



**PENINGKATAN KINERJA GENERATOR DC DENGAN SISTEM
EKSITASI BERBASIS ARDUINO UNO R3 MENGGUNAKAN
BUCK CONVERTER DAN KONTROL *PID***

SKRIPSI

oleh

**Abdul Fajar
NIM 121910201011**

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



**PENINGKATAN KINERJA GENERATOR DC DENGAN SISTEM
EKSITASI BERBASIS ARDUINO UNO R3 MENGGUNAKAN
BUCK CONVERTER DAN KONTROL *PID***

SKRIPSI

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

oleh

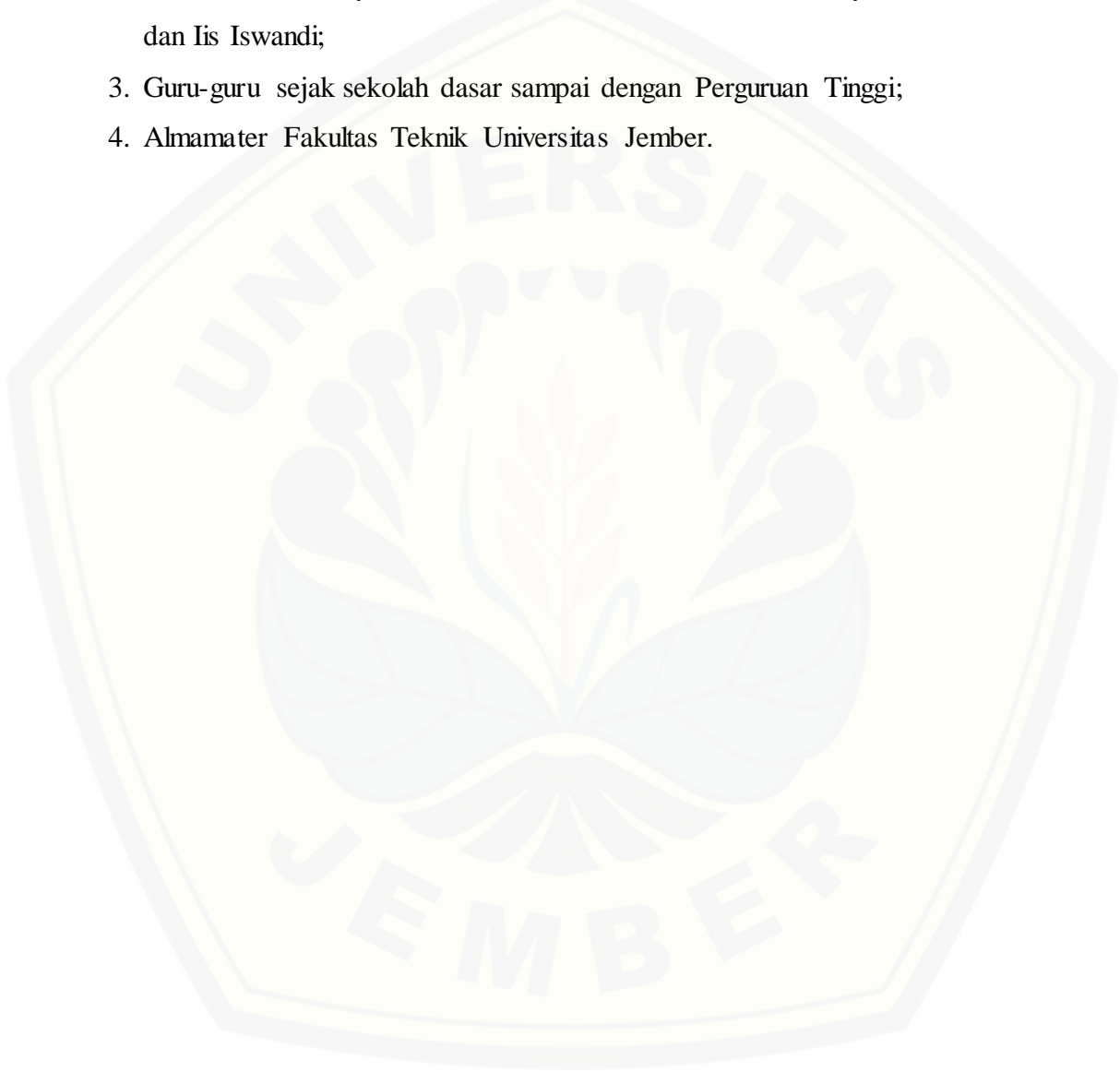
**Abdul Fajar
NIM 121910201011**

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

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Ibunda Hj. Amana dan Ayahanda H. Raliyah yang tercinta dan tersayang;
2. Adikku tercinta Ayu Nabilah dan Kakak – kakakku Nana Riyana, In Solihin dan Iis Iswandi;
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.^{**)}

Jadi diri sendiri cari jati diri dan dapatkan hidup yang mandiri optimis, karena hidup terus mengalir dan kehidupan terus berputar, sesekali lihat kebelakang untuk melanjutkan perjalanan yang tiada berujung.^{***)}

^{*)} (Q.S An Nisaa', 4:36)..

^{**)} QS Al-Baqarah: 45.

^{***)} Anonim.

