



**RANCANG BANGUN KONTROL PID PADA *SPEED OBSERVER*
GENERATOR DC BERBASIS ARDUINO UNO R3**

SKRIPSI

oleh

**Mochamad Galih Adi Prakoso
NIM 121910201083**

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2016**



**RANCANG BANGUN KONTROL PID PADA *SPEED OBSERVER*
GENERATOR DC BERBASIS ARDUINO UNO R3**

SKRIPSI

**diajukan guna melengkapi skripsi dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi S1 Teknik Elektro
dan mencapai gelar Sarjana Teknik**

oleh

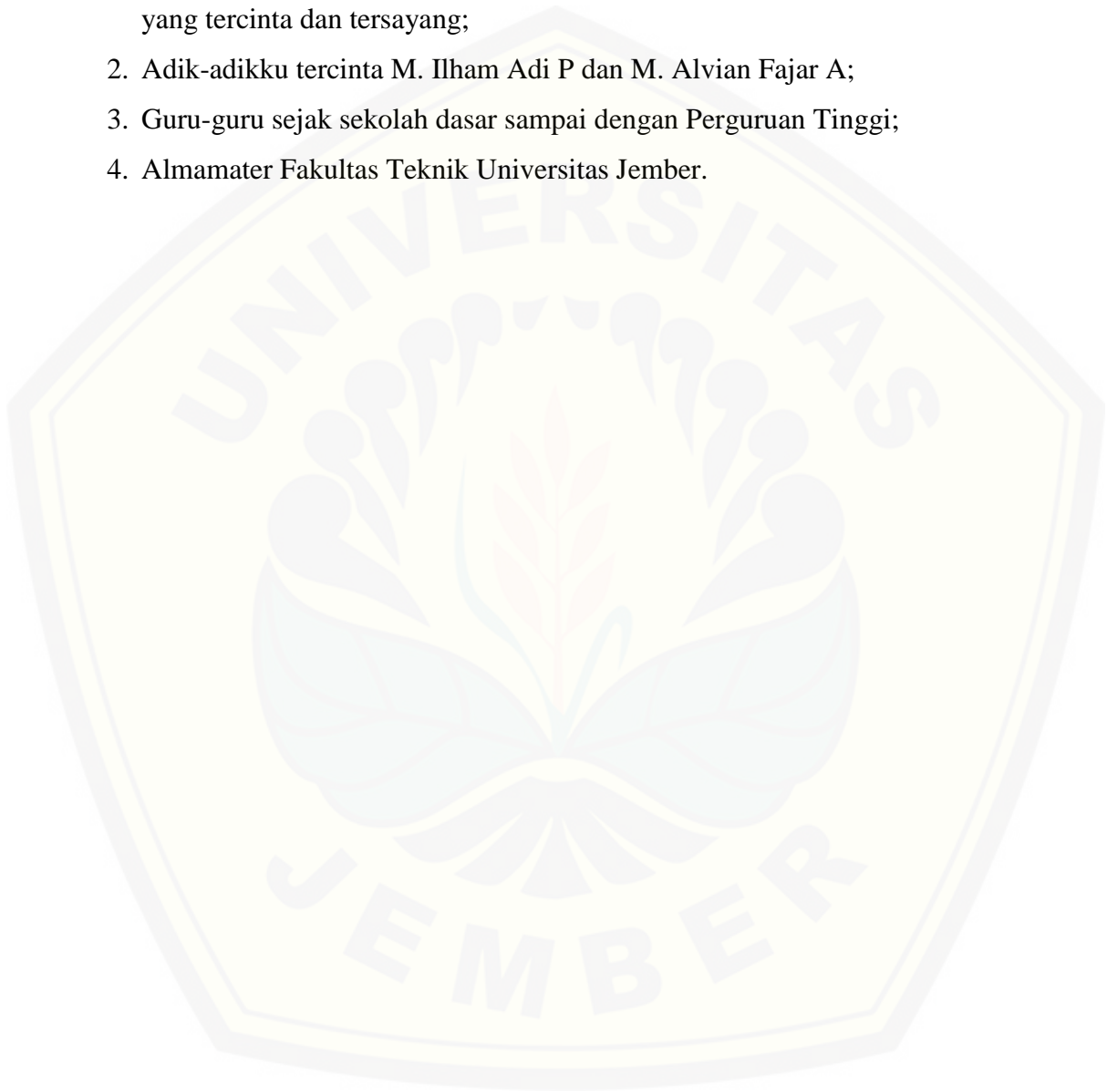
**Mochamad Galih Adi Prakoso
NIM 121910201083**

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2016**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Alm. Ibunda Yulisasi A. dan Ayahanda Suwarno serta Ibu sambung wiwik yang tercinta dan tersayang;
2. Adik-adikku tercinta M. Ilham Adi P dan M. Alvian Fajar A;
3. Guru-guru sejak sekolah dasar sampai dengan Perguruan Tinggi;
4. Almamater Fakultas Teknik Universitas Jember.



MOTO

Sembahlah Allah dan janganlah kamu mempersekutukan-Nya dengan sesuatupun. Dan berbuat baiklah kepada dua orang ibu-bapa, karib-kerabat, anak-anak yatim, orang-orang miskin, tetangga yang dekat dan tetangga yang jauh, dan teman sejawat, ibnu sabil dan hamba sahayamu. Sesungguhnya Allah tidak menyukai orang-orang yang sombong dan membangga-banggakan diri. *)

وَالصَّلَاةِ بِالصَّبْرِ وَاسْتَعِينُوا

Mintalah pertolongan dengan sabar dan shalat. **)

Masa depan bukan terletak pada pekerjaan apapun, tetapi pada orang yang mengerjakannya. ***)

*)", (Q.S An Nisaa', 4:36)..

**) QS Al-Baqarah: 45.

***) George Crane.

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Mochamad Galih Adi Prakoso

NIM : 121910201083

Menyatakan dengan sesungguhnya bahwa skripsi yang berjudul “Rancang Bangun Kontrol PID pada *Speed Observer* Generator DC Berbasis Arduino UNO R3” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan dalam institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak mana pun serta bersedia mendapat sanksi akademik jika ternyata dikemudian hari pernyataan ini tidak benar.

Jember, 03 November 2016

Yang menyatakan,

M. Galih Adi Prakoso
NIM 121910201083

SKRIPSI

**RANCANG BANGUN KONTROL PID PADA *SPEED OBSREVER*
GENERATOR DC BERBASIS ARDUINO UNO R3**

oleh

Mochamad Galih Adi Prakoso
NIM 121910201083

Pembimbing

Dosen Pembimbing Utama : Dr. Bambang Sri Kaloko S.T.,M.T

Dosen Pembimbing Anggota : Ir. Widyono Hadi, M.T.

PENGESAHAN

Skripsi berjudul “Rancang Bangun Kontrol PID pada *Speed Obserever* Generator DC Berbasis Arduino UNO R3” karya Mochamad Galih Adi Prakoso telah diuji dan disahkan pada:

hari, tanggal : Kamis, 03 November 2016

tempat : Fakultas Teknik Universitas Jember.

Tim Penguji:

Ketua,

Dr. Bambang Sri Kaloko , S.T., M.T.
NIP 197104022003121001

Anggota I,

Ir. Widyono Hadi, M.T.
NIP 196104141989021001

Anggota II,

H.R.B. Moch Gozali, S.T., M.T.
NIP 196906081999031002

Anggota III,

Dr. Azmi Saleh, S.T., M.T.
NIP 197106141997021001

Mengesahkan
Dekan,

Dr. Ir. Entin Hidayah, M.U.M
NIP 196612151995032001

Rancang Bangun Kontrol PID pada *Speed Observer*
Generator DC Berbasis Arduino UNO R3

Mochamad Galih Adi Prakoso

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember

ABSTRAK

Dalam penelitian ini membahas pengembangan observer kecepatan untuk generator arus searah (DC) yang terkontrol dengan menggunakan sensor *rotary encoder* yang dibangun menggunakan Arduino UNO dengan kemampuan *recording* dari kedua sistem ini juga dibandingkan. Pengujian dilakukan untuk melihat *error* pembacaan observer menunjukkan bahwa observer ini cukup memadai dengan *error* rata-rata 0.81 % terhadap hasil pembacaan sensor kecepatan yang dalam hal ini dijadikan sebagai standart. Penerapan kontrol PID sebagai kontrol dari *speed observer* generator DC sehingga tetap menjaga kecepatan generator pada *setpoint* meskipun dengan arus beban yang meningkat. Mengacu pada kurva hubungan kecepatan dan daya pada *prime mover setpoint* kecepatan adalah untuk meningkatkan kehandalan serta kinerja generator DC. Kontrol PID membuat kecepatan generator yang stabil pada tegangan *setpoint*. Parameter PID yang digunakan adalah $K_p = 0.77$, $K_i = 0,0083$ dan $K_d = 0,0096$. Setelah menerapkan nilai kontrol PID diperoleh *rise time* 0,51 s, *error steady-state* 1.28 %. Pengujian dilakukan dengan pemberian beban resistif pada *setpoint* kecepatan 1500 rpm, 2000 rpm dan 3000 rpm serta variasi beban yang digunakan 50 Ω dan 25 Ω . Untuk nilai *error steady-state* tertinggi yang diperoleh 17.21 % saat dengan *setpoint* 3000 rpm dan terendah adalah 0,004% dengan *setpoint* 2000 rpm. Nilai *recovery time* tertinggi pada *setpoint* 3000 rpm yang tidak bisa kembali ke *setpoint* dan terendah 0,6 s saat *setpoint* 1500 rpm.

Kata kunci: Arduino UNO R3 Generator DC, PID *Speed Observer*, ,

*Design PID Control of Speed Observer DC Generator Based
Arduino UNO R3*

Mochamad Galih Adi Prakoso

*Department of Electrical Engineering, Faculty of Engineering, University of
Jember*

ABSTRACT

In this study discusses the development of observer to the generator speed direct current (DC) that is controlled using the rotary encoder sensor built using the Arduino UNO with recording capability of the two systems is also compared. Testing is done to see readings observer error indicates that the observer is sufficient with an average error of 0.81% of the speed sensor readings in this case be used as a standard. Application of PID control as the control of the speed observer DC generators that keep the generator speed setpoint despite the load current increases. Referring to the curve of speed and power relations in the prime mover speed setpoint is to improve the reliability and performance of the DC generator. PID control makes speed generator is stable at voltage setpoint. PID parameters used are $K_p = 0.77$, $K_i = 0.0083$ and $K_d = 0.0096$. After applying PID control value obtained rise time of 0.51 s, the steady-state error 1:28%. Testing is done by giving a resistive load to the setpoint speed of 1500 rpm, 2000 rpm and 3000 rpm and load variations that are used 50 Ω and 25 Ω . For steady-state error value obtained supreme 17:21% when the setpoint of 3000 rpm and the lowest was 0.004% at 2000 rpm setpoint. The highest value of recovery time on a setpoint of 3000 rpm that can not be returned to the setpoint and the lowest was 0.6 s current setpoint 1500 rpm.

Key words: *Arduino Uno R3, DC Generator, PID, Speed Observer*

RINGKASAN

Rancang Bangun Kontrol PID pada *Speed Observer* Generator DC Berbasis Arduino UNO R3; M. Galih Adi Prakoso, 121910201083; 2016; 85 halaman; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Dalam kehidupan kita sehari – hari terutama bidang pembangkit listrik dan industri, generator DC dapat berfungsi sebagai salah satu pembangkit arus searah di bengkel – bengkel atau pabrik, sebagai pengisi accu pada perusahaan pengisi accu, bahkan dipusat – pusat tenaga listrik berfungsi sebagai *exciter* pada generator utama. Mengacu pada kurva hubungan daya dan kecepatan pada *prime mover* kehandalan serta kinerja generator sangat dibutuhkan sebagai penyuplai. Untuk meningkatkan kehandalan serta kinerja generator dibutuhkan pengaturan kecepatan *real time*. Sebuah sistem observer mampu mencuplik kecepatan generator secara *real time*. Pembangunan observer kecepatan menggunakan persamaan regresi linier, dimana pengolahan observer menggunakan *regression* pada matlab. Namun kecepatan generator akan cenderung tidak stabil apabila dalam keadaan pembebanan.

Implementasi kontrol PID dapat diterapkan pada generator DC untuk mengatasi hal tersebut, karena dengan terjadinya pembebanan kecepatan generator DC cenderung tidak stabil. Diharapkan dengan diterapkannya kontrol PID pada sistem generator DC dapat meningkatkan kehandalan serta kinerja generator dengan menstabilkan kecepatan dalam keadaan pembebanan.

Pengujian pada generator DC dengan parameter PID yang digunakan yaitu $K_p = 0.77$, $K_i = 0.0083$ dan $K_d = 0.0096$ didapatkan *rise time* 0.51 s, *settling time* 0.85 s, *error steady-state* 1.28 %. Parameter-parameter inilah yang akan digunakan kontrol PID pada *setpoint* 1500 rpm, 2000 rpm dan 3000 rpm. Variasi beban yang digunakan pada pengujian yaitu 50 Ω dan 25 Ω , dimana 25 Ω adalah beban puncak. Pada pengujian pembebanan didapatkan nilai *recovery time* tercepat terjadi saat *setpoint* 1500 yaitu sebesar 0.6 s terjadi saat pelepasan beban 50 Ω , sedangkan nilai *recovery time* terbesar terjadi pada *setpoint* 3000 rpm

dengan nilai *recovery* tak terhingga dengan kata lain tidak berhasil mencapai *setpoint* kembali. Hal tersebut diakibatkan tekanan udara mengalami pencekikan pada *valve*, sehingga kecepatan tidak bisa kembali pada *setpoint* ketika generator dalam keadaan berbeban. Untuk nilai *error steady-state* terkecil diperoleh saat *setpoint* 2000 rpm yaitu sebesar 0004 % sedangkan nilai *error steady-state* tertinggi terjadi pada *setpoint* 3000 rpm yaitu sebesar 17.21 rpm.



PRAKATA

Puji syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Rancang Bangun Kontrol PID pada *Speed Observer* Generator Berbasis Arduino UNO R3”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan strata satu (S1) pada Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Dr. Ir. Entin Hidayah, M.UM. selaku Dekan Fakultas Teknik Universitas Jember.
2. Dr. Bambang Sri Kaloko, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Universitas Jember sekaligus Dosen Pembimbing Utama dan Ir. Widyono Hadi, M.T. selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi ini.
3. Dr. Bambang Sri Kaloko S.T., M.T selaku Dosen Pembimbing Utama, Ir. Widyono Hadi, M.T. selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi ini.
4. H.R.B. Moch. Gozali, S.T., M.T. selaku Dosen Penguji I, Dr. Azmi Saleh, S.T., M.T. selaku Dosen Penguji II yang telah memberikan kritik dan saran yang sangat membangun demi penyempurnaan skripsi ini.
5. Sumardi, S.T., M.T. selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa.
6. M. Agung Prawira N., S.T., M.T. selaku Komisi Bimbingan S1 yang telah membantu penulisan skripsi secara administratif.
7. Andi Setiawan, S.T., M.T. yang tidak pernah lelah memberikan pengarahan saat pengerjaan *plan*.
8. Alm ibunda. Yulisasi A dan ayahanda Suwarno serta ibu sambung wiwik yang telah memberikan dukungan moril dan materiil serta kasih sayang yang tak terhingga sepanjang masa.

9. Adik-adikku tercinta M. Ilham Adi P. dan M. Alvian Fajar A yang telah memberikan motivasi dan semangat.
10. Keluarga besar SATE UJ 12 atas kekeluargaan yang kalian berikan.
11. Dulur teknik elektro universitas Jember atas kebersamaannya.
12. Lab Konversi Energi Listrik Universitas Jember yang telah memberikan wadah serta fasilitas dalam pengerjaan skripsi khususnya jurusan teknik elektro universitas jember.
13. Seluruh punggawa laboratorium patrang yang bersedia meluangkan waktu serta pikiran untuk membantu menyelesaikan skripsi ini.
14. Rekan-rekan satu cangkir dan satu aspal Chandra, Hendro, Erlangga, Dol, Rihuh, Mahfud, Helmy, Dio, Hery, Tutut, Agam, Danu, Umam, Latif, Aji, Dony, Rizky, Surya, Ardi, Rina, Farah, Citra, Munir, Taiwo, Saiful, Fandy, Cemeth, Ferdika, Putri, Hanso, Dovy, Jarwo dan masih banyak lagi yang selalu berkumpul selama masa kuliah;
15. Keluarga kecil Brantas 11 No 83 Dony, Mahfud, Irwan, Umam, Ardi, Aziz, Surya, Hasbul, Rizky dan yuda yang telah menjadi keluarga kedua selain keluarga di rumah.
16. Semua pihak yang tidak dapat disebutkan satu persatu.

Penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, 03 November 2016

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	ii
HALAMAN PERSEMBAHAN	iii
HALAMAN MOTO	iv
HALAMAN PERNYATAAN.....	v
HALAMAN PEMBIMBINGAN.....	vi
HALAMAN PENGESAHAN.....	vii
ABSTRAK	viii
ABSTRACT	ix
RINGKASAN	x
PRAKATA.....	xii
DAFTAR ISI.....	xiv
DAFTAR TABEL	xvii
DAFTAR GAMBAR.....	xix
DAFTAR LAMPIRAN	xxi
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Sistematika Penulisan.....	4
BAB 2. TINJAUAN PUSTAKA.....	5
2.1 Generator <i>Direct Current</i>	5
2.2 <i>Air Motor</i>	7
2.3 Sistem <i>Speed Observer</i>	9
2.3.1 Regresi Linier Berganda	10
2.4 Komponen Pengendali.....	11
2.4.1 Motor Servo.....	11

2.4.2 INA 219	14
2.4.3 Sensor Tekanan Udara.....	15
2.4.4 Arduino Uno	17
2.5 Kontrol PID	19
2.5.1 Sistem Kontrol PID.....	19
2.5.2 <i>Tunning</i> Parameter PID	22
BAB 3. METODOLOGI PENELITIAN	24
3.1 Tahapan Penelitian	24
3.2 Perancangan dan Blok Diagram Sistem	26
3.3 Perancangan <i>Hardware</i>	28
3.3.1 Perancangan Rangkaian INA 219	29
3.3.2 Perancangan Sensor Tekanan	29
3.3.3 Arduino Uno R3	30
3.3.4 Generator DC Magnet Permanen	30
3.4 Sistem <i>Speed Observer</i>	31
3.5 Hubungan Tegangan Arus dan Daya Generator	33
3.6 Kontrol PID	34
3.7 Metode <i>Manual Tunning</i>	35
BAB 4. HASIL DAN PEMBAHASAN	36
4.1 Pengujian Komponen Pengendali	37
4.1.1 Pengujian Tegangan.....	37
4.1.2 Pengujian Arus.....	38
4.1.3 Pengujian <i>Rotary Encoder</i>	39
4.1.4 Pengujian Tekanan.....	40
4.2 Pengujian <i>Plan Motor-Generator</i>	41
4.2.1 Hubungan Kecepatan dengan Tegangan Generator.....	41
4.2.2 Hubungan Tekanan dengan Kecepatan Generator.....	44
4.3 Pembuatan Persamaan <i>Observer Kecepatan</i>.....	45
4.3.1 Pengujian <i>Observer Kecepatan</i>	47

4.4 Pengujian Observer kecepatan dengan Kontrol PID	49
4.4.1 Penentuan Nilai Parameter <i>PID</i> dengan Metode <i>Tunning Manual</i>	49
4.4.2 Pengujian Kontrol PID Terhadap Observer denagan Kecepatan Bervariasi	52
4.4.3 Pengujian Kontrol PID Terhadap Observer Kecepatan Menggunakan Beban	55
4.4.3.1. Pengujian Pembebanan Pada <i>Setpoint</i> 1500 rpm	56
4.4.3.2. Pengujian Pembebanan Pada <i>Setpoint</i> 2000 rpm	60
4.4.3.3. Pengujian Pembebanan Pada <i>Setpoint</i> 3000 rpm	64
4.4.3.4. Hubungan Tegangan Arus dan Daya <i>Speed Observer</i> Terkontrol PID	68
BAB 5. PENUTUP	71
5.1. Kesimpulan	71
5.2. Saran	72
DAFTAR PUSTAKA	73
LAMPIRAN	74

DAFTAR TABEL

	Halaman
2.1 Spesifikasi <i>Air Motor</i>	9
2.2 Spesifikasi Teknis DS04-NFC	13
2.3 Karakteristik Operasi MPX5700.....	17
2.4 Spesifikasi Arduino UNO-R3	18
2.5 Karakteristik Masing-masing Kontrol P-I-D	23
3.1 Kalkulasi Parameter Generator	31
4.1 Pengujian Sensor Tegangan	37
4.2 Pengujian Sensor Arus	38
4.3 Pengujian Sensor Kecepatan.....	39
4.4 Pengujian Tekanan.....	40
4.5 Hubungan Kecepatan dengan Tegangan Generator.....	42
4.6 Hubungan Tekanan dengan Kecepatan Generator Tanpa Beban.....	44
4.7 Hubungan Tegangan dan Arus dengan Kecepatan	46
4.8 Perbandingan Antara Sensor Kecepatan dan Observer.....	48
4.9 <i>Manual Tunning</i> PID	52
4.10 Pengujian Kontrol PID dengan Variasi Kecepatan.....	53
4.11 Hubungan Tegangan dan Arus Saat Diterapkan PID Tanpa Beban	54
4.12 Pengujian Kontrol PID Terhadap Pembebanan Kecepatan Generator.....	56
4.13 Pemberian Beban 50 Ω <i>Setpoint</i> 1500 rpm.....	57
4.14 Pemberian Beban 25 Ω <i>Setpoint</i> 1500 rpm.....	58
4.15 Pelepasan Beban 25 Ω <i>Setpoint</i> 1500 rpm.....	59
4.16 Pelepasan Beban 50 Ω <i>Setpoint</i> 1500 rpm.....	60
4.17 Pemberian Beban 50 Ω <i>Setpoint</i> 2000 rpm.....	61
4.18 Pemberian Beban 25 Ω <i>Setpoint</i> 2000 rpm.....	62
4.19 Pelepasan Beban 25 Ω <i>Setpoint</i> 2000 rpm.....	63
4.20 Pelepasan Beban 50 Ω <i>Setpoint</i> 2000 rpm.....	64

4.21 Pemberian Beban 50 Ω <i>Setpoint</i> 3000 rpm.....	65
4.22 Pemberian Beban 25 Ω <i>Setpoint</i> 3000 rpm.....	66
4.23 Pelepasan Beban 25 Ω <i>Setpoint</i> 3000 rpm.....	67
4.24 Pelepasan Beban 50 Ω <i>Setpoint</i> 3000 rpm.....	68



DAFTAR GAMBAR

	Halaman
2.1 Rangkaian Ekuivalen dari PMDC	5
2.2 Skematik <i>Air Motor</i>	7
2.3 Karakteristik Tipikal <i>Air Motor</i>	8
2.4 <i>Air Motor</i> Prona.....	9
2.5 Tegangan <i>Ripple</i> Sebagai Efek Komutasi	10
2.6 Skematik Motor Servo Secara Umum	12
2.7 Pulsa Pengatur Motor Servo Standart.....	12
2.8 Motor Servo 360° DS04-NFC	13
2.9 Sinyal Kendali DS04-NFC 360°	14
2.10 Skematik INA 219	15
2.11 Konfigurasi Pin INA 219	15
2.12 Sensor Tekanan Udara MPX5700	16
2.13 Diagram Fungsional MPX5700.....	16
2.14 Modul Kontroler Arduino UNO-R3	18
2.15 Skema Pengendali PID	19
2.16 (a) Rangkaian Elektronika PID Analog (b) Fungsi Transfer.....	20
2.17 Metode Pencuplikan (a) <i>Backward Rectangular</i> Methode (b) <i>Forward Rectangular Methode</i>	21
3.1 <i>Flowchart</i> Penelitian.....	24
3.2 Rancangan Penelitian Alat.....	26
3.3 Diagram Blok Sistem.....	28
3.4 <i>Wiring</i> Rangkaian INA 219 Pada Arduino	29
3.5 Rangkaian MPX5700 ke Sistem.....	30
3.6 Generator DC Magnet Permanen.....	31
3.7 Perancangan Tegangan Arus dan Daya <i>Output</i> Generator DC	33
3.8 Diagram Perancangan Sistem PID terhadap <i>Plan</i>	35
4.1 Pengujian Kecepatan Tegangan Generator Tanpa Beban	42
4.2 Hubungan Kecepatan dengan Tegangan Generator	43

4.3	Hubungan Tekanan dengan Kecepatan Generator	45
4.4	Grafik Hubungan Tegangan Arus dan Kecepatan	47
4.5	Perbandingan Sensor Kecepatan dan Observer	49
4.6	Grafik Kecepatan Generator dengan Variasi <i>Setpoint</i>	53
4.7	Pengujian Pemberian Beban 50 Ω Pada 1500 rpm.....	57
4.8	Pengujian Pemberian Beban 25 Ω Pada 1500 rpm.....	58
4.9	Pengujian Pelepasan Beban 25 Ω Pada 1500 rpm.....	59
4.10	Pengujian Pelepasan Beban 50 Ω Pada 1500 rpm.....	60
4.11	Pengujian Pemberian Beban 50 Ω Pada 2000 rpm.....	61
4.12	Pengujian Pemberian Beban 25 Ω Pada 2000 rpm.....	62
4.13	Pengujian Pelepasan Beban 25 Ω Pada 2000 rpm.....	63
4.14	Pengujian Pelepasan Beban 50 Ω Pada 2000 rpm.....	64
4.15	Pengujian Pemberian Beban 50 Ω Pada 3000 rpm.....	65
4.16	Pengujian Pemberian Beban 25 Ω Pada 3000 rpm.....	66
4.17	Pengujian Pelepasan Beban 25 Ω Pada 3000 rpm.....	67
4.18	Pengujian Pelepasan Beban 50 Ω Pada 3000 rpm.....	68

DAFTAR LAMPIRAN

	Halaman
A. Listing Program Arduino Uno R3.....	74
B. Gambar <i>Hardware</i>	79
B1. Rangkaian Sensor.....	79
B2. <i>Air Motor</i> - Generator DC.....	79
B3. Beban Resistif	80
C. Gambar Keseluruhan Sistem dan Pengambilan Data	80
D. Data Hasil <i>Speed Observer</i>	81
E. Perhitungan Regresi	85

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Generator DC merupakan sebuah perangkat mesin listrik dinamis yang mengubah energi mekanis menjadi energi listrik searah. Secara umum generator DC tidak berbeda dengan motor DC kecuali pada arah aliran daya. Berdasarkan cara memberikan fluks pada kumparan medannya, generator arus searah DC dapat dikelompokkan menjadi dua, yaitu : generator berpenguatan terpisah dan generator berpenguatan sendiri, seperti yang dipaparkan pada buku (Hameyer, 2001).

Perkembangan teknologi di era globalisasi saat ini berimbas pada peningkatan kebutuhan energi listrik yang sangat besar terutama pada bidang industri. Penggunaan generator DC sebagai salah satu pembangkit arus searah di bengkel – bengkel atau pabrik, sebagai pengisi accu pada perusahaan pengisi accu, bahkan dipusat – pusat tenaga listrik berfungsi sebagai *exciter* pada generator utama.

Pada bidang industri dibutuhkan kecepatan generator DC yang terkontrol. Kecepatan generator DC dapat dikontrol dengan mengubah tegangan yang diberikan ke armature atau dengan mengubah arus. Pengenalan resistansi variabel dalam angker rangkaian atau sirkuit lapangan memungkinkan kontrol kecepatan. Modern Generator DC sering dikendalikan oleh sistem elektronik disebut DC drive. Perhitungan kecepatan di motor penting. Hal ini dapat digunakan dalam proses kontrol. Kecepatan pengukuran adalah mengharuskan untuk proses pengendalian, pemantauan, pengujian, dll. Pendekatan konvensional untuk mengukur kecepatan adalah speedometer, seperti *encoders*, *tachometer*, dan sebagainya (Keramat and Ildarabadi, 2014).

Pendekatan kecepatan generator DC menggunakan sensor kecepatan yang sudah banyak digunakan peneliti-peneliti sebelumnya seperti yang dipaparkan dalam jurnal (Hardiyanto, 2015) dirasa kurang efektif. Sensor kecepatan atau *hall effect* merupakan jenis sensor yang dinamis atau bergerak. Perhitungan kecepatan dengan mengikuti kecepatan putar dari *shaft* generator dapat merusak sensor bila generator dalam keadaan kecepatan tinggi atau kecepatan yang tidak stabil. Pada

bidang industri membutuhkan *plan* dalam jumlah lebih dari satu, sehingga akan kesulitan disisi perawatan sensor yang letak sensor tidak boleh berjauhan dari *plan*.

Dalam penelitian ini, untuk mengatasi ketidakefektifan dari *hall effect* dengan melakukan perhitungan kecepatan menggunakan Observer. Sebuah observer kecepatan menggunakan pendekatan nilai tegangan dan arus pada generator, dimana besar tegangan dan arus generator digunakan untuk mengetahui nilai kecepatannya. Untuk mengetahui nilai tegangan dan arus, peneliti menggunakan sensor tegangan dan arus. Dimana sensor tegangan dan arus bersifat statis atau diam, sehingga dapat dipasang dalam keadaan *plan* jarak jauh ataupun jumlah *plan* yang lebih dari satu. Monitoring akan lebih mudah karena dapat dipasang dalam satu DC *drive*.

Suatu *plan* dengan kehandalan yang tinggi membutuhkan sistem pengendali sebagai kontrol. Dalam hal ini peneliti menerapkan kontrol algoritma PID pada observer kecepatan. Kontrol konvensional PID sudah banyak dibahas pada penelitian-penelitian sebelumnya salah satu dipaparkan pada jurnal (Alghoffary, 2014). Kontrol PID dapat mengatur suatu *plan* agar mendapatkan nilai *output* yang diinginkan dengan mendapatkan nilai *feedback* dari nilai *output* pada *plan* dan stabil. Oleh karena itu kontrol PID ini dapat diimplementasikan untuk mengontrol kecepatan pada generator DC bila terjadi penurunan maupun peningkatan beban, kecepatan generator akan kembali ke *set point* yang sudah ditentukan dengan *feedback* kecepatan generator DC.

Kontrol PID akan diterapkan pada *prime mover* yang dikopel dengan generator DC. Karena suatu sistem observer kecepatan pada generator DC membutuhkan pengendali kecepatan untuk tetap menjaga kecepatan tetap stabil saat beban berubah-ubah. Diharapkan dengan diterapkannya kontrol PID pada sistem generator DC, kecepatan akan lebih stabil saat beban berubah-ubah. Seiring dengan berkembangnya waktu sistem PID dapat direalisasikan pada *embedded system* seperti arduino uno R3.

1.2 Rumusan Masalah

Dalam penelitian ini ada beberapa hal yang menjadi rumusan masalah diantaranya :

1. Bagaimana merancang *speed observer* dengan analisis hubungan tegangan dan arus generator.
2. Bagaimana penerapan kontrol PID untuk membangun sistem kendali kecepatan pada *speed observer* generator.

1.3 Batasan Masalah

Berdasarkan uraian rumusan masalah tersebut, maka pembahasan pada skripsi dibatasi pada hal-hal sebagai berikut :

1. Generator yang diuji adalah generator DC magnet permanen.
2. *Air motor* sebagai *prime mover* dari generator DC.
3. Beban yang digunakan dalam pengujian adalah beban resistif
4. Komponen dan rangkaian elektronika yang digunakan hanya dibahas pada fungsi kerjanya.

1.4 Tujuan Penelitian

Penelitian ini dilakukan karena memiliki beberapa tujuan, diantaranya :

1. Untuk menganalisa tegangan dan arus *output* generator sebagai perancang *speed observer*
2. Membangun sistem kendali pada *speed observer* dengan menerapkan algoritma PID untuk dapat melakukan pengaturan kecepatan dengan respon kendali yang baik.

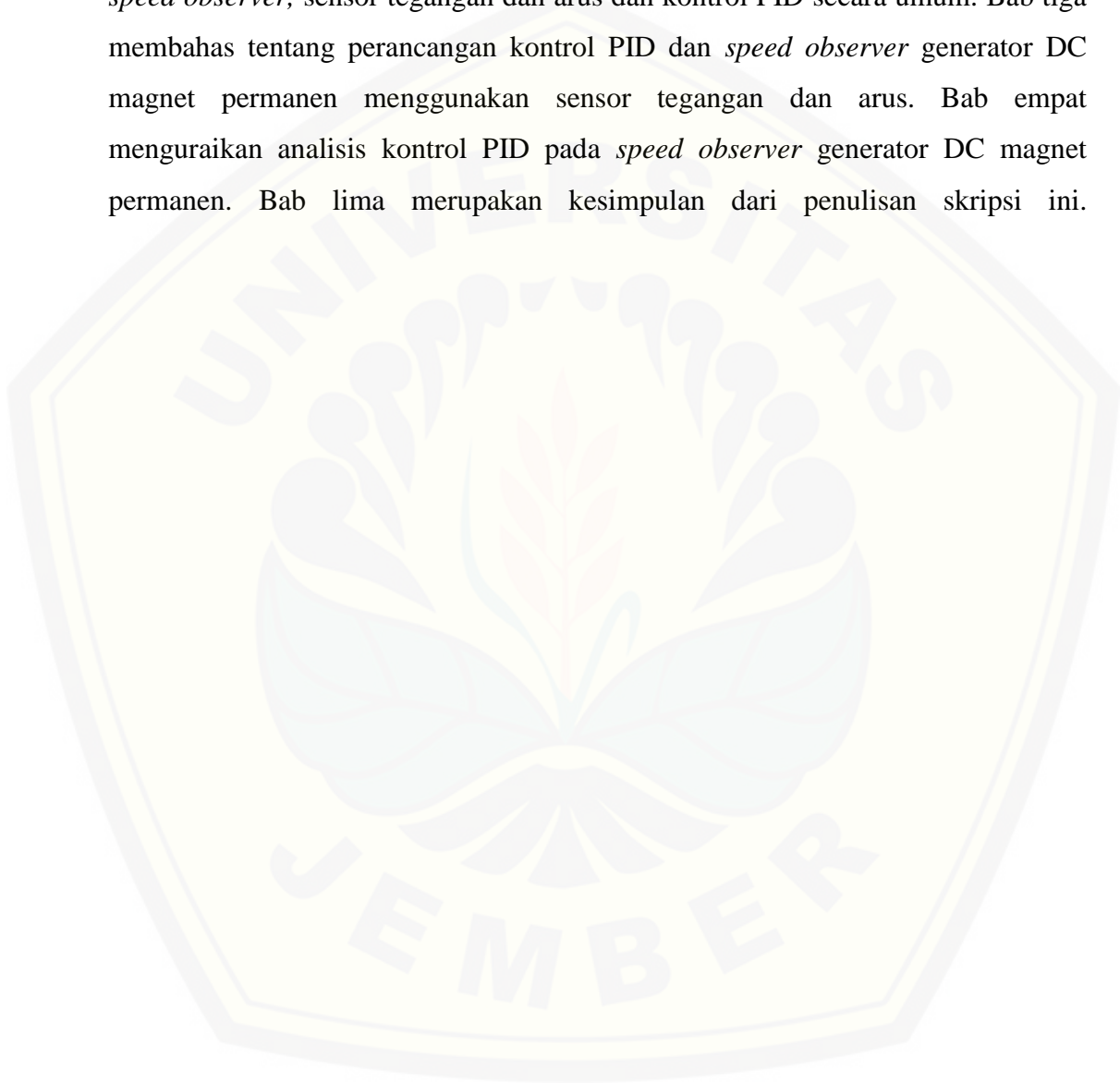
1.5 Manfaat

Adapun manfaat yang diperoleh dari penelitian ini memiliki diantaranya :

1. Sensor yang statis sebagai pengganti penggunaan sensor kecepatan yang dinamis pada generator.
2. Sebagai sumbangan pemikiran untuk pengembangan sistem *speed observer* yang diterapkan pada generator DC.

