



**TUGAS AKHIR - TE 145561**

**PEMODELAN MATEMATIKA KECEPATAN MOTOR DC  
MENGUNAKAN IDENTIFIKASI DENGAN METODE 2S**

Annona Dieni Septiarini  
NRP. 1031140000096

Dosen Pembimbing  
Ir. Josaphat Pramudijanto, M.Eng.

Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018





***FINAL PROJECT - TE 145561***

***MATHEMATICS MODELING DC MOTOR SPEED USES  
IDENTIFICATION WITH 2S METHOD***

Annona Dieni Septiarini  
NRP. 1031140000096

*Supervisor*  
Ir. Josaphat Pramudijanto, M.Eng.

*Electrical and Automation Engineering Departement  
Faculty of Vokasi  
Sepuluh Nopember Insitute of Technology  
Surabaya 2018*



## PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas akhir saya dengan judul "**Pemodelan Matematika Kecepatan Motor DC Menggunakan Identifikasi Dengan Metode 2S.**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Januari 2018



Annona Dieni Septiarini

NRP. 1031140000096

*(halaman ini sengaja dikosongkan)*

**PEMODELAN MATEMATIKA KECEPATAN MOTOR DC  
MENGUNAKAN IDENTIFIKASI DENGAN METODE 2S**

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Ahli Madya  
Pada  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember**

**Menyetujui:**

**Dosen Pembimbing**



**SURABAYA  
JANUARI, 2018**

*(halaman ini sengaja dikosongkan)*

## **Pemodelan Matematika Kecepatan Motor DC Menggunakan Identifikasi Dengan Metode 2S.**

**Nama** : Annona Dieni Septiarini  
**NRP** : 1031140000096

**Pembimbing** : Ir. Josaphat Pramudijanto, M.Eng.  
**NIP** : 196210051990031003

### **ABSTRAK**

Modul sistem kecepatan Motor DC merupakan modul yang digunakan untuk menghitung kecepatan Motor DC. Pada modul ini membutuhkan pengaturan kecepatan Motor DC dengan Arduino menggunakan teknik PWM ( *Pulse With Modulation* ). Pengaturan dengan teknik ini mengakibatkan kecepatan yang akan sulit dikendalikan sesuai yang diharapkan. Pengaturan kecepatan yang tidak tepat juga dapat mengakibatkan hasil berupa data yang berubah-ubah dengan input tegangan yg sama. Dan informasi tentang parameter Motor tidak tersedia sehingga harus dilakukan identifikasi plant dengan cara pengukuran

Dari persoalan di atas maka ada suatu cara untuk merancang dan mensimulasikan suatu sistem pengaturan kecepatan Motor DC, dengan adanya model matematika dari plant yang akan dikontrol. Sistem ini menggunakan Motor DC sebagai objek utama, dengan mencari kecepatan dan tegangan Motor tiap PWM dinaikkan dan menggunakan rem magnetik sebagai beban maka akan didapatkan daerah titik kerja yg linier. Pada daerah ini dilakukan percobaan lagi untuk mendapatkan hasil respon kecepatan tiap titik kerja, yang digunakan pada pemodelan matematika nanti.

Pada penelitian ini titik kerja atau nilai linier dari kecepatan Motor DC dengan tanpa beban dan beban mulai 25% sampai 90% . Dengan titik kerja Motor antara 8Volt – 13Volt atau 2210 – 3770rpm.

**Kata kunci** : *Motor DC, PWM, titik kerja Motor*

*(halaman ini sengaja dikosongkan)*

***Mathematics Modeling DC Motor Speed Uses Identification With 2S Method.***

***Nama*** : **Annona Dieni Septiarini**  
***Register Number*** : **10311400000096**

***Supervisor*** : **Ir. Josaphat Pramudijanto, M.Eng.**  
***ID Number*** : **196210051990031003**

***ABSTRACT***

*The DC Motor speed system module is a module used to calculate the speed of a DC Motor. In this module it requires DC Motor speed adjustment with Arduino using PWM (Pulse With Modulation) technique. The arrangement with this technique leads to speed which will be difficult to control as expected. Incorrect speed settings may also result in data being varied with the same input voltage. And information about Motor parameters is not available so that plant identification must be done by measurement*

*From the above problem there is a way to design and simulate a DC Motor speed regulation system, with the mathematical model of the plant to be controlled. This system uses the DC Motor as the main object, by looking for speed and voltage of each PWM Motor raised and using magnetic brake as a load it will get linear point of work area. In this area another experiment was conducted to obtain the response speed of each working point, which was used in later mathematical modeling.*

*In this study the working point or linear value of DC Motor speed with no load and load from 25% to 90%. With a working Motor point between 8Volt - 13Volt or 2210 - 3770rpm.*

***Keywords*** : *DC Motor, PWM, Motor working point*

*(halaman ini sengaja dikosongkan)*

## KATA PENGANTAR

Puji syukur atas kehadiran Allah SWT karenaberkat rahmat dan hidayah-Nya penulis dapat menyusun dan menyelesaikan laporan Tugas Akhir dengan judul:

### **Pemodelan Matematika Kecepatan Motor DC Menggunakan Identifikasi Dengan Metode 2S**

Penulis ingin berterima kasih kepada beberapa pihak yang telah membantu dalam penyusunan dan penyelesaian dari Tugas Akhir ini, diantaranya:

1. Kedua orang tua yang selalu memberikan dukungan, semangat, dan doa kepada penulis.
2. Bapak Ir. Josaphat Pramudijanto, M.Eng. selaku Dosen Pembimbing atas bantuan dan bimbingan hingga Tugas Akhir ini terselesaikan.
3. Keluarga besar *Andromeda* 2014 yang selalu memberikan bantuan yang tidak terkira terhadap penulis.

Harapan besar penulis bahwa buku Tugas Akhir ini dapat memberikan informasi dan manfaat bagi pembaca pada umumnya dan mahasiswa Program Studi D3 Teknik Elektro pada khususnya. Penulis juga mengharapkan kritik dan saran atas buku Tugas Akhir ini karena penulis menyadari bahwa dalam penyusunan Tugas Akhir ini banyak terdapat kekurangan.

Surabaya, Januari 2018

Penulis

*(halaman ini sengaja dikosongkan)*

## DAFTAR ISI

HALAMAN JUDUL.....	
PERNYATAAN KEASLIAN TUGAS AKHIR .....	v
LEMBAR PENGESAHAN.....	vii
ABSTRAK .....	ix
<i>ABSTRACT</i> .....	xi
KATA PENGANTAR .....	xiii
DAFTAR ISI .....	xv
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL.....	xix
BAB I .....	1
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian .....	2
1.5 Sistematika Penulisan.....	2
1.6 Relevansi.....	3
BAB II .....	5
2.1 Modul Praktikum Motor DC .....	5
2.2 Sistem Kontrol Kecepatan .....	5
2.3 Motor DC .....	7
2.4 <i>Driver</i> Motor DC .....	8
2.5 <i>Rotary Encoder</i> .....	9
2.6 Mikrokontroler.....	10
2.6.1 Daya ( <i>Power</i> ) dan Memori .....	12
2.6.2 <i>Input &amp; Output</i> .....	12
2.6.3 Komunikasi .....	13
2.6.4 <i>Software</i> Arduino .....	13
2.7 Rem Magnetik .....	14
2.8 Pemodelan Sistem .....	15
2.8.1 Metode Strejc's .....	15
2.8.2 Metode Smith's .....	17
BAB III .....	19
3.1 Blok Fungsional Sistem.....	19
3.2 Perancangan Perangkat Keras.....	20
3.2.1 Perancangan Mekanik.....	20
3.2.2 Perancangan Elektrik .....	22
3.2.2.1 Arduino UNO.....	22
3.3 Perancangan Perangkat Lunak.....	23

3.3.1	Pembuatan <i>Flowchart</i> Program .....	23
3.3.2	Pemrograman <i>Software LabVIEW</i> .....	26
BAB IV	.....	31
4.1	Gambaran Umum Pengujian Sistem.....	31
4.2	Pengujian Perangkat Keras .....	31
4.2.1	Pengujian Keseluruhan Sistem .....	31
4.2.2	Pengujian Motor Tanpa Rem.....	31
4.2.3	Pengujian Motor Dengan Rem 25% .....	32
4.2.4	Pengujian Motor Dengan Rem 50% .....	33
4.2.5	Pengujian Motor Dengan Rem 75% .....	34
4.2.6	Pengujian Motor Dengan Rem 90% .....	35
4.2.7	Hasil Keseluruhan Pengujian .....	36
4.2.8	Pengujian Titik Kerja Motor.....	37
4.2.8.1	Pengujian Tanpa Rem.....	37
4.2.8.2	Pengujian Dengan Rem 25%.....	40
4.2.8.3	Pengujian Dengan Rem 50%.....	43
4.2.8.4	Pengujian Dengan Rem 75%.....	46
4.2.8.5	Pengujian Dengan Rem 90%.....	48
4.2.8.6	Hasil Rangkuman Pengukuran.....	51
BAB V	.....	53
5.1	Kesimpulan.....	53
5.2	Saran .....	53
DAFTAR PUSTAKA	.....	55
LAMPIRAN A	.....	57
A1.	Listing Program.....	57
LAMPIRAN B	.....	61
B1.	Tampilan LabVIEW Monitoring Kecepatan Motor .....	61
RIWAYAT HIDUP	.....	63

## DAFTAR GAMBAR

<b>Gambar 2.1</b>	Pulsa Kotak dengan <i>Duty Cycle</i> 50% .....	6
<b>Gambar 2.2</b>	Bentuk Pulsa PWM .....	6
<b>Gambar 2.3</b>	Blok Diagram Sistem Kontrol Tertutup .....	7
<b>Gambar 2.4</b>	Motor DC Sederhana .....	8
<b>Gambar 2.5</b>	Rangkaian Transistor Pada <i>Driver</i> Motor <i>H-Bridge</i> .....	8
<b>Gambar 2.6</b>	Contoh Pola Keluaran <i>Incremental Encoder</i> .....	10
<b>Gambar 2.7</b>	<i>Board</i> Arduino UNO .....	10
<b>Gambar 2.8</b>	Tampilan IDE Arduino Dengan Sebuah <i>Sketch</i> .....	14
<b>Gambar 2.9</b>	Step-respon dari Sistem dengan Ditunjukkan pada $t_{1,2}, t_i, T_U$ dan $T_N$ .....	16
<b>Gambar 2.10</b>	Step-respon Sistem Ditunjukkan Dengan Waktu $t_{20}$ dan $t_{60}$ .....	18
<b>Gambar 2.11</b>	Kurva untuk Penilaian $\tau$ dan $\zeta$ .....	18
<b>Gambar 3.1.</b>	Blok Fungsional Sistem.....	19
<b>Gambar 3.2</b>	Motor DC 24 Volt .....	21
<b>Gambar 3.3</b>	Model <i>Rotary Encoder</i> .....	22
<b>Gambar 3.4</b>	Perancangan Komponen Elektrik.....	22
<b>Gambar 3.5</b>	Rangkaian Arduino UNO .....	23
<b>Gambar 3.6</b>	<i>Flowchart</i> Sistem .....	25
<b>Gambar 3.7</b>	<i>Dialog Box</i> Utama LabVIEW .....	27
<b>Gambar 3.8</b>	<i>Dialog Box</i> LabVIEW .....	27
<b>Gambar 3.9</b>	<i>Front Panel</i> .....	28
<b>Gambar 3.10</b>	<i>Block Diagram</i> .....	28
<b>Gambar 3.11</b>	Tampilan Utama LabVIEW .....	28
<b>Gambar 3.12</b>	Hasil Tampilan Monitor pada <i>Front Panel</i> 1.....	29
<b>Gambar 3.13</b>	Hasil Tampilan Monitor pada <i>Front Panel</i> 2.....	29
<b>Gambar 4.1</b>	Hasil Keseluruhan Pengujian .....	37
<b>Gambar 4.2</b>	Hasil Respon Kecepatan Motor Tanpa Rem.....	38
<b>Gambar 4.3</b>	Hasil Respon Kecepatan Motor Rem 25% .....	41
<b>Gambar 4.4</b>	Hasil Respon Kecepatan Motor Rem 50% .....	44
<b>Gambar 4.5</b>	Hasil Respon Kecepatan Motor Rem 75% .....	47
<b>Gambar 4.6</b>	Hasil Respon Kecepatan Motor Rem 90% .....	49

*(halaman ini sengaja dikosongkan)*

## DAFTAR TABEL

<b>Tabel 2.1</b> <i>Output Rotary Encoder Incremental</i> .....	9
<b>Tabel 2.2</b> <i>Datasheet</i> Mikrokontroler Arduino UNO.....	11
<b>Tabel 2.3</b> Nilai dari Konstan T.....	17
<b>Tabel 2.4</b> Penilaian Faktor n dan Ketepatan Koordinat Titik Infeksi....	17
<b>Tabel 3.1</b> Spesifikasi Motor DC .....	20
<b>Tabel 4.1</b> Data Pengujian Motor Tanpa Rem .....	32
<b>Tabel 4.2</b> Data Pengujian Motor Dengan Rem 25% .....	33
<b>Tabel 4.3</b> Data Pengujian Motor Dengan Rem 50% .....	34
<b>Tabel 4.4</b> Data Pengujian Motor Dengan Rem 75% .....	35
<b>Tabel 4.5</b> Data Pengujian Motor Dengan Rem 90% .....	36
<b>Tabel 4.6</b> Data Pengujian Tanpa Rem.....	37
<b>Tabel 4.7</b> Data Pengujian Dengan Rem 25%.....	40
<b>Tabel 4.8</b> Data Pengujian Dengan Rem 50%.....	42
<b>Tabel 4.9</b> Data Pengujian Dengan Rem 75%.....	46
<b>Tabel 4.10</b> Data Pengujian Dengan Rem 90%.....	48
<b>Tabel 4.11</b> Hasil Rangkuman Pengujian Metode Strejc's.....	51
<b>Tabel 4.12</b> Hasil Rangkuman Pengujian Metode Smith's.....	52

*(halaman ini sengaja dikosongkan)*

# BAB I

## PENDAHULUAN

Pada Bab ini akan dibahas mengenai latar belakang pembuatan Tugas Akhir, perumusan masalah, batasan masalah, tujuan, sistematika penulisan dan relevansi.

### 1.1 Latar Belakang

Motor DC merupakan motor listrik yang disuplai dari catu daya tegangan searah atau tegangan DC. Motor DC merupakan suatu perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Motor DC juga merupakan aktuator yang banyak digunakan dalam sistem kontrol. Motor menyediakan gerakan putar yang selanjutnya dapat dikopel dengan roda gigi atau belt untuk menghasilkan gerakan translasional. Pada tugas akhir ini Motor DC dipalikasikasn pada modul sistem kecepatan.

