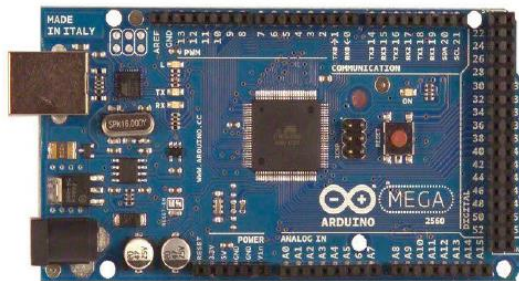
Rotary encoder (gambar 3.4) yang digunakan adalah LPD3806-400BM-G5-24C dengan kapasitas 400 pulsa/revolusi. Datasheet berada di lampiran 3.



Gambar 3.4 Rotary encoder

d. Arduino

Arduino (gambar 3.4) yang digunakan adalah arduino mega 2560.



Gambar 3.4 Arduino mega 2560

e. Catu daya

Catu daya (gambar 3.5) yang digunakan mempunyai spesifikasi 24 volt 2 ampere.



Gambar 3.5 Catu daya 24V 2A

f. Kabel *jumper*

Kabel *jumper* (gambar 3.6) digunakan untuk menghubungkan antara perangkat keras satu dengan yang lainnya.



Gambar 3.6 Kabel *jumper*

g. *Pulley dan belt*

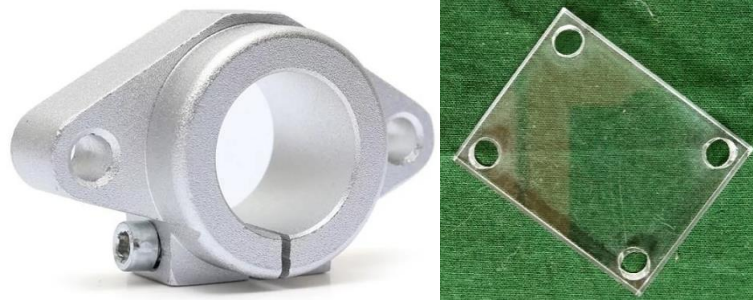
*Pulley* yang digunakan adalah *timing pulley* GT2 16 *teeth* 4 mm *boredan* *timing pulley* GT2 16 *teeth* 6 mm *bore*. *Belt* yang digunakan adalah *timing belt* GT2 65 *teeth* 130 mm *close loop*. *Pulley* dan *belt* dapat dilihat pada gambar 3.7.



Gambar 3.7 *Pulley* dan *belt*

h. *Bracket motor DC dan rotary encoder*

*Bracket motor DC* (gambar 3.8) terbuat dari alumunium dan *bracket rotary encoder* terbuat dari akrilik.



Gambar 3.8 Bracket motor DC dan *rotary encoder*

i. Lengan penunjuk

Lengan penunjuk (gambar 3.9) menggunakan material akrilik.



Gambar 3.9 Lengan penunjuk

j. *Base*

*Base* (gambar 3.10) menggunakan material besi.



Gambar 3.10 *Base*

### 3.2.2 Perangkat Lunak

Perangkat lunak yang digunakan sebagai berikut

a. Scilab

Scilab (gambar 3.11) yang digunakan versi 5.5.2 dengan *toolbox* arduino versi 1.4.



Gambar 3.11 Software Scilab

b. Arduino IDE (*Integrated Development Enviroment*)

Arduinio IDE (gambar 3.12) yang digunakan versi 1.01.



Gambar 3.12 Arduino IDE

### 3.3 Perancangan

Proses perancangan pengendalian posisi dan kecepatan sudut dijelaskan pada bagian ini yang dibagi menjadi perancangan perangkat keras dan lunak.

#### 3.3.1 Perancangan Perangkat Keras

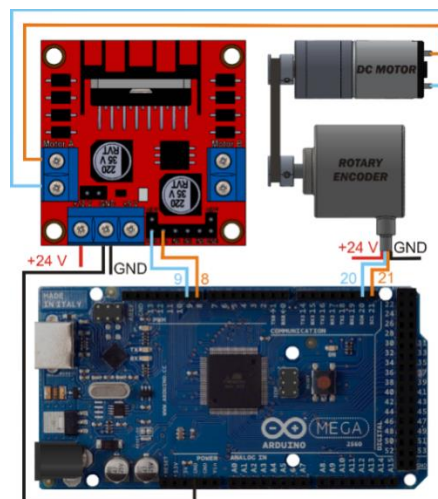
Berdasarkan referensi penulis mencoba merancang kembali *rotary servo base unit* dengan biaya yang lebih terjangkau dan memiliki fungsi yang hampir sama yaitu mengendalikan posisi atau kecepatan. Terdapat perubahan susunan perangkat keras yaitu pada motor DC dan *rotaryencoder*, pada referensi (gambar

3.13) motor DC dan enkoder terhubung pada satu poros namun komponen tersebut sulit didapatkan oleh penulis. Komponen yang tersedia motor DC dan enkoder terpisah sehingga untuk menghubungkan kedua komponen tersebut digunakan *pulley* dan *belt*. Pertimbangan penggunaan *pulley* dan *belt* adalah pembacaan enkodersesuai dengan pergerakan motor DC karena menggunakan 2 buah *pulley* yang berdiameter sama. Penggunaan *tachometer* untuk mengukur kecepatan rotasi poros motor DC dan *potentiometer* untuk mengukur posisi beban poros diabaikan karena dapat digantikan oleh *rotaryencoder*.



Gambar 3.13 Rotary servo base unit

Perancangan komponen elektrik antara motor DC, *motor shield* L298, *rotary encoder*, arduino mega 2560, dan catu daya 24V 2A dihubungkan oleh kabel *jumper* yang dapat dilihat pada gambar 3.14.



Gambar 3.14 Skema rangkaian elektrik

Motor DC dan *rotaryencoder* beroperasi pada tegangan 24 Volt yang didapatkan dari catu daya. Arduino mega 2560 mendapatkan daya dari pc sekitar 5 Volt. Arduino mengirimkan perintah kendali posisi atau kecepatan ke *motor shield* L298 yang diteruskan ke motor DC melalui pin 8 dan 9. Perputaran *rotaryencoder* dengan motor DC yang terhubung oleh *pulley* dan *belt* akan dibaca kembali di Scilab melalui pin 20 dan 21. Gambar 3.15 menunjukkan komponen yang sudah tersusun pada *base*. Bagian bawah dari arduino mega 2560, catu daya 24V 2A, dan *motor shield* L298 diberi isolator agar tidak menimbulkan arus pendek listrik karena *base* terbuat dari logam yang bersifat konduktor.

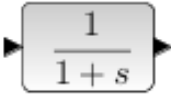
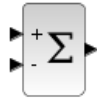
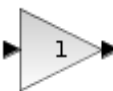


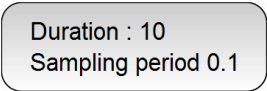
Gambar 3.15 Hasil rangkaian perangkat keras

### 3.3.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak dilakukan untuk mengetahui respon dari sistem yang diwakili oleh skema blok di Xcos Scilab. Beberapa blok yang digunakan dalam perancangan perangkat lunak pada Xcos Scilab dapat dilihat pada tabel 3.1:

| Ikon blok                                                                           | Nama blok     | Diskripsi                                                                                         |
|-------------------------------------------------------------------------------------|---------------|---------------------------------------------------------------------------------------------------|
|  | STEP_FUNCTION | Blok ini mengeluarkan sinyal langkah antara dua level nilai awal dan akhir yang dapat ditentukan. |

|                                                                                     |            |                                                                                                                                                      |
|-------------------------------------------------------------------------------------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | CLR        | Blok ini merealisasikan sistem linear SISO yang diwakili oleh fungsi transfer.                                                                       |
|    | CSCOPE     | Blok ini digunakan untuk menampilkan respon sistem simulasi.                                                                                         |
|    | CLOCK_C    | Blok ini digunakan untuk mengatur besar periode dan parameter waktu inisialisasi.                                                                    |
|    | SUMMATION  | Blok ini melakukan penambahan atau pengurangan input skalar, vektor, atau matriks.                                                                   |
|   | MUX        | Blok ini digunakan untuk menggabungkan beberapa input menjadi output tunggal                                                                         |
|  | INTEGRAL_m | Blok melakukan proses integral dari input yang diberikan.                                                                                            |
|  | DERIV      | Blok melakukan proses turunan dari input yang diberikan.                                                                                             |
|  | GAINBLK_f  | Blok ini menghitung produk dari matriks kuadrat A oleh vektor U dimana jumlah baris A sama dengan jumlah baris U.                                    |
|  | SATURATION | Blok ini digunakan untuk membatasi sinyal. Ketika sinyal input tidak berada di kisaran yang ditentukan maka sinyal akan terpotong ke atas atau bawah |

|                                                                                    |               |                                                                                             |
|------------------------------------------------------------------------------------|---------------|---------------------------------------------------------------------------------------------|
|   | TIME_SAMPLE   | Blok ini menentukan durasi komunikasi dengan arduino dan durasi langkah pengambilan sampel. |
|   | ARDUINO_SETUP | Blok ini digunakan untuk mengatur port komunikasi antara arduino dengan Xcos.               |
|   | ENCODER_SB    | Blok ini digunakan untuk mengoperasikan enkoder.                                            |
|   | DCMOTOR_SB    | Blok ini digunakan untuk mengoperasikan motor DC melalui <i>motor driver</i> .              |
|  | ARDUINO_SCOPE | Blok ini digunakan untuk menampilkan respon sistem simulasi dengan arduino.                 |

Tabel 3.1 Tabel penjelasan blok Xcos Scilab

## BAB 4

### HASIL DAN PEMBAHASAN

#### 4.1 Hasil Perancangan

Hasil perancangan terdiri dari perhitungan fungsi transfer (eksperimen dan *datasheet*), hasil perancangan perangkat keras, dan hasil perancangan perangkat lunak.