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Abdul Fajar

NIM : 121910201011

Menyatakan dengan sesungguhnya bahwa skripsi yang berjudul “Peningkatan Kinerja Generator DC dengan Sistem Eksitasi Berbasis Arduino UNO R3 Menggunakan *Buck Converter* dan Kontrol *PID*” 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, 2 Juni 2016
Yang menyatakan,

Abdul Fajar
NIM 121910201011

SKRIPSI

**PENINGKATAN KINERJA GENERATOR DC DENGAN SISTEM
EKSITASI BERBASIS ARDUINO UNO R3 MENGGUNAKAN
BUCK CONVERTER DAN KONTROL *PID***

oleh

Abdul Fajar
NIM 121910201011

Pembimbing

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

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

PENGESAHAN

Skripsi berjudul “Peningkatan Kinerja Generator DC dengan Sistem Eksitasi Berbasis Arduino UNO R3 Menggunakan *Buck Converter* dan Kontrol *PID*” telah diuji dan disahkan pada:

hari, tanggal : Kamis, 2 Juni 2016

tempat : Fakultas Teknik Universitas Jember.

Tim Penguji:

Pembimbing Utama,

Pembimbing Anggota,

Ir. Widyono Hadi, M.T.
NIP 196104141989021001

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

Penguji I,

Penguji II,

Dr. Triwahju Hardiatio, S.T., M.T.
NIP 197008261997021001

Dr. Ir. Bambang Sujanarko M.M.
NIP 196312011994021002

Mengesahkan
Dekan Fakultas Teknik,

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

Peningkatan Kinerja Generator DC dengan Sistem Eksitasi Berbasis Arduino UNO R3 Menggunakan *Buck Converter* dan Kontrol *PID*

Abdul Fajar

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember

ABSTRAK

Dalam kehidupan kita, terutama sektor listrik dan industri, generator DC dapat berfungsi sebagai generator arus searah di bengkel atau pabrik, sebagai pengisi baterai pada perusahaan *charger* baterai, bahkan berfungsi sebagai sumber penguat di *exciter* generator utama. Namun, tegangan *output* akan cenderung turun ketika arus beban meningkat. Untuk mengatasi hal ini dapat dilakukan dengan meningkatkan tegangan induksi (E_a) di jangkar generator DC. Untuk meningkatkan tegangan induksi digunakan kontrol *PID* sehingga nilai dari tegangan *output* tidak terlalu tinggi atau terlalu rendah dengan umpan balik pembangkit tegangan *output*. Kontrol *PID* membuat tegangan *output* generator yang stabil pada tegangan 220 V. parameter *PID* yang digunakan adalah $K_p = 3$, $K_i = 0,00005$ dan $K_d = 0,00002$. Setelah menerapkan nilai kontrol *PID* diperoleh *rise time* 0,8 s, *error steady-state* 0,21% dan *peak* 241 V. Pengujian dilakukan dengan beban 15, 25, 40, 50 dan 65 Watt, yang tertinggi nilai *error steady-state* yang diperoleh 0,88% dan terendah adalah 0,21%, nilai *recovery time* tertinggi 3,1 s dan terendah 0,6 s. Pengujian pelepasan beban nilai puncak tertinggi 228 V dan 223 V terendah puncak.

Kata kunci: Generator DC, Kontrol *PID*, Arduino UNO R3, *Buck Converter*

Increased performance with the DC Generator Excitation System Based Arduino UNO R3 Using Buck Converter and PID Control

Abdul Fajar

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

ABSTRACT

In our lives, especially the power sector and industry, the DC generator can serve as a direct current generator in the workshop or factory, as the battery charger on the battery charger company, and even serves as a center of power in the main generator exciter. However, the output voltage will tend to go down when the load current increases. To overcome this can be done by increasing the induced voltage (E_a) at anchor DC generator. To increase the induced voltage used PID control so that the value of the induced voltage field is not too high or too low with the generator output voltage feedback. PID control makes the generator output voltage stabilized at a voltage of 220 V. PID parameters used are $K_p = 3$, $K_i = 0.00005$ and $K_d = 0.00002$. After applying PID control values obtained rise time 0.8 s, error s-s 0.21 % and peak loading 241 V. Testing is done with a load of 15, 25, 40, 50 and 65 Watt, the highest value obtained s-s error 0.88% and the lowest was 0.21%, the recovery value The highest time of 3.1 s and 0.6 s lows. Testing unburdening the highest peak value of 228 V and 223 V peak lows.

Key words: *DC Generator, PID Control, Arduino UNO R3, Buck Converter*

RINGKASAN

Peningkatan Kinerja Generator DC dengan Sistem Eksitasi Berbasis Arduino UNO R3 Menggunakan Buck Converter dan Kontrol PID;