Suatu *plant* merupakan suatu sistem yang mungkin disusun dari bermacam-macam komponen/elemen yang saling berhubungan dalam menjalankan suatu aksi. Tergantung dari komponen/elemen yang dipergunakan, suatu sistim dapat berupa mekanik, pneumatik, elektrik atau jenis elektro-mekanik. Motor DC adalah termasuk dalam sistim komponen elektro-mekanik. Operasi Motor DC dengan kontrol armature menghasilkan rasio steady state kecepatan yang hampir linier terhadap tegangan masukannya dan arah putaran Motor DC tergantung pada polaritas tegangan masukan. Suatu prasyarat dasar pada kebanyakan strategi yang dipergunakan dalam membangun suatu sistim kontrol adalah kemampuan dalam memodelkan secara matematika dari plant yaitu sistim yang akan dikontrol. Model matematika dari suatu plant dapat berupa persamaan differensial, *transfer function* atau *state space*. Persamaan differensial menjelaskan suatu performa dinamik dari suatu sistem. Model ini dapat termasuk turunan (derivative) sekian orde dari masukan dan keluaran. Parameter sistim muncul dalam persamaan model matematika sebagai koefisien. Pada prinsipnya terdapat dua macam pemodelan matematika yang dapat dipilih, yaitu berdasarkan pada teori pengetahuan dengan menggunakan hukum fisika atau dengan proses eksperimen yaitu melakukan pengukuran.

Pada kebanyakan kasus adalah tidak mungkin untuk membuat model yang sempurna hanya dengan menggunakan pengetahuan fisika saja. Beberapa parameter harus ditentukan dari suatu eksperimen Pendekatan ini disebut dengan Identifikasi sistem. Ada banyak metode untuk menganalisa data yang didapatkan dari eksperimen. Pada eksperimen ini proses identifikasi dilakukan dengan sistem statis menggunakan metode yang digunakan adalah dengan metode Strej'c dan metode Smith's, dengan kedua

metode ini dapat dilakukan perbandingan pada kecepatan Motor dengan memberikan tegangan input yang berbeda kemudian dicari nilai RMSE terkecil untuk mendapatkan data hasil yang terbaik.

## **1.2 Perumusan Masalah**

Melihat latar belakang di atas dalam Tugas Akhir ini, maka dirumuskan permasalahan yaitu belum adanya informasi tentang parameter Motor, sehingga harus dilakukan identifikasi plant dengan cara pengukuran yaitu dengan pemodelan matematika dengan suatu metode.

## **1.3 Batasan Masalah**

Dari perumusan masalah tersebut, maka batasan masalah dari tugas akhir ini adalah :

1. Modul ini digunakan untuk mengatur kecepatan pada Motor DC maksimal hingga 5000 rpm
2. Hasil dari pengontrol kecepatan , pengereman, *counter* dan tegangan pada Motor ditampilkan pada monitor dengan *software* LabVIEW.
3. Dalam modul ini kecepatan pada Motor DC dapat dikendalikan.
4. Pada Monitor dapat menampilkan data grafik kecepatan berupa rpm, pengereman, *counter* dan tegangan.

## **1.4 Tujuan Penelitian**

Tujuan membuat Tugas Akhir ini adalah sebagai berikut ;

1. Merancang dan merealisasikan sistem kontrol Motor DC menggunakan metode 2S (Strejc's dan Smith's)
2. Merancang dan merealisasikan model matematika dan *performance* kecepatan dan rem Motor DC
3. Merealisasikan supaya modul kecepatan Motor DC dapat dikontrol melalui mikrokontroler, dimana hasil dari pengukuran Motor DC di tampilkan pada *software* labview.

## **1.5 Sistematika Penulisan**

Sistematika pembahasan tugas akhir ini terdiri dari lima bab, yaitu Pendahuluan, Dasar Teori, Perancangan dan Pembuatan Alat, Pengujian dan Analisa Alat, serta Penutup.

### **BAB 1 : PENDAHULUAN**

Membahas tentang latar belakang, permasalahan, batasan masalah, maksud dan tujuan, sistematika laporan, serta relevansi.

### **BAB II : TINJAUAN PUSTAKA**

Membahas dasar teori pada Tugas Akhir ini yaitu Modul Praktikum Motor DC, Sistem Kontrol

Kecepatan, Motor DC, DC Motor *Driver*, Mikrokontroler, *Rotary Encoder*, Rem Magnetik, Pemodelan Matematika, Metode Strejc's dan Metode Smith's.

**BAB III : PERANCANGAN SISTEM**

Membahas tentang perencanaan dan pembuatan perangkat keras (*hardware*) yang terdiri dari perancangan elektronik dan perancangan mekanik serta pembuatan dan perancangan perangkat lunak (*software*).

**BAB IV : PENGUJIAN DAN ANALISA**

Membahas tentang pengujian alat yang terdiri dan pengujian perangkat keras dan juga perangkat lunak. Begitu pula dengan pengukuran.

**BAB V : PENUTUP**

Menjelaskan tentang kesimpulan dari tugas akhir ini dan saran-saran untuk pengembangan alat ini lebih lanjut

### **1.6 Relevansi**

Manfaat dari Pengerjaan Tugas Akhir (TA) ini untuk memenuhi kurikulum dari kuliah Diploma 3 (tiga) Teknik Elektro Fakultas Vokasi Institut Sepuluh Nopember. Juga diharapkan dapat untuk mendapatkan suatu model dari suatu plant. Dengan model dari suatu *plant*, kita dapat memperoleh keluaran tanpa *noise* yang jika menggunakan proses fisika menghasilkan keluaran dengan *noise* untuk mengoptimalkan suatu kontroler. Untuk mendapatkan *self-tuning* dan kontrol adaptif secara otomatis. Untuk memprediksi suatu keluaran yang akan terjadi pada waktu kedepan. Juga diharapkan dapat memudahkan perancangan sistem kontrol kecepatan dengan Motor DC melalui model matematika.

*(halaman ini sengaja dikosongkan)*

## **BAB II**

### **TEORI DASAR**

Pada Bab ini akan dibahas mengenai materi dasar dalam penyusunan Tugas Akhir. Pada Tugas Akhir ini melanjutkan alat dari Tugas Akhir Catur Andianto dan Amanda Dwi P. Beberapa hal yang di bahas meliputi teori dasar mengenai Tugas Akhir ini diantaranya adalah Modul Praktikum Motor DC, Sistem Kontrol Kecepatan, Motor DC, DC Motor *Driver*, Mikrokontroler, *Rotary Encoder*, Rem Magnetik, Pemodelan Matematika, Metode Strejc's dan Metode Smith's.

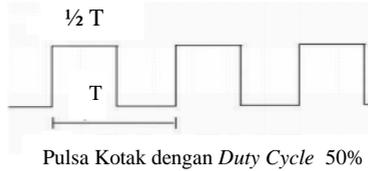
#### **2.1 Modul Praktikum Motor DC [1]**

Modul adalah satuan program pembelajaran yang terkecil, yang dapat dipelajari oleh mahasiswa sendiri secara perseorangan (*self instructional*) setelah mahasiswa menyelesaikan satu satuan dalam modul, selanjutnya mahasiswa dapat melangkah maju dan mempelajari satuan modul berikutnya. Pembelajaran dengan menggunakan modul, merupakan strategi tertentu dalam menyelenggarakan pembelajaran individual. Modul pembelajaran, sebagaimana yang dikembangkan di Indonesia, merupakan suatu paket bahan pembelajaran (*learning materials*) yang memuat deskripsi tentang tujuan pembelajaran, lembaran petunjuk dosen yang menjelaskan cara mengajar yang efisien, bahan bacaan bagi mahasiswa, lembaran kunci jawaban pada lembar kertas kerja mahasiswa, dan alat-alat evaluasi pembelajaran.

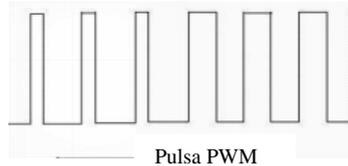
#### **2.2 Sistem Kontrol Kecepatan[2]**

Sistem kontrol kecepatan atau pengaturan kecepatan Motor DC adalah dengan menggunakan cara analog. Pada tugas akhir ini mengatur kecepatan Motor DC dengan menggunakan mikrokontroler. Mikrokontroler yang digunakan adalah Arduino UNO. Data hasil dari kecepatan Motor akan ditampilkan pada *software* LabVIEW. Kemudian sebagai *driver* motor menggunakan modul *driver* Motor IC L298N. Cara pengaturan kecepatan yang digunakan adalah dengan menggunakan teknik PWM (*Pulse Width Modulation*), salah satu teknik untuk mengatur kecepatan Motor DC yang digunakan. Dengan menggunakan PWM dapat mengatur kecepatan yang diinginkan dengan mudah. Teknik PWM untuk pengaturan kecepatan Motor adalah, pengaturan kecepatan Motor dengan cara merubah-ubah besarnya *duty cycle* pulsa. Pulsa yang berubah ubah *duty cycle*-nya inilah yang menentukan kecepatan Motor. Besarnya amplitudo dan frekuensi pulsa. adalah tetap, sedangkan besarnya *duty cycle* berubah-ubah sesuai dengan kecepatan yang diinginkan, semakin besar *duty cycle* maka semakin cepat pula kecepatan Motor, dan sebaliknya semakin kecil *duty cycle* maka

semakin pelan pula kecepatan Motor. Sebagai contoh bentuk pulsa yang dikirimkan adalah seperti pada **Gambar 2.1**, pulsa kotak dengan *duty cycle* pulsa 50%. Sedangkan sebagai contoh bentuk pulsa PWM adalah seperti **Gambar 2.2**

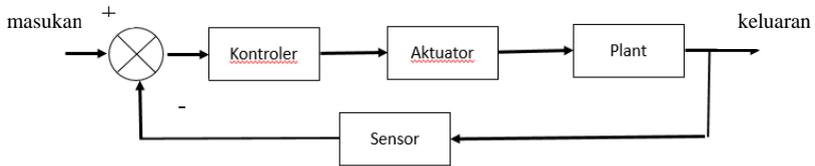


**Gambar 2.1** Pulsa Kotak dengan *Duty Cycle* 50%



**Gambar 2.2** Bentuk Pulsa PWM

Seperti pada **Gambar 2.1**, semakin besar *duty cycle* pulsa kotak, maka semakin lama pula posisi logika *high*. Jika Motor diatur agar berjalan ketika diberi logika *high*, maka jika memberi pulsa seperti pada gambar 1 diatas, maka Motor akan berada pada kondisi “nyala-mati-nyala-mati” sesuai dengan bentuk pulsa tersebut. Semakin lama Motor berada pada kondisi “nyala” maka semakin cepat pula kecepatan Motor tersebut. Motor akan berputar dengan kecepatan maksimum jika mendapat pulsa dengan *duty cycle* 100%. Dengan kata lain Motor mendapat logika *high* terus menerus. Dengan mengatur besarnya *duty cycle* pulsa kotak yang dikirimkan, kita dapat mengatur banyaknya logika *high* yang diberikan pada Motor, dengan kata lain mengatur lamanya waktu motor untuk berputar dalam satu periode pulsa. Jika lamanya waktu Motor untuk berputar dalam satu periode pulsa ini berubah maka kecepatan putaran Motor juga akan berubah, sesuai dengan *duty cycle* atau waktu Motor untuk berputar dalam satu periode pulsa. Pada sistem kontrol kecepatan ini menggunakan sistem kontrol loop tertutup dengan blok diagram seperti **Gambar 2.3**.



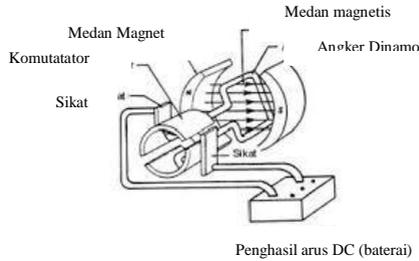
**Gambar 2.3** Blok Diagram Sistem Kontrol Tertutup

Sistem kontrol tertutup adalah sistem kontrol yang sinyal keluarannya mempunyai pengaruh langsung pada aksi pengontrolan, sistem kontrol tertutup juga merupakan sistem kontrol berumpan balik. Sinyal kesalahan penggerak, yang merupakan selisih antara sinyal masukan dan sinyal umpan balik yang dapat berupa sinyal keluaran atau suatu fungsi sinyal keluaran atau turunannya, diumpungkan ke kontroler untuk memperkecil kesalahan dan membuat agar keluaran sistem mendekati harga yang diinginkan. Dengan kata lain, istilah “kontrol tertutup” berarti menggunakan aksi umpan – balik untuk memperkecil kesalahan sistem. Jika dalam hal ini manusia bekerja sebagai operator, maka manusia ini akan menjaga sistem agar tetap pada keadaan yang diinginkan, ketika terjadi perubahan pada sistem maka manusia akan melakukan langkah – langkah awal pengaturan sehingga sistem kembali bekerja pada keadaan yang diinginkan.

### 2.3 Motor DC [3]

Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Kumparan medan pada Motor DC disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Jika terjadi putaran pada kumparan jangkar dalam pada medan magnet, maka akan timbul tegangan (GGL) yang berubah-ubah arah pada setiap setengah putaran, sehingga merupakan tegangan bolak-balik.

Prinsip kerja dari arus searah adalah membalik fasa tegangan dari gelombang yang mempunyai nilai positif dengan menggunakan komutator, dengan demikian arus yang berbalik arah dengan kumparan jangkar yang berputar dalam medan magnet. Bentuk Motor paling sederhana memiliki kumparan satu lilitan yang bisa berputar bebas di antara kutub-kutub magnet permanen.



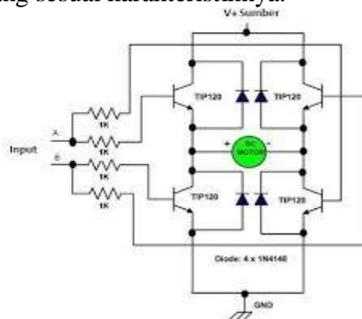
**Gambar 2.4** Motor DC Sederhana

Catu tegangan DC dari baterai menuju ke lilitan melalui sikat yang menyentuh komutator, dua segmen yang terhubung dengan dua ujung lilitan. Kumparan satu lilitan pada gambar di atas disebut anker dinamo. Anker dinamo adalah sebutan untuk komponen yang berputar di antara medan magnet.