1.6 Sistematika Penulisan

Sistematika penulisan pada skripsi ini meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan di bab satu. Bab dua menjelaskan tentang penjelasan tentang, generator DC magnet permanen, *speed observer*, sensor tegangan dan arus dan kontrol PID secara umum. Bab tiga membahas tentang perancangan kontrol PID dan *speed observer* generator DC magnet permanen menggunakan sensor tegangan dan arus. Bab empat menguraikan analisis kontrol PID pada *speed observer* generator DC magnet permanen. Bab lima merupakan kesimpulan dari penulisan skripsi ini.



BAB 2. TINJAUAN PUSTAKA

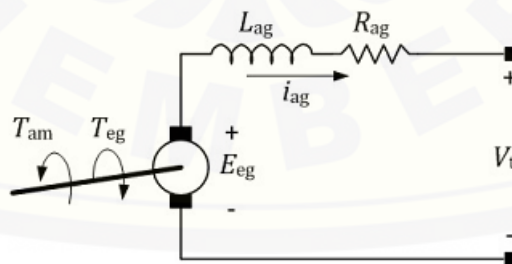
2.1 Generator *Direct Current* (DC)

Generator DC merupakan sebuah perangkat mesin listrik dinamis yang mengubah energi mekanis menjadi energi listrik yang menghasilkan arus DC atau arus searah. Generator DC dibedakan menjadi beberapa jenis berdasarkan dari rangkaian belitan magnet atau penguat eksitasinya terhadap jangkar (angker), jenis generator DC yaitu: generator penguat terpisah, generator shunt, generator kompon.

Berdasarkan cara dimana fluks medan yang dihasilkan, generator DC diklarifikasikan menjadi lima jenis antara lain:

1. *Separately excited generator* / Generator penguatan terpisah
2. *Shunt generator* / Generator Shunt
3. *Series generator* / Generator Series
4. *Cumulatively compounded generator*
5. *Differentially compounded generator*

Gambar 2.1 menunjukkan rangkaian untuk memodelkan PMDC yang dirangkai dengan sebuah penggerak mula dengan torsi mekanik pada poros sebesar T_{am} (Setiawan et al. 2015).



Gambar 2.1 Rangkaian Ekuivalen dari PMDC

$$T_{am} - (B_m + B_g)\omega_r - T_{eg} = (J_m - J_g)\frac{d\omega_r}{dt} \quad (2.1)$$

Dengan: T_{am} = torsi mekanik input dari penggerak (N-m)

T_{eg} = torsi elektromagnetik (N-m)

B_m = koefisien gesek air motor

B_g = koefisien gesek pada poros generator

J_m = konstanta inersia air motor

J_g = konstanta inersia generator

ω_r = kecepatan sudut (rad/s)

Persamaan (2.1) menyatakan hubungan dinamis antara daya mekanis *input* pada generator dengan kecepatan. Kecepatan ini penting pada generator karena tegangan *output* yang dihasilkan sebanding dengan kecepatan poros generator sebagaimana dinyatakan dalam persamaan-persamaan (2.2) (2.3) dan (2.4).

$$E_{eg} = k_m \omega_r \quad (2.2)$$

$$T_{eg} = k_e i_{ag} \quad (2.3)$$

$$V_t = E_{eg} - i_{ag} R_{ag} - L_{ag} \frac{di_{ag}}{dt} \quad (2.4)$$

Dengan: E_{eg} = gaya gerak listrik balik dari generator (V)

k_m = konstanta kecepatan generator (V/(rad/s))

k_e = konstanta torsi (N-m/A)

i_{ag} = arus pada jangkar generator (A)

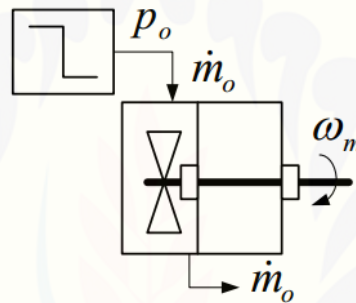
R_{ag} = resistansi jangkar generator (Ω)

L_{ag} = induktansi jangkar generator (H)

V_t = tegangan *output* terminal generator (V)

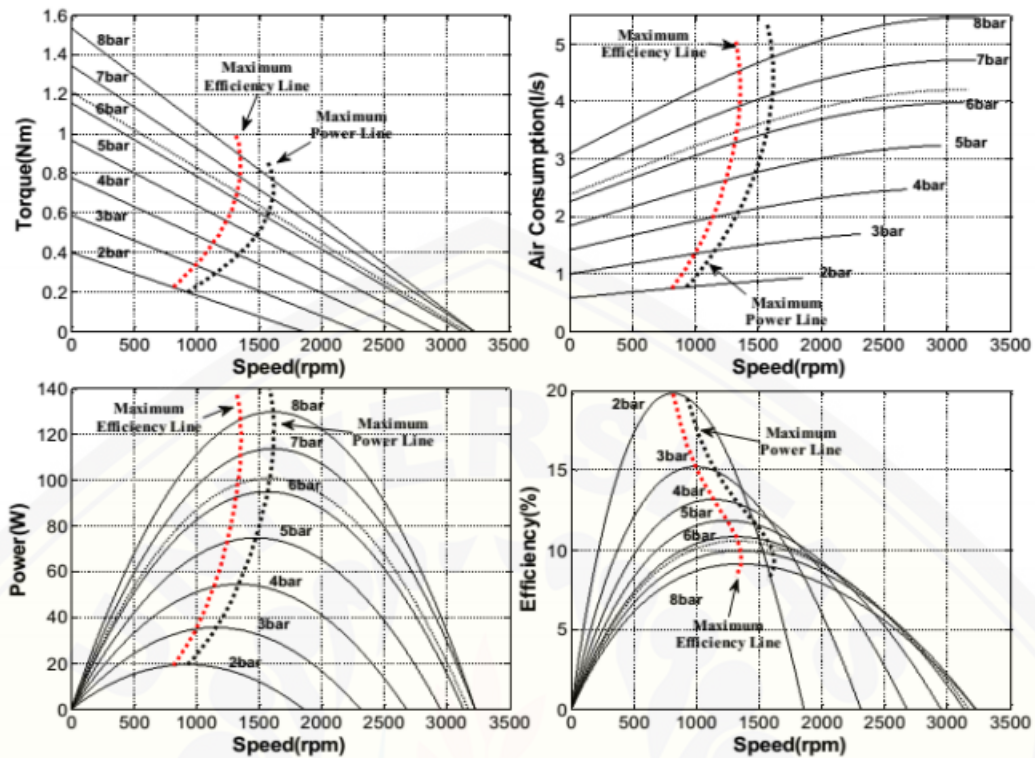
2.2 Air Motor

Secara umum, turbin bisa didefinisikan sebagai mesin putar yang bergerak dengan aliran fluida, dalam hal ini fluida tersebut adalah udara yang bergerak keluar dari tangki penyimpanan udara dan digunakan untuk menggerakkan turbin. Sistem *speed observer* dalam penelitian ini menggunakan turbin untuk mengubah energi pada aliran massa udara menjadi energi gerak putar. Poros turbin dikopel dengan poros generator untuk mengubah energi putar menjadi energi listrik. Gambar 2.2 menunjukkan gambar skematik dari air motor yang digunakan sebagai turbin pada generator DC dalam sistem *speed observer*.



Gambar 2.2 Skematik Air Motor

Dalam penelitian ini digunakan sebuah air motor sebagai turbin. Gambar 2.3 menunjukkan karakteristik sebuah air motor yang tipikal (Kokaewet *al*, 2013). Dalam gambar tersebut juga ditunjukkan bahwa terdapat garis efisiensi maksimum pada setiap kurva karakteristik yang dapat dipetakan pada nilai tekanan (*pressure*) dan kecepatan (*speed*).



Gambar 2.3 Karakteristik Tipikal Air Motor

(Kokaew *et al*, 2013)

Dalam grafik karakteristik, aliran massa udara yang masuk dan keluar dari air motor dinyatakan dalam aliran volume (*air consumption*) dengan satuan liter/second. Hal ini dikarenakan grafik tersebut adalah grafik dari sebuah produk air motor dengan standar industrial sehingga istilah yang digunakan untuk menyatakan satuan adalah istilah yang umum digunakan dalam industri.

Air motor atau motor udara yang digunakan pada penelitian ini adalah air motor atau motor udara dc yang bertipe prona, dimana sistem kerja alat ini adalah pada saat udara yang berasal dari kompresor disalurkan kepada air motor ini untuk menghasilkan putaran untuk memutar generator, gambar dan spesifikasi dari *air motor* ini dapat dilihat pada gambar 2.4 dan pada tabel 2.1 dibawah ini.



Gambar 2.4 Air Motor Prona

Berikut spesifikasi dari *air motor* prona yang akan dijelaskan seperti terlihat pada Tabel 2.1

Tabel 2.1 Spesifikasi Air Motor Prona

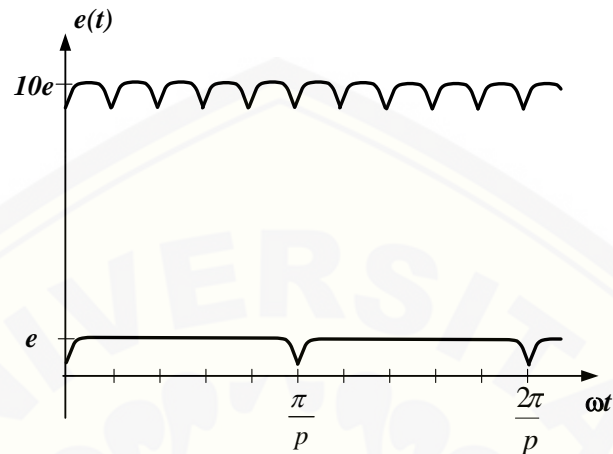
Type	Power		Pressure		Air Consumption	Shaft	Weight
	Hp	Kw	Bar (Mpa)	Speed rpm	l/Min	Diameter	Gram
M-10	0,08	0,06	3,5(0,34)	2200	200	7	450
M-20	0,16	0,12	7(0,68)	3000	380	10	1800
M-30	0,25	0,19	7(0,68)	2500	360	10	1600

2.3 Sistem Speed Observer

Sampai saat ini belum ada publikasi yang khusus membahas tentang observer yang digunakan untuk menggantikan sensor seperti yang akan dilakukan dalam penelitian yang diusulkan. Pembahasan dalam bagian ini berkaitan dengan observer yang sudah dikembangkan dalam penelitian lain yang dapat digunakan sebagai rujukan dan diaplikasikan dalam kegiatan penelitian ini.

Salah satu model observer kecepatan adalah observer yang bekerja berdasarkan prinsip komutasi. Hal ini dikarenakan pada generator arus searah yang bekerja dengan komutator dan brush selalu terjadi ripple, baik pada arus maupun pada tegangan *output* generator. Sistem observer yang sudah dikembangkan dan bekerja dengan prinsip ini adalah sebagaimana dibahas dalam

jurnal (Keramat and Ildarabadi, 2014). Gambar 2.5 memperlihatkan bentuk gelombang tegangan *output brushed dc generator* yang dapat dimanfaatkan sebagai dasar *speed observer*.



Gambar 2.5 Tegangan *Ripple* Sebagai Efek Komutasi

Metode analitis untuk *state observer* (posisi rotor, kecepatan dan arus) dengan menggunakan model analitis dari *brush motor dc* dibahas dalam makalah (Patil *et al*, 2013). Penelitian-penelitian ini murni menggunakan persamaan-persamaan dinamis dari mesin dc untuk digunakan membangun *state observer* dan mensimulasikannya dengan Matlab.

2.3.1 Regresi Linier Berganda

Regresi linier berganda adalah hubungan secara linear antara dua atau variabel independen (X_1, X_2, \dots, X_n) dengan variabel dependen (Y). Analisis ini untuk mengetahui arah hubungan antara variabel independen dengan variabel dependen apakah masing-masing variabel independen berhubungan positif atau negatif dan untuk memprediksi nilai dari variabel dependen apabila nilai variabel independen mengalami kenaikan atau penurunan. Data yang digunakan biasanya berskala interval atau rasio. Persamaan regresi linier berganda adalah sebagai berikut.

$$Y = a + b_1X_1 + b_2X_2 + \dots + b_nX_n \quad (2.5)$$

Dengan: Y = variabel terikat yang diduga
 X_1 dan X_2 = variabel bebas
 a = intersep
 b = koefisien regresi

Sedangkan untuk melihat bentuk korelasi antar variabel dengan persamaan regresi tersebut maka nilai a dan b harus ditentukan terlebih dahulu

$$b_1 = \frac{(\sum x_2^2)(\sum x_1 y) - (\sum x_1 x_2)(\sum x_2 y)}{(\sum x_1^2)(\sum x_2^2) - (\sum x_1 x_2)^2} \quad (2.6)$$

$$b_2 = \frac{(\sum x_1^2)(\sum x_2 y) - (\sum x_1 x_2)(\sum x_1 y)}{(\sum x_1^2)(\sum x_2^2) - (\sum x_1 x_2)^2}$$

$$a = Y(\text{avg}) - b_1 X_1(\text{avg}) - b_2 X_2(\text{avg}) \quad (2.7)$$

Persamaan regresi linier sederhana tersebut akan menjadi dasar dari pembentukan *speed observer* pada penelitian ini (Hasan, 2004).

2.4 Komponen Pengendali

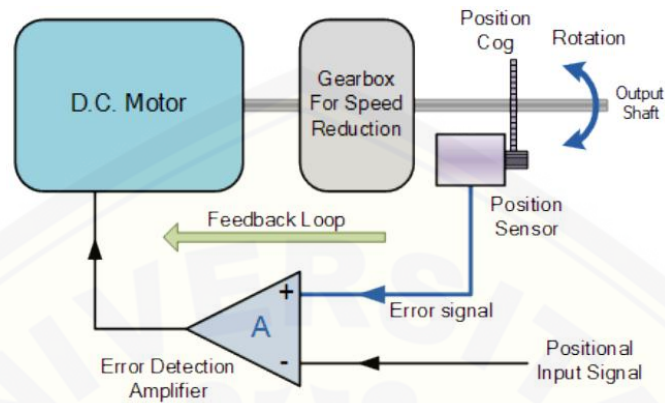
Dalam penelitian yang dilakukan terdapat komponen pengendali yang berfungsi sebagai pengatur kecepatan pada *speed observer* generator DC. Komponen pengendali tersebut antara lain.

2.4.1 Motor Servo

Motor servo adalah sebuah motor dengan sistem umpan balik tertutup dimana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor DC, serangkaian gear, potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor.

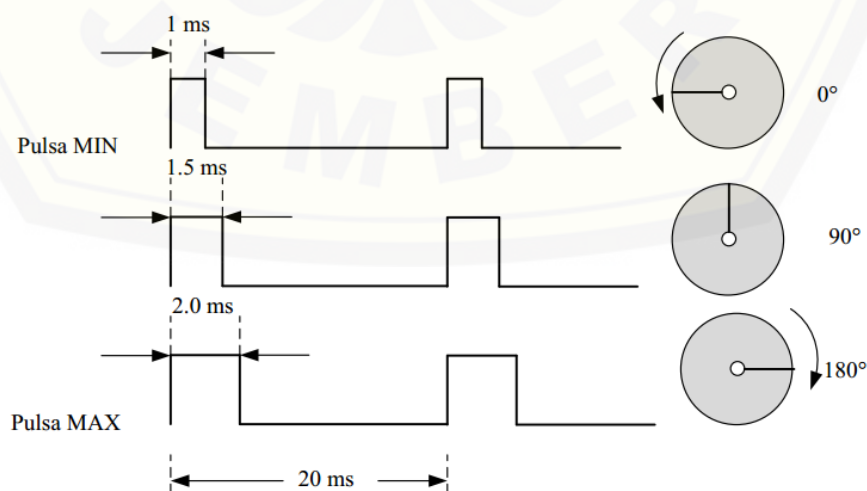
Secara umum, motor servo adalah dc, ac, atau brushless motor dc yang dikombinasikan dengan perangkat penginderaan posisi (misalnya dekode digital). Motor servo dilengkapi dengan circuit kendali dengan sistem umpan-balik yang menjadi satu kesatuan, motor servo. Gambar 2.6 menunjukkan diagram skematik

motor servo secara umum. Gambar tersebut menunjukkan komponen-komponen yang dikemas menjadi satu kesatuan dalam sebuah motor servo yang tipikal (ElectronicTutorial, 2015).



Gambar 2.6 Skematik Motor Servo Secara Umum

Umumnya DC servo memiliki tiga buah kabel yaitu untuk power (+VCC, biasanya +5V), Ground (0 volt) dan Control (pulsa) (Mechatronics, 2015). Dalam kaitannya dengan pengaturan putaran dan kecepatan, dikenal dua macam motorservo, yaitu motor servo standar (Standard Servo) yang dapat berputar dan mengatur posisi dengan sudut $0-180^\circ$ dan motor servo kontinyu (Continues Rotation Servo) (PmWiki 2015). Gambar 2.7 menunjukkan pulsa pengendali servo motor standar. Pengaturan posisi sebanding dengan ON-State dari pulsa dan kecepatan sebanding dengan tinggi dari pulsa.



Gambar 2.7 Pulsa Pengatur Motor Servo Standart

Dalam penelitian ini digunakan motor servo yang dapat berputar secara kontinyu (360°) dengan kode tipe DS04-NFC dapat dilihat pada Gambar 2.8 dengan spesifikasi sebagaimana terlihat dalam Tabel 2.2 dibawah ini.