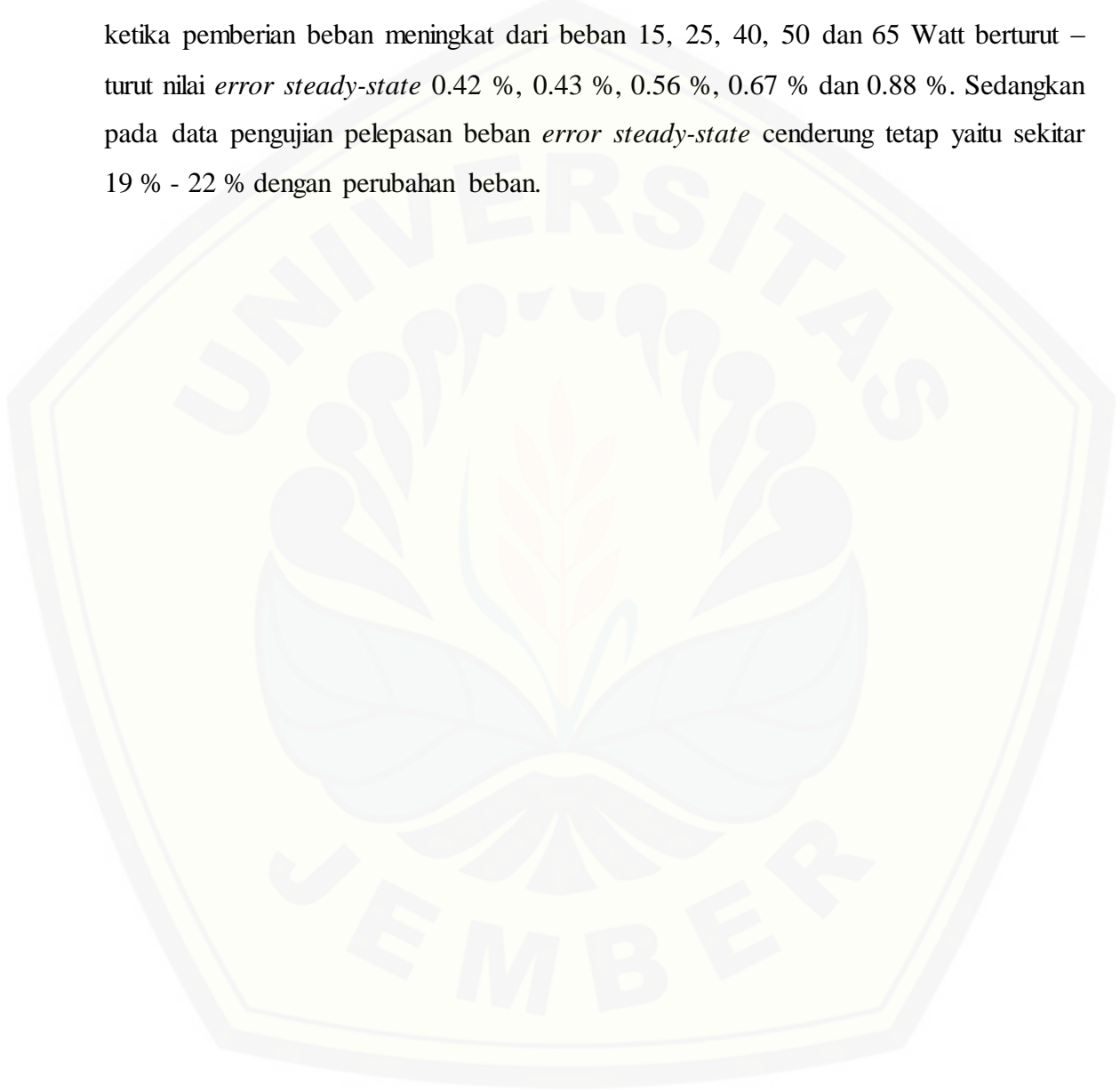
Abdul Fajar, 121910201011; 2016; 52 halaman; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

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. Namun tegangan *output* akan cenderung turun bila arus beban semakin besar. Untuk mengatasi hal tersebut dapat dilakukan dengan cara meningkatkan tegangan induksi (E_a) pada jangkar generator DC.

Implementasi kontrol *PID* dapat diterapkan pada generator DC penguatan terpisah untuk mengatasi hal tersebut, karena sebagai sumber tegangan, karakteristik dari generator DC penguatan terpisah memiliki tegangan *output* tidak konstan dan cenderung turun bila arus beban naik. Diharapkan dengan diterapkannya kontrol *PID* pada sistem generator DC penguatan terpisah, tegangan *output* akan lebih stabil.

Pengujian pada generator DC penguatan terpisah yang telah diberikan kontrol *PID* pembebanan 15 W, 25 W, 40 W, 50 W dan 65 W menghasilkan nilai kestabilan yang cukup baik sebagai sumber tegangan listrik. Nilai setpoint tegangan *output* yang digunakan adalah 220 V. Nilai $K_p = 3$, $K_i = 0.00005$ dan $K_d = 0.00002$ didapatkan *rise time* 0.2 s, *settling time* 0.8 s, *error steady-state* 0.21 % dan *peak* 241 Volt. Pada pengujian pembebanan *peak* terbesar pada beban 15 Watt 216 Volt dan *peak* terendah pada beban 65 Watt yaitu 201 Volt. Pada pengujian pelepasan beban *peak* terkecil pada pelepasan beban 15 Watt yaitu 223 Volt dan *peak* terbesar pada pelepasan beban 65 Watt 228 Volt. *Recovery time* tercepat terjadi pada beban 15 Watt ketika

pemberian beban 0.6 s dan pelepasan beban 0.6 s. Sedangkan *recovery time* paling lambat terjadi pada beban 65 Watt ketika pemberian beban 3.1 s dan pelepasan beban 1.5 s. *Error steady-state* pada data pengujian pemberian beban semakin meningkat ketika pemberian beban meningkat dari beban 15, 25, 40, 50 dan 65 Watt berturut – turut nilai *error steady-state* 0.42 %, 0.43 %, 0.56 %, 0.67 % dan 0.88 %. Sedangkan pada data pengujian pelepasan beban *error steady-state* cenderung tetap yaitu sekitar 19 % - 22 % dengan perubahan beban.