### 2.4 Driver Motor DC [4]

Motor DC tidak dapat dikendalikan secara langsung oleh mikrokontroler, karena kebutuhan arus listrik yang besar pada Motor DC. Sedangkan arus keluaran pada mikro sangat kecil. Jika Motor langsung dihubungkan ke mikrokontroler, maka hal ini akan menyebabkan kerusakan pada mikrokontroler tersebut. *Driver* Motor merupakan pilihan alternatif yang harus digunakan untuk mengendalikan Motor DC.

*Driver* Motor merupakan suatu rangkaian khusus yang memiliki fungsi untuk mengatur arah ataupun kecepatan pada Motor DC. Bentuk rangkaian *Driver* Motor yang umum digunakan yaitu *H-Bridge*. Berbentuk seperti huruf H yang memiliki perbedaan fungsi di setiap sisinya. *Driver* Motor *H-Bridge* adalah rangkaian yang tersusun dari transistor yang digunakan untuk menggerakkan Motor DC. Komponen utamanya adalah transistor yang dipasang sesuai karakteristiknya.



**Gambar 2.5** Rangkaian Transistor pada *Driver* Motor *H-Bridge*

Pada saat *input* A berlogika 1, maka ada arus yang mengalir pada rangkaian, akibatnya transistor 1 dan 3 on karena basis terbias, sehingga Motor berputar. Pada saat *input* B berlogika 1, maka ada arus yang mengalir pada rangkaian, akibatnya transistor 2 dan 4 on karena basis terbias, sehingga Motor berputar tapi dengan arah yang berlawanan. Pada saat membuat rangkaian tersebut, jika *input* A berlogika 1 maka *input* B harus berlogika 0 dan juga sebaliknya. Hal ini agar rangkaian tersebut dapat bekerja dengan baik dan juga agar transistor tidak mudah rusak. Jadi hanya ada 1 *input* yang berlogika 1 sedangkan yang lain berlogika 0. Untuk komponen yang digunakan seperti resistor dan transistor dapat diganti sesuai kebutuhan dari rangkaian tersebut.

## 2.5 Rotary Encoder [5]

*Rotary Encoder* disebut juga *encoder* poros, adalah perangkat elektromekanis yang mengubah sudut\_posisi/gerakan poros atau mengubah poros ke kode *analog* ataupun *digital*. Ada dua jenis utama *rotary encoder*, yaitu : absolut dan *incremental* (relatif). Sebuah *rotary encoder incremental* menyediakan *output* siklus (hanya) ketika *encoder* diputar. *Rotary encoder* dapat berupa mekanik atau optik. Jenis pada benda mekanik membutuhkan *debouncing* dan biasanya digunakan sebagai potensiometer *digital* pada peralatan termasuk perangkat konsumen.

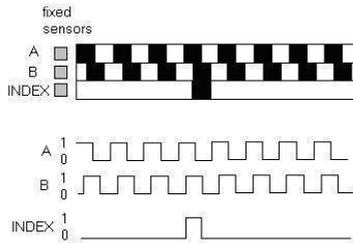
*Rotary encoder incremental* adalah yang paling banyak digunakan dari semua *rotary encoder* karena biaya rendah dan kemampuan untuk memberikan sinyal yang dapat dengan mudah ditafsirkan untuk memberikan informasi terkait seperti gerak kecepatan. *Encoder Incremental* digunakan untuk melacak gerakan dan dapat digunakan untuk menentukan posisi dan kecepatan. Hal ini dapat berupa linear atau gerakan berputar. Karena arah dapat ditentukan, pengukuran yang sangat akurat dapat dibuat.

*Rotary Encoder Incremental* memiliki dua buah *output* yang disebut *channel A* dan *channel B*, yang disebut *output quadrature*, karena pada *channel A* dan *channel B* memiliki perbedaan sebesar 90 derajat keluar dari fase.

**Tabel 2.1** output Rotary Encoder Incremental

Coding untuk rotasi berlawanan arah jarum jam		
Tahap	A	B
1	1	0
2	1	1
3	0	1
4	0	0

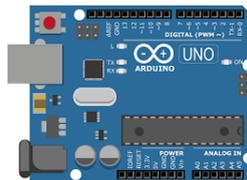
Coding untuk rotasi searah jarum jam		
Tahap	A	B
1	0	0
2	0	1
3	1	1
4	1	0



**Gambar 2.6** Contoh Pola Keluaran *Incremental Encoder*

## 2.6 Mikrokontroler[6]

Pada tugas akhir ini, mikrokontroler yang dipakai ialah Arduino UNO. Arduino UNO adalah sebuah *board* mikrokontroler yang didasarkan pada ATmega328. Arduino UNO mempunyai 14 pin *digital input* atau *output* (6 diantaranya dapat digunakan sebagai *output PWM*), 6 *input analog*. Sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah *power jack*, sebuah *ISCP header*, dan sebuah tombol *reset*. Arduino UNO memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkan ke sebuah komputer dengan sebuah kabel USB.



**Gambar 2.7** *Board* Arduino UNO

Arduino memiliki kelebihan tersendiri dibanding *board* mikrokontroler yang lain selain bersifat *open source*, arduino juga mempunyai bahasa pemrogramannya sendiri yang berupa bahasa C. Selain itu dalam *board*

Arduino sendiri sudah terdapat *loader* yang berupa USB sehingga memudahkan kita ketika kita memprogram mikrokontroler didalam arduino. Sedangkan pada kebanyakan *board* mikrokontroler yang lain yang masih membutuhkan rangkaian *loader* terpisah untuk memasukkan program ketika kita memprogram mikrokontroler. *Port* USB tersebut selain untuk *loader* ketika memprogram, bisa juga difungsikan sebagai *port* komunikasi *serial*.

Arduino menyediakan 20 *pin* I/O, yang terdiri dari 6 *pin* *input analog* dan 14 *pin* *digital input/output*. Untuk 6 *pin* *analog* sendiri bisa juga difungsikan sebagai *output digital* jika diperlukan *output digital* tambahan selain 14 *pin* yang sudah tersedia. Untuk mengubah *pin* *analog* menjadi *digital* cukup mengubah konfigurasi *pin* pada program. Pada dalam *board* kita bisa lihat *pin* *digital* diberi keterangan 0-13, jadi untuk menggunakan *pin* *analog* menjadi *output digital*, *pin* *analog* yang pada keterangan *board* 0-5 kita ubah menjadi *pin* 14-19. dengan kata lain *pin* *analog* 0-5 berfungsi juga sebagai *pin* *output digital* 14-16

**Tabel 2.2** *Datasheet* Mikrokontroler Arduino UNO

Mikrokontroler	AT mega 328
Tegangan Pengoperasian	5 Volt
Batas tegangan yg disarankan	7 – 12 Volt
Batas tegangan <i>input</i>	6 – 20 Volt
Jumlah <i>pin</i> I/O <i>digital</i>	14 <i>pin</i> <i>digital</i> (6 diantaranya menyediakan keluaran PWM)
Jumlah <i>pin</i> <i>input Analog</i>	6 <i>pin</i>
Arus DC tiap <i>pin</i> I/O	40 mA
Arus DC untuk <i>pin</i> 3,3 V	50 mA
<i>Memory Flash</i>	32KB (AT mega 328) sekitar 0,5KB digunakan oleh <i>Bootloader</i>
SRAM	2KB (AT mega 328)
EEPROM	1KB (AT mega 328)
<i>Clockspeed</i>	16MHz

Sifat *open source* arduino juga banyak memberikan keuntungan tersendiri untuk kita dalam menggunakan *board* ini, karena dengan sifat *open source* komponen yang kita pakai tidak hanya tergantung pada satu merek, namun memungkinkan kita bisa memakai semua komponen yang ada dipasaran. Bahasa pemrograman arduino merupakan bahasa C yang sudah disederhanakan *syntax* bahasa pemrogramannya sehingga mempermudah kita dalam mempelajari dan mendalami mikrokontroler.

### 2.6.1 Daya (*Power*) dan Memori

Arduino dapat diberikan *power* melalui koneksi USB atau *power supply*. *Power*nya diseleksi secara otomatis. *Power supply* dapat menggunakan adaptor DC atau baterai. Adaptor dapat dikoneksikan dengan menyambung *jack* adaptor pada koneksi *port input supply*. *Board* arduino dapat dioperasikan menggunakan *supply* dari luar sebesar 6 - 20 Volt. Jika *supply* kurang dari 7 Volt, kadang kala *pin* 5 Volt akan menyuplai kurang dari 5 Volt dan *board* bisa menjadi tidak stabil. Jika menggunakan lebih dari 12 Volt, tegangan di regulator bisa menjadi sangat panas dan menyebabkan kerusakan pada *board*. Rekomendasi tegangan ada pada 7 sampai 12 Volt.

Penjelasan pada *pin* daya (*power*) adalah sebagai berikut :

- a. **Vin.** Tegangan *input* ke *board* arduino ketika menggunakan tegangan dari luar (seperti yang disebutkan 5 Volt dari koneksi dengan USB atau tegangan yang diregulasikan). Pengguna dapat memberikan tegangan melalui *pin* ini, atau jika tegangan suplai menggunakan *power jack*, aksesnya menggunakan *pin* ini.
- b. **5V.** Regulasi *power supply* digunakan untuk *power* mikrokontroler dan komponen lainnya pada *board*. 5 Volt dapat melalui *Vin* menggunakan regulator pada *board*, atau *supply* oleh USB atau *supply* regulasi 5 Volt lainnya.
- c. **3.3V.** Suplai 3.3 Volt didapat oleh FTDI *chip* yang ada di *board*. Arus maksimumnya adalah 50mA
- d. **Pin Ground** berfungsi sebagai jalur *ground* pada Arduino.
- e. **Memori.** ATmega328 memiliki 32 KB *flash* memori untuk menyimpan kode, juga 2 KB yang digunakan untuk *bootloader*. ATmega328 memiliki 2 KB untuk SRAM dan 1 KB untuk EEPROM.

### 2.6.2 *Input & Output*

Setiap 14 *pin digital* pada arduino dapat digunakan sebagai *input* atau *output*, menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. *Input/output* dioperasikan pada 5 Volt. Setiap *pin* dapat menghasilkan atau menerima maksimum 40 mA dan memiliki internal *pull-up* resistor (*disconnected* oleh *default*) 20-50 KOhm. Beberapa *pin* memiliki fungsi sebagai berikut ;

- a. *Serial* : 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirim (TX) TTL data *serial*. *Pin* ini terhubung pada *pin* yang koresponding dari USB ke TTL *chip serial*
- b. *Interrupt* eksternal : 2 dan 3 *Pin* ini dapat dikonfigurasi untuk *trigger* sebuah interap pada *low value*, *rising* atau *falling edge*, atau perubahan nilai

- c PWM : 3, 5, 6, 9, 10, dan 11. Mendukung 8-bit *output* PWM dengan fungsi *analogWrite()*.
- d SPI : 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mensupport komunikasi SPI, yang mana masih mendukung *hardware*, yang tidak termasuk pada bahasa arduino
- e LED : 13. Ini adalah dibuat untuk koneksi LED ke *digital pin* 13. Ketika *pin* bernilai *HIGH*, LED hidup, ketika *pin* *LOW*, LED mati

### 2.6.3 Komunikasi

Arduino UNO memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lain. ATmega328 ini menyediakan UART TTL (5 Volt) komunikasi *serial*, yang tersedia pada *pin digital* 0 (RX) dan 1 (TX). *Firmware* Arduino menggunakan USB *driver* standar COM, dan tidak ada *driver* eksternal yang dibutuhkan. Namun, pada *Windows*, file Ini diperlukan. Perangkat lunak Arduino termasuk monitor serial yang memungkinkan data sederhana yang akan dikirim ke *board* Arduino. RX dan TX LED di *board* akan berkedip ketika data sedang dikirim melalui *chip* USB-to-serial dan koneksi USB ke komputer.

### 2.6.4 Software Arduino

Arduino Uno dapat diprogram dengan perangkat lunak Arduino. Pada ATmega328 di Arduino terdapat *bootloader* yang memungkinkan Anda untuk meng-*upload* kode baru untuk itu tanpa menggunakan *programmer hardware eksternal*. IDE Arduino adalah *software* yang sangat canggih ditulis dengan menggunakan *Java*. IDE Arduino terdiri dari:

- a. *Editor* : Program sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *Processing*.
- b. *Compiler* : Sebuah modul yang mengubah kode program (bahasa *Processing*) menjadi kode *biner*. Bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa *Processing*. Yang bisa dipahami oleh mikrokontroler adalah kode *biner*. Itulah sebabnya *compiler* diperlukan dalam hal ini.
- c. *Uploader* : Sebuah modul yang memuat kode *biner* dari komputer ke dalam memori didalam papan Arduino.

Sebuah kode program Arduino umumnya disebut dengan istilah *sketch*. Kata “*sketch*” digunakan secara bergantian dengan “kode program” dimana keduanya memiliki arti yang sama.



**Gambar 2.8** Tampilan IDE Arduino dengan Sebuah *Sketch*

## 2.7 Rem Magnetik [7]

Rem adalah suatu alat yang digunakan untuk melakukan aksi diselerasi yang akan menurunkan kecepatan dalam selang waktu yang ditentukan. Tipe rem yang umumnya digunakan adalah rem yang menggunakan gaya gesek untuk memberikan gaya lawan terhadap gaya gerak. Model yang digunakan bisa berupa piringan cakram ataupun berupa drum (tromol). Rem tipe ini diperlukan apabila diinginkan penurunan kecepatan hingga mencapai nol (atau terkunci).

Namun selain memiliki kelebihan tersebut, rem tipe ini juga memiliki kekurangan antara lain panas yang timbul bias berlebihan sehingga mengurangi gaya gesek yang terjadi, dan juga karena memanfaatkan gaya gesek maka bagian yang saling bergesekan menuntut pemeliharaan dan perawatan berkala. Ada tipe rem lain yang tidak memanfaatkan gesekan dua permukaan untuk menghasilkan gaya lawan terhadap gaya penyebab gerak, yaitu rem yang menggunakan gaya magnet untuk menimbulkan gaya lawan. Rem ini disebut Rem Arus *Eddy* (Rem Magnetik).