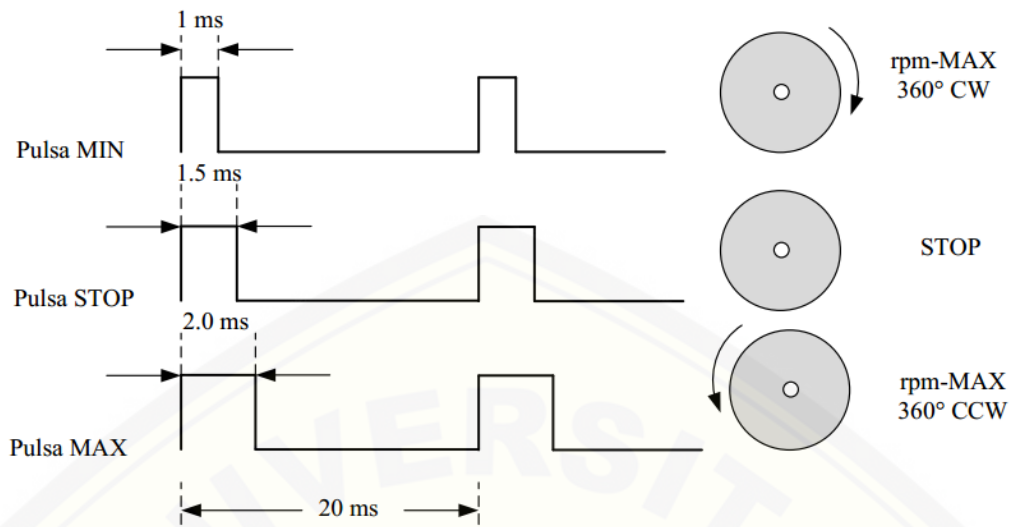
Gambar 2.8 Motor Servo 360° DS04-NFC

Tabel 2.2 Spesifikasi Teknis DS04-NFC

Type	DS04-NFC
Berat	38 g
Deimensi	39.5 mm× 20 ×40.8
Torsi	5.5 kg-cm (pada 4.8V)
Kecepatan	0.22 sec/60°(pada 4.8 V)
Range Tegangan Motor	4.8-6 Vdc

Diterjemahkan dari: (DafLabs 2015)

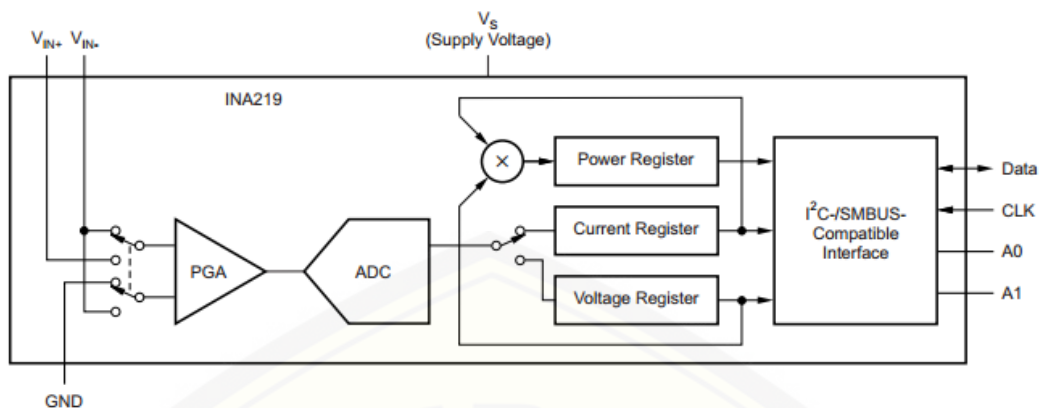
DS04-NFC dikendalikan gelombang persegi (kotak) dengan frekuensi 50Hz. Durasi dari sinyal tinggi (*high*) dapat menentukan kecepatan, pembalikan, dan penghentian. Sinyal tinggi dengan durasi 1 ms sampai dengan 1.5 ms akan menggerakkan gear maju (CW). Semakin kecil selisih durasi sinyal tinggi terhadap 1.0 ms, gerakan akan semakin cepat. Sebaliknya, semakin mendekati 1.5 ms, kecepatan akan semakin rendah. Dapat dilihat seperti Gambar 2.9.



Gambar 2.9 Sinyal Kendali DS04-NFC 360°

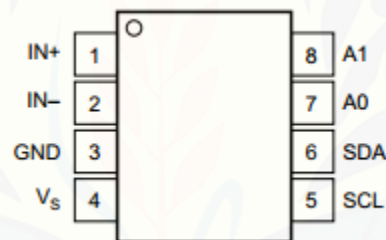
2.4.2 INA 219

INA 219 merupakan modul sensor yang dapat memonitoring tegangan dan arus pada suatu rangkaian listrik. INA 219 didukung dengan *interface* I²C atau SMBUS-COMPATIBLE dimana peralatan ini mampu memonitor tegangan shunt dan suplai tegangan bus, dengan konversi program *times* dan *filtering*. INA 219 memiliki sebuah amplifier *input* maksimum adalah $\pm 320\text{mV}$ ini berarti dapat mengukur sampai $\pm 3,2\text{ A}$. Dengan internal 12 bit ADC, resolusi pada \pm kisaran 3.2 A adalah 0.8 mA. Dengan gain internal yang ditetapkan pada minimum div8, maks saat ini adalah $\pm 400\text{mA}$ dan resolusi 0.1 mA. INA 219 mengidentifikasi tegangan shunt pada bus 0 – 26 V. Berikut merupakan gambar 2.10 skematik sederhana dari INA219. (Lady, 2016)



Gambar 2.10 Skematik INA 219

Dalam gambar 2.10 skematik INA 219 memiliki Pin I/O data, *clock*, analog 0, analog 1, V_{in+} , V_{in-} , ground, dan suplai tegangan. Berikut gambar 2.11 yang menjelaskan pin I/O dari INA.



Gambar 2.11 Konfigurasi Pin INA 219

Pin $IN+$ dan $IN-$ merupakan pin positif dan negatif input dari tegangan shunt dimana pin positif dihubungkan dengan hambatan shunt sedangkan yang negatif dihubungkan dengan *ground*. Pin SCL dan SDA adalah pin *serial bus clock line* dan *serial bus data line*. pin A0 dan A1 merupakan *address* dari pin analog input.

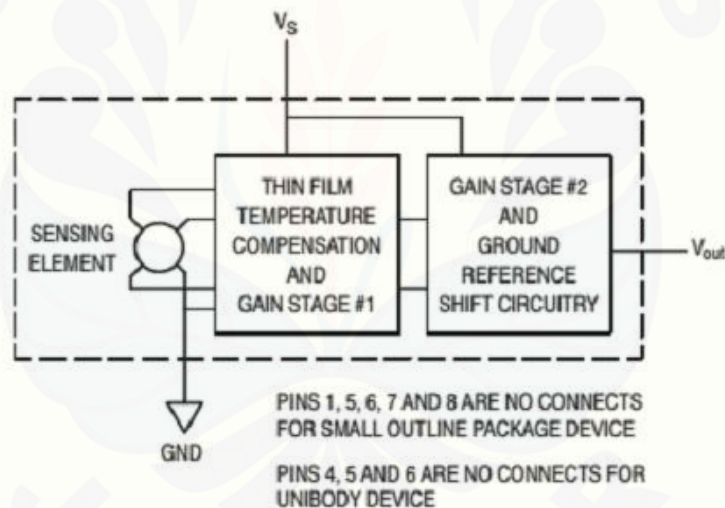
2.4.3 Sensor Tekanan Udara

Tranducer tekanan adalah perangkat sensitif yang berisi sirkuit diferensial tekanan yang terus-menerus mengirimkan pembacaan tekanan di dalam ruang. Kontroler menggunakan informasi ini untuk mengatur tekanan atau menghindari kondisi *over pressure*. Informasi tekanan juga dapat digunakan untuk

pengumpulan data, untuk memicu alarm, dan untuk melakukan tugas kontrol eksternal lainnya (anonymous 2015). Salah satu jenis sensor tekanan yang juga digunakan dalam penelitian ini adalah silicon pressure sensor yang bekerja dengan piezosensitive. Untuk membatasi pembahasan diambil acuan dari *datasheet* sensor MPX5700 yang digunakan dalam penelitian ini (Freescale 2015)



Gambar 2.12 Sensor Tekanan Udara MPX5700



Gambar 2.13 Diagram Fungsional MPX5700

MPX5700 adalah transduser tekanan yang bekerja dengan sistem piezoresistive dan merupakan hasil rancang bangun sistem sensor mutakhir yang dirancang untuk berbagai macam aplikasi, terutama yang menggunakan mikrokontroler atau mikroprosesor dengan A/D input. Transduser ini menggabungkan teknik-teknik canggih *micromachining*, metalisasi thin film, dan pengolahan bipolar untuk menghasilkan tegangan analog yang sebanding dengan tekanan diterapkan. Gambar 2.13 menunjukkan diagram fungsional dari sensor

tekanan ini, sedangkan tabel 2.3 memperlihatkan karakteristik operasi dari MPX5700 pada $V_S = 5.0 \text{ Vdc}$, $T_A = 25^\circ\text{C}$ (Freescale 2015).

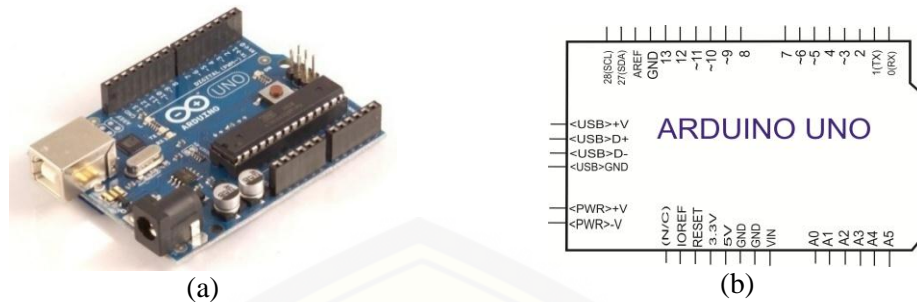
Tabel 2.3 Karakteristik Operasi MPX5700

Karakteristik	Simbol	Min	Tipikal	Maks	Satuan
Range Tekanan	Pop	15	-	700	kPa
Tegangan Suplai	V_S	4.75	5	525	Vdc
Arus Suplai	I_o	-	7	10	mADC
Offset Tekanan Nol Absolut (0 t0 85)	V_{off}	0.184	-	0.409	Vdc
Output Skala Penuh (0 t0 85)	VFSS	4.587	4.7	4.813	Vdc
Rentang Skala Penuh (0 t0 85)	VFSS	-	4.5		Vdc
Akurasi (0 t0 85)	-	-	-	± 25	%VFSS
Sensitivitas	V/P	-	6/4		mV/kPa
Waktu Tanggap	tR	-	1		ms
Arus Output Pada Output Skala Penuh	I_{o+}	-	0.1		mADC
Waktu Pemanasan		-	20		ms

2.4.4 Arduino Uno

Arduino Uno adalah modul mikrokontroler berbasis ATmega 328. Modul ini memiliki empat belas pin *input-output* (14 I/O) di mana enam diantaranya dapat digunakan sebagai *output pulse width modulation* (PWM), enam *input* analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, tombol *reset* dan koneksi ke catu daya. Kontroler ini semua yang dibutuhkan untuk mendukung mikrokontroler.

“Uno” berarti satu dalam bahasa Itali dan hal ini untuk menandai versi pertama (Arduino 1.0) yang akan menjadi versi acuan bagi Arduino berikutnya. Uno adalah seri terakhir dari seri papan atau modul USB Arduino. Uno tidak lagi menggunakan *chip* FTDI USB-to-serial tetapi menggunakan Atmega 8U2 sebagai konverter USB ke serial. Gambar 2.14 (a) adalah modul Arduino UNO-R3 sedangkan Gambar 2.14 (b) menunjukkan skema dari modul Arduino UNO-R3 yang merupakan penyederhanaan dari gambar modul aslinya (Arduino UNO 2015).



Gambar 2.14 Modul Kontroler Arduino UNO-R3

(Sumber : www.arduino.com)

Masing-masing pin I/O dapat digunakan sebagai *input* maupun *output* dengan menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Masing-masing pin ini beroperasi pada tegangan 5 volt dan dapat menerima arus 40 mA serta sudah memiliki *pull-up* resistor 20-50 k Ω yang secara *default* tidak tersambung (*disconnected*) (Arduino_My Weblog, 2012).

Berikut ini adalah tabel spesifikasi dari Arduino Uno R3 dapat kita lihat pada Tabel 2.4.

Tabel 2.4 Spesifikasi Arduino UNO-R3

Microcontroller	ATmega328
Tegangan Operasi	5 V
Tegangan Input	7-12 V
Tegangan Input(Batas)	60-20 V
I/O Digital	14 (6 buah menyediakan PWM output)
Input Analog	6 buah
Arus DC per I/O pin	40 Ma
Arus DC untuk pin 3.3	50 Ma
Flash Memory	Bootloader
SRAM	2 Kb
EEPROM	1 Kb
Kecepatan Klok	16 MHz

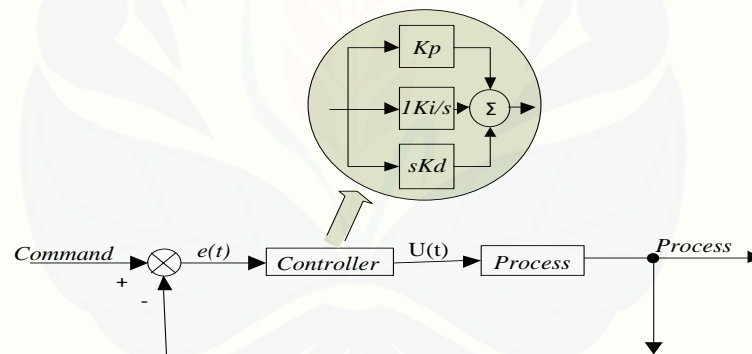
Diterjemahkan dari: (Arduino UNO 2015)

2.5 Kontrol PID

Fungsi kontrol adalah mengendalikan sistem dengan memanipulasi sinyal *error*, sehingga respon sistem (*output*) sama dengan yang kita inginkan (*input*). Komponen PID terdiri dari *propotional*, *integratif* dan *derivatif*.

2.5.1 Sistem Kontrol PID

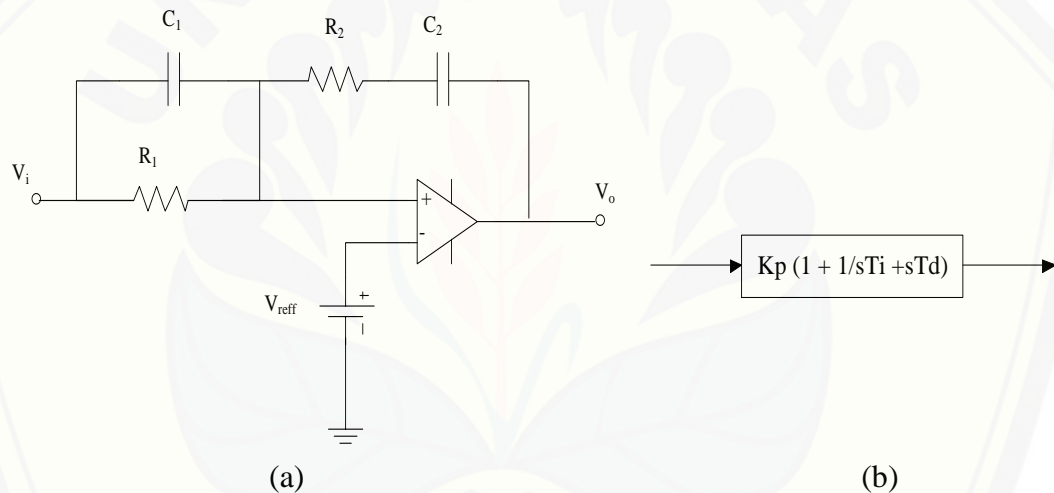
Pengendali PID telah secara luas digunakan dalam industri, yaitu hampir sebesar 90 % (Gunawan, 2009). Hal ini dikarenakan jika pengendali ini diatur dengan baik maka akan menghasilkan pengendalian PID yang sangat baik. Selain itu pengendali ini juga sederhana dan diimplementasikan dengan mudah. Skema pengendalian PID ditunjukkan oleh gambar 2.15. Pada gambar 2.15 menunjukkan bahwa pengendali PID merupakan jumlah dari tiga aksi kendali yaitu *propotional* yang ditunjukkan oleh K_p , integral yang ditunjukkan oleh K_i dan *derivative* yang ditunjukkan oleh K_d .



Gambar 2.15 Skema Pengendali PID

Masing – masing parameter dari PID ini memiliki fungsi dan perannya. *Propostional* akan mengurangi *rise time* dan akan mengurangi *steady state error*, *integral* dapat menghilangkan *steady state error*. Namun jika hanya kontrol *PI* saja yang digunakan maka akibatnya *transient response* akan bertambah buruk. Untuk mengatasi hal tersebut diperlukan parameter *derivative*. Peramater *D* ini akan memperbaiki *transient response* seperti *overshoot* dan *settling time*. Gabungan dari parameter – parameter ini disebut kontrol PID yang akan meningkatkan stabilitas sistem, mengurangi *overshoot* dan mempercepat *transient response*.

Bentuk kontrol PID dapat diimplementasikan dalam bentuk analog atau digital. Implementasi PID dalam bentuk analog secara dominan digunakan dalam bentuk komponen elektronika analog. Sedangkan bentuk PID digital diimplementasikan dalam bentuk digital, yaitu dengan *programming system* di dalam mikroprosesor atau *DSP*. PID analog dalam bentuk rangkaian elektronika analog ditunjukkan pada gambar 2.16, selain parameter PID dalam bentuk K_p , K_i dan K_d dapat juga dalam bentuk K_p , T_i , dan T_d . Untuk *tunning* parameter – parameter pada bentuk ini dapat dilakukan dengan mengatur nilai dari komponen pada rangkaian analog tersebut. Bentuk PID ini memiliki keunggulan murah, efisien yang tinggi dan kinerja dinamik yang cukup baik (Gunawan, 2009).