PRAKATA

Puji syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Peningkatan Kinerja Generator DC dengan Sistem Eksitasi Berbasis Arduino UNO R3 Menggunakan *Buck Converter* dan Kontrol *PID*”. 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. Ir. Widyono Hadi, M.T. selaku Dosen Pembimbing Utama, Dr. Bambang Sri Kaloko S.T., M.T. selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi ini;
2. Dr. Triwahju Hardiato, S.T., M.T. selaku Dosen Penguji I, Dr. Ir. Bambang Sujanarko M.M. selaku Dosen Penguji II yang telah memberikan kritik dan saran yang sangat membangun demi penyempurnaan skripsi ini;
3. Ir. Widyono Hadi, M.T. selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa;
4. M. Agung Prawira N., S.T., M.T. selaku Komisi Bimbingan S1 yang telah membantu penulisan skripsi secara administratif;
5. Bunda Hj. Amena dan ayahanda H. Ralياهو yang telah memberikan dukungan moril dan materiil serta kasih sayang yang tak terhingga sepanjang masa;
6. Adikku tercinta Ayu Nabilah dan Kakak – kakakku Nana Riyana, Iin Solihin dan Iis Iswandi terkasih yang telah memberikan motivasi dan semangat;
7. Keluarga besar SATE UJ 12 atas kekeluargaan yang kalian berikan.
8. Dulur teknik elektro universitas Jember atas kebersamaannya.

9. Keluarga besar Robotika Universitas Jember yang telah memberikan ilmu lebih dan rasa kekeluargaan yang besar.
10. Keluarga besar Lab Konversi Energi Listrik, Universitas Jember yang telah memberikan dukungan besar dalam penulisan skripsi terutama mas sugianto, A.md.
11. Rekan-rekan ngopi dan touring Chandra, Hendro, Erlangga, Galih, Rih, 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, Hikmah dan masih banyak lagi yang selalu berkumpul selama masa kuliah;
12. Rekan-rekan 65 Watt *squad*, Aditya Chandra Purnama dan Hendro Rosyidi yang telah meluangkan waktu dan tenaga untuk membantu menyelesaikan skripsi ini.
13. 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, 2 Juni 2016