Prinsip dasar rem magnetik ini menggunakan hukum *Faraday* dan hukum *Lenz* yang terkenal didunia elektromagnetik. Kedua hukum ini menimbulkan arus *eddy* yang melingkar dan menginduksi medan magnet yang melawan medan magnet penyebabnya. Hukum – hukum ini berlaku bila ada permukaan yang memotong medan magnet, dengan artian gaya lawan hanya dihasilkan apabila permukaan tersebut memiliki kecepatan. Semakin tinggi kecepatan maka gaya lawan yang dihasilkan semakin besar. Namun semakin rendah kecepatan maka gaya lawan semakin kecil. Dengan melihat

teori yang berlaku maka rem magnetik ini akan optimal untuk memberikan penurunan kecepatan, bukan untuk menghentikan gerak suatu objek. Rem magnetik sering diaplikasikan untuk sistem pengereman pada *roller coaster*, kereta api dan juga digunakan pada alat dinamometer untuk pengukuran torsi suatu motor.

## 2.8 Pemodelan Sistem[8]

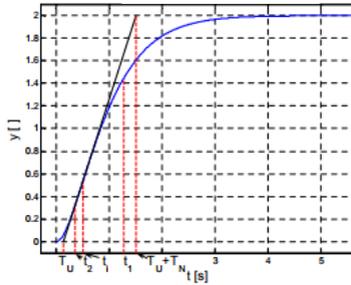
Suatu plant merupakan suatu sistem yang mungkin disusun dari bermacam-macam komponen/elemen yang saling berhubungan dalam menjalankan suatu aksi. Tergantung dari komponen/elemen yang dipergunakan, suatu sistem dapat berupa mekanik, pneumatic, elektrik atau jenis elektro-mekanik. Motor DC adalah termasuk dalam sistem komponen elektro-mekanik. Operasi Motor DC dengan kontrol armature menghasilkan rasio steady state kecepatan yang hampir linier terhadap tegangan masukannya dan arah putaran Motor DC tergantung pada polaritas tegangan masukan. Suatu prasyarat dasar pada kebanyakan strategi yang dipergunakan dalam membangun suatu sistem kontrol adalah kemampuan dalam memodelkan secara matematika dari plant yaitu sistem yang akan dikontrol. Model matematika dari suatu plant dapat berupa persamaan differensial, transfer function atau state space. Persamaan differensial menjelaskan suatu performa dinamik dari suatu system. Model ini dapat termasuk turunan (*derivative*) sekian orde dari masukan dan keluaran. Parameter sistem muncul dalam persamaan model matematika sebagai koefisien. Pada prinsipnya terdapat dua macam pemodelan matematika yang dapat dipilih, yaitu berdasarkan pada teori pengetahuan dengan menggunakan hukum fisika atau dengan proses eksperimen yaitu melakukan pengukuran. Pada kebanyakan kasus adalah tidak mungkin untuk membuat model yang sempurna hanya dengan menggunakan pengetahuan fisika saja. Beberapa parameter harus ditentukan dari suatu eksperimen. Pendekatan ini disebut dengan Identifikasi sistem.

### 2.8.1 Metode Strejc's [9]

Metode Strejc melakukan pendekatan pada sistem yang tidak stabil tanpa ada penundaan waktu dengan bantuan perhitungan waktu  $T_U$  dan  $T_N$  dengan membuat garis singgung yang memotong/ bersinggungan dengan respon sistem. Rasio waktu ini memberi parameter  $\tau$ .

$$\tau = \frac{T_U}{T_N} \dots \dots \dots (2.1)$$

### Step-response



**Gambar 2.9** Step respon dari Sistem dengan Ditunjukkan pada  $t_{1,2}, t_i, T_U$  dan  $T_N$

Berdasarkan pada nilai dari  $\tau$  perkiraan pendekatan dapat ditentukan dengan:

- a. Dalam kasus:  $\tau < 0$  sistemnya didekati dengan model urutan kedua :

$$G_{ST1(s)} = \frac{K}{(\tau_{ST1} s + 1)(\tau_{ST2} s + 1)} \dots \dots \dots (2.2)$$

Konstanta  $\tau_{ST1}$  dan  $\tau_{ST2}$  ditentukan sebagai berikut :

Tentukan waktu  $t_1$  terkait dengan nilai  $0,72y^\infty$  pada grafik sistem step-respons, tentukan jumlah konstanta  $\tau_{ST1}$  dan  $\tau_{st2}$  sesuai Persamaan :

$$\tau_{ST1} + \tau_{ST2} = \frac{t_1}{1,2564} \dots \dots \dots (2.3)$$

Tentukan waktu  $t_2$  :

$$t_2 = 0,3574(\tau_{ST1} + \tau_{ST2}) \dots \dots \dots (2.4)$$

Tentukan nilai  $y(t_2)$  pada grafik step-respons system

Tentukan rasio T pada **Tabel 2.3**

**Tabel 2.3** Nilai dari Konstan T

$y(t_2)$	T	$y(t_2)$	T
0,30	0,000	0,22	0,183
0,29	0,023	0,21	0,219
0,28	0,043	0,20	0,264
0,27	0,063	0,19	0,322
0,26	0,084	0,18	0,403
0,25	0,105	0,17	0,538

$y(t_2)$	T	$y(t_2)$	T
0,24	0,128	0,16	1,000
0,23	0,154		

$$T = \frac{\tau_{ST1}}{\tau_{ST2}} \dots\dots\dots(2.5)$$

Tentukan nilai yang tidak diketahui  $\tau_{ST1}$  dan  $\tau_{ST2}$  dari rasio (2.4) dan jumlah (2.5)

b. Dalam kasus:  $\tau \geq 0$  sistem didekati dengan model urutan ke-n:

$$G_{ST2}(s) = \frac{K}{\tau_{ST}(s+1)^n} \dots\dots\dots(2.6)$$

**Tabel 2.4** Penilaian Faktor n dan Ketepatan Koordinat Titik Infleksi

n	2	3	4	5	6	7	8	9	10
$\tau$	0,104	0,218	0,319	0,41	0,493	0,57	0,642	0,709	0,773
$y_i$	0,264	0,327	0,359	0,371	0,384	0,394	0,401	0,407	0,413

- Rancanglah sebuah garis singgung, yang memotong titik infleksi, temukan nilai  $T_U$  dan  $T_N$ , tentukan rasio  $\tau$  :  

$$\tau = \frac{T_U}{T_N} \dots\dots\dots(2.7)$$
- Menurut nilai  $\tau$  cari urutan yang sesuai dari perkiraan model dan koordinat titik koordinat  $y_i$
- Berdasarkan koordinat yang telah dinilai sebelumnya, temukan di grafik step-respons system
- Tentukan konstanta waktu konstan dari Persamaan(2.8):  

$$\tau_{ST} = \frac{t_1}{n-1} \dots\dots\dots(2.8)$$

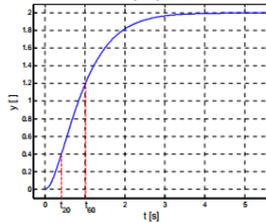
### 2.8.2 Metode Smith's [9]

Metode Smith, berdasarkan dua titik respon fraksional sistem pada 20% dan 60% dari nilai akhir kondisi akhirnya, mendekati sistem dengan model orde 2 dengan kemungkinan waktu inklusi waktu. Metode Smith memerlukan waktu dimana nilai normal respon masing-masing mencapai

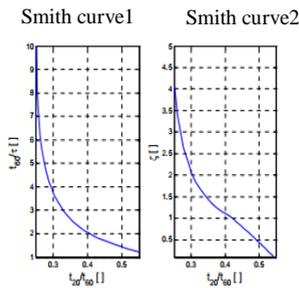
20% dan 60%. Dengan menggunakan rasio  $t_{20} / t_{60}$  memberi nilai  $\zeta$ . Perkiraan  $\tau$  dapat diperoleh dari plot  $t_{60} / \tau$  vs  $t_{20} / t_{60}$ .

$$G_{SM}(s) = \frac{K}{(\tau_{SM1} s + 1)(\tau_{SM2} s + 1)} \dots\dots\dots(2.9)$$

Step-response



**Gambar 2.10** Step-respon Sistem Ditunjukkan dengan Waktu  $t_{20}$  dan  $t_{60}$



**Gambar 2.11** Kurva untuk Penilaian  $\tau$  dan  $\zeta$

$$\tau_{SM1} = \tau \zeta 2 + \tau \sqrt{(\zeta 2 - 1)} \dots\dots\dots(2.10)$$

$$\tau_{SM2} = \tau \zeta 2 + \tau \sqrt{(\zeta 2 - 1)} \dots\dots\dots(2.11)$$

## BAB III PERANCANGAN SISTEM

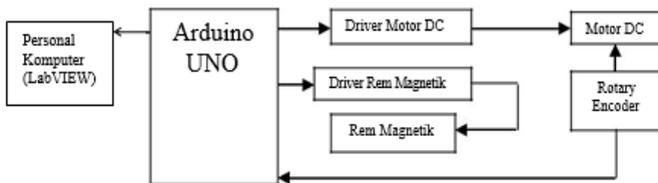
Dalam bab ini akan dibahas mengenai perancangan alat yang meliputi perencanaan perangkat keras (*hardware*) dan perangkat lunak (*software*). Pada Tugas Akhir ini melanjutkan alat dari Tugas Akhir Catur Andianto dan Amanda Dwi P. Hal tersebut guna mewujudkan tugas akhir ini yang berjudul “Pemodelan Kecepatan Motor DC menggunakan Identifikasi dengan Metode 2S”. Perancangan alat akan dibahas perbagian yang disertai dengan gambar skematik.

Untuk memudahkan dalam pembahasan bab ini akan dibagi menjadi 2 yaitu:

1. Perancangan perangkat keras yang terdiri dari perancangan mekanik dan elektrik,  
Perancangan mekanik meliputi:
  - a. Motor DC
  - b. Rem Magnetik
  - c. *Rotary Encoder*Perancangan elektrik meliputi:
  - a. Arduino UNO
2. Perancangan perangkat lunak dengan *software* Arduino UNO yang menggunakan bahasa Arduino UNO , *software* LabVIEW.

### 3.1 Blok Fungsional Sistem

Pada bab ini dibahas mengenai perancangan alat secara keseluruhan. Dijelaskan pula lebih terinci tiap-tiap bagian atau blok-blok penyusun alat ini berupa blok fungsional sistem. Dapat dilihat pada **Gambar 3.1**.



**Gambar 3.1.** Blok Fungsional Sistem

Perancangan sistem dalam pembuatan alat ini secara garis besar disertai urutan dan cara kerja alat ini di ilustrasikan pada **Gambar 3.1**. Dari Gambar

tersebut dapat dilihat bahwa sistem tersebut terdiri dari beberapa blok fungsional yaitu:

1. *Driver* Motor DC, berfungsi sebagai pengendali Motor DC.
2. Motor DC, berfungsi sebagai objek pada tugas akhir ini.
3. Arduino UNO, berfungsi untuk sebagai kontroler yang mengontrol kecepatan Motor DC dengan memberikan signal PWM (*Pulse With Modulation*).
4. *Driver* Rem Magnetik, berfungsi sebagai pengaturan pengereman pada rem magnetik.
5. *Rotary Encoder*, berfungsi sebagai sensor kecepatan pada Motor DC.
6. Personal Komputer, berfungsi sebagai tampilan pengaturan kecepatan melalui *software* LabVIEWW.

### 3.2 Perancangan Perangkat Keras

Pada perancangan perangkat keras ini, prosesnya dibagi menjadi dua bagian, yaitu perancangan mekanik dan elektrik. Masing-masing perancangan tersebut selanjutnya akan dibahas lebih mendalam pada sub bab berikutnya.

#### 3.2.1 Perancangan Mekanik

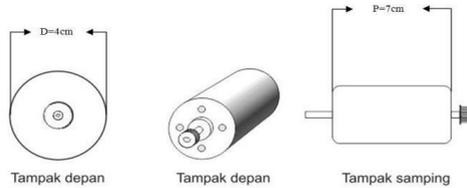
Pada perancangan mekanik ini terdiri dari Motor DC, rem magnetik, dan *rotary encoder*.

##### a. Motor DC

Pada Motor DC digunakan sebagai penggerak utama untuk menanggung beban rem magnetik. Salah satu faktor pemilihan dari Motor DC yang nantinya akan dipakai yaitu kecepatan putar yang mampu menggerakkan beban agar gerakan yang didapat dapat seimbang dengan baik. Spesifikasi Motor DC yang digunakan dapat dilihat pada **Tabel 3.1** dan **Gambar 3.2**.

**Tabel 3.1** Spesifikasi Motor DC

No.	Spesifikasi	Nilai
1	Tegangan	24 Volt
2	Arus	1 Ampere
3	Kecepatan putar	5000rpm



**Gambar 3.2** Motor DC 24 Volt

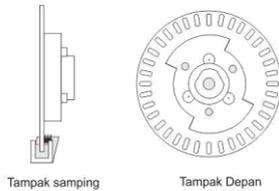
#### b. Rem Magnetik

Piringan besi dihubungkan dengan poros utama. Piringan ini diapit oleh dua penghasil medan magnet. Sisi penghasil medan magnet utara dan sisi penghasil medan magnet selatan. Saat poros utama berputar maka piringan besi ikut berputar. Pada saat kecepatan putar tinggi, medan magnet diberikan, sehingga akan menimbulkan arus *eddy* yang menimbulkan gaya lawan yang melawan arah putar piringan besi. Gaya lawan ini yang disebut sebagai beban, yang harus ditanggung Motor DC.

Pada sisi penghasil medan magnet utara digunakan 8 buah kumparan yang bila dialiri arus listrik akan menimbulkan medan magnet. Begitu juga pada sisi penghasil medan magnet selatan yang menggunakan 8 buah kumparan. Semua kumparan identik baik jumlah lilitannya maupun arah lilitannya. Kemudian semua kumparan dihubung seri. Sumber tenaga yang digunakan untuk rem magnetik ini sudah terkonfigurasi sama dengan sumber yang digunakan pada Motor DC. Besar pengaturan tegangan yang masuk pada rangkaian rem magnet akan mempengaruhi banyak arus yang mengalir dan mempengaruhi besar medan magnet yang dihasilkan, yang berarti juga mempengaruhi gaya lawan yang dihasilkan.

#### c. Rotary Encoder

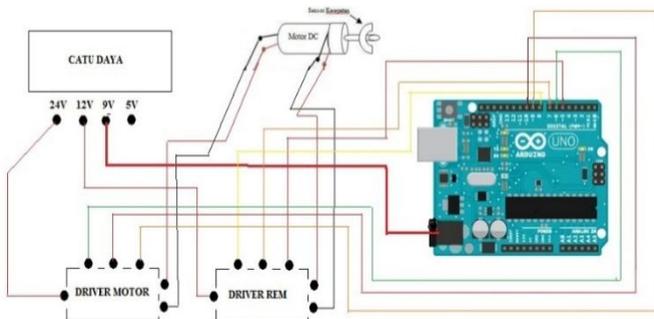
*Rotary encoder* digunakan untuk menentukan banyaknya putaran poros tiap menit (rpm). *Encoder* ini akan menghasilkan gelombang kotak yang frekuensinya akan bertambah bila kecepatan putar poros bertambah. *Encoder* ini diletakkan pada poros utama.