Gambar 2.16 (a) Rangkaian Elektronika PID Analog (b) Fungsi Transfer

$$K_p = -\frac{R_1 \cdot C_1 + R_2 \cdot C_2}{R_1 \cdot C_2} \quad (2.8)$$

$$T_i = R_1 \cdot C_1 + R_2 \cdot C_2 \quad (2.9)$$

$$T_d = \frac{R_1 \cdot C_1 \cdot R_2 \cdot C_2}{R_1 \cdot C_1 + R_2 \cdot C_2} \quad (2.10)$$

Penurunan persamaan PID digital didapat dengan melihat pada bentuk kontinunya. Persamaan PID dalam bentuk *laplace* adalah

$$U(s) = K_p \left[1 + \frac{1}{T_i s} + T_d s \right] E(s) \quad (2.11)$$

jika persamaan ini dikembalikan dalam fungsi waktu akan menjadi

$$U(t) = K_p \left[e(t) + \frac{1}{T_i} \int e(t) + T_d \frac{de(t)}{dt} \right] \quad (2.12)$$

Untuk mengubah persamaan ini ke dalam bentuk diskrit. Dapat dilakukan pendekatan diskrit dengan perioda T_c , sehingga didapatkan bentuk *derivative*

$$\frac{de}{dt} = \frac{e(k) - e(k-1)}{T_c} \quad (2.13)$$

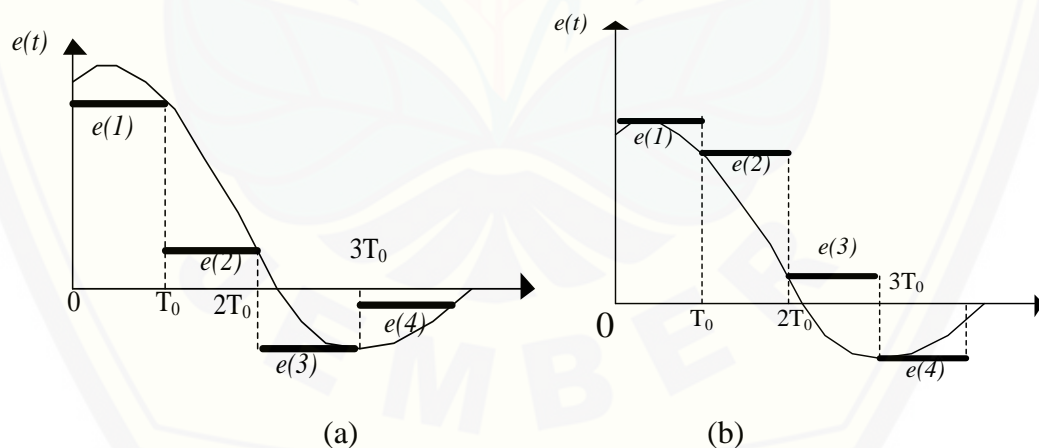
sedangkan untuk integralnya dilakukan dengan beberapa metode, diantaranya adalah *forward rectangular method* (FRM) dan *Backward Rectangular Method* (BRM). Pencuplikan dengan FRM dan BRM digambarkan oleh gambar 2.17. Dari gambar 2.17 bisa didapatkan algoritma *integral* untuk masing – masing metode.

Untuk FRM didapatkan

$$\int_0^T e(t)dt \approx T_c \sum_{i=1}^k e(i-1) \quad (2.14)$$

sedangkan untuk BRM didapatkan

$$\int_0^T e(t)dt \approx T_c \sum_{i=1}^k e(i) \quad (2.15)$$



Gambar 2.17 Metode Pencuplikan (a) *Backward Rectangular Methode* (b) *Forward Rectangular Methode*

Persamaan 2.11 lebih sering digunakan dalam menulis persamaan PID. Memakaipersamaan 2.11 ini dan persamaan 2.9 untuk menggantikan persamaan 2.9didapatkan *absolute/position algorithm* untuk pengendali PID

$$u(k) = K_p \left\{ e(k) + \frac{T_c}{T_i} \sum_{i=1}^k e(i) + \frac{T_d}{T_c} [e(k) - e(k-1)] \right\} \quad (2.16)$$

Pengendali PID pada persamaan 2.14 disebut *nonrecurrent algorithm* dimana nilai semua nilai error perlu diketahui untuk menghitung parameter integral yang mengakibatkan ketidakefisienan penggunaan memori. Untuk mengatasi hal ini, salah satu cara yang digunakan adalah dengan cara memecah komponen pengendali menjadi tiga komponen yaitu

$$u(k) = U_p(k) + U_i(k) + U_D(k) \quad (2.17)$$

dimana,

$$\begin{aligned} U_p(k) &= K_p \cdot e(k) \\ U_i(k) &= K_p \frac{T_c}{T_i} \sum_{i=1}^k e(a) = U_i(k-1) + K_p \frac{T_c}{T_i} e(k) \\ U_D(k) &= K_p \frac{T_d}{T_c} [e(k) - e(k-1)] \end{aligned} \quad (2.18)$$

dengan cara ini, hanya nilai terakhir dari komponen *integral* saja yang perlu disimpan untuk komponen *integral*. Kelebihan dari cara ini adalah komponen *integral* yang sudah terpisah juga bisa dibatasi untuk mencegah terjadinya integral *wind-up* (Gunawan, 2009).

2.5.2 Tuning Parameter PID

Tuning kontrol PID ini bertujuan untuk menentukan parameter aksi kontrol *Proportional*, *Integratif*, *Derivatif* pada *plan*. Proses ini dapat dilakukan dengan cara *trial and error*. Keunggulan cara ini kita tidak perlu mengidentifikasi *plan*, membuat model matematis *plan*, menentukan parameter *plan* dengan grafis ataupun analitis, cukup dengan cara coba-coba memberikan konstanta P-I-D pada formula PID hingga diperoleh hasil yang diinginkan, dengan mengacu pada karakteristik masing – masing kontrol P-I-D.

Tabel 2.5 Karakteristik masing – masing kontrol P-I-D

<i>Respon loop tertutup</i>	<i>Rise time</i>	<i>Overshoot</i>	<i>Settling time</i>	<i>Error steady-state</i>
K_p	Mengurangi	Meningkatkan	Mengurangi	Mengurangi
K_i	Mengurangi	Meningkatkan	Meningkatkan	Menghilangkan
K_d	Perubahan Kecil	Mengurangi	Menurun	Perubahan Kecil

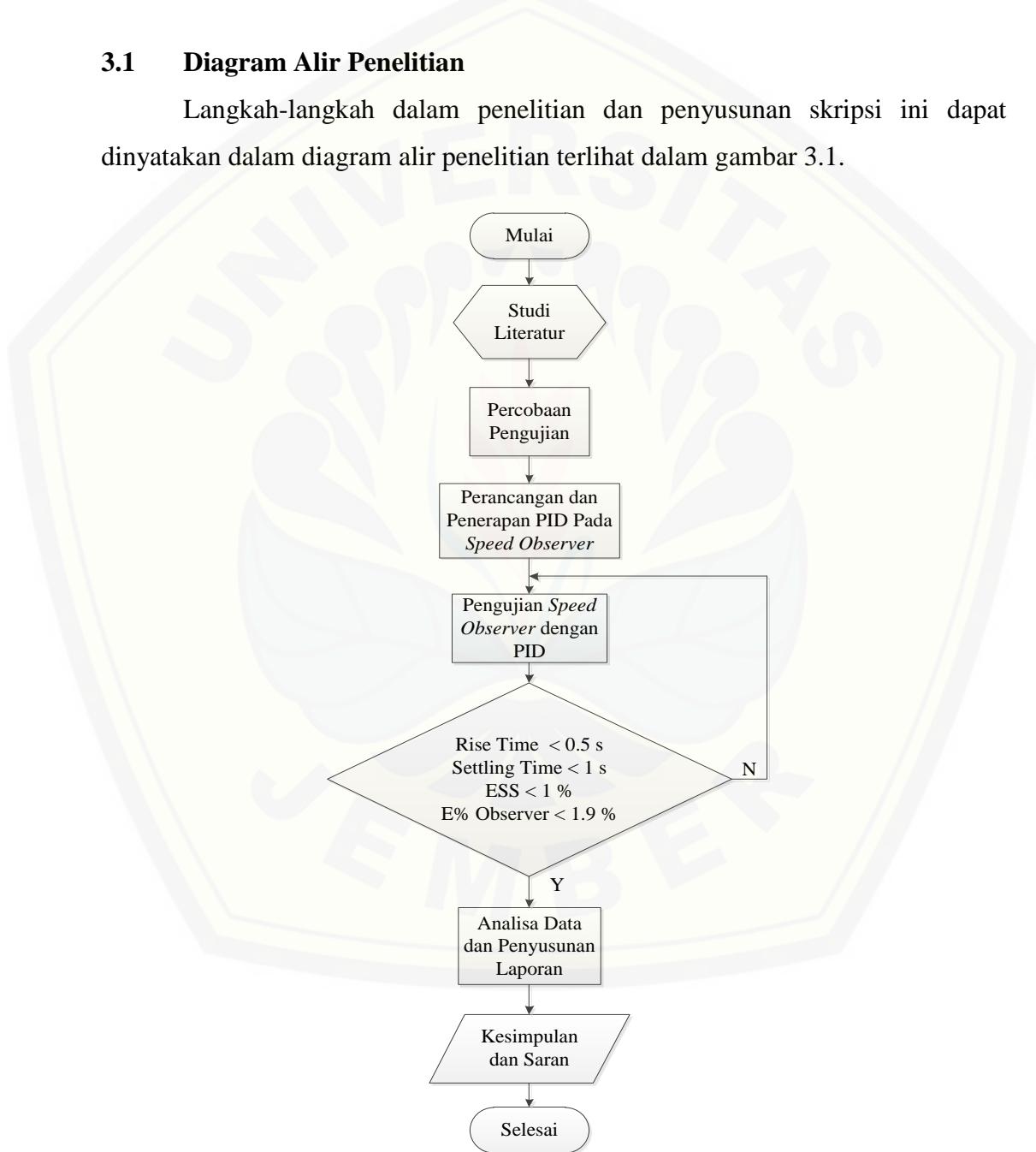
Rise time adalah ukuran waktu yang menyatakan keberadaan suatu respon, yang di ukur mulai respon 5 % s/d 95 % dari respon *steady state* (dapat pula 10 % s/d 90 %). *Overshoot* adalah nilai tertinggi dari *output* respon. *Settling time* adalah ukuran waktu yang menyatakan respon telah masuk 5 % atau 2 % atau 0,5% dari respon *steady state*. *Error steady-state* adalah persentase kesalahan pada saat respon *steady state*. Pada penelitian sebelumnya yang berjudul “Implementasi kontroler PID pada AVR (*Automatic Voltage Regulator*) untuk pengaturan tegangan eksitasi generator sinkron 3 fasa”, kondisi *transient* dari *plan* memerlukan 1.3 s untuk mencapai *steady-state* saat generator dibebani dan 1.1 s untuk mencapai *steady-state* saat beban pada generator dilepas. *Overshoot* dan *undershoot* dari respon generator memenuhi standar toleransi tegangan yaitu -10 % dan +5 %. Saat dibebani tegangan turun 9.5 % dan saat beban dilepas tegangan naik 5.2% dari tegangan nominal generator (Jayaet al, 2011).

BAB 3. METODOLOGI PENELITIAN

Mencapai hasil penelitian yang sesuai dengan tujuan yang diharapkan, tahapan dari penelitian ini selanjutnya direncanakan dalam tahapan – tahapan yang berisi kegiatan penelitian sebagaimana dijelaskan dalam bagian berikut.

3.1 Diagram Alir Penelitian

Langkah-langkah dalam penelitian dan penyusunan skripsi ini dapat dinyatakan dalam diagram alir penelitian terlihat dalam gambar 3.1.



Gambar 3.1 Flowchart Penelitian

Tahapan yang dilakukan pada penelitian implementasi PID *speed observer* generator DC magnet permanen berbasis Arduino Uno R3

1. Studi literatur

Tahap pertama memulai penelitian ini dilakukan tahapan mencari literatur yang berkaitan dengan penelitian ini, dengan harapan perancangan dan pelaksanaan penelitian dapat sesuai dengan yang diharapkan.

2. Percobaan pengujian

Tahap kedua adalah percobaan pengujian. Pengujian ini dilakukan terhadap sensor arus, sensor tegangan, dan kecepatan generator.

3. Perancangan dan penerapan PID pada *speed observer*

Tahap ketiga adalah proses perancangan kontrol PID pada *speed observer*. Pembuatan parameter – parameter agar kontrol PID bekerja dengan baik pada sistem. Parameter terdiri dari *rise time*, *settling time*, *error steady-state* dan *peak*. Penerapan PID pada *speed observer* generator DC magnet permanen sebagai pengendali kecepatan yang memiliki respon kontrol yang baik

4. Pengujian dan pengambilan data

Tahap keempat adalah pengujian dan pengambilan data terhadap sistem yang telah dibuat seperti sensor tegangan, sensor arus, dan generator yang sudah disertai dengan kontrol PID.

5. Analisa data dan pembuatan laporan

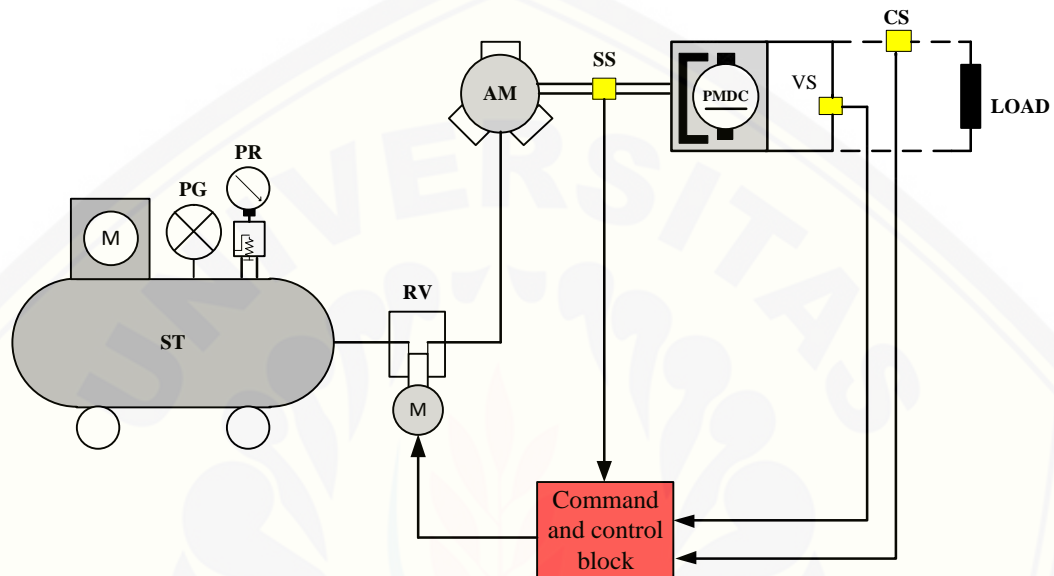
Tahap kelima adalah melakukan analisa pada data yang diperoleh dari tahap sebelumnya dan melaporkan hasil yang telah dicapai dalam bentuk laporan skripsi.

6. Penarikan Kesimpulan

Tahap keenam adalah tahap terakhir pada penelitian ini yaitu memberikan kesimpulan dari pengujian dan analisa data yang diperoleh, memberikan saran untuk penelitian dengan tahap selanjutnya.

3.2 Perancangan dan Blok Diagram Sistem

Dari penelitian yang akan dilakukan, terlebih dahulu membuat perancangan alat dan *hardware*. Perancangan alat yang akan di uji pada penelitian dapat dilihat perancangan alat yang akan didesain seperti gambar 3.2 sebagai berikut.



Gambar 3.2 Rancangan Penelitian Alat

Keterangan Gambar

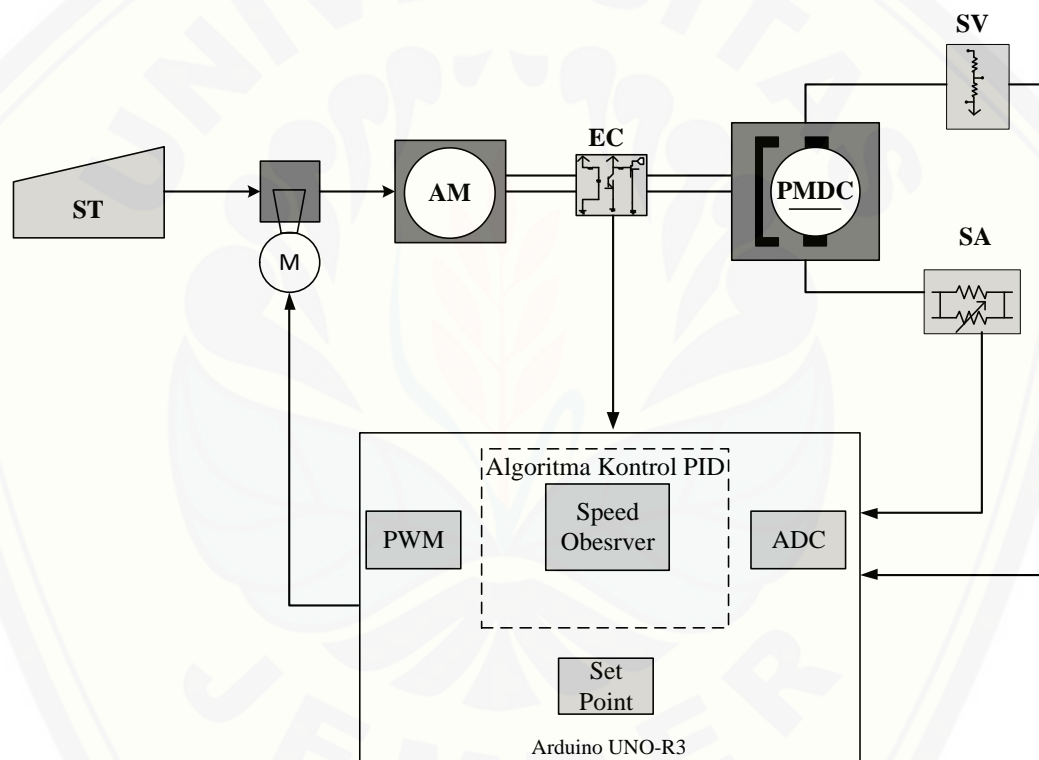
- ST = *Storage Tank* (Tangki penyimpan udara)
 PG = *Pressure gauge* digunakan untuk mengamati tekanan
 PR = *Pressure Relief* untuk membatasi tekanan yang keluar dari tangki penyimpan sebelum masuk ke *regulated valve* (RV)
 RV = *Regulated Valve* adalah katub pengatur tekanan dan aliran udara.
 AM = *Air Motor* yang digunakan sebagai prime mover dari generator.
 PMDC = *Permanent Magnet DC generator* adalah NC5475C
 SS = *Encoder* digunakan untuk deteksi kecepatan dan memberikan umpan balik pada kontroler.

- CS = Sensor arus digunakan untuk deteksi arus output generator sebagai informasi ke kontroler.
- VS = Sensor tegangan untuk mendeteksi tegangan output generator sebagai informasi ke kontroler.
- LOAD = Rangkaian beban DC berupa rangkaian resistor maupun resistor geser.
- Kontrol = Kontroler algoritma PID pada Arduino UNO R-3.