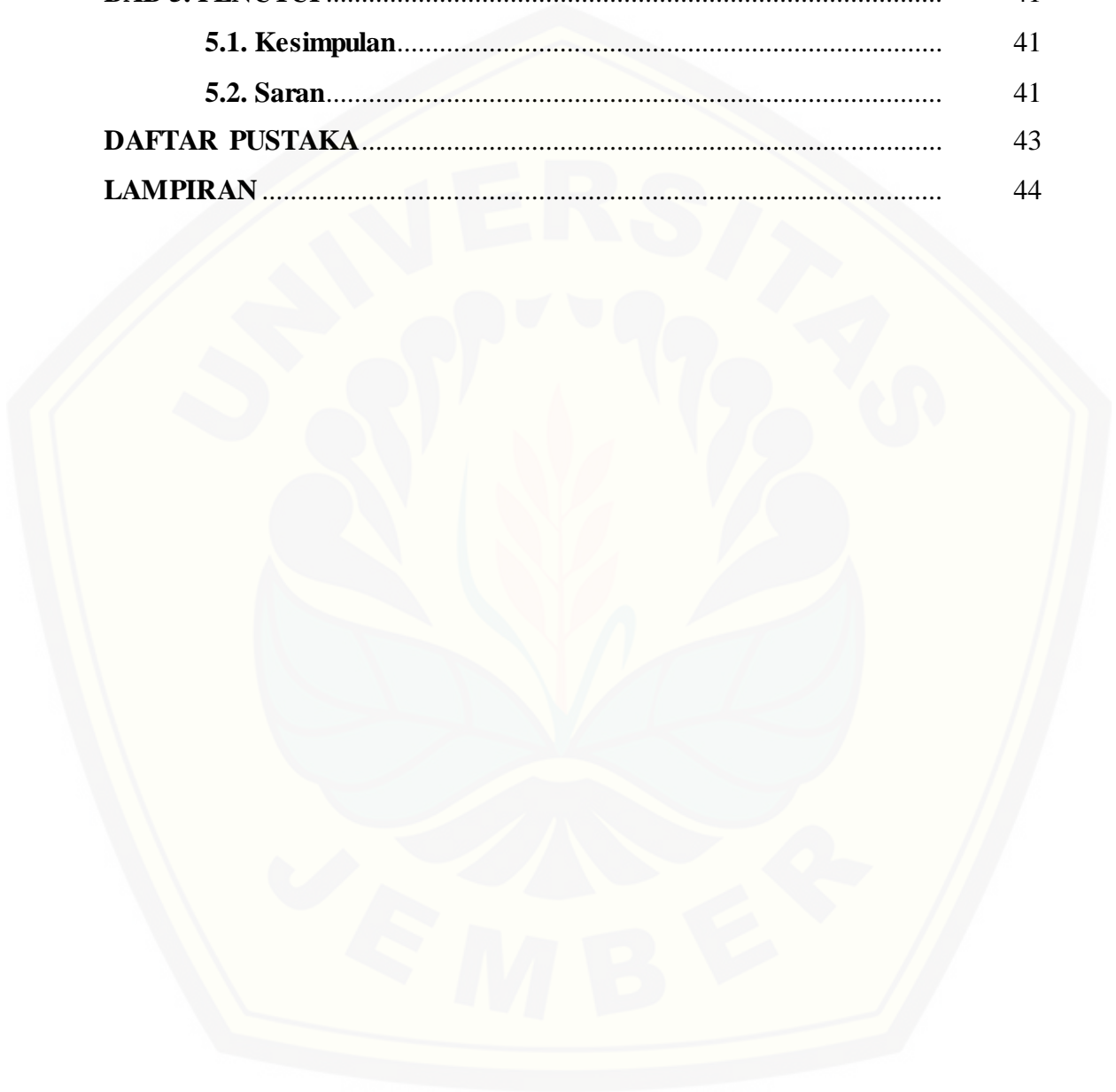
Penulis

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 | xviii |
| DAFTAR LAMPIRAN | xx |
| BAB 1. PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 2 |
| 1.3 Batasan Masalah | 2 |
| 1.4 Tujuan | 3 |
| 1.5 Manfaat | 3 |
| BAB 2. TINJAUAN PUSTAKA | 4 |
| 2.1 Generator DC Penguatan Terpisah | 4 |
| 2.1.1 Prinsip Kerja Generator DC penguatan terpisah..... | 4 |
| 2.1.2 Teori Kontrol Tegangan <i>Output</i> Generator DC Penguatan Terpisah..... | 6 |
| 2.2 Sensor Tegangan DC | 7 |

| | |
|---|----|
| 2.3 Buck Converter | 7 |
| 2.4 Arduino UNO R3 | 12 |
| 2.5 Kontrol PID | 14 |
| 2.5.1 Sistem Kontrol <i>PID</i> | 14 |
| 2.5.2 <i>Tunning</i> Parameter <i>PID</i> | 18 |
| BAB 3. METODOLOGI PENELITIAN | 19 |
| 3.1 Alat dan Bahan | 19 |
| 3.2 Tahapan Penelitian | 20 |
| 3.3 Blok Diagram Sistem | 22 |
| 3.4 Perancangan Hardware | 23 |
| 3.4.1 Sensor Tegangan..... | 23 |
| 3.4.2 <i>Buck Converter</i> | 24 |
| 3.4.3 Arduino UNO R3..... | 25 |
| 3.4.4 Sistem Generator DC Penguatan Terpisah..... | 25 |
| 3.5 Perancangan Kontrol PID Diskrit | 26 |
| 3.5.1 Perancangan Kontrol <i>PID</i> Diskrit..... | 26 |
| 3.5.2 Metode <i>Manual Tunning</i> | 28 |
| 3.6 Aplikasi Monitoring Generator | 28 |
| BAB 4. HASIL DAN PEMBAHASAN | 29 |
| 4.1 Pengujian Sensor Tegangan | 30 |
| 4.2 Pengujian <i>Buck Converter</i> | 31 |
| 4.3 Pengujian Pembebanan dan Pelepasan Beban Generator DC Penguatan Terpisah Tanpa Kontrol | 33 |
| 4.4 Pengujian Generator DC Penguatan Terpisah dengan Kontrol <i>PID</i> | 33 |
| 4.4.1 Penentuan Nilai Parameter <i>PID</i> dengan Metode <i>Tunning Manual</i> | 33 |
| 4.4.2 Pengujian Pembebanan dan Pelepasan Beban..... | 35 |
| 4.4.2.1. Pengujian Pemberian Beban..... | 35 |

| | |
|--|----|
| 4.4.2.2. Pengujian Pelepasan Beban..... | 37 |
| 4.5 Perbandingan Sistem <i>Open Loop</i> dan <i>Close Loop PID</i> | 40 |
| BAB 5. PENUTUP | 41 |
| 5.1. Kesimpulan..... | 41 |
| 5.2. Saran..... | 41 |
| DAFTAR PUSTAKA | 43 |
| LAMPIRAN | 44 |



DAFTAR TABEL

| | Halaman |
|--|---------|
| 2.1 Spesifikasi Arduino UNO-R3 | 14 |
| 2.2 Karakteristik Masing – Masing Kontrol <i>P-I-D</i> | 18 |
| 4.1 Pengujian Sensor Tegangan | 30 |
| 4.2 Pengujian <i>Buck Converter</i> | 31 |
| 4.3 Daya <i>Input</i> dan Daya <i>Output</i> | 31 |
| 4.4 Pengujian Pemberian Beban Generator <i>Open Loop</i> | 33 |
| 4.5 Pengujian Pelepasan Beban Generator <i>Open Loop</i> | 33 |
| 4.6 <i>Tunning Manual</i> Parameter <i>PID</i> | 35 |
| 4.7 Pengujian Pemberian Beban..... | 35 |
| 4.8 Pengujian Pelepasan Beban..... | 38 |
| 4.9 Perbandingan <i>Drop</i> Tegangan Sistem <i>Open Loop</i> dan <i>Close Loop PID</i> | 40 |
| 4.10 Perbandingan <i>Over Voltage</i> Sistem <i>Open Loop</i> dan <i>Close Loop PID</i> | 40 |