**Gambar 3.3** Model *Rotary Encoder*

### 3.2.2 Perancangan Elektrik

Masukan yang diperlukan Motor DC untuk menghasilkan torsi didapat dari jala-jala PLN berupa sinyal AC 220Volt 50Hz yang kemudian disearahkan dengan rangkaian *full wave rectifier*. Caranya adalah dengan mengatur waktu penyalan *rectifier* sehingga tegangan rata-rata yang masuk ke Motor DC dapat bervariasi. Untuk mengatur waktu penyalan maka diperlukan sinyal PWM (*Pulse Width Modulation*) yang tersinkronisasi dengan jala-jala PLN dengan frekuensi 50Hz. Sinyal PWM tersinkronisasi ini yang kemudian memicu waktu penyalan *rectifier*.



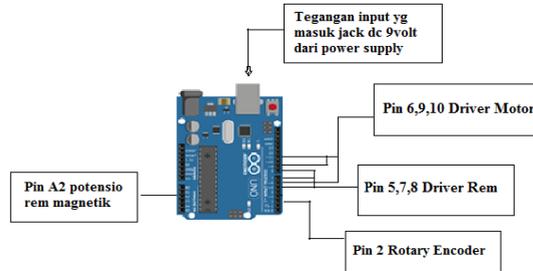
**Gambar 3.4** Perancangan Komponen Elektrik

#### 3.2.2.1 Arduino UNO

Dalam perancangan perangkat keras terdapat rangkaian Arduino UNO. Arduino UNO berfungsi untuk sebagai kontroler yang mengontrol kecepatan Motor DC dengan memberikan sinyal PWM (*Pulse With Modulation*). Spesifikasi pada arduino UNO ini ialah hanya menerima *inputan* sekitar 0-5 Volt, jika lebih dari 5 Volt maka Arduino UNO tersebut akan rusak.

Untuk penggunaan kaki - kaki pada pin Arduino UNO yaitu yang pertama pada pin A2 digunakan untuk potensio pada rem magnetik sebagai pengatur beban, yang kedua pada *pin* 5, 7, dan 8 sebagai *driver* rem,

kemudian pada *pin* 6, 10 dan 9 digunakan untuk *driver* Motor, *pin* 2 digunakan sebagai *rotary encoder*, sebagai PWM digunakan *pin* 9 dan sumber *input* nya masuk melalui *jack* DC dengan tegangan *input* 9 Volt dari *power supply*. Rangkaian mikrokontroler Arduino UNO dapat dilihat pada **Gambar 3.5**.



**Gambar 3.5.** Rangkaian Arduino UNO

### 3.3 Perancangan Perangkat Lunak

Dalam pembuatan perangkat lunak ada beberapa program yang harus dibuat agar sistem monitoring dapat berjalan dengan baik. Tahapan pembuatan tersebut adalah sebagai berikut:

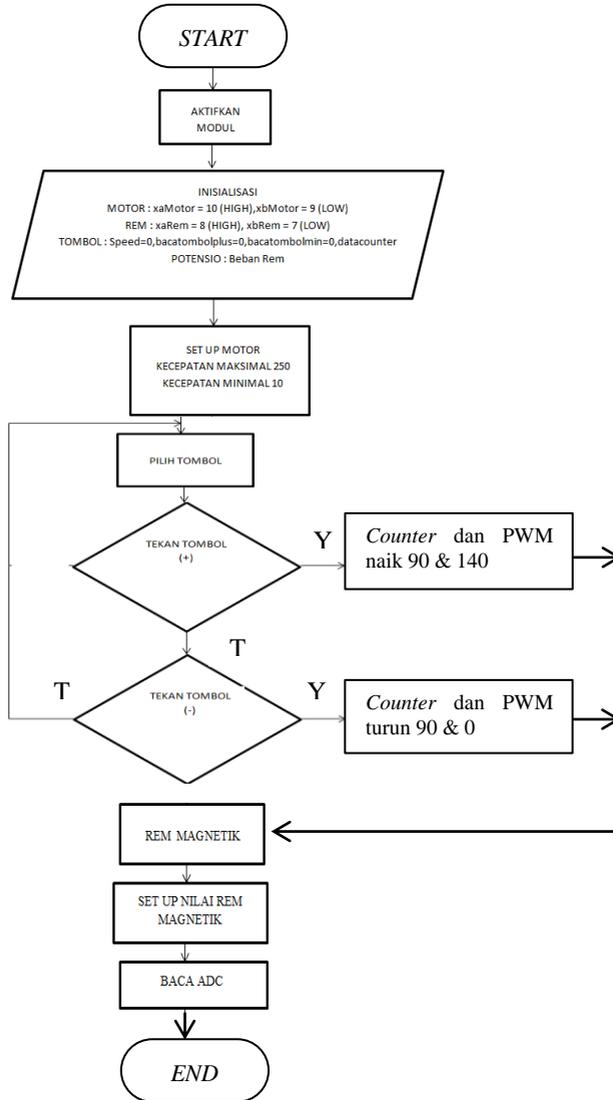
#### 3.3.1 Pembuatan *Flowchart* Program

*Flowchart* merupakan bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya. Bagan ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan hubungan antar proses digambarkan dengan garis penghubung. *Flowchart* merupakan langkah awal pembuatan program. Dengan adanya *flowchart* urutan poses kegiatan menjadi lebih jelas.

Untuk pengolahan data dengan komputer, dapat dirangkum urutan dasar untuk pemecahan suatu masalah, yaitu;

- a. *START* : Berisi instruksi untuk persiapan peralatan yang diperlukan sebelum menangani pemecahan masalah.
- b. *READ* : Berisi instruksi untuk membaca data dari suatu peralatan.
- c. *PROCESS*: Berisi kegiatan yang berkaitan dengan pemecahan persoalan sesuai dengan data yang dibaca.
- d. *WRITE* : Berisi instruksi untuk merekam hasil kegiatan ke peralatan *output*.
- e. *END* : Mengakhiri kegiatan pengolahan.

*Flowchart* program dari Tugas Akhir ini meliputi seluruh sistem jalannya alat ini. Sistem yang dimaksud adalah sistem umum secara keseluruhan. Berikut merupakan *flowchart* program *driver* Motor dan rem magnetik , ditunjukkan pada **Gambar 3.6**.



**Gambar 3.6.** Flowchart Sistem

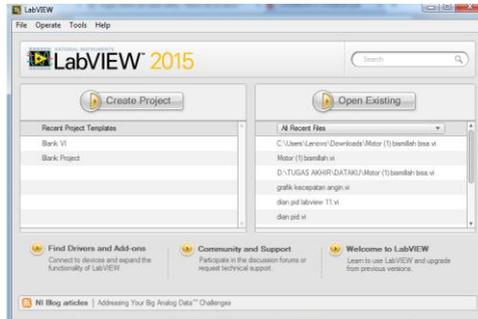
Dari *flowchart* diatas dapat dijelaskan alur kerja modul kecepatan Motor yaitu Pada saat modul diaktifkan, langsung terjadi inisialisasi Motor, rem, tombol, dan potensiometer. Lalu untuk *set up* Motor kecepatan maksimal 140 dan kecepatan minimum 90. Untuk mengambil data pada Motor pilih tombol (+) atau (-). Jika pilih tombol (+) kecepatan harus <140, maka *counter* dan PWM akan bertambah 90 untuk kecepatan Motor. Jika pilih tombol (-) kecepatan harus <0 maka *counter* dan PWM akan turun menjadi 90 untuk kecepatan Motor. Pada beban Motor digunakan rem magnetik dengan cara *set up* nilai rem pada potensiometer kemudian arduino membaca nilai ADC yg nanti di *convert* ke nilai digital pada monitor untuk mengetahui berapa persen beban rem.

### 3.3.2 Pemrograman Software LabVIEW

LabVIEW adalah suatu bahasa pemrograman yang menggunakan berbagai macam ikon yang merepresentasikan suatu instruksi. Jika bahasa pemrograman *text based* mengsekusi instruksi sesuai dengan urutan yang ditulis, LabVIEW menggunakan metode data *flow programming* dimana alur data melalui berbagai ikon akan menentukan urutan eksekusi dari setiap instruksi. Dalam LabVIEW, VI adalah program yang menyerupai instrumen yang sesungguhnya. Karena fleksibilitasnya sifat yang modular, dan kemudian programnya.

LabVIEW juga membantu teknisi untuk semakin mudah dalam mengaplikasikan sistem *programmable logic* kontroler dengan cara penggabungan PC pada aplikasi mereka dengan bantuan perlengkapan HMI, atau SCADA. Dengan LabVIEW teknisi dapat memprogram HMI dan *logic* pada daerah yang sama sehingga dapat meminimalkan biaya pembuatan dan waktu pembelajaran dan dapat memaksimalkan ketrampilan pemrograman. Perancangan Program Kontrol Kecepatan pada Labview:

- a. Klik program LabVIEW
- b. Pada LabVIEW akan muncul *dialog box* seperti yang ditunjukkan pada **Gambar 3.7**, kemudian klik *Create Project*.



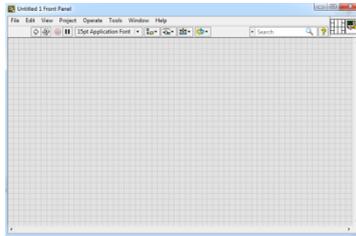
**Gambar 3.7.** Dialog Box Utama LabVIEW

- c. Setelah itu akan muncul *dialog box* seperti pada **Gambar 3.8.** Lalu klik *Blank VI*.

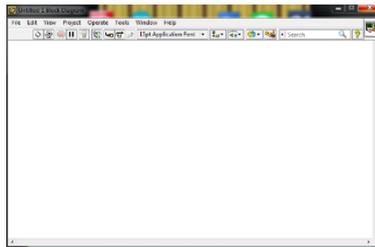


**Gambar 3.8** Dialog Box LabVIEW

- d. Maka akan muncul tampilan *front panel* dan *block diagram* seperti pada **Gambar 3.9** dan **Gambar 3.10**
- e. *Front Panel* dari VI : *User Interface* atau *front panel*, adalah bagian *window* yang berlatar belakang abu-abu serta mengandung *controls* dan *indicators*.
- f. *Block Diagram* dari VI : *Block Diagram* adalah bagian *window* yang berlatar belakang putih berisi *source code* yang dibuat dan berfungsi sebagai intruksi untuk *front panel*.



**Gambar 3.9** *Front Panel*



**Gambar 3.10** *Block Diagram*

- g. Kemudian pada *Block Diagram* klik *view* dan pilih *function palette*. Lalu pilih *item* mana saja yang akan digunakan.
- h. Setelah membuat program pada LabVIEW, tentukan *port* yang akan dihubungkan dengan mikrokontroler.
- i. Maka, hasil tampilan monitor pada *front panel* tampak pada **Gambar 3.11**.



**Gambar 3.11** Tampilan Utama LabVIEW



*(halaman ini sengaja dikosongkan)*

## **BAB IV**

### **PENGUJIAN DAN ANALISA**

Pada Bab ini dibahas tentang pengujian dan analisa sistem yang telah dibuat. Pengujian dan analisa meliputi pengujian perangkat keras sistem dan pengujian perangkat lunak sistem.

#### **4.1 Gambaran Umum Pengujian Sistem**

Pada pengerjaan Tugas Akhir ini, akan dilakukan beberapa pengujian pada sistem dengan beberapa kondisi kecepatan tegangan, *counter* dan rem. Sebelumnya, akan terlebih dahulu dilakukan pengujian terhadap beberapa komponen sistem. Tahapan selanjutnya adalah melakukan monitoring hasil grafik dari beberapa kondisi diatas pada LabVIEW dan juga pemodelan matematika 2S.

#### **4.2 Pengujian Perangkat Keras**

Pengujian perangkat keras bertujuan untuk mengetahui perangkat keras yang dirancang telah berfungsi baik, dan mengetahui performa dari perangkat tersebut. Pengujian tersebut meliputi pengujian Motor DC, rem magnetik, tombol *counter*.

##### **4.2.1 Pengujian Keseluruhan Sistem**

Pengujian keseluruhan sistem ini menggunakan modul kecepatan Motor DC dan Motor DC sebagai objek utama pada alat ini. Saat modul telah aktif, tekan tombol plus(+) untuk menambah kecepatan pada motor dan tombol minus(-) untuk mengurangi kecepatan Motor. *Rotary encoder* digunakan sebagai sensor kecepatan Motor DC, saat pengambilan data dilakukan pada beberapa kondisi rem magnetik sebagai beban pada Motor yang dirubah-rubah mulai dari tanpa rem dan dengan rem 25% 50% 75% dan 90% untuk mengetahui respon grafik untuk mencari nilai linier dari Motor untuk pemodelan. Berikut tabel hasil pengujian dan grafik respon pada Motor.

##### **4.2.2 Pengujian Motor Tanpa Rem**

Pengujian ini dilakukan untuk mengetahui kecepatan normal Motor DC tanpa rem yang digunakan pada sistem. Pengujian dilakukan dengan mengambil data dari beberapa kondisi pada *counter*, dan kecepatan. Motor mulai aktif pada *counter* 40 dengan kecepatan 156rpm sampai 4991rpm. Data dapat dilihat pada **Tabel.4.1**.

**Tabel 4.1** Data Pengujian Motor Tanpa Rem

No.	Counter (pwm)	Tegangan (Volt)	Kecepatan (rpm)
1	10	1	0
2	20	1,8	0
3	30	2	0
4	40	3	156
5	50	4	191
6	60	5	820
7	70	6	1087
8	80	7	2028
9	90	8	2443
10	100	9	2905
11	110	10	3197
12	120	11	3392
13	130	12	3572
14	140	13	3789
15	150	14	3824
16	160	15	4029
17	170	16	4220
18	180	16,9	4282
19	190	17	4309
20	200	18	4430
21	210	19	4510
22	220	20	4649
23	230	21	4836
24	240	22	4950
25	250	23,5	4991

### 4.2.3 Pengujian Motor dengan Rem 25%

Pengujian ini dilakukan untuk mengetahui tingkat beban mulai dari terendah sampai tertinggi yang digunakan pada sistem. Pengujian ini dilakukan untuk mencari titik kerja atau garis linier pada kerja Motor. Pengujian dilakukan dengan mengambil data dari beberapa kondisi pada *counter*, dan kecepatan dan rem. Pada kecepatan normal terlihat jelas kecepatan Motor menurun 12 sampai 15 rpm saat diberi rem 25% .Data dapat dilihat pada **Tabel.4.2**.