##### 4.1.1 Perhitungan Fungsi Transfer

Perhitungan fungsi transfer dari eksperimen menggunakan persamaan (2.18), dimana  $k_m$  diperoleh dari perhitungan persamaan (2.20)

$$k_m = \frac{w(\infty)}{a}$$

$$k_m = \frac{51.81}{24}$$

$$k_m = 2.15$$

dan  $\tau_m$  diperoleh dari perhitungan persamaan (2.22)

$$\tau_m = \frac{-t_0}{\ln\left(1 - \frac{w(t)}{w(\infty)}\right)}$$

$$\tau_m = \frac{-0.03}{\ln\left(1 - \frac{38.01}{51.81}\right)}$$

$$\tau_m = 0.02$$

sehingga fungsi transfernya adalah

$$\frac{W(s)}{U(s)} = \frac{k_m}{\tau_m s + 1}$$

$$\frac{W(s)}{U(s)} = \frac{2.15}{0.02s + 1}$$

$$\frac{W(s)}{U(s)} = \frac{107.5}{s + 50}$$

Perhitungan fungsi transfer dari *datasheet* menggunakan persamaan (2.32)

dimana,

$$K_t = \text{motor torque constant} = 30.7 \text{ mNm/A} = 30.7 \times 10^{-3} \text{ Nm/A}$$

$$R_a = \text{armature resistance} = 21.6 \Omega$$

$$J_m = \text{armature inertia} = 5.68 \text{ gcm}^2 = 5.68 \times 10^{-7} \text{ kgm}^2$$

$$D_m = \frac{1}{\Delta n / \Delta M} = \text{viscous damping} = \frac{1}{219 \text{ rpm/mNm}} = \frac{10^{-3}}{219 \times 2 \times 3.14 / 60} \frac{\text{Nm}}{\text{rad/s}} = 4.36 \times 10^{-5} \frac{\text{Nm}}{\text{rad/s}}$$

$$K_b = \text{back emf}^4 = \frac{1}{\text{speed constant}} = \frac{1}{311 \text{ rpm/V}} = \frac{1}{311 \times 2 \times 3.14 / 60} \frac{\text{V}}{\text{rad/s}} = 0.03 \frac{\text{V}}{\text{rad/s}}$$

$$\frac{\theta_m(s)}{E_a(s)} = \frac{K_t / R_a J_m}{s[s + \frac{1}{J_m}(D_m + \frac{K_t K_b}{R_a})]}$$

untuk memperoleh fungsi transfer orde 1 maka diturunkan sehingga menjadi,

$$\begin{aligned} \frac{\theta_m(s)}{E_a(s)} &= \frac{K_t / R_a J_m}{s + \frac{1}{J_m}(D_m + \frac{K_t K_b}{R_a})} \\ &= \frac{\frac{30.7 \times 10^{-3}}{21.6 \times 5.68 \times 10^{-7}}}{s + \frac{1}{5.68 \times 10^{-7}}(4.36 \times 10^{-5} + \frac{30.7 \times 10^{-3} \times 0.03}{21.6})} \\ &= \frac{2502.28}{s + \frac{1}{5.68 \times 10^{-7}}(4.36 \times 10^{-5} + 4.26 \times 10^{-5})} \\ &= \frac{2516.39}{s + 151.76} \end{aligned}$$

Reduksi gir dapat dihitung seperti berikut,

$$v = \frac{s}{t}$$

$$v = \frac{3170.68 \text{ rad}}{60 \text{ s}}$$

$$v = 52.84 \frac{\text{rad}}{\text{s}}$$

$$v = 52.84 \times \frac{60}{2 \times 3.14}$$

$$v = 504.84 \text{ rpm}$$

---

<sup>4</sup> Lihat lampiran 2 pada *line* 15

Reduksi

$$\frac{7190 \text{ rpm}}{504.84 \text{ rpm}} = 14.24$$

Kemudian fungsi transfer tersebut dibagi dengan reduksi gir dan diperoleh fungsi

$$\text{transfer} \frac{\theta_m(s)}{E_a(s)} = \frac{176.71}{s + 151.76}$$

#### 4.1.2 Perhitungan RL *Low-Pass Filter*

Perhitungan RL *low-pass filter* menggunakan persamaan (2.35) seperti berikut ini,

$$\begin{aligned} \frac{V_0}{V_{in}} &= \frac{R}{R + sL} \\ &= \frac{\frac{R}{L}}{s + \frac{R}{L}} \\ &= \frac{\frac{21.6}{1.97 \times 10^{-3}}}{s + \frac{21.6}{1.97 \times 10^{-3}}} \\ &= \frac{10964.46}{s + 10964.46} \end{aligned}$$

RL *low-pass filter* digunakan untuk meminimalisir atau menghilangkan *noise* (gangguan) yang terjadi pada respon objek kendali. Jika terjadi *noise* maka perhitungan tersebut ditambahkan pada akhir (sebelum blok *ARDUINO\_SCOPE*) skema blok X-cos di blok CLR

#### 4.1.3 Hasil Perancangan Perangkat Keras

Gambar 4.1 adalah hasil perancangan perangkat keras yang digunakan. *Base* dicat menggunakan cat berwarna hitam dengan list warna merah dan bagian atas menggunakan kertas stiker yang sudah memiliki desain sedemikian rupa sehingga memudahkan dalam pembacaan sudut.

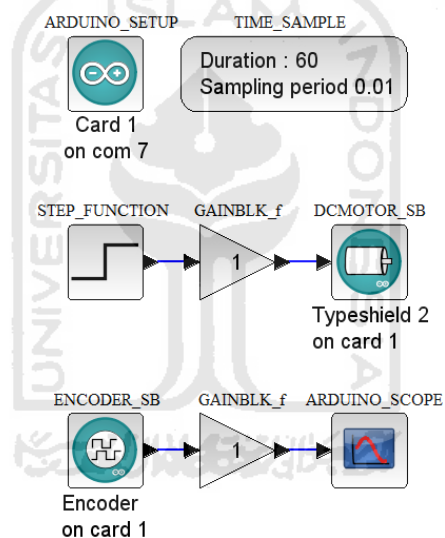


Gambar 4.1 Hasil perancangan perangkat keras

#### 4.1.4 Hasil Perancangan Perangkat Lunak

Terdapat 8 skema blok yang digunakan pada perancangan perangkat lunak seperti berikut :

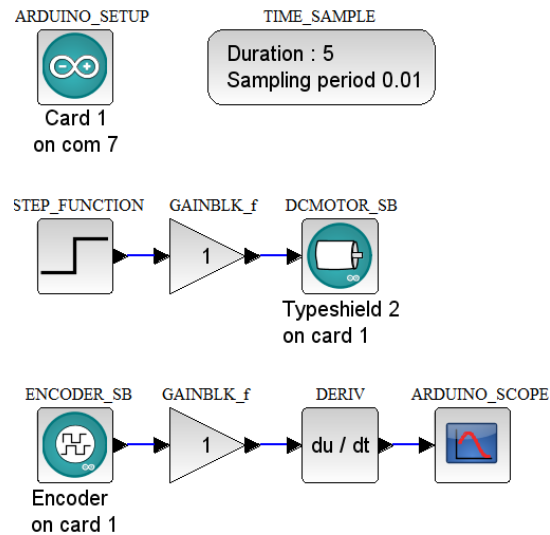
- a. Skema blok pengukuran gir reduksi



Gambar 4.2 Skema blok pengukuran gir reduksi

Pada gambar 4.2, blok *STEP\_FUNCTION* merupakan blok untuk memasukkan nilai berupa tegangan (volt) yang selanjutnya akan diproses menuju blok *GAINBLK\_f* yang berisi nilai maksimal pwm untuk memutar motor DC pada blok *DCMOTOR\_SB*. Blok *ENCODER\_SB* membaca putaran poros motor DC melalui *rotary* enkoderyang kemudian menuju blok *GAINBLK\_f*, blok tersebut berisi perbandingan antara sudut satu putaran dengan jumlah pulsa perangkat keras enkoder(konversi pulsa ke radian). Blok *ARDUINO\_SCOPE* akan menampilkan respon posisi.

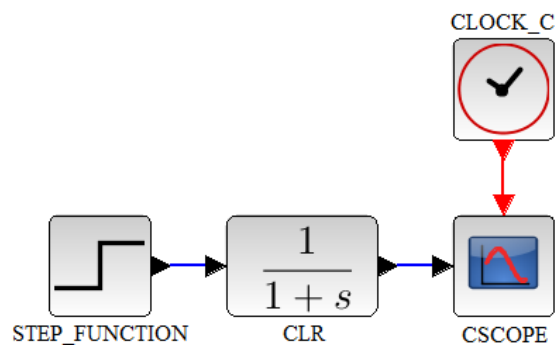
b. Skema blok pengukuran fungsi transfer eksperimen



Gambar 4.3 Skema blok pengukuran fungsi transfer eksperimen

Pada gambar 4.3, blok *STEP\_FUNCTION* merupakan blok untuk memasukkan nilai berupa tegangan (volt) yang selanjutnya akan diproses menuju blok *GAINBLK\_f* yang berisi nilai maksimal pwm untuk memutar motor DC pada blok *DCMOTOR\_SB*. Blok *ENCODER\_SB* membaca putaran poros motor DC melalui *rotary encoder* yang kemudian menuju blok *GAINBLK\_f*, blok tersebut berisi perbandingan antara sudut satu putaran dengan jumlah pulsa perangkat keras enkoder (konversi pulsa ke radian). Blok *DERIV* akan menurunkan dari posisi (radian) menjadi kecepatan sudut (radian/detik) yang responnya akan ditampilkan pada blok *ARDUINO\_SCOPE*.

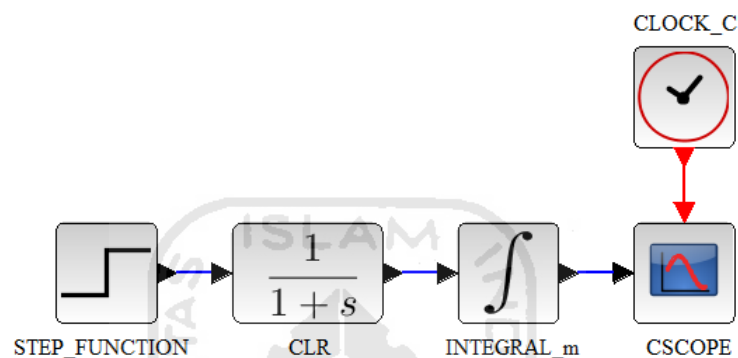
c. Skema blok fungsi transfer orde 1 *open loop*



Gambar 4.4 Skema blok fungsi transfer orde 1 *open loop*

Pada gambar 4.4, blok *STEP\_FUNCTION* merupakan blok untuk memasukkan nilai berupa tegangan (volt) yang selanjutnya akan diproses menuju blok CLR yang berisi fungsi transfer dari *datasheet* atau eksperimen. Respon dari fungsi transfer diproses di blok CSCOPE untuk mengetahui responnya.

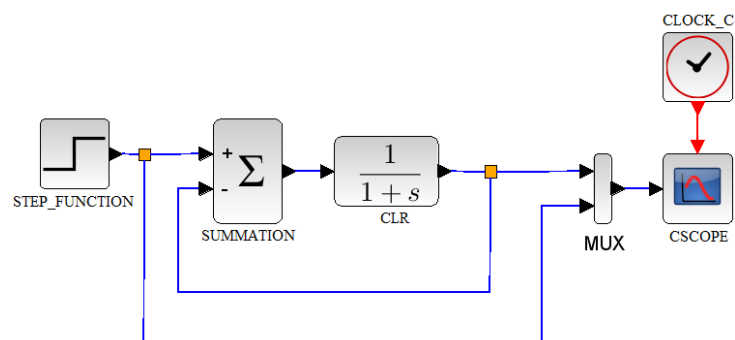
d. Skema blok fungsi transfer orde 2 *open loop*



Gambar 4.5 Skema blok fungsi transfer orde 2 *open loop*

Pada gambar 4.5, blok *STEP\_FUNCTION* merupakan blok untuk memasukkan nilai berupa tegangan (volt) yang selanjutnya akan diproses menuju blok CLR yang berisi fungsi transfer dari *datasheet* atau eksperimen. Fungsi transfer diintegral agar menjadi orde 2 dengan blok *INTEGRAL<sub>m</sub>*. Respon dari fungsi transfer orde 2 diproses di blok CSCOPE untuk mengetahui responnya.