DAFTAR GAMBAR

| | Halaman |
|--|---------|
| 2.1 Rangkaian Generator DC Penguatan Terpisah..... | 4 |
| 2.2 Grafik hubungan $I_f - \phi$ | 5 |
| 2.3 Grafik Hubungan $E_a - I_f$ | 6 |
| 2.4 Rangkaian Sensor Tegangan..... | 7 |
| 2.5 Rangkaian <i>Buck Converter</i> | 8 |
| 2.6 (a) Rangkaian Ekivalen Saat <i>Switch</i> Tertutup (b) Rangkaian Ekivalen Saat <i>Switch</i> Tertutup (c) Tegangan Induktor pada Satu Perioda <i>Switching</i> | 9 |
| 2.7 Grafik hubungan $V_o - D$ | 10 |
| 2.8 Arus Induktor..... | 11 |
| 2.9 Tegangan <i>Ripple</i> | 12 |
| 2.10 (a) Modul Kontroler Arduino UNO-R3 (b) Pinout Arduino UNO-R3 | 13 |
| 2.11 Skema Pengendali <i>PID</i> | 14 |
| 2.12 (a) Rangkaian Elektronika <i>PID</i> Analog (b) Fungsi Transfer..... | 15 |
| 2.13 Metoda Pencuplikan. (a) <i>Backward Rectangular Methode</i> (a) <i>Foward Rectangular Methode</i> | 17 |

| | | |
|------|---|----|
| 3.1 | <i>Flowchart</i> Penelitian..... | 21 |
| 3.2 | Blok Diagram Sistem..... | 22 |
| 3.3 | Skematik Sensor Tegangan..... | 24 |
| 3.4 | Skematik <i>Buck Converter</i> | 24 |
| 3.5 | Generator DC..... | 26 |
| 3.6 | <i>Flowchart</i> Program Kontrol <i>PID</i> Diskrit..... | 27 |
| 3.7 | Aplikasi Monitoring Generator | 28 |
| 4.1 | Grafik Hubungan Antara Tegangan <i>Output</i> dan <i>Duty Cycle</i> | 32 |
| 4.2 | Pemberian Beban 15 Watt..... | 36 |
| 4.3 | Pemberian Beban 25 Watt..... | 36 |
| 4.4 | Pemberian Beban 40 Watt..... | 36 |
| 4.5 | Pemberian Beban 50 Watt..... | 37 |
| 4.6 | Pemberian Beban 65 Watt..... | 37 |
| 4.7 | Pengujian Pelepasan Beban 15 Watt..... | 38 |
| 4.8 | Pengujian pelepasan beban 25 Watt..... | 38 |
| 4.9 | Pengujian Pelepasan Beban 40 Watt..... | 39 |
| 4.10 | Pengujian Pelepasan Beban 50 Watt..... | 39 |
| 4.11 | Pengujian Pelepasan Beban 65 Watt..... | 39 |

DAFTAR LAMPIRAN

| | Halaman |
|--|---------|
| A. Listing Program Arduino UNO R3..... | 44 |
| B. Listing Program Monitoring C#..... | 47 |
| C. Gambar <i>Hardware</i> | 50 |
| C1. <i>Buck Converter</i> dan <i>Voltage Sensore</i> | 50 |
| C2. Generator DC Penguatan Terpisah..... | 50 |
| C3. Beban Lampu Pijar..... | 51 |
| D. Gambar Keseluruhan Sistem dan Pengambilan Data..... | 51 |

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 (Hameyer, 2001).

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.

Namun tegangan *output* akan cenderung turun bila arus beban semakin besar. Untuk mengatasi hal tersebut dapat dilakukan dengan cara meningkatkan tegangan induksi (E_a) pada jangkar generator DC. Untuk meningkatkan tegangan induksi jangkar dapat dilakukan dengan cara meningkatkan kecepatan putaran atau meningkatkan flux pada medan generator DC. Pada kecepatan putaran generator DC sulit dikendalikan karena dibutuhkan pengontrol kecepatan yang terdiri dari kontrol mekanis atau elektris yang rumit, terlebih pada kecepatan yang tinggi. Oleh karena itu untuk meningkatkan tegangan induksi pada jangkar generator DC akan lebih mudah bila diatur arus eksitasi pada medan generator DC sehingga nilai flux pada medan akan meningkat (Chapman, 2005)

Untuk meningkatkan arus eksitasi diperlukan kontrol agar nilai tegangan induksi medan tidak terlalu tinggi atau pun rendah. 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 arus eksitasi pada medan secara otomatis bila

terjadi penurunan tegangan *output* karena meningkatnya arus beban, dengan *feedback* tegangan *output* generator DC.

Implementasi kontrol *PID* ini diterapkan pada generator DC penguatan terpisah, karena sebagai sumber tegangan, karakteristik dari generator DC penguatan terpisah memiliki tegangan *output* tidak konstan dan cenderung turun bila arus beban naik. Diharapkan dengan diterapkannya kontrol *PID* pada sistem generator DC penguatan terpisah, tegangan *output* akan lebih stabil. Seiring dengan berkembangnya waktu sistem *PID* dapat direalisasikan pada *embedded system* seperti arduino UNO R3, sistem *PID* ini akan lebih sederhana dari pada sistem elektronika analog, mekanis atau pneumatis murni.