Prinsip kerja dari penelitian ini dapat digambarkan pada blok diagram sistem seperti terlihat pada gambar 3.3. Ketika *prime mover* yaitu *air motor* diberi aliran udara dari kompresor akan berputar dan juga memutar generator, maka akan menghasilkan tegangan dan arus *output* tertentu. Dalam perancangan ini akan diberi beban yang bervariasi sehingga didapat arus *output* generator yang berubah-ubah. Nilai tegangan dan arus yang terbaca oleh sensor tegangan (SV) dan sensor arus (SA), kemudian dikonversikan menjadi nilai kecepatan putaran generator. Persamaan kecepatan dirubah ke persamaan deskrit agar dapat diolah ke dalam arduino uno R3 yang akan dijadikan *speed observer*. Dalam perancangan ini digunakan motor servo yang bekerja seperti *valve* (RV) guna mengatur tekanan udara yang masuk ke *air motor*. Untuk mengatur kecepatan *air motor* yaitu mengatur putaran motor servo pada *valve*(RV).

Putaran motor servo tergantung pada besarnya sinyal PWM yang diberikan. Agar putaran motor servo lebih halus maka diberikan algoritma PID yang diprogram pada arduino uno R3, yang akan memberikan sinyal PWM ke motor servo untuk mengatur sudut putar motor servo. Saat terjadi perubahan beban, kecepatan generator bisa berubah dan jika diinginkan kembali ke nilai kecepatan sesuai dengan yang diinginkan (*setpoint*), maka diperlukan sistem kendali umpan balik yang akan mengubah RV agar menghasilkan tekanan yang menyebabkan kecepatan sesuai dengan nilai yang diinginkan. Informasi kecepatan aktual didapat dari sensor encoder (EC). Sinyal ini akan diolah dalam kontroler sehingga dapat memberikan perintah pada RV. Dengan adanya kontrol PID sistem pengaturan kecepatan pada *speed observer* akan lebih terintegrasi dan menjadikan

suatu *plan* yang lebih baik. Penggunaan dari algoritma PID sebagai kontrol kecepatan pada *Speed observer* generator ditinjau dari pemodelan *plan* yang menggunakan *air motor* sebagai *prime mover* generator DC. Dibutuhkan tekanan udara sebagai tenaga untuk pemutar *air motor*, dengan tekanan udara yang kurang stabil maka diterapkan kontrol PID pada motor servo yang bekerja sebagai pengatur tekanan udara yang akan masuk ke *air motor* agar kecepatan dari generator menjadi lebih stabil. Peralatan ini digunakan untuk mensimulasikan sistem pengaturan kecepatan pada *speed observer*. Secara diagram blok sistem perancangan ini dapat dilihat seperti pada gambar 3.3.



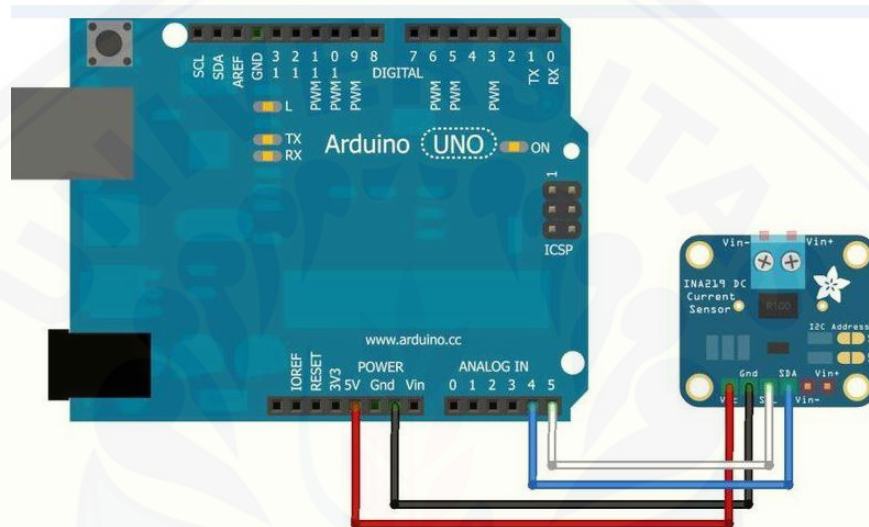
Gambar 3.3 Diagram Blok Sistem

3.3 Perancangan Hardware

Dalam bagian ini akan dibahas mengenai pengujian perangkat keras yang diperlukan dan dibuat dalam penelitian ini. Hasil pengujian disajikan dan dibahas dalam Bab 4.

3.3.1 Perancangan Rangkaian INA 219

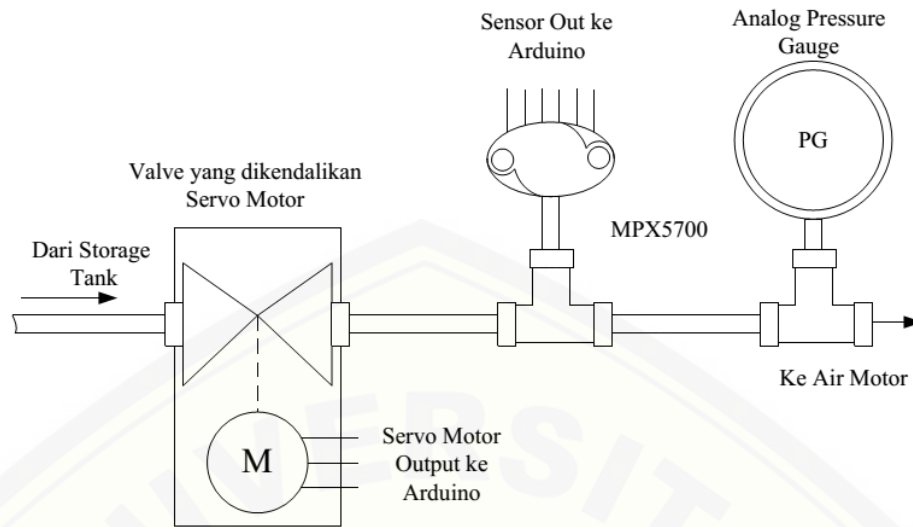
INA 219 pada penelitian ini digunakan untuk mengidentifikasi tegangan dan arus *output* pada generator DC. Dimana pin I/O sudah dijelaskan pada bab sebelumnya. Pin yang digunakan adalah VCC 5 V, GND, A0 dan A1, SCL dan SDA yang akan dihubungkan dengan *board microcontroller* berupa arduino uno R3. Berikut *wiring* rangkaian INA 219 dengan arduino UNO R3.



Gambar 3.4 *Wiring* Rangkaian INA 219 Pada Arduino

3.3.2 Perancangan Sensor Tekanan

Sensor tekanan dibuat dengan berbasis pada komponen MPX5700 yang merupakan sensor tekanan silikon yang bekerja dengan piezosystem. Gambar 3.5 menunjukkan penempatan dan rangkaian sensor ini ke sistem. Sensor ditempatkan setelah sistem *valve* yang diatur dengan menggunakan motor servo. Hal ini dimaksudkan agar tekanan yang terukur adalah tekanan udara yang akan masuk inlet dari air motor, bukan lagi tekanan output dari relief yang masih harus melewati selang udara. Untuk dapat menguji dan melakukan kalibrasi, pada percobaan ini digunakan analog *pressure gauge* untuk membandingkan hasil pengukuran sistem sensor MPX5700. Hasil pengukuran ini dibandingkan dilakukan kalibrasi pada rumus konversi bit ke nilai tekanan dalam kPa yang ditampilkan pada PC melalui Arduino UNO-R3.



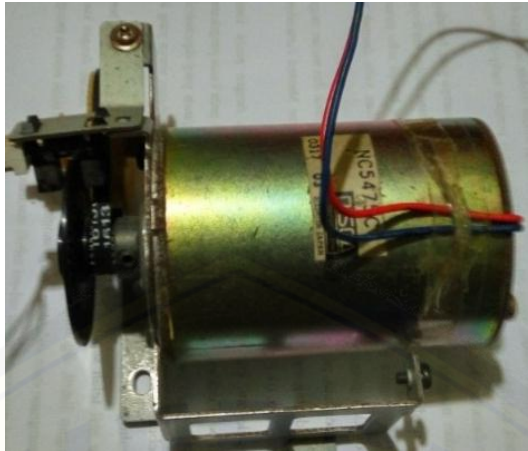
Gambar 3.5 Rangkaian MPX5700 Ke Sistem

3.3.3 Arduino UNO R3

Sistem mikrokontroler yang digunakan dalam perancangan *embedded system* pada penelitian ini adalah arduino. Tipe arduino yang digunakan adalah arduino UNO R3. Untuk menerima sinyal *feedback* tegangan yang dikeluarkan oleh generator DC, dapat menggunakan fasilitas ADC yang terdapat pada pin A0 – A5. Pada penelitian ini digunakan pada fasilitas ADC pada pin A0 dan A2. Untuk memberikan sinyal PWM pada motor servo dapat menggunakan pin – pin PWM pada arduino UNO R3, yaitu pada pin 3, 5, 6, 9 dan 10. Pada penelitian ini digunakan pin 3, karena menurut website resmi arduino frekuensi PWM yang dapat dihasilkan pada pin PWM ini adalah 31,250 Hz (Arduino LLC, 2015).

3.3.4 Generator DC Magnet Permanen

Generator yang digunakan dalam penelitian ini adalah sebuah mesin DC NISCA NC5475 yang berdasarkan spesifikasi yang bisa diperoleh adalah NISCA NC5475. 24V, 3200 rpm, 2.5A *full load*, 0.33A *No-Load*, *Starting torque* 0.392 N-m, *Nominal torque* 0.147 N-m Bentuk fisik dari mesin DC NISCA NC5475 seperti terlihat pada gambar 3.6.



Gambar 3.6 Generator DC Magnet Permanen NC7554

3.4 Sistem *Speed Observer*

Untuk menentukan estimasi nilai kecepatan dari generator DC magnet permanen, maka dapat dilakukan dengan menggunakan persamaan yang sudah dijelaskan pada bab sebelumnya yaitu persamaan (2.2) dan (2.4). sehingga didapatkan persamaan estimasi nilai kecepatan (Patil, Jogakelar, Jagtap, & Jadhav, 2013):

$$\omega_r = \frac{V_t + (i_{ag}R_{ag}) + (L_{ag} \frac{di_{ag}}{dt})}{k_m} \quad (3.1)$$

Dengan parameter dan pengujian generator yang akan dilakukan pada penelitian ini dapat dilihat pada tabel 3.1.

Tabel 3.1 Kalkulasi Parameter Generator

Parameter	Pengujian
R_{ag}	Resistansi angker dihitung $\frac{V_{ag}}{I_{ag}}$
L_{ag}	menggunakan LCR meter
k_m	Konstanta kecepatan generator

Persamaan kecepatan generator 3.1 masih dalam satuan rad/s yang harus dirubah ke dalam satuan rpm dengan (Iqbal Hasan, 2004)

$$SO = \frac{30}{\pi} \omega r \quad 3.2$$

Sehingga

$$SO = \frac{Vt + ia.Ra + La \frac{dia}{dt}}{K1} \quad 3.3$$

Dimana

$$K1 = Km \frac{\pi}{30} \quad 3.4$$

Untuk kondisi *steady state* $\frac{dia}{dt} = 0$ maka

$$SO = \frac{Vt + ia.Ra}{K1} \quad 3.5$$

$$SO = Vt.K2 + Ia.Ra.K2 \quad 3.6$$

$$K2 = \frac{1}{K1}$$

Karena Ra adalah konstan maka

$$SO = Vt.K2 + Ia.K3 \quad 3.7$$

$$K3 = Ra.K2$$

Dengan regresi linier maka hubungan kecepatan dengan tegangan dan arus dapat ditulis dengan persamaan linier sebagai berikut.

$$SO = a + b1.Vt + b2 Ia \quad 3.8$$

Dimana : $b1 = K2$

$$b2 = K3$$

Keterangan : SO = Variabel Terikat

a = Intersep

b = Koefisien Regresi

Vt = Variabel Bebas (Vt)

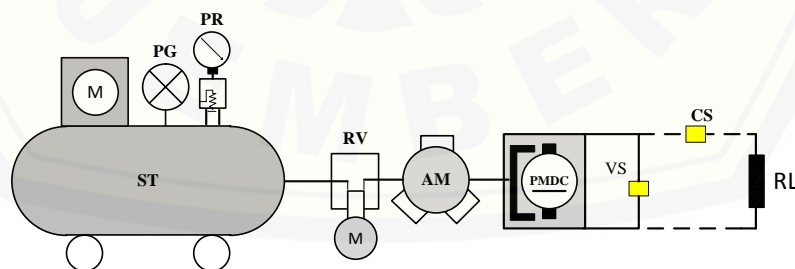
Ia = Variabel Bebas (Ia)

Pada persamaan 3.8 SO merupakan *speed observer* dimana dalam regresi merupakan variabel terikat, variabel yang di duga, b_1 dan b_2 merupakan koefisien regresi sedangkan V_t dan I_a merupakan tegangan dan arus output yang dibaca oleh sensor, pada persamaan regresi biasa disebut Variabel bebas.

Variabel-variabel dari persamaan 3.3 disesuaikan dengan yang dibutuhkan dalam peneliatn ini. Dimana persamaan regresi dari hubungan arus dan tegangan dapat dilihat pada persamaan 3.3 (Math Works). Sebelum melakukan *running* pada matlab membuat *workspace* data kecepatan, tegangan dan arus *output* pada generator DC. Persamaan – persamaan regresi linier pada matlab dapat diulis seperti “ $X = [\text{ones}(\text{size}(\text{volt})) \text{ volt curr}]$,” dengan X adalah variabel terikat atau variabel yang akan diduga dan “ $b = \text{regress}(\text{speed}, X)$ ” sebagai penentu nilai intersep dari regresi.

3.5 Hubungan Tegangan Arus dan Daya Generator DC

Daya dalam fisika adalah laju energi yang dihantarkan atau kerja yang dilakukan per satuan waktu. Dalam hal ini generator DC sebagai penghasil daya listrik. Daya akan bernilai tinggi apabila tegangan dan arus *output* generator DC bernilai tinggi pula. Untuk mengetahui daya yang dihasilkan dibutuhkan sensor INA 219 sebagai pencuplik tegangan dan arus *output* generator DC. Berikut perancangan hubungan tegangan arus dan daya generator dapat dilihat pada gambar 3.7



Gambar 3.7 Perancangan Tegangan Arus dan Daya *Output* Generator DC.

Dapat dijelaskan bahwa untuk dapat mencuplik tegangan dan arus *output* yaitu untuk tegangan dirangkai secara paralel sedangkan untuk arus dirangkai secara seri. Persamaan – persamaan hubungan tegangan arus dan daya dapat ditulis sebagai berikut

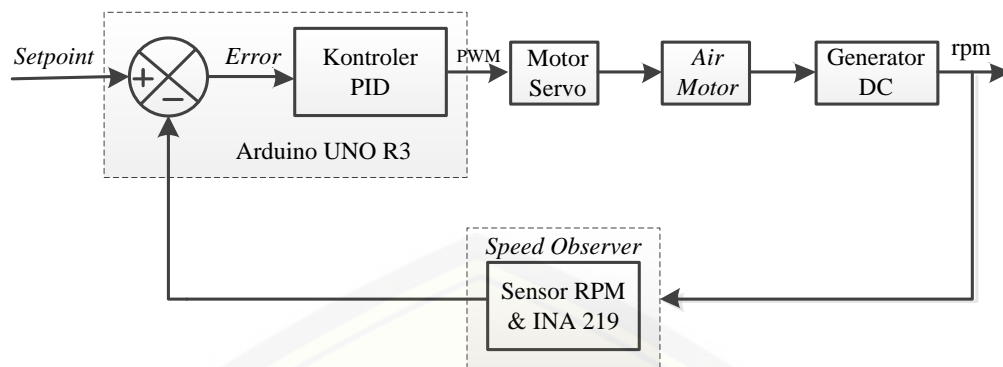
$$P(\text{watt}) = V * I ; P(\text{watt}) = I^2 * R ; I = V/R \quad 3.9$$

Dimana persamaan 3.9 menunjukkan hubungan tegangan, arus dan daya *output* generator dc dengan keterangan P merupakan daya yang dihasilkan generator DC dalam satuan watt, V merupakan tegangan *output* dalam satuan volt, I merupakan arus *output* dalam satuan ampere dan R merupakan beban resistansi dalam satuan ohm. Beban yang akan digunakan dalam penelitian ini merupakan beban RL atau resistor dengan dua variasi beban yaitu sebesar 50 Ω dan 25 Ω

3.6 Kontrol PID

Perancangan kontrol PID diskrit diperlukan algoritma program kontrol PID agar menjadi kontrol yang sesuai yang akan dimasukkan ke dalam program arduino. Program algoritma PID pada arduino meliputi proses inialisasi, tuning parameter, akumulasi error dan perhitungan PID sebagai proses berjalannya motor servo yang digunakan untuk mengatur aliran tekanan udara yang masuk ke *air motor* sehingga kecepatannya dapat diatur.

Pada penelitian ini untuk menentukan parameter – parameter PID digunakan metode *manual tuning* yaitu dengan cara *trial and error*. Kontrol PID yang akan digunakan pada penelitian ini menggunakan *delay* pada arduino. PID ini akan diterapkan pada motor servo yang berguna sebagai pembuka atau penutup tekanan udara *storage tank* untuk memutar *air motor* yang digunakan sebagai *prime mover* pada penelitian ini. Untuk perancangan sistem kontrol PID dapat dilihat pada gambar 3.8 yaitu diagram perancangan sistem PID pada *plan*.



Gambar 3.8 Diagram Perancangan Sistem PID terhadap *Plan*

Masukan dari kontrol PID adalah *error* RPM generator DC. *Error* akan diolah oleh Algoritma kontrol PID sehingga menghasilkan sinyal PWM yang diumpankan ke motor servo, dimana motor servo akan bekerja sesuai dengan sinyal kontroler PWM yang diterima. Untuk program sistem secara lengkap dapat dilihat pada lampiran A *listing* program arduino

3.7 Metode *Manual Tunning*

Metode *manual tuning* adalah metode untuk mencari nilai parameter *PID* dengan cara *trial and error* dengan mengacu pada karakteristik dari setiap parameter yang dicari seperti yang dijelaskan pada bab sebelumnya. Pada penelitian ini dilakukan *manual tuning* parameter dengan langkah sebagai berikut.