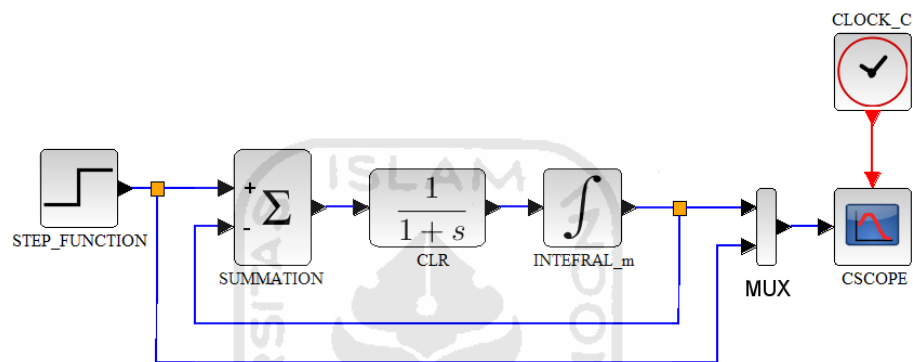
e. Skema blok pengendali kecepatan *close loop*



Gambar 4.6 Skema blok pengendali kecepatan *close loop*

Pada gambar 4.6, blok *STEP\_FUNCTION* merupakan blok untuk memasukkan nilai berupa kecepatan sudut (rpm) yang selanjutnya akan diproses menuju blok CLR yang berisi fungsi transfer dari *datasheet* atau eksperimen. Proses tersebut mengalami perulangan (looping). Blok *MUX* digunakan untuk membandingkan nilai masukan kecepatan sudut (rpm) dengan respon *close loop*.

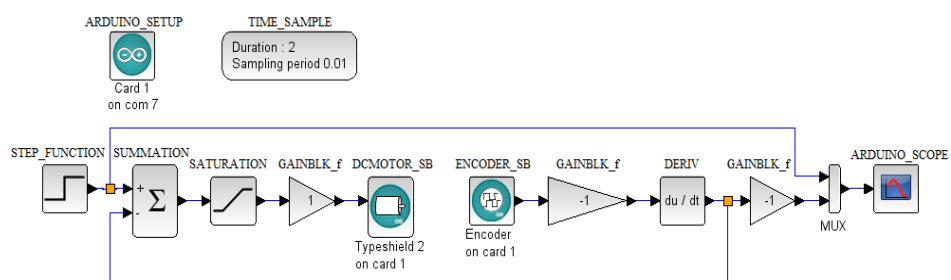
f. Skema blok pengendali posisi *close loop*



Gambar 4.7 Skema blok pengendali posisi *close loop*

Pada gambar 4.7, blok *STEP\_FUNCTION* merupakan blok untuk memasukkan nilai berupa posisi (derajat) yang selanjutnya akan diproses menuju blok CLR yang berisi fungsi transfer dari *datasheet* atau eksperimen. Proses tersebut mengalami perulangan (looping). Blok *MUX* digunakan untuk membandingkan nilai masukan posisi (derajat) dengan respon *close loop*.

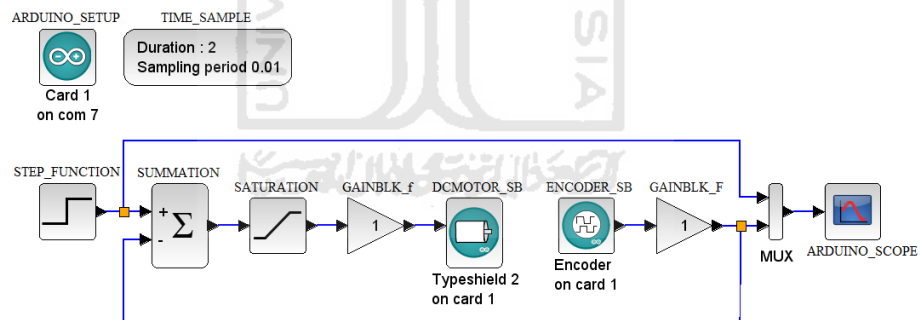
g. Skema blok pengendali kecepatan pada perangkat keras



Gambar 4.8 Skema blok pengendali kecepatan pada perangkat keras

Pada gambar 4.8, blok *STEP\_FUNCTION* merupakan blok untuk memasukkan nilai berupa kecepatan sudut (rpm) yang selanjutnya akan diproses menuju blok *SATURATION* untuk membatasi nilai maksimal dan minimal input, *GAINBLK\_f* yang berisi nilai pwm, dan *DCMOTOR\_SB* yang menggerakkan motor DC sesuai nilai kecepatan sudut (rpm) yang dimasukkan. Blok *ENCODER\_SB* membaca putaran poros motor DC yang kemudian menuju blok *GAINBLK\_f* (minus menunjukkan arah putaran berlawanan arah jarum jam), blok tersebut berisi perbandingan antara jumlah putaran dengan jumlah pulsa perangkat keras enkoder (konversi pulsa ke jumlah putaran). *DERIV* digunakan agar output yang dihasilkan kecepatan. Proses ini mengalami perulangan (*looping*). Blok *GAINBLK\_f* berisi konversi rps ke rpm, tanda minus digunakan untuk membalik tampilan respon (ke bawah menjadi ke atas) di *ARDUINO\_SETUP*. Blok *MUX* digunakan untuk membandingkan nilai masukan kecepatan (rpm) dengan respon perangkat keras *close loop*.

h. Skema blok pengendali posisi pada perangkat keras



Gambar 4.9 Skema blok pengendali posisi pada perangkat keras

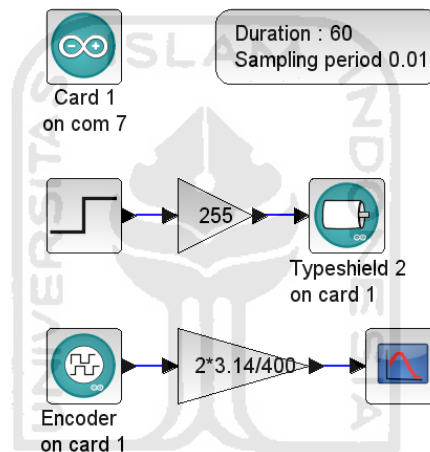
Pada gambar 4.9, blok *STEP\_FUNCTION* merupakan blok untuk memasukkan nilai berupa posisi (derajat) yang selanjutnya akan diproses menuju blok *SATURATION* untuk membatasi nilai maksimal dan minimal input, *GAINBLK\_f* yang berisi nilai pwm, dan *DCMOTOR\_SB* yang menggerakkan motor DC sesuai nilai posisi (derajat) yang dimasukkan. Blok *ENCODER\_SB* membaca putaran poros motor DC yang kemudian menuju blok *GAINBLK\_f*, blok tersebut berisi perbandingan antara jumlah

sudut satu putaran dengan jumlah pulsa perangkat keras encoder (konversi pulsa ke derajat). Proses ini mengalami perulangan (*looping*). Blok *MUX* digunakan untuk membandingkan nilai masukan posisi (derajat) dengan respon perangkat keras *close loop*. Blok *ARDUINO\_SCOPE* digunakan untuk menampilkan respon sistem.

## 4.2 Hasil Pengujian

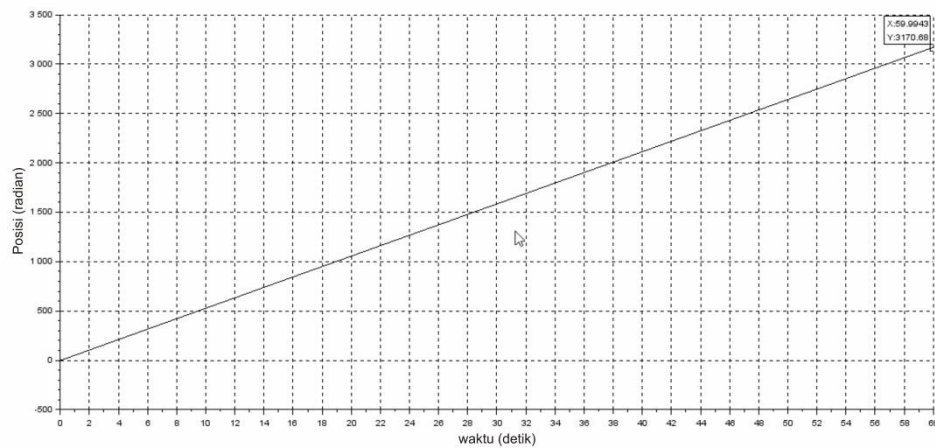
Terdapat 8 hasil perancangan perangkat lunak dan respon skema blok seperti berikut :

- Hasil perancangan perangkat lunak dan respon skema blok pengukuran gir reduksi



Gambar 4.10 Hasil perancangan skema blok pengukuran gir reduksi

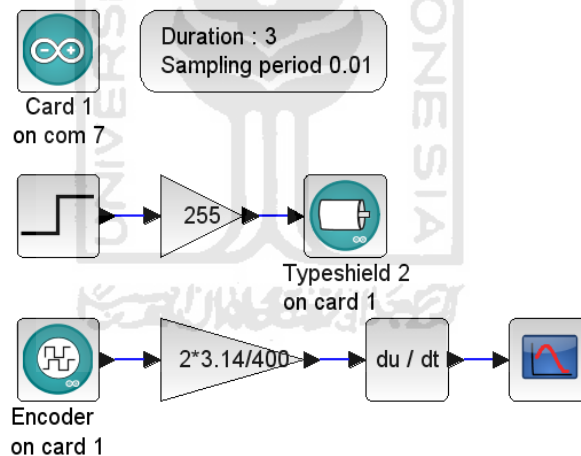
Pada gambar 4.10, blok *STEP\_FUNCTION* memiliki nilai 24. *Type of shield* pada blok *DCMOTOR\_SB* adalah *PMODHB5 or L298 based* dengan pengaturan *direction pin* 8 dan *enable (speed) pin* 9. Pada blok *ENCODER\_SB counting mode* adalah *up chantA* (1) dengan pengaturan *pin channel A* 20 dan *pin direction* 21.