1.2 Rumusan Masalah

Dalam penelitian ini ada beberapa hal yang menjadi rumusan masalah untuk menjadi rujukan apa yang akan dilakukan dan diteliti, diantaranya :

1. Bagaimana merancang sistem eksitasi generator DC berbasis arduino UNO R3 menggunakan *buck converter* dan kontrol *PID*.
2. Bagaimana peningkatan kinerja generator DC dengan sistem eksitasi berbasis arduino UNO R3 menggunakan *buck converter* dan kontrol *PID*.

1.3 Batasan Masalah

Berdasarkan uraian rumusan masalah tersebut, maka pembahasan pada penelitian ini dibatasi pada pembahasan berikut agar pembahasan lebih terfokuskan :

1. Peningkatan kinerja generator DC hanya dengan mengatur arus eksitasi medan menggunakan *buck converter*
2. Generator yang digunakan adalah generator DC penguatan terpisah dan *Prime mover* menggunakan mesin DC *compound*.
3. Pengujian kinerja pada kestabilan tegangan *output* menggunakan pengujian pemberian beban dan pelepasan beban dengan variasi beban lampu pijar 15 W, 25 W, 40 W, 50 W dan 65 W.

4. *PID* yang digunakan adalah *PID digital* dan *Tunning Parameter PID* menggunakan *manual tuning*.
5. Pembahasan parameter kestabilan tegangan *output* dengan kontrol *PID* pada *peak*, *rise time*, *settling time*, *error steady-state* dan *recovery time*.

1.4 Tujuan

Penelitian yang dilakukan ini memiliki beberapa tujuan untuk dicapai sebagai pencapaian akhir, diantaranya :

1. Membuat kontrol *PID* untuk tegangan *output* generator DC penguatan terpisah menggunakan kontrol *PID* berbasis arduino UNO R3.
2. Penerapan kontrol *PID* dapat membuat tegangan *output* generator DC lebih konstan di 220 Volt ketika diberikan variasi beban.

1.5 Manfaat

Penelitian ini memiliki beberapa manfaat diantaranya :

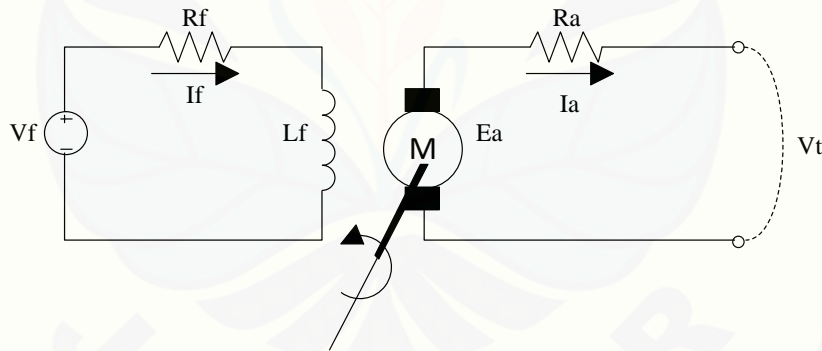
1. Mengatasi *drop* tegangan dari generator DC penguatan terpisah ketika diberikan beban dan *over voltage* ketika terjadi pelepasan beban.
2. Tegangan *output* generator DC penguatan terpisah akan diatur secara otomatis dengan sistem *close loop*.
3. Penggunaan beban pada generator DC penguatan terpisah yang lebih besar dengan tidak menurunkan tegangan *output*.

BAB 2. TINJAUAN PUSTAKA

2.1 Generator DC Penguatan Terpisah

2.1.1. Prinsip Kerja Generator DC penguatan terpisah

Generator DC penguatan terpisah adalah salah satu mesin DC yang memiliki lilitan medan yang secara listrik tidak tergantung dari mesin itu sendiri. Untuk membangkitkan fluks pada medan diperlukan arus searah yang dipasangkan pada kumparan medan yang mempunyai tahanan R_f sehingga menghasilkan fluks pada kedua kutub. Untuk rangkaian pada generator penguatan terpisah ditunjukkan pada gambar 2.1, dari gambar 2.1 didapatkan persamaan – persamaan berdasarkan hukum kirchhoff II tentang total tegangan pada rangkaian tertutup adalah nol, dari hal tersebut persamaan generator dc penguatan terpisah dapat dilihat pada persamaan 2.1 dan persamaan 2.2.



Gambar 2.1 Rangkaian Generator DC Penguatan Terpisah

Pada sisi rangkaian eksitasi atau stator, arus eksitasi dilambangkan dengan I_f , arus eksitasi didapatkan dengan membagi nilai tegangan eksitasi (V_f) dengan hambatan pada kumparan eksitasi (R_f) sesuai dengan hukum ohm, sedangkan pada sisi jangkar atau rotor didapatkan perhitungan bahwa tegangan *output* (V_t) dipengaruhi oleh nilai tegangan jangkar (E_a) dan nilai dari tegangan yang dihasilkan oleh arus yang mengalir pada jangkar (I_a) dan hambatan jangkar (R_a).