1. Memberikan *gain* parameter *proportional* sehingga *plan* memberikan nilai *output* yang mencapai *setpoint*. Namun dengan memberikan parameter ini akan meningkatkan *overshoot* pada *output* dari *plan* dan *error steady-state* masih tinggi.
2. Memberikan nilai parameter *integral* sehingga nilai *error steady-state* dapat dikurangi. Namun pemberian parameter ini akan meningkatkan *overshoot* dan *settling time* meningkat.
3. Memberikan nilai parameter *derivative* sehingga nilai *overshoot* dan *settling time* dapat dikurangi. Namun untuk memberikan nilai parameter ini tidak diperbolehkan terlalu tinggi karena akan membuat sistem lebih tidak stabil.

BAB 5. PENUTUP

Dari hasil-hasil pengujian dan pembahasan yang telah dilakukan dapat diambil beberapa kesimpulan dan saran yang dapat digunakan sebagai dasar untuk penelitian selanjutnya.

5.1 Kesimpulan

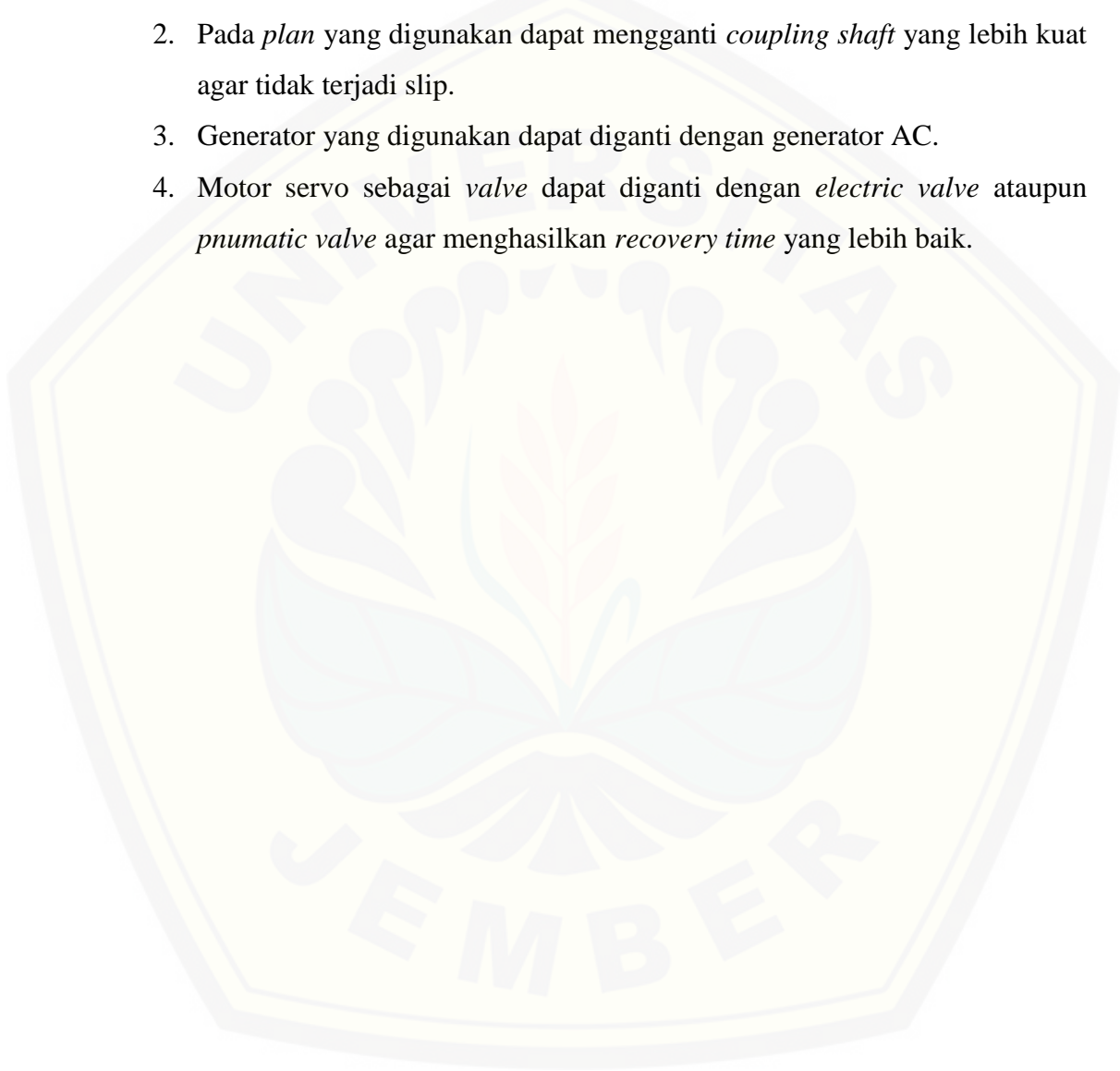
Setelah melakukan pengujian dan analisa pada penelitian yang berjudul “Rancang Bangun Kontrol PID Pada *Speed Observer* Generator DC Berbasis Arduino UNO R3” didapatkan kesimpulan sebagai berikut.

1. Hubungan dari tegangan dan arus sangat penting untuk membentuk sebuah observer kecepatan. Seperti yang sudah dijelaskan sebelumnya pada persamaan 4.1, untuk mencari nilai *speed observer* yaitu melalui analisis hubungan tegangan dan arus output generator dengan nilai tegangan 11.86 V dan arus 1.23 A menghasilkan *speed observer* 2211 rpm dengan pembacaan sensor kecepatan 2213 rpm. Pembangunan persamaan *speed observer* menggunakan teori regresi linier menggunakan perhitungan matlab.
2. Perancangan kontrol PID pada pengujian ini menggunakan metode *manual tuning*. Kontrol PID yang digunakan sebagai pengaturan kecepatan pada *speed observer* generator DC didapatkan nilai parameter kontrol PID yang sesuai dengan *plan* adalah $K_p = 0.77$, $K_i = 0.0083$, dan $K_d = 0.0096$. Pada pengujian kontrol PID saat digunakan *setpoint* yang bervariasi tanpa beban, *rise time* tercepat yaitu sekitar 0.48 s dan *error steady-state* terkecil 0.02 % terjadi saat *setpoint* 2000 rpm akan tetapi membutuhkan *setling time* yang cukup lama yaitu sekitar 2.28 s.

5.2 Saran

Setelah melakukan penelitian, penulis memberikan saran untuk mengembangkan penelitian ini untuk penelitian berikutnya antara lain.

1. Kontrol PID dapat dikembangkan dengan kontrol *hybrid* yaitu PID-Fuzzy agar dapat mengurangi presentase *overshoot* dan *peak*.
2. Pada *plan* yang digunakan dapat mengganti *coupling shaft* yang lebih kuat agar tidak terjadi slip.
3. Generator yang digunakan dapat diganti dengan generator AC.
4. Motor servo sebagai *valve* dapat diganti dengan *electric valve* ataupun *pnumatic valve* agar menghasilkan *recovery time* yang lebih baik.



DAFTAR PUSTAKA

- Chapman. (2005). *Electric Machinery Fundamentals*. New York: Library of Congress Cataloging.
- Hameyer, K. (2001). *Electrical Machines*. Aachen: Institut für Elektrische Maschinen der RWTH Aachen.
- Hardiyanto, R. (2015). Prototype Sistem Compressed Air Energy Storage Dengan Pengaturan Valve Menggunakan Fuzzy Logic Controller.
- Hasan, I. (2004). *Analisis Data Penelitian Dengan Statistik*. Jakarta: PT Bumi Aksara.
- Jamil, R., Jamil, I., Jinquan, Z., Ming, L., Dong, W. Y., & Jamil, R. (2013, November). Control and Configuration of Generator Excitation System as Current Mainstream. *International Journal of Innovation and Applied Studies*, 4(3).
- Kokaew, V., torbati, m. m., & sharkh, s. m. (2013). Maximum Efficiency or Power Tracking of Stand-Alone. *energy procedia*.
- Mansoor, A.-K. Z., Salih, T. A., & Abdullah, F. S. (2013, Maret). Speed Control of Separately Excited D.C. Motor using Self-Tuned. *Tikrit Journal of Engineering Sciences*, 20(1), 1-9.
- Patil, Jogalekar, S., R. J., & S. J. (2013). Tachogeneratorless Speed Control of Separately Excited DC/PMDC Motor Using Observer. *International Journal of Innovative Research in Science, Engineering and Technology*, 1086-1091.
- Suhariningsih, Soebagio, & Purnomo, M. H. (2008). Pengaturan Motor Induksi Menggunakan Observer Self Constructing Fuzzy Neural Network dengan Metode Algoritma Pelatihan Levenberg Marquardt. *Jurnal Teknik Elektro Vol. 8, No. 2*, 79-80.
- Widjonarko, Setiawan, A., & Afiliasi. Pengembangan Observer untuk Pendeteksi Kecepatan Brushed DC Generator NC475. *elektronik Jurnal Arus Elektro Indonesia (eJAEI)*.

LAMPIRAN

A. Listing Program Arduino Uno R3

```
#include <Servo.h>
#include <Wire.h>
#include <Adafruit_INA219.h>

Adafruit_INA219 ina219;

Servo myservo;
int pulse_ = 0;

String inData = "";
String data[10];
int i;
boolean parsing = false;

int start = 0;

double pulse = 0.00;
float SPEED = 0.00;

float voltage = 0.00;
float voltage1 = 0.00;
float current = 0.00;
float current1 = 0.00;

float speed_observer = 0.00;

int setPoint = 0;
float KP = 0.77;
float KI = 0.0083;
float KD = 0.0096;

float last_error_I = 0.00;
float last_error_D = 0.00;
float Tc = 0.1;
int error;
int error_I;
int error_D;
float out_P;
float out_I;
float out_D;
float out_PID;
float LpulseH = 300;
float LpulseL = -500;

float waktu = 0.00;
float waktu_akhir = 0.00;

float vMPX = 0;
float pt = 0;
```

```
float Power = 0;

float VKalman = 0.0;
float IKalman = 0.0;

void setup() {
  Serial.begin(9600);
  myservo.attach(3);

  ina219.begin();
  pinMode(2, INPUT_PULLUP);
  initTimer1();
  attachInterrupt(0, Interrupt0, FALLING);
}

void loop() {
  receive_serial();
  if (start == 1) {
    //access_volt();
    //access_current();

    current = ina219.getCurrent_mA();
    current = abs(current / 1000);
    voltage = ina219.getBusVoltage_V();
    voltage = abs(voltage);
    Power = voltage*current;

    observer();
    PID();
    servo();
    MPX5700();

    waktu = (float)(millis() - waktu_akhir) / 1000;

    disp();
  } else {
    waktu_akhir = millis();
  }
}

//-----//observer
void PID() {
  //error = setPoint - SPEED;
  error = setPoint - speed_observer;

  if (error >= LpulseH)
  {
    error = LpulseH;
  }
  if (error <= (LpulseL))
  {
    error = LpulseL;
  }
  out_P = KP * error;
```

```
error_I = error + last_error_I;
out_I = KI * error_I * Tc;
last_error_I = error_I;

error_D = error - last_error_D;
out_D = (KD * error_D) / Tc;
last_error_D = error_D;

out_PID = out_P + out_I + out_D;
}

//-----//observer
void observer() {
  if (voltage <= 0) {
    speed_observer = 0;
  } else {
    speed_observer = (float)(-2.1361 + 150.1774 * voltage + 351.8041 * current);
  }
}

//-----//Speed
void initTimer1() {
  TCCR1A = 0x00;
  TCCR1B = 0x05;
  TCNT1H = 0xC2;
  TCNT1L = 0xE8;
  ICR1H = 0x00;
  ICR1L = 0x00;
  OCR1AH = 0x00;
  OCR1AL = 0x00;
  OCR1BH = 0x00;
  OCR1BL = 0x00;
  TIMSK1 = 0x01;

  sei();
}

ISR(TIMER1_OVF_vect) {
  TCNT1H = 0xC2;
  TCNT1L = 0xE8;

  SPEED = (float)pulse * 60 / 36;
  pulse = 0;
  waktu++;
}

void Interrupt0() {
  pulse++;
}

//-----//Servo
void servo() {
  pulse_ = out_PID;
```

```
if (pulse_ < LpulseL) {
  pulse_ = LpulseL;
}
if (pulse_ > LpulseH) {
  pulse_ = LpulseH;
}

if (pulse_ < 0)
{
  digitalWrite(3, HIGH);
  delayMicroseconds(1480 + pulse_);
  digitalWrite(3, LOW);
  delayMicroseconds(20000 - (1480 + pulse_));
}
if (pulse_ == 0)
{
  digitalWrite(3, HIGH);
  delayMicroseconds(1480);
  digitalWrite(3, LOW);
  delayMicroseconds(20000 - 1480);
}
if (pulse_ > 0)
{
  digitalWrite(3, HIGH);
  delayMicroseconds(1480 + pulse_);
  digitalWrite(3, LOW);
  delayMicroseconds(20000 - (1480 + pulse_));
}
}
//-----//Tekanan
void MPX5700(){
  vMPX = analogRead (A2);
  vMPX = vMPX*5/1024;
  pt = vMPX*700.0/4.62;
}
//-----//Display
void disp()
{
  Serial.print(waktu);
  Serial.print("\t");
  Serial.print(pt);
  Serial.print("\t");
  Serial.print(speed_observer);
  Serial.print("\t");
  Serial.print("\t");
  Serial.print(KP);
  Serial.print("\t");
  Serial.print(KI,5);
  Serial.print("\t");
  Serial.print(KD,5);
  Serial.print("\t");
  Serial.print(setPoint);
  Serial.print("\t");
  Serial.print(current);
  Serial.print("\t");
```

```
Serial.print(voltage);
Serial.print("\t");
Serial.println(Power);

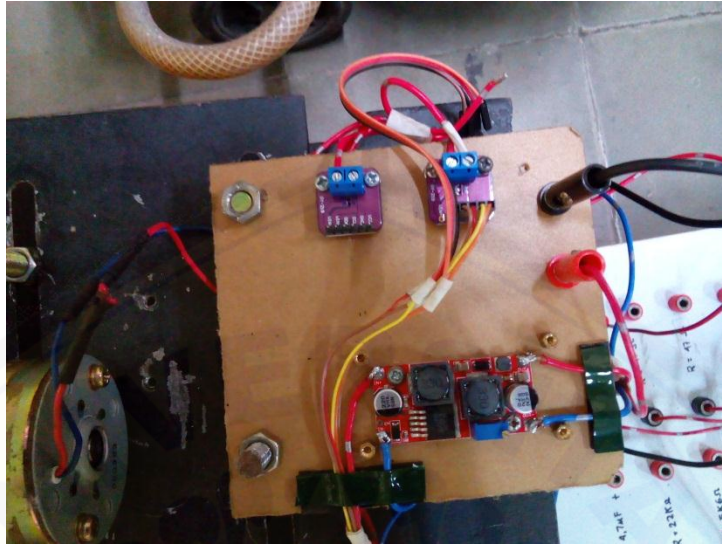
}

//-----//recieving data
void receive_serial() {
  while (Serial.available() > 0) {
    char inChar = (char)Serial.read();
    if (inChar == '\n') {
      parsing = true;
    } else {
      if (inChar != '\r') {
        inData += inChar;
      }
    }
  }
}

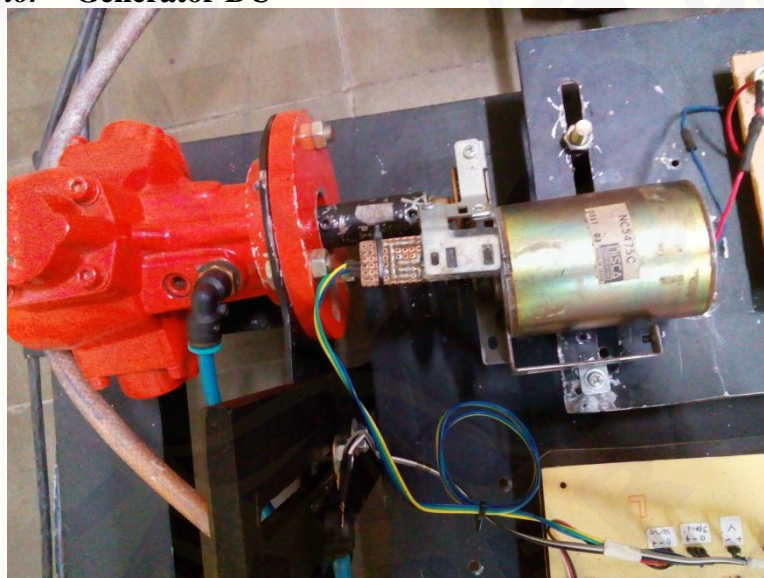
if (parsing) {
  int j = 0;
  data[j] = "";
  for (i = 0; i < inData.length(); i++) {
    if (inData[i] == ' ') {
      j++;
      data[j] = "";
    } else {
      data[j] += inData[i];
    }
  }
  start = data[0].toInt();
  //KP = data[1].toFloat();
  //KI = data[2].toFloat();
  //KD = data[3].toFloat();
  setPoint = data[1].toInt();
  parsing = false;
  inData = "";
}
}
```