Gambar 4.11 Respon skema blok pengukuran gir reduksi

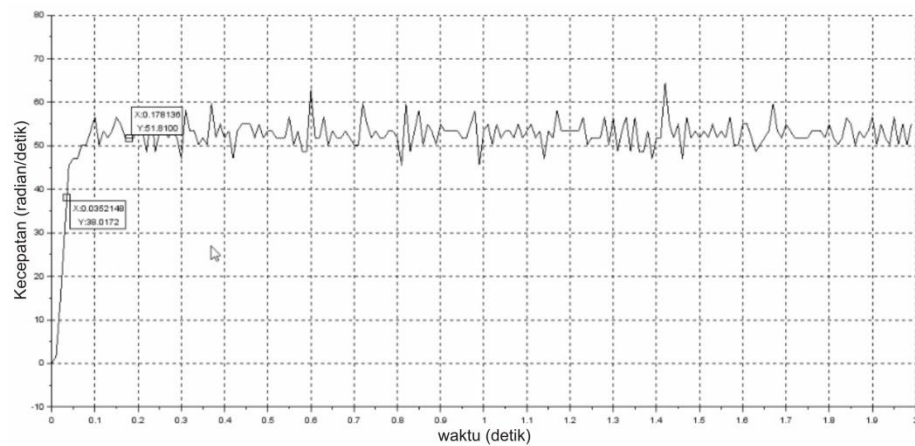
Berdasarkan gambar 4.11 diperoleh posisi 3170.68 radian dalam 60 detik.

- b. Hasil perancangan perangkat lunak dan respon skema blok pengukuran fungsi transfer eksperimen



Gambar 4.12 Hasil perancangan skema blok pengukuran fungsi transfer eksperimen

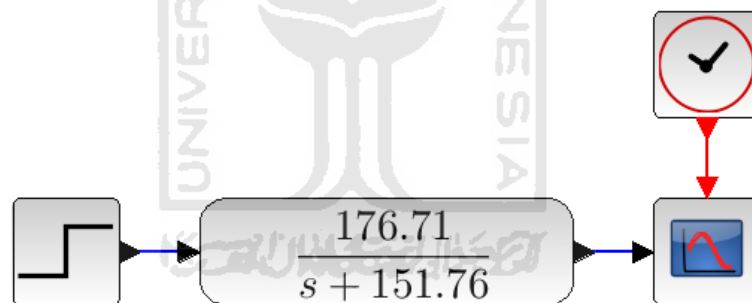
Pada gambar 4.12, blok *STEP\_FUNCTION* memiliki nilai 24. *Type of shield* pada blok *DCMOTOR\_SB* adalah *PMODHB5 or L298 based* dengan pengaturan *direction pin* 8 dan *enable (speed) pin* 9. Pada blok *ENCODER\_SB counting mode* adalah *up chantA (1)* dengan pengaturan *pin channel A* 20 dan *pin direction* 21.



Gambar 4.13 Respon skema blok pengukuran fungsi transfer eksperimen

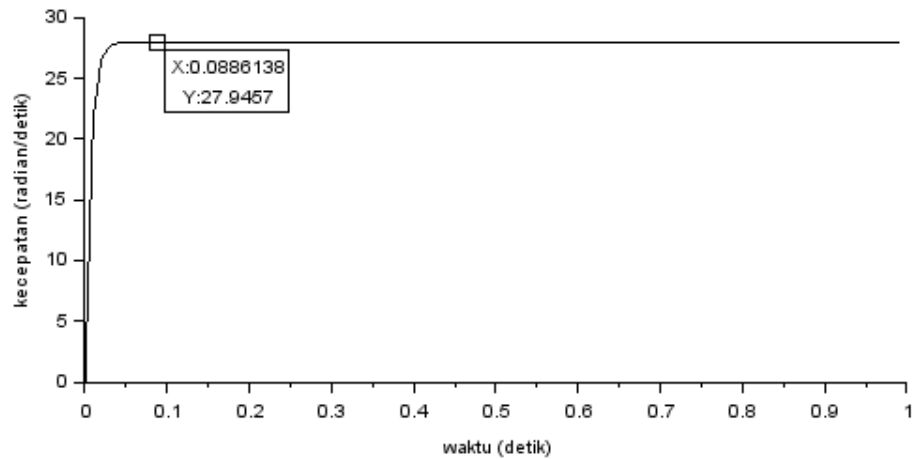
Berdasarkan gambar 4.13 diperoleh  $w(\infty) = 51.81$  radian/detik,  $w(t) = 38.01$  radian/detik, dan  $t_0 = 0.03$  detik.

- c. Hasil perancangan perangkat lunak dan respon skema blok fungsi transfer orde 1 *open loop*

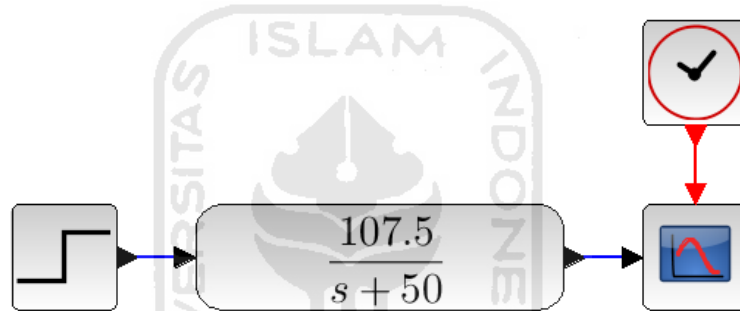


Gambar 4.14 Hasil perancangan skema blok fungsi transfer dari *datasheet* orde 1 *open loop*

Pada gambar 4.14, blok *STEP\_FUNCTION* memiliki nilai 24, blok *CLR* berisi fungsi transfer dari *satasheet*. Blok *CSCOPE* diatur dengan *Y max* 30, *Y min* 0, dan *refresh period* 1. Blok *CLOCK\_c* diatur dengan *period* 0.01 dan *initialisation time* 0. Simulasi dilakukan selama 1 detik.

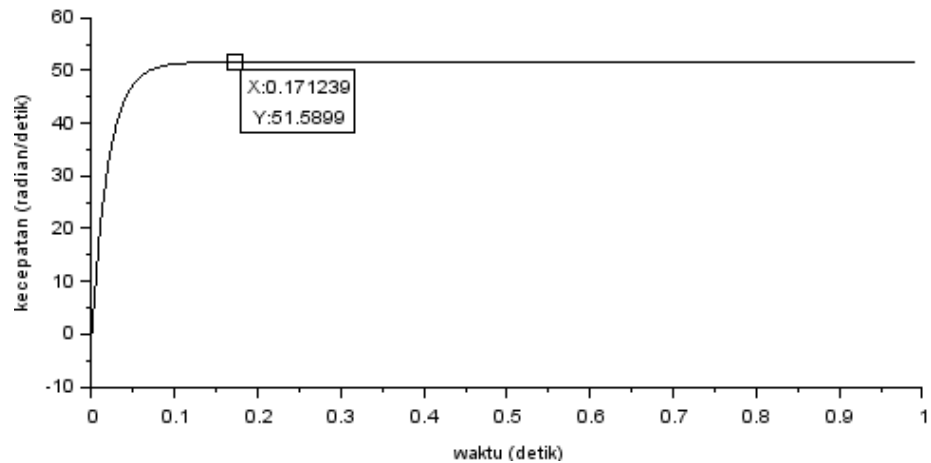


Gambar 4.15 Respon skema blok fungsi transfer datasheet orde 1 *open loop*



Gambar 4.16 Hasil perancangan skema blok fungsi transfer dari eksperimen orde 1 open loop

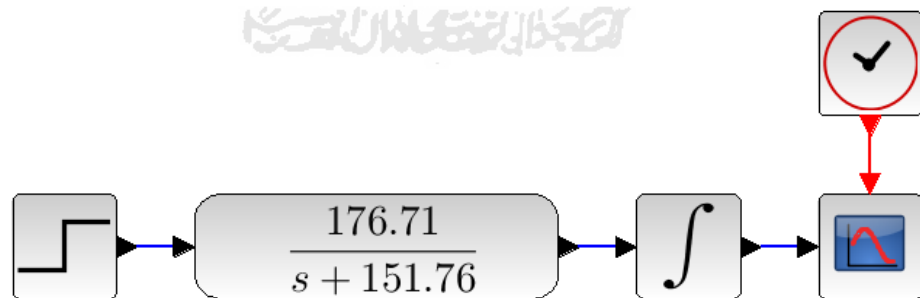
Pada gambar 4.16, blok *STEP\_FUNCTION* memiliki nilai 24, blok *CLF* berisi fungsi transfer dari eksperimen. Blok *CSCOPE* diatur dengan *Y max* 60, *Y min* 0, dan *refresh period* 1. Blok *CLOCK\_c* diatur dengan *period* 0.01 dan *initialisation time* 0. Simulasi dilakukan selama 1 detik.



Gambar 4.17 Respon skema blok fungsi transfer eksperimen orde 1 open loop

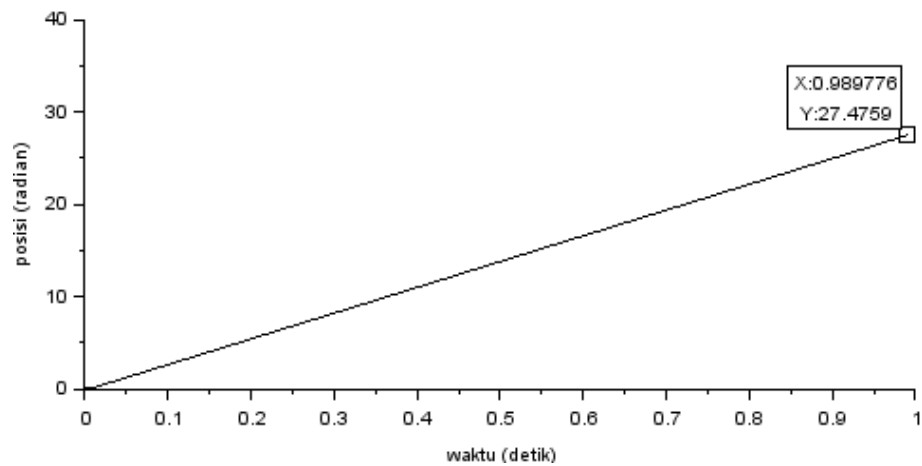
Gambar 4.15 dan 4.17 merupakan respon dari fungsi transfer dan *datasheet* orde 1 atau relasi antara kecepatan (radian/detik) dengan waktu (detik). Kecepatan yang dihasilkan dari fungsi transfer *datasheet* 27.94 radian/detik dan kecepatan yang dihasilkan fungsi transfer eksperimen 51.58 radian/detik.