**Tabel 4.2** Data Pengujian Motor dengan Rem 25%

No.	Counter (pwm)	Tegangan (Volt)	Kecepatan (rpm)
1	10	1	0
2	20	1,8	0
3	30	2	0
4	40	3	145
5	50	4	179
6	60	5	807
7	70	6	1076
8	80	7	2016
9	90	8	2430
10	100	9	2892
11	110	10	3187
12	120	11	3381
13	130	12	3561
14	140	13	3776
15	150	14	3812
16	160	15	4018
17	170	16	4208
18	180	16,9	4271
19	190	17	4296
20	200	18	4418
21	210	19	4499
22	220	20	4436
23	230	21	4824
24	240	22	4939
25	250	23,5	4979

#### 4.2.4 Pengujian Motor Dengan Rem 50%

Pengujian ini dilakukan untuk mengetahui tingkat beban mulai dari terendah sampai tertinggi yang digunakan pada sistem. Pengujian ini dilakukan untuk mencari titik kerja atau garis linier pada kerja Motor. Pengujian dilakukan dengan mengambil data dari beberapa kondisi pada *counter* dan kecepatan dan rem. Terlihat bahwa Motor mulai aktif saat *counter* 60 dengan kecepatan 721rpm, kecepatan menurun hingga 150rpm. Karena saat rem 50% *counter* 40 dan 50 Motor sudah tidak aktif. Data dapat dilihat pada Tabel.4.3.

**Tabel 4.3** Data Pengujian Motor dengan Rem 50%

No.	Counter (pwm)	Tegangan (Volt)	Kecepatan (rpm)
1	10	1	0
2	20	1,8	0
3	30	2	0
4	40	3	0
5	50	4	0
6	60	5	721
7	70	6	998
8	80	7	1939
9	90	8	2353
10	100	9	2815
11	110	10	3107
12	120	11	3303
13	130	12	3483
14	140	13	3699
15	150	14	3735
16	160	15	3939
17	170	16	4131
18	180	16,9	4193
19	190	17	4219
20	200	18	4341
21	210	19	4422
22	220	20	4559
23	230	21	4747
24	240	22	4861
25	250	23,5	4901

#### 4.2.5 Pengujian Motor Dengan Rem 75%

Pengujian ini dilakukan untuk mengetahui tingkat beban mulai dari terendah sampai tertinggi yang digunakan pada sistem. Pengujian ini dilakukan untuk mencari titik kerja atau garis linier pada kerja Motor. Pengujian dilakukan dengan mengambil data dari beberapa kondisi pada *counter*, dan kecepatan dan rem. Pada saat rem 75% kecepatan menurun hingga 300rpm dan Motor masih aktif saat *counter* 60 Data dapat dilihat pada **Tabel.4.4**

**Tabel 4.4** Data Pengujian Motor dengan Rem 75%

No.	Counter (pwm)	Tegangan (Volt)	Kecepatan (rpm)
1	10	1	0
2	20	1,8	0
3	30	2	0
4	40	3	0
5	50	4	0
6	60	5	179
7	70	6	876
8	80	7	1608
9	90	8	1730
10	100	9	2092
11	110	10	2517
12	120	11	2681
13	130	12	2751
14	140	13	3039
15	150	14	3164
16	160	15	3489
17	170	16	3549
18	180	16,9	3700
19	190	17	3809
20	200	18	4010
21	210	19	4290
22	220	20	4310
23	230	21	4506
24	240	22	4620
25	250	23,5	4656

#### 4.2.6 Pengujian Motor Dengan Rem 90%

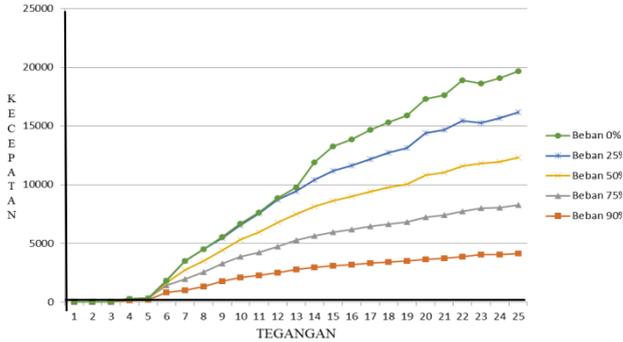
Pengujian ini dilakukan untuk mengetahui tingkat beban mulai dari terendah sampai tertinggi yang digunakan pada sistem. Pengujian ini dilakukan untuk mencari titik kerja atau garis linier pada kerja Motor. Pengujian dilakukan dengan mengambil data dari beberapa kondisi pada *counter*, dan kecepatan dan rem. Pada saat rem 90% kecepatan Motor aktif mulai *counter* 90 dengan kecepatan 74rpm, dari kecepatan normal terlihat kecepatan menurun hingga 500rpm saat diberi rem 90%. Data dapat dilihat pada **Tabel.4.5**

**Tabel 4.5** Data Pengujian Motor dengan Rem 90%

No.	Counter (pwm)	Tegangan (Volt)	Kecepatan (rpm)
1	10	1	0
2	20	1,8	0
3	30	2	0
4	40	3	0
5	50	4	0
6	60	5	0
7	70	6	0
8	80	7	0
9	90	8	74
10	100	9	96
11	110	10	1016
12	120	11	1042
13	130	12	1079
14	140	13	1289
15	150	14	2263
16	160	15	3029
17	170	16	3220
18	180	16,9	3451
19	190	17	3560
20	200	18	3726
21	210	19	3720
22	220	20	4149
23	230	21	4216
24	240	22	4201
25	250	23,5	4339

#### 4.2.7 Hasil Keseluruhan Pengujian

Pada hasil keseluruhan pengujian ini terlihat semakin besar beban rem magnetik yang diberikan pada Motor maka semakin rendah pula kecepatan pada Motor. Untuk grafik dapat dilihat pada **Gambar 4.1**



**Gambar 4.1** Grafik Hasil Keseluruhan Pengujian

#### 4.2.8. Pengujian pada Titik Kerja Motor

Pada pengujian ini didapatkan daerah linier atau titik kerja Motor pada 8-13 Volt pada grafik hasil pengujian keseluruhan, dapat dilihat pada Gambar 4.6. Pada pengujian ini dilakukan saat Motor dengan tegangan 8 Volt dilanjutkan dengan 13 Volt yg masing masing dalam keadaan *steady* selama 10 detik dengan rem 0% hingga 90% untuk mendapatkan grafik hasil respon .

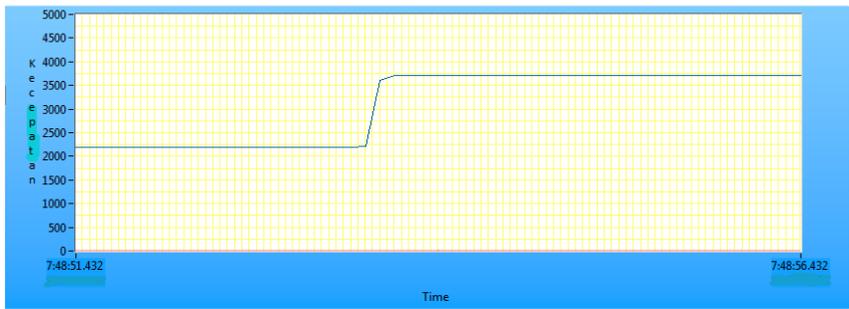
##### 4.2.8.1 Pengujian Tanpa Rem

Pengujian ini dilakukan untuk mengetahui kecepatan normal Motor DC yang digunakan pada sistem. Pengujian dilakukan dengan mengambil data pengukuran pada titik kerja Motor yaitu 8-13Volt.

**Tabel 4.6** Data Pengujian Tanpa Rem

Counter (pwm)	Tegangan (Volt)	Rem (%)	Kecepatan (rpm)
90	8,470588	0	1756
90	8,470588	0	1756
90	8,470588	0	2246
90	8,470588	0	2246
90	8,470588	0	2260
90	8,470588	0	2260
90	8,470588	0	2268
90	8,470588	0	2268

Counter (pwm)	Tegangan (Volt)	Rem (%)	Kecepatan (rpm)
90	8,470588	0	2261
90	8,470588	0	2261
140	13,176471	0	3498
140	13,176471	0	3498
140	13,176471	0	3753
140	13,176471	0	3758
140	13,176471	0	3758
140	13,176471	0	3758
140	13,176471	0	3772
140	13,176471	0	3772
140	13,176471	0	3761
140	13,176471	0	3761



**Gambar 4.2** Hasil Respon Kecepatan Motor Tanpa Rem

Dari **Tabel 4.6** dan **Gambar 4.2** dapat dilihat bahwa pada saat tegangan 8Volt Motor memiliki kecepatan rata rata sebesar 2260rpm, setelah keadaan sudah *steady* Motor naik pada tegangan 13Volt dengan kecepatan rata rata sebesar 3758rpm. Kemudian dari **Gambar 4.2** dapat dicari pemodelan matematika dengan cara identifikasi statis. Dari grafik **Gambar 4.2** diperoleh data sebagai berikut ;

1. Strejc's

$$\tau = \frac{T_u}{T_N} \quad (4.1)$$

$T_N$  dapat dicari menggunakan rumus pada (4.2) :

$$\begin{aligned}\tau &= \frac{\tau \ln 2}{T_N} \\ T_N &= \ln 2 \\ T_N &= 0,69\end{aligned}\tag{4.2}$$

karena *time sampling* yang digunakan pada waktu pengambilan data adalah 10s, maka :

$$\begin{aligned}10 &= \tau \ln 9 \\ \tau &= 4,55\end{aligned}\tag{4.3}$$

$$\begin{aligned}td &= \tau \cdot \ln 2 \\ &= 4,55 \cdot \ln 2 \\ &= 3,15\end{aligned}$$

$$\tau_{ST} = 4,55\tag{4.4}$$

A (*Input*) = 2260 Rpm.

Oss (*Output steady state*) = 3758 Rpm

Perbandingan *output steady state* dan *input* :

$$K = \frac{3758}{2260} = 1,66\tag{4.5}$$

sehingga dari data di atas diperoleh model matematis *plant* sebagai berikut :

$$G_{(s)} = \frac{1,66}{4,55s+1}\tag{4.6}$$

2. Smith's

$$\tau = \frac{T_u}{T_N}\tag{4.7}$$

$T_N$  dapat dicari menggunakan rumus pada (4.8) :

$$\begin{aligned}\tau &= \frac{\tau \ln 2}{T_N} \\ T_N &= \ln 2\end{aligned}\tag{4.8}$$

karena nilai waktu yang digunakan pada waktu pengambilan data adalah 20% dan 60% pada kondisi akhirnya/*steady state* , maka :

$$\tau_{sm1}=5,24 \quad \text{dan} \quad \tau_{sm2}=2,76 \quad (4.9)$$

A (Input) = 2260 Rpm.

Oss(Output steady state) = 3758 Rpm

Perbandingan *output steady state* dan *input* :

$$K = \frac{3758}{2260} = 1,66 \quad (4.10)$$

sehingga dari data di atas diperoleh model matematis *plant* sebagai berikut

$$G(s) = \frac{1,66}{(5,24s+1)(2,76s+1)} \quad (4.11)$$

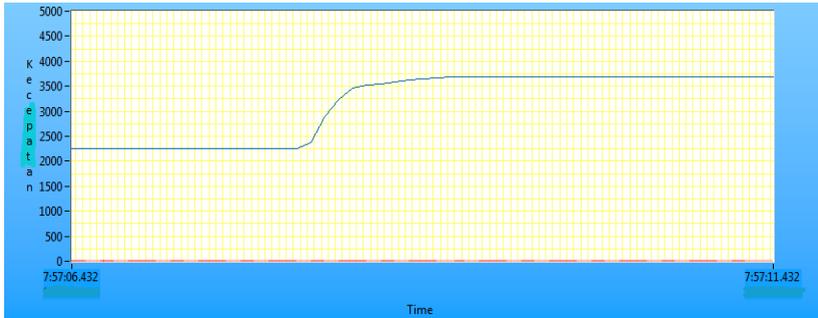
#### 4.2.8.2 Pengujian Dengan Rem 25%

Pengujian ini dilakukan untuk mengetahui kecepatan Motor DC dengan beban 25% yang digunakan pada sistem. Pengujian ini dilakukan dengan mengambil data pengukuran pada titik kerja Motor yaitu 8-13Volt.

**Tabel 4.7** Data Pengujian Dengan Rem 25%

Counter (pwm)	Tegangan (Volt)	Rem (%)	Kecepatan (rpm)
90	8,470588	24,8	1723
90	8,470588	24,39	1723
90	8,470588	25,61	2241
90	8,470588	24,8	2241
90	8,470588	24,8	2242
90	8,470588	24,8	2242
90	8,470588	23,98	2242
90	8,470588	23,17	2240
90	8,470588	23,17	2260
90	8,470588	23,17	2260
140	13,176471	23,98	3442
140	13,176471	23,58	3442
140	13,176471	24,8	3689
140	13,176471	24,8	3689

Counter (pwm)	Tegangan (Volt)	Rem (%)	Kecepatan (rpm)
140	13,176471	24,39	3708
140	13,176471	23,98	3708



**Gambar 4.3** Hasil Respon Kecepatan Motor Rem 25%

Dari **Tabel 4.7** dan **Gambar 4.3** dapat dilihat bahwa pada saat tegangan 8Volt Motor memiliki kecepatan rata rata sebesar 2240rpm, setelah keadaan sudah *steady* Motor naik pada tegangan 13Volt dengan kecepatan rata rata sebesar 3689rpm. Kemudian dari **Gambar 4.3** dapat dicari pemodelan matematika dengan cara identifikasi statis. Dari grafik **Gambar 4.3** diperoleh data sebagai berikut ;

1. Strej'c

$$\tau = \frac{T_u}{T_N} \quad (4.12)$$

$T_N$  dapat dicari menggunakan rumus pada (4.13) :

$$\tau = \frac{\tau \ln_2}{T_N}$$

$$T_N = \ln_2$$

$$T_N = 0,69 \quad (4.13)$$

karena *time sampling* yang digunakan pada waktu pengambilan data adalah 10s, maka :

$$10 = \tau \ln_9$$

$$\tau = 4,55$$

$$td = \tau \cdot \ln 2$$

$$= 4,55 \cdot \ln 2$$

$$= 3,15 \quad (4.14)$$

$$\tau_{ST} = 4,65 \quad (4.15)$$

A (*Input*) = 2240 Rpm.