B. Gambar Hardware

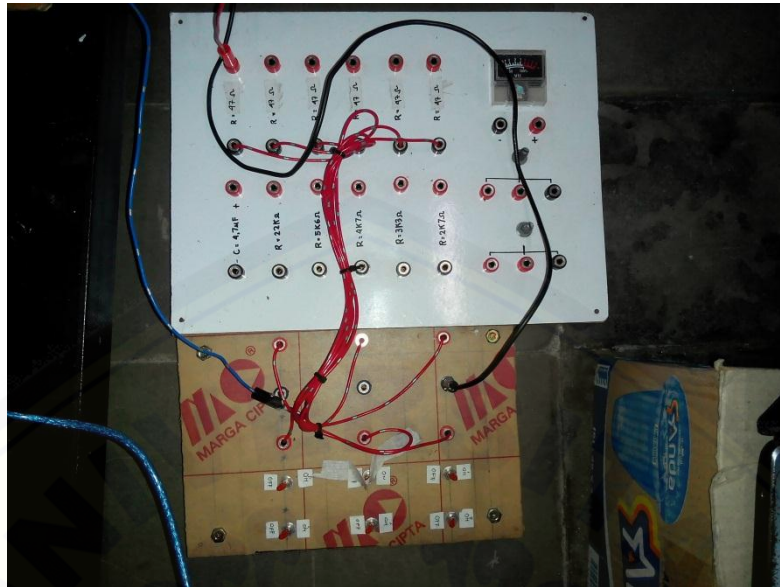
B1. Rangkaian Sensor



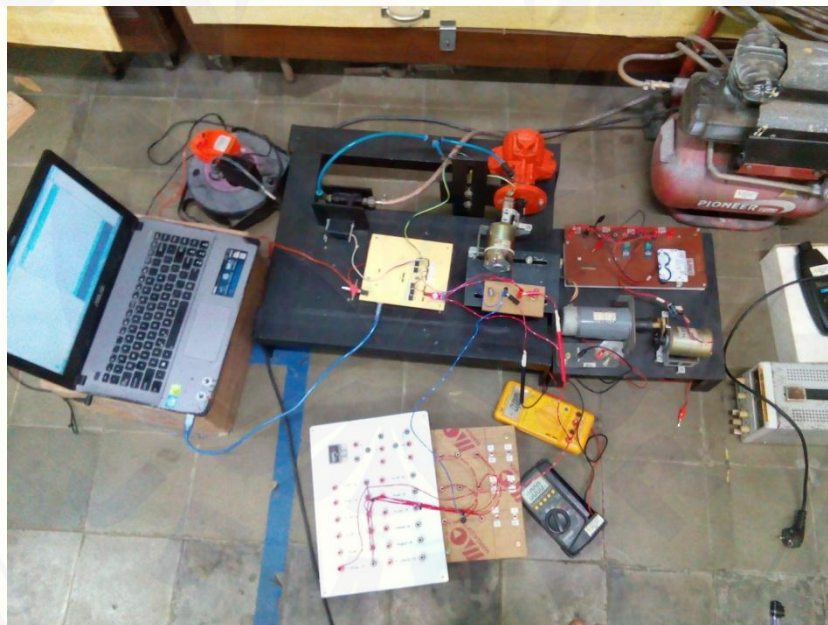
B2. Ait Motor – Generator DC



B3. Beban Resistif



C. Gambar Keseluruhan Plan



D. Data Hasil Speed Observer

SS (rpm)	CS (Amps)	VS (V)	SO (rpm)	Error (%)					
					433	0	2.9	432	0.30%
195	0	1.31	193	1.03%	442	0	2.96	441	0.14%
198	0	1.3	191	3.45%	447	0	3.09	461	3.12%
200	0	1.32	194	2.75%	600	0	4.04	603	0.57%
202	0	1.35	199	1.32%	602	0	4.06	606	0.79%
203	0	1.37	202	0.65%	602	0	4.02	600	0.21%
203	0	1.32	194	4.34%	603	0	4.12	615	2.01%
205	0	1.38	204	0.72%	603	0	4.07	608	0.77%
207	0	1.36	201	2.98%	607	0	4.08	609	0.46%
210	0	1.38	204	3.09%	608	0	4.09	611	0.43%
212	0	1.44	213	0.41%	615	0	4.1	612	0.41%
217	0	1.46	216	0.52%	645	0	4.26	637	1.32%
218	0	1.46	216	1.27%	788	0	5.29	791	0.39%
220	0	1.47	217	1.34%	792	0.2	4.83	792	0.10%
223	0	1.48	219	2.14%	798	0.2	4.87	798	0.02%
223	0	1.52	225	0.55%	800	0.2	4.87	798	0.19%
225	0	1.53	226	0.48%	802	0	5.34	799	0.35%
230	0	1.57	232	0.91%	803	0.2	4.9	803	0.04%
232	0	1.58	234	0.83%	805	0	5.36	802	0.39%
235	0	1.59	235	0.04%	807	0	5.42	811	0.53%
247	0	1.71	253	2.62%	808	0	5.4	808	0.05%
277	0	1.86	276	0.36%	810	0	5.41	809	0.07%
392	0	2.63	391	0.06%	615	0	4.1	612	0.41%
393	0	2.61	388	1.24%	645	0	4.26	637	1.32%
398	0	2.66	396	0.59%	788	0	5.29	791	0.39%
400	0	2.64	393	1.76%	792	0.2	4.83	792	0.10%
402	0	2.63	391	2.54%	798	0.2	4.87	798	0.02%
403	0	2.7	402	0.34%	800	0.2	4.87	798	0.19%
405	0	2.72	405	0.00%	802	0	5.34	799	0.35%
407	0	2.73	406	0.05%	803	0.2	4.9	803	0.04%
410	0	2.68	399	2.69%	805	0	5.36	802	0.39%
412	0	2.81	419	1.66%	807	0	5.42	811	0.53%
413	0	2.76	411	0.56%	808	0	5.4	808	0.05%
415	0	2.78	414	0.24%	810	0	5.41	809	0.07%
417	0	2.8	417	0.08%	810	0	5.42	811	0.11%
420	0	2.8	417	0.71%	812	0.11	5.17	812	0.04%
422	0	2.84	423	0.32%	813	0.21	4.95	814	0.08%
423	0	2.84	423	0.07%	815	0.11	5.19	815	0.00%
427	0	2.87	428	0.20%	817	0.11	5.2	816	0.02%
428	0	2.9	432	0.87%	818	0.2	4.98	815	0.41%
432	0	2.91	434	0.43%	822	0.11	5.16	810	1.36%

823	0	5.52	826	0.32%	1218	0.17	7.73	1218	0.03%
833	0.21	5.08	834	0.03%	1218	0	8.12	1217	0.12%
837	0.11	5.32	835	0.26%	1220	0.17	7.73	1218	0.17%
980	0.35	5.73	980	0.05%	1220	0	8.15	1221	0.11%
980	0.35	5.72	979	0.11%	1222	0.17	7.75	1221	0.06%
987	0.26	6.09	1003	1.65%	1222	0	8.16	1223	0.10%
990	0.36	5.79	993	0.30%	1223	0.31	7.46	1226	0.26%
992	0.35	5.8	991	0.07%	1225	0.45	7.12	1225	0.04%
993	0.36	5.78	991	0.19%	1225	0.17	7.8	1228	0.28%
993	0.36	5.79	993	0.04%	1227	0.45	7.14	1228	0.07%
995	0.35	5.82	994	0.10%	1227	0.32	7.44	1227	0.03%
997	0.14	6.33	997	0.02%	1228	0.45	7.12	1225	0.31%
998	0	6.66	997	0.10%	1228	0.32	7.44	1227	0.11%
1000	0.36	5.84	1000	0.05%	1222	0.17	7.75	1221	0.06%
1002	0.14	6.36	1001	0.03%	1222	0	8.16	1223	0.10%
1003	0	6.69	1002	0.15%	1223	0.31	7.46	1226	0.26%
1005	0	6.71	1005	0.01%	1225	0.45	7.12	1225	0.04%
1007	0	6.72	1006	0.03%	1225	0.17	7.8	1228	0.28%
1008	0	6.71	1005	0.34%	1227	0.45	7.14	1228	0.07%
1010	0	6.74	1009	0.06%	1227	0.32	7.44	1227	0.03%
1012	0.26	6.15	1012	0.00%	1228	0.45	7.12	1225	0.31%
1015	0.26	6.17	1015	0.00%	1228	0.32	7.44	1227	0.11%
1018	0.14	6.5	1022	0.40%	1228	0	8.2	1229	0.04%
1025	0.27	6.2	1023	0.19%	1230	0.32	7.45	1228	0.12%
1030	0.27	6.24	1029	0.10%	1232	0.45	7.18	1234	0.15%
1033	0	6.91	1035	0.15%	1233	0.45	7.2	1237	0.26%
1177	0.43	6.91	1186	0.79%	1235	0.32	7.49	1235	0.04%
1180	0.43	6.87	1180	0.01%	1235	0	8.24	1235	0.01%
1187	0.43	6.92	1187	0.07%	1272	0.32	7.67	1262	0.79%
1195	0.43	6.97	1195	0.00%	1380	0.64	7.71	1380	0.00%
1197	0.44	6.92	1191	0.48%	1383	0.19	8.94	1407	1.70%
1202	0.43	7.03	1204	0.19%	1383	0.64	7.75	1386	0.19%
1207	0	8.15	1221	1.22%	1388	0.64	7.76	1387	0.06%
1208	0.43	7.05	1207	0.11%	1395	0.65	7.78	1394	0.07%
1208	0	8.06	1208	0.04%	1397	0	9.32	1397	0.04%
1212	0	8.11	1215	0.30%	1397	0.51	8.12	1396	0.05%
1213	0	8.1	1214	0.04%	1400	0.65	7.83	1402	0.11%
1215	0.17	7.69	1212	0.26%	1403	0	9.36	1403	0.00%
1215	0	8.1	1214	0.10%	1403	0.51	8.17	1403	0.01%
1217	0.44	7.12	1221	0.36%	1407	0	9.38	1406	0.03%
1217	0.17	7.72	1216	0.02%	1407	0.51	8.2	1408	0.09%
1218	0.17	7.74	1219	0.09%	1412	0	9.42	1412	0.04%

1412	0.19	8.99	1414	0.19%	1805	0.25	11.44	1804	0.07%
1413	0.19	8.98	1413	0.03%	1805	0.65	10.51	1804	0.03%
1417	0.37	8.58	1416	0.05%	1805	0.84	10.04	1801	0.25%
1422	0	9.45	1417	0.34%	1815	0	12.1	1815	0.01%
1422	0.19	9.03	1420	0.09%	1815	0.66	10.54	1812	0.14%
1423	0	9.49	1423	0.04%	1817	0	12.11	1817	0.00%
1568	0.73	8.74	1566	0.12%	1817	0.25	11.52	1816	0.05%
1587	0.73	8.88	1587	0.05%	1817	0.66	10.58	1818	0.10%
1593	0.73	8.91	1592	0.08%	1823	0.67	10.59	1824	0.01%
1597	0.74	8.91	1596	0.07%	1833	0	12.26	1839	0.32%
1602	0.75	8.94	1604	0.12%	1842	0.48	11.14	1839	0.12%
1610	0	10.75	1612	0.14%	1850	0.48	11.22	1851	0.08%
1612	0.22	10.28	1619	0.45%	1855	0.48	11.23	1853	0.11%
1615	0	10.78	1617	0.11%	1863	0.48	11.28	1861	0.15%
1620	0.76	9.03	1621	0.04%	1957	0.92	10.92	1961	0.22%
1620	0	10.81	1621	0.08%	1987	0.91	11.01	1971	0.79%
1627	0.42	9.87	1627	0.05%	1993	0.93	11.13	1996	0.13%
1627	0	10.84	1626	0.05%	2002	0	13.36	2005	0.15%
1633	0.76	9.12	1634	0.05%	2010	0	13.4	2011	0.03%
1633	0.6	9.47	1631	0.17%	2010	0.93	11.21	2008	0.10%
1638	0.22	10.41	1638	0.00%	2012	0	13.43	2015	0.17%
1638	0.23	10.39	1639	0.03%	2013	0	13.41	2012	0.06%
1643	0.43	9.95	1643	0.02%	2013	0.93	11.29	2020	0.33%
1643	0	10.96	1644	0.03%	2020	0.73	11.72	2014	0.27%
1650	0.61	9.6	1654	0.22%	2020	0	13.45	2018	0.09%
1650	0.43	10.02	1654	0.21%	2048	0.74	11.93	2050	0.06%
1655	0.77	9.23	1654	0.05%	2077	0.75	12.07	2074	0.12%
1660	0.77	9.26	1659	0.08%	2085	0.54	12.62	2083	0.09%
1662	0.43	10.07	1661	0.04%	2168	1	12.05	2159	0.43%
1673	0.23	10.6	1670	0.17%	2187	1.01	12.22	2188	0.06%
1677	0.77	9.37	1675	0.09%	2190	0.78	12.75	2187	0.14%
1682	0.78	9.38	1680	0.09%	2197	0.56	13.35	2200	0.14%
1698	0.44	10.26	1693	0.31%	2205	0.55	13.43	2208	0.15%
1710	0.63	9.91	1707	0.16%	2212	0.56	13.5	2222	0.48%
1738	0.46	10.74	1772	1.95%	2212	1.2	11.93	2211	0.03%
1775	0.84	9.93	1784	0.51%	2220	0	14.78	2218	0.08%
1785	0.47	10.77	1780	0.26%	2220	1.04	12.38	2223	0.12%
1790	0.66	10.52	1809	1.09%	2230	0	14.85	2229	0.06%
1798	0.25	11.4	1798	0.03%	2230	1.03	12.46	2231	0.05%
1798	0.83	10.02	1794	0.24%	2232	1.02	12.48	2231	0.05%
1800	0.25	11.43	1802	0.13%	2240	1.02	12.53	2238	0.09%
1800	0.25	11.42	1801	0.04%	2240	0.57	13.52	2229	0.50%

2240	0.29	14.26	2242	0.08%	440	0	16.24	2438	0.10%
2252	0.82	13.09	2252	0.02%	2440	1.09	13.68	2436	0.18%
2252	0.56	13.66	2246	0.23%	2452	0.62	14.88	2451	0.03%
2253	0.81	13.12	2253	0.01%	2453	0.32	15.6	2454	0.02%
2260	0.57	13.74	2262	0.09%	2485	0.63	15.05	2480	0.20%
2260	0.3	14.39	2265	0.22%	2500	0.33	15.82	2490	0.38%
2272	0.57	13.8	2271	0.03%	2555	1.19	14.25	2556	0.05%
2287	0.3	14.53	2286	0.03%	2590	0.94	15.04	2587	0.10%
2313	0.3	14.7	2311	0.08%	2600	0.67	15.74	2598	0.09%
2342	0.84	13.64	2342	0.00%	2640	0.68	16	2640	0.01%
2313	0.3	14.7	2311	0.08%	2662	0	17.72	2660	0.06%
2342	0.84	13.64	2342	0.00%	2777	1	16.06	2762	0.54%
2388	1.33	12.8	2388	0.03%	2795	0.71	16.99	2800	0.17%
2397	1.33	12.88	2400	0.12%	2828	0.71	17.14	2822	0.21%
2407	0.61	14.6	2405	0.06%	2900	0.73	17.62	2901	0.05%
2420	0.61	14.71	2422	0.08%	3050	0.79	18.49	3053	0.11%
2420	0.88	14.08	2422	0.08%	3087	0.8	18.83	3108	0.69%
2430	0.88	14.17	2435	0.23%					
2430	1.12	13.6	2434	0.17%					

E. Perhitungan Regresi

Y(SS)	X ₁ (VS)	X ₂ (CS)	X ₁ *Y	X ₂ *Y	X ₁ *X ₂	
15257184	94184,74	3218,34	190479935,1	6636124,36	38960,7085	SUM
8435	8435	8435	8435	8435	8435	N
1808,794816	11,16594428	0,38154594	22582,0907	786,7367351	4,618934025	M
626,7815231	3,876725951	0,348618746	13763,71313	839,5500075	4,841504144	SD
3313339726	126754,6206	1025,026441				USS

$$\sum x_1 y$$

$$\begin{aligned}\sum x_1 y &= \sum X_1 Y - \frac{(\sum X_1)(\sum Y)}{N} \\ &= (190479935,1) - \frac{(94184,74)(15257184)}{8435} \\ &= 20119065,65\end{aligned}$$

$$\sum x_2 y$$

$$\begin{aligned}\sum x_2 y &= \sum X_2 Y - \frac{(\sum X_2)(\sum Y)}{N} \\ &= (6636124,36) - \frac{(3218,34)(15257184)}{8435} \\ &= 814807,6534\end{aligned}$$

$$\sum x_1 x_2$$

$$\begin{aligned}\sum x_1 x_2 &= \sum X_1 X_2 - \frac{(\sum X_1)(\sum X_2)}{N} \\ &= (38960,7085) - \frac{(94184,74)(3218,34)}{8435} \\ &= 3024,903387\end{aligned}$$

$$\sum y y$$

$$\begin{aligned}\sum y y &= \sum Y Y - \frac{(\sum Y)(\sum Y)}{N} \\ &= (30910455535) - \frac{(15257184)(15257184)}{8435} \\ &= 3313339726\end{aligned}$$

$$\sum x_1 x_1$$

$$\begin{aligned}\sum x_1 x_1 &= \sum X_1 X_1 - \frac{(\sum X_1)(\sum X_1)}{N} \\ &= (1178416,179) - \frac{(94184,74)(94184,74)}{8435} \\ &= 126754,6206\end{aligned}$$

$$\sum x_2 x_2$$

$$\sum x_2 x_2 = \sum X_2 X_2 - \frac{(\sum X_2)(\sum X_2)}{N}$$

$$= (30910455535) - \frac{(15257184)(15257184)}{8435}$$

$$= 3313339726$$

$$b1 = \frac{(\sum x2^2)(\sum x1y) - (\sum x1x2)(\sum x2y)}{(\sum x1^2)(\sum x2^2) - (\sum x1x2)^2}$$

$$= \frac{(1025,026441)(20119065) - (3024,903387)(814807,6534)}{(126754,6206)(1025,026441) - (3024,903387)(3024,903387)}$$

$$= 150,3422865$$

$$b2 = \frac{(\sum x1^2)(\sum x2y) - (\sum x1x2)(\sum x1y)}{(\sum x1^2)(\sum x2^2) - (\sum x1x2)^2}$$

$$= \frac{(126754,6206)(814807,6534) - (3024,903387)(20119065,65)}{(126754,6206)(1025,026441) - (3024,903387)(3024,903387)}$$

$$= 351,2463167$$

$$a = Y(\text{avg}) - b1X1(\text{avg}) - b2X2(\text{avg})$$

$$= 1808,794816 - (150,34322865)(11,16594428) -$$

$$(351,2463167)(0,348618746)$$

$$a = -3,93538408$$

Jadi

$$Y' = -3,935 + 150,342 (\text{VS}) + 351,246 (\text{CS})$$