- d. Hasil perancangan perangkat lunak dan respon skema blok fungsi transfer orde 2 *open loop*

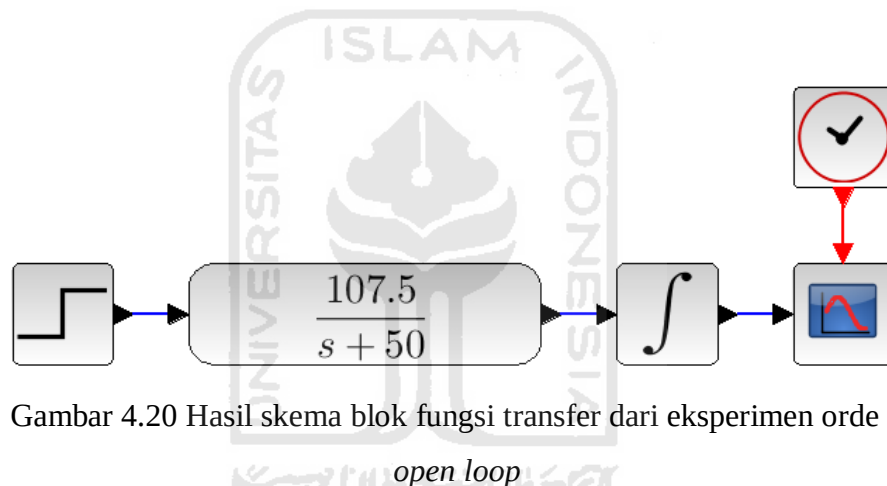


Gambar 4.18 Hasil perancangan skema blok fungsi transfer dari *datasheet* orde 2 *open loop*

Pada gambar 4.18, blok *STEP\_FUNCTION* memiliki nilai 24, blok *CLR* berisi fungsi transfer dari *datasheet*. Blok *CSCOPE* diatur dengan *Y max* 40, *Y min* 0, dan *refresh period* 1. Blok *CLOCK\_c* diatur dengan *period* 0.01 dan *initialisation time* 0. Simulasi dilakukan selama 1 detik.

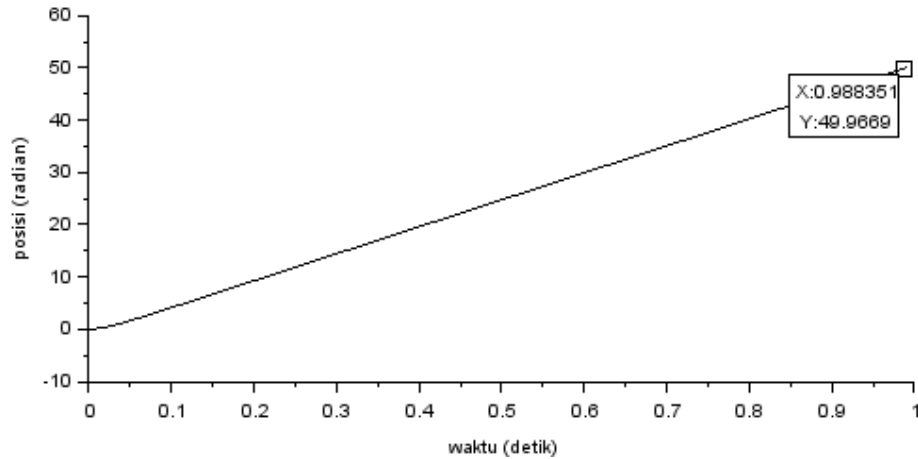


Gambar 4.19 Respon skema blok fungsi transfer *datasheet* orde 2 open loop



Gambar 4.20 Hasil skema blok fungsi transfer dari eksperimen orde 2 open loop

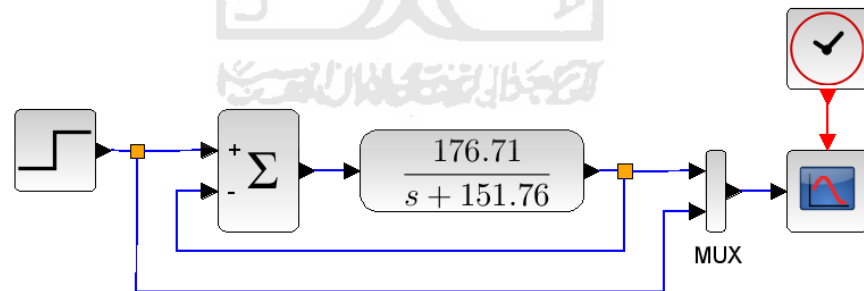
Pada gambar 4.20, blok *STEP\_FUNCTION* memiliki nilai 24, blok *CLR* berisi fungsi transfer dari eksperimen. Blok *CSCOPE* diatur dengan *Y max* 60, *Y min* 0, dan *refresh period* 1. Blok *CLOCK\_c* diatur dengan *period* 0.01 dan *initialisation time* 0. Simulasi dilakukan selama 1 detik.



Gambar 4.21 Respon skema blok fungsi dari eksperimen transfer orde 2 open loop

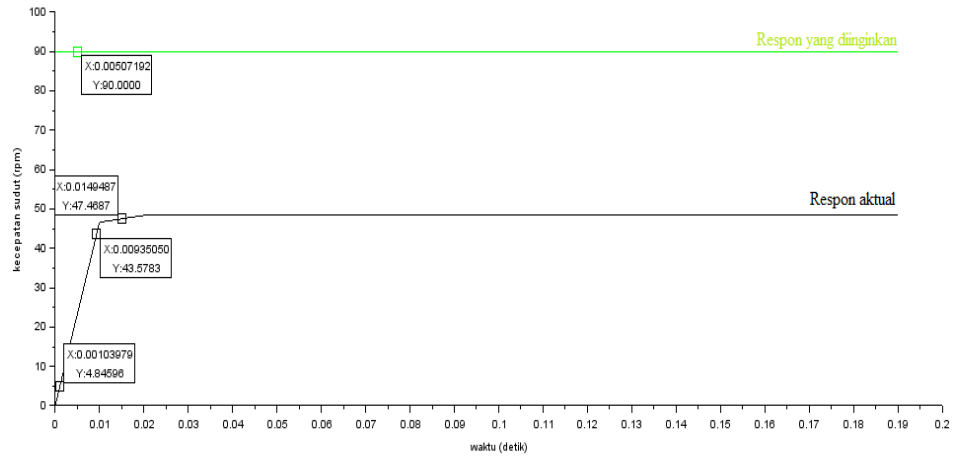
Gambar 4.19 dan 4.21 merupakan respon dari fungsi transfer dan *datasheet* orde 2 atau relasi antara posisi (radian) dengan waktu (detik). Posisi yang dihasilkan dari fungsi transfer *datasheet* dan eksperimen selama 1 detik adalah 27.47 radian dan 49.96 radian.

- e. Hasil perancangan perangkat lunak dan respon skema blok pengendali kecepatan *close loop*



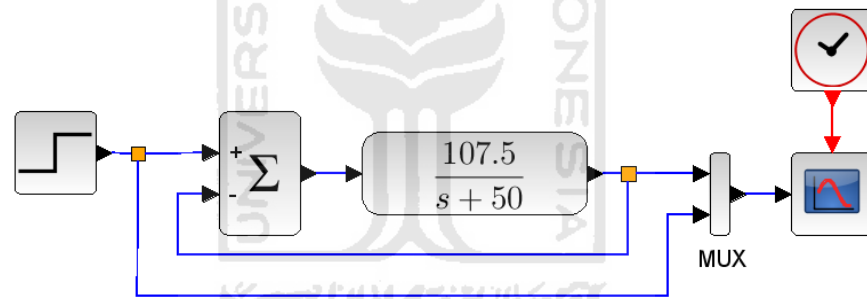
Gambar 4.22 Hasil perancangan skema blok pengendali kecepatan *datasheet close loop*

Pada gambar 4.22, blok *STEP\_FUNCTION* memiliki nilai 90, blok *CLR* berisi fungsi transfer dari *datasheet*. Blok *CSCOPE* diatur dengan *Y max* 95, *Y min* 0, dan *refresh period* 1. Blok *CLOCK\_c* diatur dengan *period* 0.01 dan *initialisation time* 0. Simulasi dilakukan selama 1 detik.



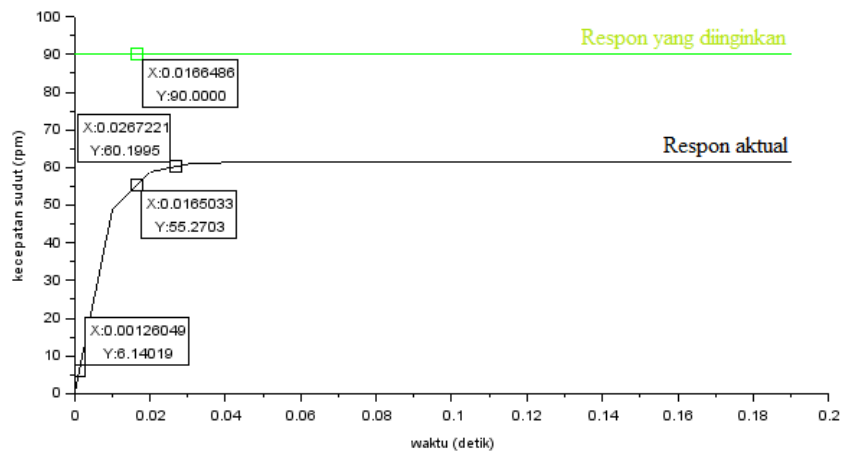
Gambar 4.23 Respon skema blok pengendali kecepatan *datasheet close loop*

Berdasarkan gambar 4.23 diperoleh data  $T_r = 0.0083$  detik,  $T_s = 0.0149$  detik dan  $sse = 42.5313$  rpm.



Gambar 4.24 Hasil perancangan skema blok pengendali kecepatan eksperimen *close loop*

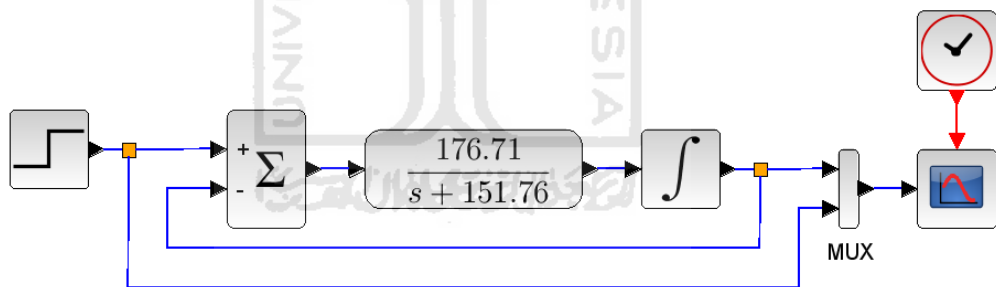
Pada gambar 4.24, blok *STEP\_FUNCTION* memiliki nilai 90, blok *CLR* berisi fungsi transfer dari eksperimen. Blok *CSCOPE* diatur dengan  $Y_{max}$  95,  $Y_{min}$  0, dan *refresh period* 1. Blok *CLOCK\_c* diatur dengan *period* 0.01 dan *initialisation time* 0. Simulasi dilakukan selama 1 detik.