Oss(*Output steady state*) = 3689 Rpm

Perbandingan *output steady state* dan *input* :

$$K = \frac{3689}{2240} = 1,64 \quad (4.16)$$

sehingga dari data di atas diperoleh model matematis plant sebagai berikut :

$$G_{(s)} = \frac{1,64}{4,65s+1} \quad (4.17)$$

2.Smith's

$$\tau = \frac{T_u}{T_N} \quad (4.18)$$

$T_N$  dapat dicari menggunakan rumus pada (4.19) :

$$\tau = \frac{\tau \ln_2}{T_N} \quad (4.19)$$

karena nilai waktu yang digunakan pada waktu pengambilan data adalah 20% dan 60% pada kondisi akhirnya/*steady state* , maka :

$$\tau_{sm1}=5,2 \quad \text{dan} \quad \tau_{sm2}=2,8 \quad (4.20)$$

A (*Input*) = 2240 Rpm.

Oss(*Output steady state*) = 3689 Rpm

Perbandingan *output steady state* dan *input* :

$$K = \frac{3758}{2260} = 1,64 \quad (4.21)$$

sehingga dari data di atas diperoleh model matematis *plant* sebagai berikut

$$G_{(s)} = \frac{1,64}{(0,5,2s+1)(2,8s+1)} \quad (4.22)$$

#### 4.2.8.3 Pengujian Dengan Rem 50%

Pengujian ini dilakukan untuk mengetahui kecepatan Motor DC dengan beban 50% yang digunakan pada sistem. Pengujian ini dilakukan dengan mengambil data pengukuran pada titik kerja Motor yaitu 8-13Volt.

**Tabel 4.8** Data Pengujian Dengan Rem 50%

<i>Counter</i> (pwm)	Tegangan (Volt)	Rem (%)	Kecepatan (rpm)
90	8,470588	53,66	2012
90	8,470588	53,25	2012
90	8,470588	53,25	2274
90	8,470588	53,66	2274
90	8,470588	52,85	2290
90	8,470588	51,22	2290
90	8,470588	50,81	2288
90	8,470588	53,25	2288
90	8,470588	52,44	2295
90	8,470588	50,81	2295
90	8,470588	49,19	2297
90	8,470588	53,66	2297
90	8,470588	54,88	2285
90	8,470588	53,25	2285
90	8,470588	53,66	2272
140	13,176471	53,25	3495
140	13,176471	52,85	3495
140	13,176471	52,03	3718
140	13,176471	51,22	3718
140	13,176471	55,69	3719
140	13,176471	53,25	3719
140	13,176471	52,03	3720
140	13,176471	51,22	3720
140	13,176471	53,25	3725

Counter (pwm)	Tegangan (Volt)	Rem (%)	Kecepatan (rpm)
140	13,176471	53,25	3725



**Gambar 4.4** Hasil Respon Kecepatan Motor Rem 50%

Dari **Tabel 4.8** dan **Gambar 4.4** dapat dilihat bahwa pada saat tegangan 8Volt Motor memiliki kecepatan rata rata sebesar 2290pm, setelah keadaan sudah *steady* Motor naik pada tegangan 13Volt dengan kecepatan rata rata sebesar 3720rpm. Kemudian dari **Gambar 4.4** dapat dicari pemodelan matematika dengan cara identifikasi statis. Dari grafik **Gambar 4.4** diperoleh data sebagai berikut ;

1. Strejc's

$$\tau = \frac{T_u}{T_N} \quad (4.23)$$

$T_N$  dapat dicari menggunakan rumus pada (4.24) :

$$\tau = \frac{\tau \ln 2}{T_N}$$

$$T_N = \ln 2$$

$$T_N = 0,69 \quad (4.24)$$

karena *time sampling* yang digunakan pada waktu pengambilan data adalah 10s, maka :

$$10 = \tau \ln 9$$

$$\tau = 4,55$$

$$td = \tau \cdot \ln 2$$

$$= 4,55 \cdot \ln 2$$

$$= 3,15 \quad (4.25)$$

$$\tau_{ST} = 4,32 \quad (4.26)$$

A (*Input*) = 2290 Rpm.

Oss(*Output steady state*) = 3720 Rpm

Perbandingan *output steady state* dan *input* :

$$K = \frac{3720}{2290} = 1,62 \quad (4.27)$$

sehingga dari data di atas diperoleh model matematis *plant* sebagai berikut :

$$G_{(s)} = \frac{1,62}{4,32s+1} \quad (4.28)$$

2. Smith's

$$\tau = \frac{T_u}{T_N} \quad (4.29)$$

$T_N$  dapat dicari menggunakan rumus pada (4.30) :

$$\tau = \frac{\tau \ln 2}{T_N} \\ T_N = \ln 2 \quad (4.30)$$

karena nilai waktu yang digunakan pada waktu pengambilan data adalah 20% dan 60% pada kondisi akhirnya/*steady state* , maka :

$$\tau_{sm1}=5,32 \quad \text{dan} \quad \tau_{sm2}=2,68 \quad (4.31)$$

A (*Input*) = 2290 Rpm.

Oss(*Output steady state*) = 3720 Rpm

Perbandingan *output steady state* dan *input* :

$$K = \frac{3758}{2260} = 1,62 \quad (4.32)$$

sehingga dari data di atas diperoleh model matematis *plant* sebagai berikut

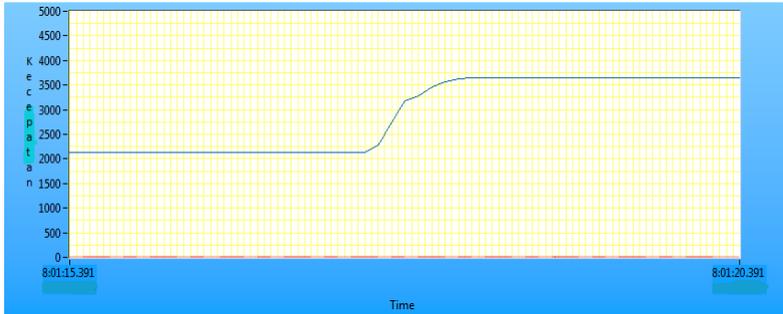
$$G_{(s)} = \frac{1,62}{(5,32s+1)(2,68s+1)} \quad (4.33)$$

#### 4.2.8.4 Pengujian Dengan Rem 75%

Pengujian ini dilakukan untuk mengetahui kecepatan Motor DC dengan beban 75% yang digunakan pada sistem. Pengujian ini dilakukan dengan mengambil data pengukuran pada titik kerja Motor yaitu 8-13Volt.

**Tabel 4.9** Data Pengujian Dengan Rem 75%

Counter (pwm)	Tegangan (Volt)	Rem (%)	Kecepatan (rpm)
90	8,470588	77,64	1760
90	8,470588	77,64	1760
90	8,470588	74,39	2276
90	8,470588	72,76	2276
90	8,470588	77,24	2266
90	8,470588	77,64	2266
90	8,470588	77,24	2266
90	8,470588	77,24	2266
90	8,470588	73,58	2269
90	8,470588	80,89	2269
140	13,176471	77,64	3540
140	13,176471	77,24	3540
140	13,176471	73,98	3732
140	13,176471	77,24	3732
140	13,176471	73,98	3731
140	13,176471	77,24	3731
140	13,176471	77,24	3730
140	13,176471	77,24	3730
140	13,176471	75,2	3732
140	13,176471	74,8	3732



**Gambar 4.5** Hasil Respon Kecepatan Motor Rem 75%

Dari **Tabel 4.9** dan **Gambar 4.5** dapat dilihat bahwa pada saat tegangan 8Volt Motor memiliki kecepatan rata rata sebesar 2270rpm, setelah keadaan sudah *steady* Motor naik pada tegangan 13Volt dengan kecepatan rata rata sebesar 3609rpm. Kemudian dari **Gambar 4.5** dapat dicari pemodelan matematika dengan cara identifikasi statis. Dari grafik **Gambar 4.5** diperoleh data sebagai berikut ;

1. Strejc's

$$\tau = \frac{T_U}{T_N} \quad (4.34)$$

$T_N$  dapat dicari menggunakan rumus pada (4.35) :

$$\tau = \frac{\tau \ln_2}{T_N}$$

$$T_N = \tau \ln_2$$

$$T_N = 0,69 \quad (4.35)$$

karena *time sampling* yang digunakan pada waktu pengambilan data adalah 10s, maka :

$$10 = \tau \ln_9$$

$$\tau = 4,55$$

$$\begin{aligned} \tau d &= \tau \cdot \ln 2 \\ &= 4,55 \cdot \ln 2 \\ &= 3,15 \end{aligned} \quad (4.36)$$

$$\tau_{ST} = 4,37 \quad (4.37)$$

A (Input) = 2270 Rpm.  
 Oss(Output steady state) = 3609 Rpm  
 Perbandingan output steady state dan input :

$$K = \frac{3609}{2270} = 1,59 \quad (4.38)$$

sehingga dari data di atas diperoleh model matematis plant sebagai berikut :

$$G_{(s)} = \frac{1,59}{4,375s+1} \quad (4.39)$$

2.Smith

$$\tau = \frac{T_u}{T_N} \quad (4.40)$$

$T_N$  dapat dicari menggunakan rumus pada (4.41) :

$$\tau = \frac{\tau \ln_2}{T_N} \quad (4.41)$$

$$T_N = \ln_2$$

karena nilai waktu yang digunakan pada waktu pengambilan data adalah 20% dan 60% pada kondisi akhirnya/steady state , maka :

$$\tau_{sm1}=5,28 \quad \text{dan} \quad \tau_{sm2}=2,72 \quad (4.42)$$

A (Input) = 2270 Rpm.  
 Oss(Output steady state) = 3609 Rpm  
 Perbandingan output steady state dan input :

$$K = \frac{3609}{2270} = 1,59 \quad (4.43)$$

sehingga dari data di atas diperoleh model matematis plant sebagai berikut

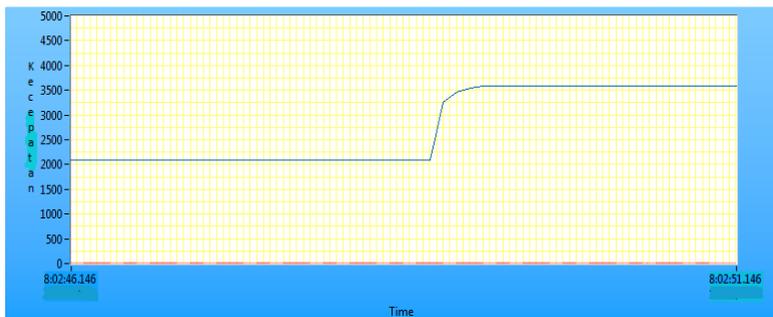
$$G_{(s)} = \frac{1,59}{(5,28+1)(2,72s+1)} \quad (4.44)$$

#### 4.2.8.5 Pengujian Dengan Rem 90%

Pengujian ini dilakukan untuk mengetahui kecepatan Motor DC dengan beban 90% yang digunakan pada sistem. Pengujian ini dilakukan dengan mengambil data pengukuran pada titik kerja Motor yaitu 8-13Volt.

**Tabel 4.10** Data Pengujian Dengan Rem 90%

Counter (pwm)	Tegangan (Volt)	Rem (%)	Kecepatan (rpm)
90	8,470588	94,72	1804
90	8,470588	94,72	1804
90	8,470588	93,9	2346
90	8,470588	93,5	2346
90	8,470588	91,06	2361
90	8,470588	89,84	2361
90	8,470588	96,75	2368
90	8,470588	94,31	2368
90	8,470588	94,31	2362
90	8,470588	94,31	2362
90	8,470588	94,31	2367
140	13,176471	90,24	3569
140	13,176471	94,31	3569
140	13,176471	94,31	3609
140	13,176471	93,9	3609
140	13,176471	91,87	3609
140	13,176471	90,65	3587
140	13,176471	95,93	3587
140	13,176471	94,72	3612
140	13,176471	93,9	3612
140	13,176471	93,09	3612



**Gambar 4.6** Hasil Respon Kecepatan Motor Rem 90%

Dari **Tabel 4.10** dan **Gambar 4.6** dapat dilihat bahwa pada saat tegangan 8Volt Motor memiliki kecepatan rata rata sebesar 2370rpm, setelah keadaan sudah *steady* Motor naik pada tegangan 13Volt dengan kecepatan rata rata sebesar 3600rpm. Kemudian dari **Gambar 4.6** dapat dicari pemodelan matematika dengan cara identifikasi statis. Dari grafik **Gambar 4.6** diperoleh data sebagai berikut ;

1. Strejc's

$$\tau = \frac{T_U}{T_N} \quad (4.45)$$

$T_N$  dapat dicari menggunakan rumus pada (4.46) :

$$\tau = \frac{\tau \ln 2}{T_N}$$

$$T_N = \tau \ln 2$$

$$T_N = 0,69 \quad (4.46)$$

karena *time sampling* yang digunakan pada waktu pengambilan data adalah 10s, maka :

$$10 = \tau \ln 9$$

$$\tau = 4,55$$

$$td = \tau \cdot \ln 2$$

$$= 4,55 \cdot \ln 2$$

$$= 3,15$$

$$(4.47)$$

$$\tau_{ST} = 4,42$$

$$(4.48)$$

A (*Input*) = 2370 Rpm.

Oss (*Output steady state*) = 3600 Rpm

Perbandingan *output steady state* dan *input* :

$$K = \frac{3600}{2370} = 1,51 \quad (4.49)$$

sehingga dari data di atas diperoleh model matematis *plant* sebagai berikut :

$$G_{(s)} = \frac{1,51}{4,42s+1} \quad (4.50)$$

## 2. Smith

$$\tau = \frac{T_u}{T_N} \quad (4.51)$$

$T_N$  dapat dicari menggunakan rumus pada 4.52) :

$$\tau = \frac{\tau \ln_2}{T_N}$$

$$T_N = \ln_2 \quad (4.52)$$

karena nilai waktu yang digunakan pada waktu pengambilan data adalah 20% dan 60% pada kondisi akhirnya/*steady state* , maka :

$$\tau_{sm1}=5,33 \quad \text{dan} \quad \tau_{sm2}=2,70 \quad (4.53)$$

A (Input) = 2370 Rpm.