Gambar 4.25 Respon skema blok pengendali kecepatan eksperimen  
*close loop*

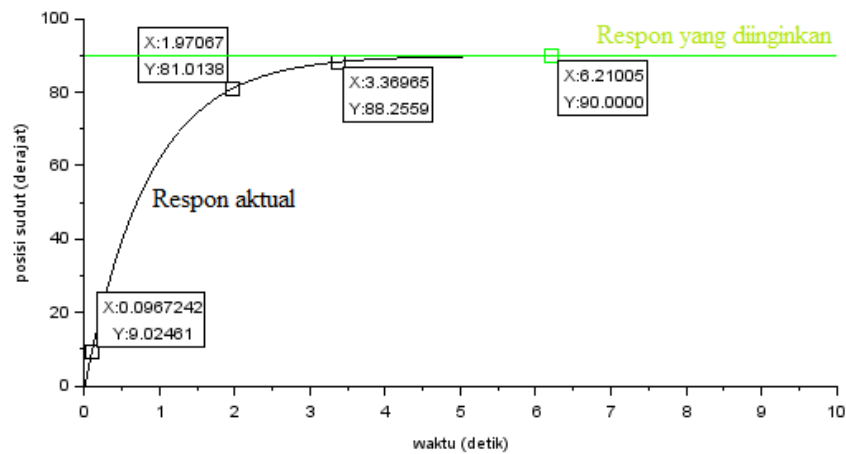
Berdasarkan gambar 4.25 diperoleh data  $T_r = 0.0153$  detik,  $T_s = 0.0267$  detik dan  $sse = 29.8005$  rpm

- f. Hasil perancangan perangkat lunak dan respon skema blok pengendali posisi *close loop*



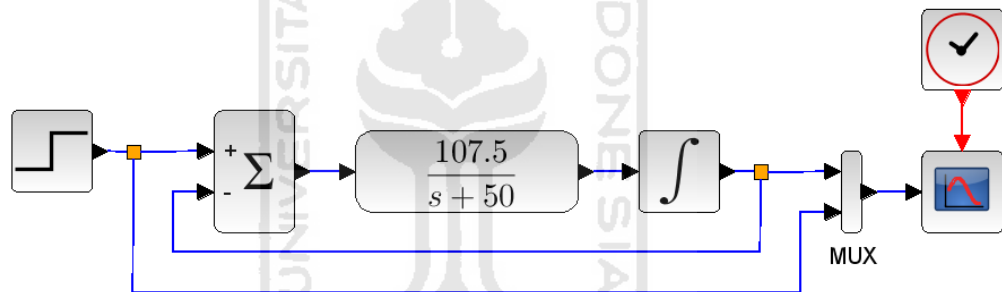
Gambar 4.26 Hasil perancangan skema blok pengendali posisi *datasheet*  
*close loop*

Pada gambar 4.26, blok *STEP\_FUNCTION* memiliki nilai 90, blok *CLR* berisi fungsi transfer dari *datasheet*. Blok *CSCOPE* diatur dengan *Y max* 95, *Y min* 0, dan *refresh period* 10. Blok *CLOCK\_c* diatur dengan *period* 0.01 dan *initialisation time* 0. Simulasi dilakukan selama 10 detik.



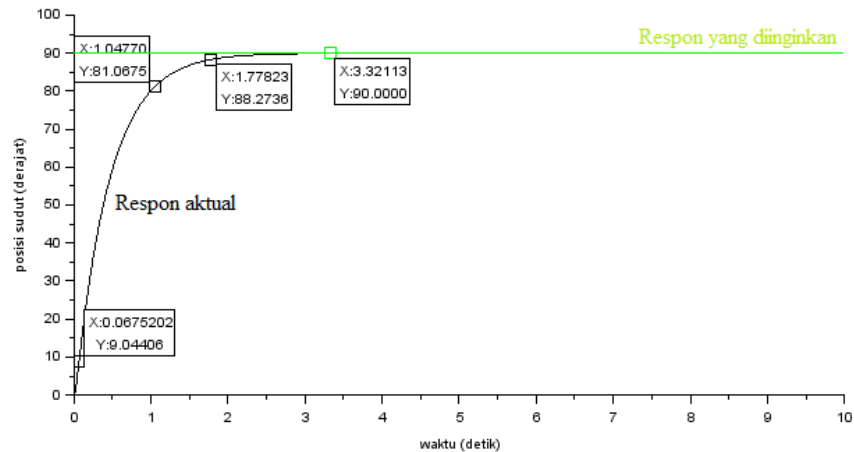
Gambar 4.27 Respon skema blok pengendali posisi *datasheet close loop*

Berdasarkan gambar 4.27 diperoleh data  $T_r = 1.8739$  detik dan  $T_s = 3.3696$  detik.



Gambar 4.28 Hasil perancangan skema blok pengendali posisi eksperimen close loop

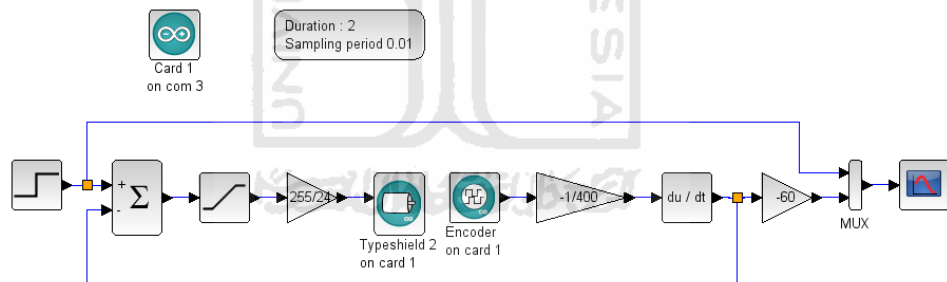
Pada gambar 4.28, blok *STEP\_FUNCTION* memiliki nilai 90, blok *CLR* berisi fungsi transfer dari eksperimen. Blok *CSCOPE* diatur dengan  $Y_{max}$  95,  $Y_{min}$  0, dan *refresh period* 10. Blok *CLOCK\_c* diatur dengan *period* 0.01 dan *initialisation time* 0. Simulasi dilakukan selama 10 detik.



Gambar 4.29 Respon skema blok pengendali posisi eksperimen *close loop*

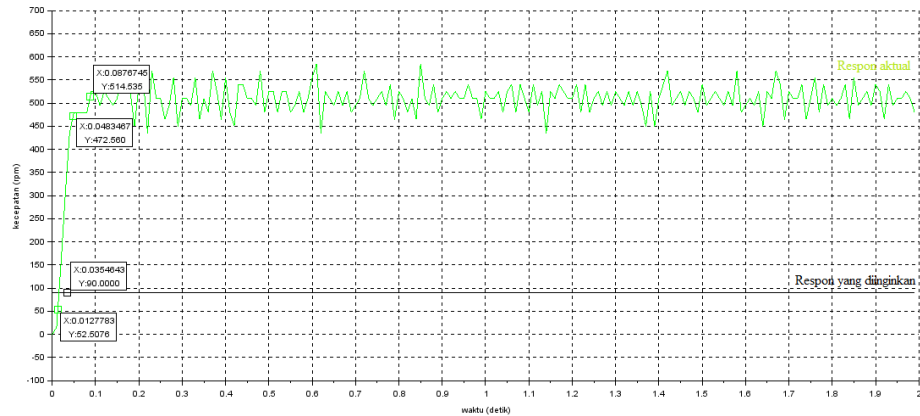
Berdasarkan gambar 4.29 diperoleh data  $T_r = 0.9802$  detik dan  $T_s = 1.7782$  detik.

- g. Hasil perancangan perangkat lunak dan respon skema blok pengendali kecepatan pada perangkat keras



Gambar 4.30 Hasil skema blok pengendali kecepatan pada perangkat keras

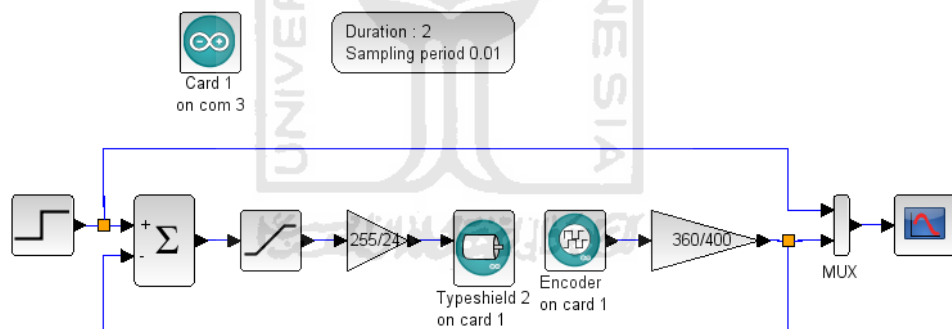
Pada gambar 4.30, blok *STEP\_FUNCTION* memiliki nilai 90, blok *SATURATION* memiliki *upper limit* 24 dan *lower limit* -24. *Type of shield* pada blok *DCMOTOR\_SB* adalah *PMODHB5 or L298 based* dengan pengaturan *direction pin* 8 dan *enable (speed) pin* 9. Pada blok *ENCODER\_SB* *counting mode* adalah *up chantA* (1) dengan pengaturan *pin channel A* 20 dan *pin direction* 21. Simulasi dilakukan selama 2 detik.



Gambar 4.31 Respon skema blok pengendali kecepatan pada perangkat keras

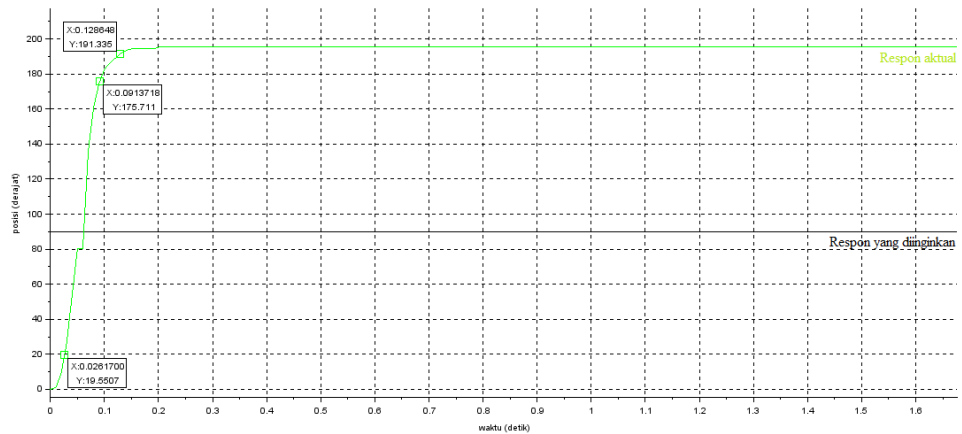
Berdasarkan gambar 4.31 diperoleh data  $T_r = 0.0356$  detik,  $T_s = 0.0876$  detik, dan  $sse = 424.535$  rpm.

- h. Hasil perancangan perangkat lunak dan respon skema blok pengendali posisi pada perangkat keras



Gambar 4.32 Hasil perancangan skema blok pengendali posisi pada perangkat keras

Pada gambar 4.32, blok *STEP\_FUNCTION* memiliki nilai 90, blok *SATURATION* memiliki *upper limit* 24 dan *lower limit* -24. *Type of shield* pada blok *DCMOTOR\_SB* adalah *PMODHB5 or L298 based* dengan pengaturan *direction pin* 8 dan *enable (speed) pin* 9. Pada blok *ENCODER\_SB* *counting mode* adalah *up chantA* (1) dengan pengaturan *pin channel A* 20 dan *pin direction* 21. Simulasi dilakukan selama 2 detik.



Gambar 4.33 Respon skema blok pengendali posisi pada perangkat keras

Berdasarkan gambar 4.33 diperoleh data  $T_r = 0.0652$  detik,  $T_s = 0.1286$  detik, dan  $sse = 101.335$  derajat.

### 4.3 Analisis dan Pembahasan

Berikut ini adalah tabel data respon skema blok hasil perancangan perangkat lunak pengendali kecepatan dan pengendali posisi.

| Parameter     | Respon berdasarkan <i>datasheet</i> | Respon berdasarkan eksperimen | Respon berdasarkan implementasi |
|---------------|-------------------------------------|-------------------------------|---------------------------------|
| $T_r$ (detik) | 0.0083                              | 0.0153                        | 0.0356                          |
| $T_p$ (detik) | -                                   | -                             | -                               |
| %OS (%)       | -                                   | -                             | -                               |
| $T_s$ (detik) | 0.0149                              | 0.0267                        | 0.0876                          |
| Sse (rpm)     | 42.5313                             | 29.8005                       | 424.535                         |

Tabel 4.1 Perbandingan respon pengendalian kecepatan

| Parameter     | Respon berdasarkan <i>datasheet</i> | Respon berdasarkan eksperimen | Respon berdasarkan implementasi |
|---------------|-------------------------------------|-------------------------------|---------------------------------|
| $T_r$ (detik) | 1.8739                              | 0.9802                        | 0.0652                          |
| $T_p$ (detik) | -                                   | -                             | -                               |
| %OS (%)       | -                                   | -                             | -                               |
| $T_s$ (detik) | 3.3696                              | 1.7782                        | 0.1286                          |
| Sse (derajat) | -                                   | -                             | 101.335                         |

Tabel 4.2 Perbandingan respon pengendalian posisi

Berdasarkan tabel 4.1 nilai  $T_r$  terkecil adalah respon berdasarkan *datasheet*, nilai  $T_s$  terkecil adalah respon berdasarkan *datasheet*, dan sse terkecil adalah respon berdasarkan eksperimen. Berdasarkan tabel 4.2 nilai  $T_r$  terkecil adalah respon berdasarkan implementasi, nilai  $T_s$  terkecil adalah respon berdasarkan implementasi, dan nilai sse terkecil adalah respon berdasarkan eksperimen dan *datasheet*. Respon sistem yang baik tidak memiliki selisih respon yang diinginkan dan respon aktual, *percent overshoot*, dan *peak time* atau nilainya kecil mendekati nilai respon yang diinginkan. Nilai besar kecilnya *settling time* dan *rise time* tergantung dari kebutuhan produk. Nilai *settling time* dan *rise time* berbanding lurus, jika nilai *settling time* besar maka nilai *rise time* besar, begitu juga sebaliknya.

Pada pengujian fungsi transfer berdasarkan *datasheet* dan eksperimen menunjukkan hasil atau respon yang berbeda. Perbedaan respon skema blok fungsi transfer berdasarkan *datasheet* dan eksperimen disebabkan karena beberapa kemungkinan seperti pada perhitungan fungsi transfer berdasarkan *datasheet* ada parameter yang belum teridentifikasi, parameter yang tidak sesuai dengan *datasheet* yang diberikan perusahaan motor DC, dan motor DC yang digunakan dalam pembuatan *datasheet* tidak sama persis dengan yang penulis gunakan karena semua motor DC yang diproduksi tidak 100% sama pada tipe dan merk yang sama. Oleh karena itu, fungsi transfer berdasarkan *datasheet* tidak dapat mewakili fungsi transfer dari prototipe.

## BAB 5

### PENUTUP

#### 5.1 Kesimpulan

Berdasarkan hasil penelitian diperoleh kesimpulan sebagai berikut:

1. Pembuatan prototipe sistem kendali posisi dan kecepatan sudut berhasil terealisasi dengan arduino sebagai perangkat komunikasi antara objek kendali dengan PC, material base dari logam, dan lengan penunjuk dari akrilik.
2. Pembuatan model matematika berupa fungsi transfer berhasil dengan fungsi transfer  $\frac{W(s)}{U(s)} = \frac{107.5}{s+50}$ .
3. Sistem kendali posisi dan kecepatan sudut *close loop* berhasil dibuat dengan menghasilkan perbandingan parameter respon kendali berdasarkan *datasheet*, eksperimen, dan implementasi.

#### 5.2 Saran atau Penelitian Selanjutnya

Pada respon skema blok pengukuran fungsi transfer eksperimen tidak menunjukkan hasil yang konsisten. Penulis mencoba menggunakan 2 laptop dan komputer untuk menguji konsistennya respon namun hasilnya menunjukkan tidak konsisten. Penulis juga melakukan simulasi ini dengan Scilab 6.1.0 dan toolbox 1.8 (perangkat lunak dan *toolbox* terbaru) dengan hasil motor DC dapat berputar namun enkoder tidak terbaca. Perangkat lunak Scilab merupakan *open source* sehingga diharapkan kepada pihak Scilab atau seorang yang ahli dalam bidang ini dapat mengatasi *bug* tersebut.

Penelitian yang dilakukan penulis menghasilkan beberapa respon pengendalian posisi atau kecepatan memiliki *steady state error*. *Steady state error* tersebut dapat diminimalkan atau dihilangkan dengan menambahkan kontroler. Kontroler yang digunakan dapat berupa PID, *fuzzy*, *state space*, dan sebagainya. Kontroler juga dapat dimanfaatkan untuk mengatur *overshoot* dan *settling time*.

## DAFTAR PUSTAKA

- Adriansyah, A., & Hidyatama, O. (2013). Rancang Bangun Prototipe Elevator Menggunakan Microcontroller Arduino ATmega 328P. *Jurnal Teknologi Elektro*, 100-112.
- Aulia, F. (2013). Pengenalan Scilab. Dalam F. Aulia, *Scilab (5.4.0)* (hal. 1). Malang: Laboratorium Informatika dan Komputer Teknik Elektro Universitas Brawijaya.
- Chapman, S. (1991). *Electric Machinery Fundamentals Second Edition*. New York: McGraw-Hill.
- Chen, C. T. (1995). Motors. Dalam *Analog & Digital Control System Design* (hal. 70-77). New York: Oxford University Press, Inc.
- Fitriansyah, A., Waluyo, & Syahrial. (2013). Analisis Penalaan Kontroller PID pada Simulasi Kendali Kecepatan Putaran Motor DC. *Jurnal Reka Elkomika*, 360-372.
- Izadian, A. (2019). *Fundamental of Modern Electric Circuit Analysis and Filter Synthesis*. Switzerland: Springer Nature Switzerland AG.
- Junaidi, & Prabowo, Y. D. (2018). Dasar-Dasar Mikrokontroler dan Arduino. Dalam *Project Sistem Kendali Elektronik Berbasis Arduino* (hal. 1-4). Bandar Lampung: Aura.
- Nise, N. S. (2011). Electromechanical System Transfer Function. Dalam *Control Systems Engineering Sixth Edition* (hal. 79-81). California: John Wiley & Sons, INC.
- Ogata, K. (2010). *Modern Control Engineering Fifth Edition*. New Jersey: Pearson Education, Inc.
- Rizqiawan, A. (2009, Juni 12). *Sekilas Rotary Encoder*. Diambil kembali dari Konversi ITB: <https://konversi.wordpress.com/2009/06/12/sekilas-rotary-encoder/>
- Sailan, K., & Kuhnert, K. D. (2013). DC Motor Angular Position Control using PID Controller for the porpuse of controlling the Hydraulic Pump. *International Conference on Control, Enginering & Information Technology (CEIT'13)* (hal. 22-26). Germany: IPCO.

Sitohang, E. P., Mamahit, D. J., & Tulung , N. S. (2018). Rancang Bangun Catu Daya DC Menggunakan Mikrokontroler ATmega 8535. *Jurnal Teknik Elektro dan Komputer Vol.7*, 135-142.

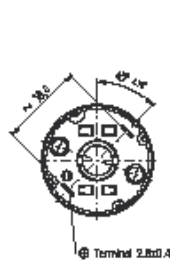


# LAMPIRAN 1

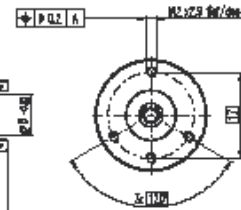
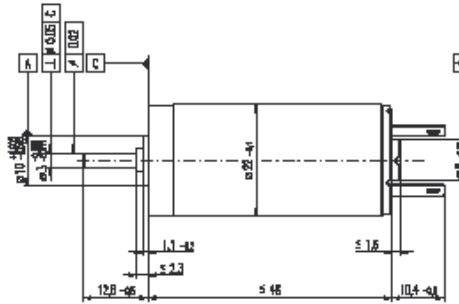
## S 2322 Ø22 mm, Graphite Brushes, 6 Watt

**NRND** See page 12  
Not recommended for New Design

maxon special program



M 1:1



- Stock program
- Standard program
- Special program (on request)

### Part Numbers

2322 ... -11,225-200 (insert winding number)

Winding number

980 981 990 982 983 985 987

### Motor Data

#### Values at nominal voltage

|                                             | V                | 12    | 15    | 18    | 18    | 24    | 36    | 48    |  |
|---------------------------------------------|------------------|-------|-------|-------|-------|-------|-------|-------|--|
| 1 Nominal voltage                           | V                | 12    | 15    | 18    | 18    | 24    | 36    | 48    |  |
| 2 No load speed                             | rpm              | 7010  | 7000  | 7530  | 6690  | 7190  | 7240  | 6370  |  |
| 3 No load current                           | mA               | 42.2  | 33.3  | 36.6  | 25.8  | 21.3  | 14.3  | 8.88  |  |
| 4 Nominal speed                             | rpm              | 4300  | 4310  | 4840  | 3970  | 4400  | 4500  | 3580  |  |
| 5 Nominal torque (max. continuous torque)   | mNm              | 11.8  | 12.2  | 12.2  | 12.3  | 12.4  | 12.3  | 12.4  |  |
| 6 Nominal current (max. continuous current) | A                | 0.807 | 0.654 | 0.583 | 0.520 | 0.420 | 0.279 | 0.184 |  |
| 7 Stall torque                              | mNm              | 32.9  | 33.5  | 35.9  | 31.4  | 34.1  | 33.4  | 28.9  |  |
| 8 Starting current                          | A                | 2.14  | 1.72  | 1.84  | 1.28  | 1.11  | 0.724 | 0.411 |  |
| 9 Max. efficiency                           | %                | 70    | 71    | 72    | 71    | 73    | 73    | 73    |  |
| <b>Characteristics</b>                      |                  |       |       |       |       |       |       |       |  |
| 10 Terminal resistance                      | Ω                | 5.81  | 8.71  | 10.9  | 14.1  | 21.6  | 49.7  | 117   |  |
| 11 Terminal inductance                      | mH               | 0.492 | 0.790 | 0.995 | 1.26  | 1.97  | 4.43  | 10.2  |  |
| 12 Torque constant                          | mNm/A            | 15.4  | 19.4  | 21.8  | 24.6  | 30.7  | 46.1  | 69.9  |  |
| 13 Speed constant                           | rpm/V            | 622   | 491   | 437   | 389   | 311   | 207   | 137   |  |
| 14 Speed/torque gradient                    | rpm/mNm          | 227   | 220   | 219   | 223   | 219   | 224   | 228   |  |
| 15 Mechanical time constant                 | ms               | 13.9  | 13.5  | 13.3  | 13.3  | 13.0  | 12.9  | 12.8  |  |
| 16 Rotor inertia                            | gcm <sup>2</sup> | 5.84  | 5.86  | 5.70  | 5.70  | 5.68  | 5.50  | 5.37  |  |