Oss(Output *steady state*) = 3600 Rpm

Perbandingan *output steady state* dan *input* :

$$K = \frac{3758}{2260} = 1,51 \quad (4.54)$$

sehingga dari data di atas diperoleh model matematis *plant* sebagai berikut

$$G(s) = \frac{1,51}{(5,33s+1)(2,70s+1)} \quad (4.55)$$

### 4.2.8.6 Hasil Rangkuman Pengukuran

Pada Hasil Rangkuman Pengukuran ini, 2 metode yang diidentifikasi pada titik kerja motor untuk mendapatkan respon model yang paling mendekati respon *plant*. Model matematika dari 2 metode identifikasi perlu di uji dengan metode ISE (*Integral Square Error*) untuk membuktikan bahwa model memiliki kesamaan dengan *plant* dalam kondisi nyata. Hasil dapat dilihat pada Tabel 4.11 dan Tabel 4.12

**Tabel 4.11** Hasil Rangkuman Pengukuran Metode Strejc's

Rem	Transfer Function (Strejc's)	ISE
0%	$G(s) = \frac{1,66}{(4,55s+1)}$	18,75
25%	$G(s) = \frac{1,64}{(4,65s+1)}$	13,75

Rem	Transfer Function (Strej'c)	ISE
50%	$G_{(s)} = \frac{1,62}{(4,3s+1)}$	27,5
75%	$G_{(s)} = \frac{1,59}{(4,37s+1)}$	26,25
90%	$G_{(s)} = \frac{1,51}{(4,42s+1)}$	24,1

**Tabel 4.12** Hasil Rangkuman Pengukuran Metode Smith's

Rem	Transfer Function (Smith's)	ISE
0%	$G_{(s)} = \frac{1,66}{(5,24s+1)(2,76s+1)}$	15
25%	$G_{(s)} = \frac{1,64}{(5,2s+1)(2,8s+1)}$	9,72
50%	$G_{(s)} = \frac{1,62}{(5,32s+1)(2,68s+1)}$	13,95
75%	$G_{(s)} = \frac{1,59}{(5,28s+1)(2,72s+1)}$	12,68
90%	$G_{(s)} = \frac{1,51}{(5,33s+1)(2,70s+1)}$	16,18

Berdasarkan **Tabel 4.11** dan **Tabel 4.12**

1. Metode Strej'c dengan Rem 25% memiliki Transfer Function  $G_{(s)} = \frac{1,64}{(4,65s+1)}$  dan mempunyai ISE paling kecil yaitu 13,75.
2. Metode Smith's dengan Rem 25% memiliki Transfer Function  $G_{(s)} = \frac{1,64}{(5,2s+1)(2,8s+1)}$  dan ISE paling kecil yaitu 9,72.

## BAB V PENUTUP

### 5.1 Kesimpulan

Dari hasil pengujian dan analisa terhadap sistem Identifikasi pada Motor dapat diambil beberapa kesimpulan sebagai berikut:

1. Pada pengujian data pertama dilakukan pengujian tanpa rem menunjukkan Motor aktif mulai *counter* 40 dengan kecepatan 156 – 4991rpm
2. Pada pengujian Motor dengan beban 25% sampai 90% terdapat penurunan rpm mulai 40rpm sampai 800rpm
3. Daerah titik kerja kecepatan Motor terdapat pada rentang 8-13 Volt
4. Hasil identifikasi dengan metode 2S adalah suatu Transfer Function, pada Metode Strejc's didapatkan  $G_{(s)} = \frac{1,64}{(4,65s+1)}$  dengan ISE terkecil 13,75. Pada Metode Smith's didapatkan  $G_{(s)} = \frac{1,64}{(5,2s+1)(2,8s+1)}$  dengan ISE terkecil 9,72.
5. Dari hasil simulasi, dalam identifikasi menggunakan metode 2S memberikan sinyal ouput dengan *performance* yang baik.
6. Dengan identifikasi menggunakan metode 2S, memudahkan dalam membuat pengaturan kecepatan Motor, pengaturan beban dan saat *starting* Motor.

### 5.2 Saran

Dari hasil perancangan tugas akhir ini masih kurang sempurna sehingga ada yang harus diperbaiki agar hasil tugas akhir ini mendekati sempurna yaitu:

1. Pemodelan yang dibuat belum maksimal dalam Motor DC karena masih terdapat nilai *error*. Diharapkan untuk pengembangan selanjutnya perlu dilakukan lebih banyak pemodelan supaya didapatkan nilai *error* yang paling kecil dan model paling mendekati respon *plant*.

*(halaman ini sengaja dikosongkan)*

## DAFTAR PUSTAKA

- [1] Andianto, Catur dan Amanda Dwi .*Pembuatan Kontrol Kecepatan pada Modul Praktikum Motor DC dengan Metode PI Menggunakan Mikrokontroler dengan Media Komunikasi Wireless*. Surabaya. **Tugas Akhir**, FTI-ITS.2013
- [2] Izal, Fahmi. *Pengaturan Kecepatan Motor DC*. Bandung. Grasindo. 2005.
- [3] Zuhail, Mahfud. *Dasar Tenaga Listrik dan Elektronika Daya*. Jakarta. Gramedia. 1998.
- [4] Ali, Amrizal. *Pengertian Driver Motor DC*. Jakarta. Media Pressindo. 2011.
- [5] Eitel, Elisabeth. *Basics of Rotary Encoders: Overview and New Technologies*. London. *Machine Design Magazine*. 2014.
- [6] Kadir, Abdul. *Panduan Mempelajari Arduino*. Jogjakarta. ANDI OFFSET. 2013.
- [7] Bachtiar, Muhammad Fachri dan Priyatna, Alif Gigah. *Perancangan Rem Magnetik Pada Motor DC dengan Menggunakan Arduino*. Surabaya. **Tugas Akhir**, FTI-ITS. 2012.
- [8] Harrington J., and K. Tumay. *Simulation Modeling Methods: An Interactive Guide to Results-Based Decision*, McGraw-Hill, 1998.
- [9] Ing. Pavel Jakoubek, *Experimental Identification of Stable Nonoscillatory Systems from Steps-Responses by Selected Method* ,**Konference Students**,2009.

*(halaman ini sengaja dikosongkan)*

## LAMPIRAN A

### A1. Listing Program

```
unsigned long counts; //variable for GM Tube events
unsigned long previousMillis; //variable for time measurement
unsigned long currentMillis;

int enRem = 5, xaRem = 8, xbRem = 7;
int enMotor = 6, xaMotor = 10, xbMotor = 9;
const int plus=3;
const int minus=4;
unsigned int Speed=0, bacatombolplus=0, bacatombolmin=0, datacounter,
data=0;
float persen, potensio;
#define LOG_PERIOD 1000 // cetak tiap detik
void impulse() { // dipanggil setiap ada sinyal FALLING di pin 2
  counts++;
}

void setup() {
  pinMode(plus, INPUT);
  pinMode(minus, INPUT);
  pinMode(xaRem, OUTPUT);
  pinMode(xbRem, OUTPUT);
  pinMode(xaMotor, OUTPUT);
  pinMode(xbMotor, OUTPUT);
  counts = 0;
  Serial.begin(9600);
  pinMode(2, INPUT); // tidak perlu sebenarnya, untuk jaga-jaga saja.
  digitalWrite(2, HIGH); // mengaktifkan internal pull up resistor
  attachInterrupt(digitalPinToInterrupt(2), impulse, RISING); //define
external interrupts
  Serial.println("Start counter");
}

void loop() {
  utama:
  digitalWrite(xaRem, LOW);
  digitalWrite(xbRem, HIGH);
  digitalWrite(xaMotor, LOW);
```

```

digitalWrite(xbMotor,HIGH);
bacatombolplus=digitalRead(plus);
bacatombolmin=digitalRead(minus);
if (bacatombolplus==LOW)
{

    if(data>20)
    {
        data=20;
        if (bacatombolmin==LOW)
        {
            goto tunggu;
        }
    }

    data=data+10;
    if (bacatombolplus==LOW)
    {
        goto tunggu;
    }
}

if (bacatombolmin==LOW)
{
    if(data<=0)
    {
        data=0;
        if (bacatombolmin==LOW)
        {
            goto tunggu;
        }
    }
}

if(data>0)
{
    data=data-10;
    if (bacatombolmin==LOW)
    {
        goto tunggu;
    }
}

```

```

}

if(data==0)
{
  Speed=0;
}

if(data==10)
{
  Speed=90;
}

if(data==20)
{
  Speed=140;
}

potensio = (analogRead(A2)/4);
persen= (potensio/246)*100;
analogWrite(enRem,potensio);
analogWrite(enMotor,Speed);
Serial.print("x");
Serial.print(Speed);
Serial.print(",");
Serial.print(persen);
Serial.print(",");
currentMillis = millis();
if (currentMillis - previousMillis > LOG_PERIOD) {
  previousMillis = currentMillis;
  datacounter = counts;
  counts = 0;
}
Serial.print(datacounter);
// Serial.print("50");
Serial.println("y");
delay(500);
goto utama;

tunggu:
bacatombolplus=digitalRead(plus);
bacatombolmin=digitalRead(minus);

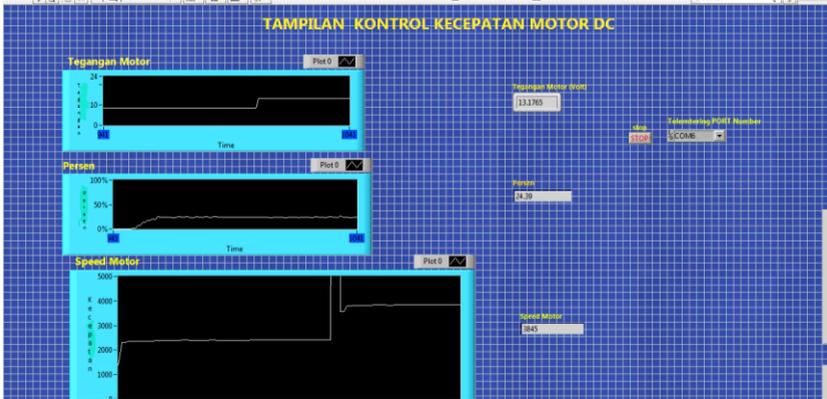
```

```
if (bacatombolplus==HIGH && bacatombolmin==HIGH)
{
    goto lanjut;
}
delay(100);
goto tunggu;

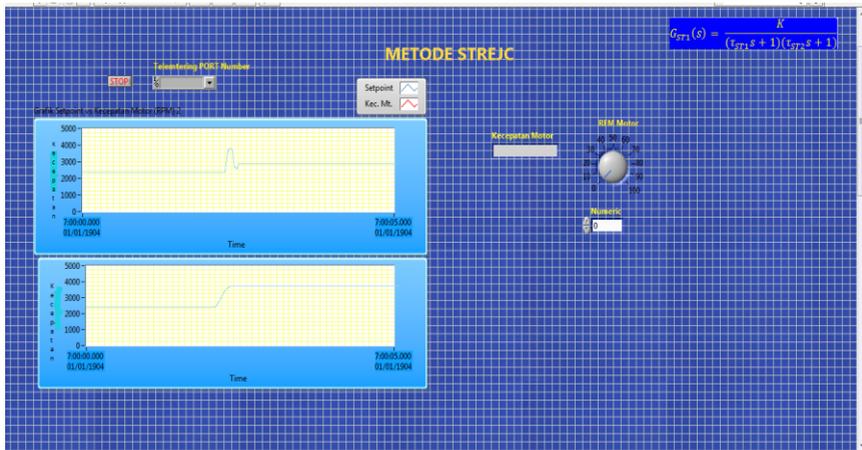
lanjut:
goto utama;
}
```

## LAMPIRAN B

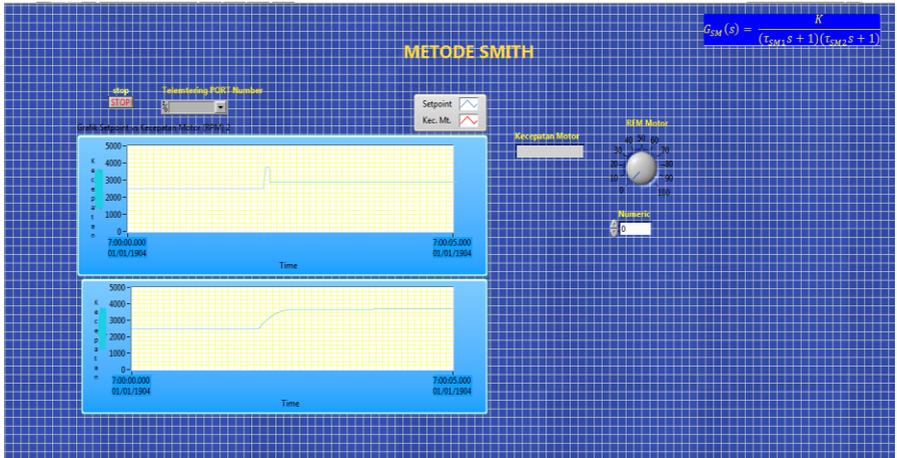
### B1. Tampilan LabVIEW Monitoring Kecepatan Motor



Gambar 1. Tampilan Utama LabVIEW



Gambar 2. Tampilan Metode Strejc's



Gambar 3. Tampilan Metode Smith

## RIWAYAT HIDUP



Penulis, Annona Dieni Septiarini, lahir di Surabaya, 10 September 1996. Merupakan anak pertama dari tiga bersaudara dari pasangan Bapak Suyadi dan Ibu Enie Yulianti. Bertempat tinggal di Girilaya gang 8 no. 61 kecamatan sawahan Surabaya, Jawa Timur, Indonesia. Penulis memulai Studinya di TK AL IKHLAS lulus tahun 2002, SD Bluru Kidul II lulus tahun 2008, dilanjutkan di SMP Muhammadiyah 1 lulus tahun 2011, dan MAN Sidoarjo lulus tahun 2014. Kemudian pada tahun 2014 melanjutkan pendidikan tinggi di Diploma III Teknik Elektro ITS dan berkonsentrasi pada Bidang Studi Komputer Kontrol. Pada bulan Januari 2018 penulis mengikuti ujian Tugas Akhir di Bidang Studi Komputer Kontrol Program Studi D3 Teknik Elektro ITS Surabaya sebagai salah satu persyaratan untuk memperoleh gelar Ahli Madya Teknik Elektro. Penulis dapat dihubungi pada alamat email sebagai berikut: [dieniannona96@gmail.com](mailto:dieniannona96@gmail.com)

*(halaman ini sengaja dikosongkan)*