### Specifications

#### Thermal data

|                                         |              |
|-----------------------------------------|--------------|
| 17 Thermal resistance housing-ambient   | 14.1 K/W     |
| 18 Thermal resistance winding-housing   | 3.6 K/W      |
| 19 Thermal time constant winding        | 0.24 s       |
| 20 Thermal time constant motor          | 640 s        |
| 21 Ambient temperature                  | -20...+100°C |
| 22 Max. permissible winding temperature | +125°C       |

#### Mechanical data (ball bearings)

|                                          |                |
|------------------------------------------|----------------|
| 23 Max. permissible speed                | 9500 rpm       |
| 24 Axial play                            | 0.05 - 0.15 mm |
| 25 Radial play                           | 0.025 mm       |
| 26 Max. axial load (dynamic)             | 2.8 N          |
| 27 Max. force for press fits (static)    | 64 N           |
| 28 Max. radial loading, 5 mm from flange | 14 N           |

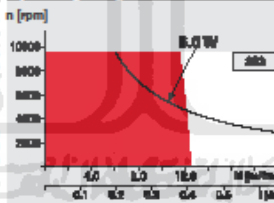
#### Other specifications

- 29 Number of pole pairs
- 30 Number of commutator segments
- 31 Weight of motor

Values listed in the table are nominal.  
Explanation of the figures on page 71.

Option  
Sleeve bearings in place of ball bearings

### Operating Range



### Comments

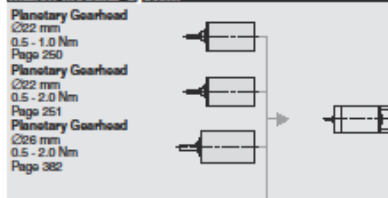
■ Continuous operation  
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.  
= Thermal limit.

Short term operation  
The motor may be briefly overloaded (recurring).

Assigned power rating

### maxon Modular System

Overview on page 20 - 25

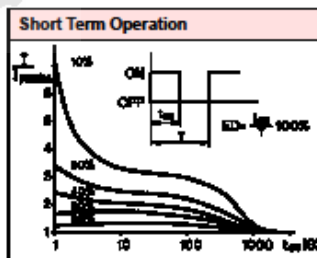
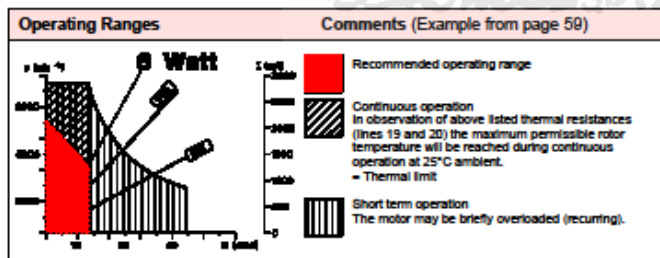


## LAMPIRAN 2

### maxon motor

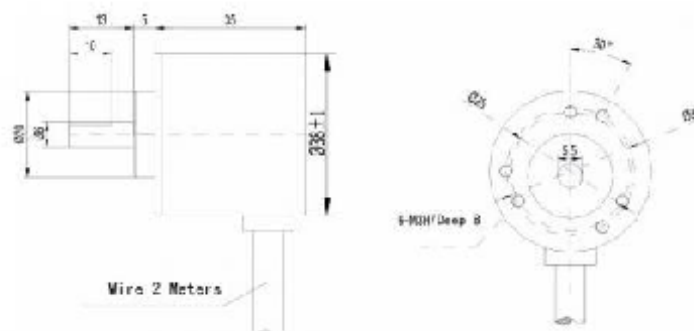
#### maxon DC motor – Comments regarding motor data (pages 37-84)

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                           |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Line 1 <b>Assigned power rating <math>P_{2T}</math> [W]</b><br>● This figure represents the maximum power output when operating within the recommended output power range. It is dependent on motor type and can be correlated with the presentation in our Selection Guide (Please refer also to pages 37-81 under «Operating Ranges»).                                                                                                                                                   | Line 7 <b>Starting current <math>I_s</math> [mA], [A]</b><br>○ Is the quotient «Voltage at terminals/DC-resistance» and is proportional to stall torque.                                                                                                                                                                                                   | The higher the efficiency, the nearer to no load speed the operating point will be. Maximum efficiency is usually at approximately $1/3$ rd of stall torque. Operating at this point is not necessarily synonymous to operating at the motor's optimum load point.                                        |
| Line 2 <b>Nominal voltage <math>U</math> [V]</b><br>All nominal values refer to operation at this voltage. It was chosen such as not to exceed the maximum recommended speed in no load condition. Motor application is of course not limited to this voltage. To reach the assigned power rating (Line 1) higher operating voltages are permissible. Maximum power output (Line 12) will increase respectively (see also page 30 «Important considerations» – bottom of right hand side). | Line 8 <b>Terminal resistance <math>R</math> [Ohm]</b><br>Is the resistance at the terminals at 25°C and determines the starting current at a given voltage. In the case of graphite brushes it must be noted that contact resistance varies depending on the load.                                                                                        | Line 14 <b>Torque constant <math>k_M</math> [mNm/A]</b><br>This may also be referred to as «specific torque» and represents the quotient from generated torque and applicable current.                                                                                                                    |
| Line 3 <b>No load speed <math>n_0</math> [rpm]</b><br>This is the speed at which the motor turns at nominal voltage and without load. For practical purposes, this speed is proportional to the voltage applied.                                                                                                                                                                                                                                                                           | Line 9 <b>Maximum permissible speed <math>n_{max}</math> [rpm]</b><br>This speed should not be exceeded during normal operation. Commutation problems may be expected at higher speeds, in turn, this could lead to premature motor failure.                                                                                                               | Line 15 <b>Speed constant <math>k_n</math> [rpm/V]</b><br>shows the specific speed per volt of applied voltage, disregarding any frictional losses. The reciprocal of the speed constant is known as the voltage constant or back-EMF constant.                                                           |
| Line 4 <b>Stall torque <math>M_H</math> [mNm]</b><br>This is the torque produced by the motor in a standstill condition, also called starting torque. The rapidly rising rotor temperature leads to a corresponding decrease in stall torque (see also «Technology – short and to the point»).                                                                                                                                                                                             | Line 10 <b>Maximum continuous current <math>I_{permiss}</math> [mA], [A]</b><br>Operating the motor continuously at this current level and at 25°C ambient will cause the winding to ultimately reach the specified max. winding temperature. This assumes no heat sinking. Depending how the motor is mounted, this value can be increased substantially. | Line 16 <b>Mechanical time constant <math>\tau_m</math> [ms]</b><br>Is the time required by the rotor to accelerate from standstill to 63% of its no load speed. In 4x the rotor will have reached more than 99% of no load speed.                                                                        |
| Line 5 <b>Speed/torque gradient <math>\Delta n/\Delta M</math> [rpm/mNm]</b><br>This gradient says a lot about the power capability of a motor. The flatter the gradient, the less speed variation is experienced during load variations. The speed/torque gradient is calculated at 25°C winding temperature.                                                                                                                                                                             | Line 11 <b>Max. continuous torque <math>M_{perm}</math> [mNm]</b><br>Is the torque that can be supplied continuously or on an average, thereby heating up the winding to the maximum permissible temperature; based on an ambient temperature of 25°C.                                                                                                     | Line 17 <b>Moment of inertia <math>J_R</math> [gcm<sup>2</sup>]</b><br>Is the polar mass moment of inertia of the rotor.                                                                                                                                                                                  |
| Line 6 <b>No load current <math>I_0</math> [mA]</b><br>This is the current the unloaded motor draws. It depends on brush and bearing friction and varies slightly with varying speed.                                                                                                                                                                                                                                                                                                      | Line 12 <b>Maximum power output <math>P_{max}</math> [mW], [W]</b><br>Is the theoretical maximum output at 25°C rotor temperature. Permissible limits are frequently below this level (see max. continuous current and max. permissible speed).                                                                                                            | Line 18 <b>Terminal inductance <math>L</math> [mH]</b><br>Is the winding inductance when stationary and measured at 1 kHz, sinusoidal.                                                                                                                                                                    |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Line 13 <b>Maximum efficiency <math>\eta_{max}</math> [%]</b><br>Efficiency is derived from the relationship between no load and starting current.                                                                                                                                                                                                         | Line 19 <b>Thermal resistance <math>R_{\theta 2}</math> [K/W]</b><br>from housing to ambient air. Characteristic value of thermal contact resistance without additional heat sinking. The sum of lines 19 and 20 define the max. admissible power loss. Heat sinking may substantially reduce this value. |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                            | Line 20 <b>Thermal resistance <math>R_{\theta 1}</math> [K/W]</b><br>from rotor to housing.                                                                                                                                                                                                               |



## LAMPIRAN 3

### LPD3806-400BM-G5-24C AB Two Phase 5-24V 400 Pulses Incremental Optical Rotary Encoder



This module is ideally suited to adding rotation detection to your project and is compatible with microcontrollers such as the Arduino. The encoder lets you add control to your project similar to radio volume control knobs in cars and detecting rotation direction, speed in machines. The rotary 6mm shaft provides a large range of compatible knobs to choose from allowing you to get the right knob for your job.

#### Uses:

For measuring the rotational speed of the object, angle, acceleration, and length measurements.

Suitable for intelligent control of various displacement measurement, automatic fixed-length leather automatic guillotine machine, steel fixed length cutting controller, civil measured height human scale, college game with robots.

#### Feature:

- AB two-phase incremental optical rotary encoder 400 pulses, NPN open collector output . (We also have 360 pulses and 600 pulses in stock, including NPN open collector output and voltage output two kinds of output. If you need please leave message for which one, or we will send by default 400 pulses and NPN open collector output ).
- Standard with 1.5 m cable
- Performance: 400 pulses / rev.
- Operating voltage: DC5-24V
- Maximum mechanical speed: 5000 rev / min
- The electrical response frequency: 20K / sec
- Integrated speed: 2000 rev / min
- Size: Encoder body size:  $\phi 38\text{mm}$ ; shaft  $\phi 6 \times 13\text{mm}$ ; axis platform: High 5mm,  $\phi 20\text{mm}$ ; fixing holes for: M3 screws
- Three mounting holes on the circle 30, and the other three mounting holes on the 28 circle ; side qualify.