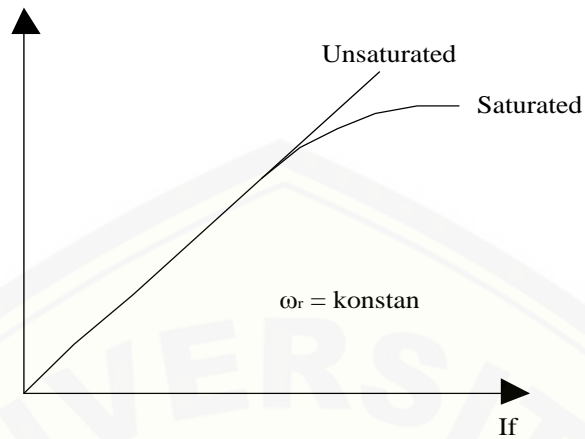
$$I_f = \frac{V_f}{R_f} \quad (2.1)$$

$$V_t = E_a -) \quad (2.2)$$

Dapat diketahui dari persamaan 2.2 bila beban semakin tinggi maka nilai tegangan *output* semakin kecil karena ada drop tegangan dari. Namun untuk meningkatkan tegangan *output* dapat meningkatkan tegangan jangkar (), pada persamaan 2.3 diketahui bahwa untuk meningkatkan tegangan jangkar, dapat meningkatkan nilai fluks () atau kecepatan putar () dengan adalah konstanta.

$$E_a = K\phi\omega_r \quad (2.3)$$

Tegangan jangkar pada generator dipengaruhi oleh nilai fluks dan kecepatan putar dari generator. Untuk meningkatkan fluks dapat dilakukan dengan cara meningkatkan arus eksitasi atau tegangan eksitasi pada medan generator. karakteristik fluks dengan arus eksitasi dapat dikatakan linier sebelum mencapai saturasi. Semakin tinggi nilai arus eksitasi semakin tinggi nilai fluks yang akan mengakibatkan tegangan induksi pada generator akan meningkat, sama halnya dengan kecepatan. Semakin tinggi tinggi kecepatan maka tegangan induksi semakin tinggi begitu pula dengan sebaliknya. Untuk menentukan nilai K pada mesin DC menurut Prof. Dr.Ing. Dr. h.c. kay hameyer pada buku berjudul “*electrical mechine 1*” yang ditentukan dengan karakteristik tanpa beban.



Gambar 2.2 Grafik hubungan –

2.1.2. Teori Kontrol Tegangan *Output* Generator DC Penguatan Terpisah

Tegangan *output* pada generator DC penguatan terpisah dapat dilakukan dengan merubah tegangan induksi pada mesin DC. menurut hukum Kirchoff II tegangan pada rangkaian tertutup akan bernilai sama dengan nol menghasilkan persamaan berikut $V_t = E_a$. Jadi, jika I_f meningkat maka tegangan *output* akan meningkat dan jika I_f menurun maka tegangan *output* akan menurun. Pada persamaan 2.3 adalah $E_a = k \omega_r \phi$ sehingga dapat dilakukan dua kemungkinan untuk mengontrol tegangan *output* yang pertama adalah ω_r (kecepatan putar) dan ϕ (fluks) :

- a. Merubah kecepatan putar (ω_r).

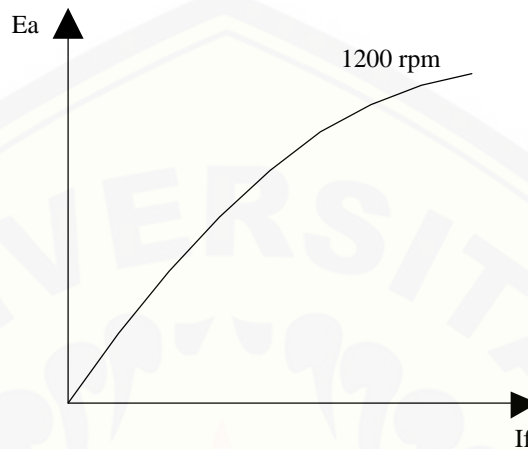
Jika ω_r dinaikan dari $E_a = k \omega_r \phi$ akan meningkatkan tegangan *output* karena persamaan tegangan *output* $V_t = E_a$.

- b. Merubah fluks (ϕ)

Jika tegangan eksitasi (V_e) pada medan dinaikan maka arus eksitasi (I_e) akan naik. Dimana jika arus medan meningkat, maka akan meningkatkan fluks dan akan

meningkatkan juga tegangan induksi $E_a =$ dan meningkatkan tegangan *output*

$$V_t = E_a.$$



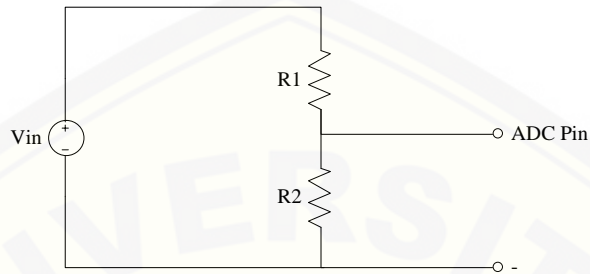
Gambar 2.3 Grafik Hubungan –

Dalam banyak aplikasi, kecepatan putar *prime mover* sangat terbatas kecepataannya dan sukar untuk dikendalikan. Dalam generator DC penguatan terpisah akan lebih mudah mengontrol tegangan *output* dengan mengendalikan arus (Chapman, 2005)

2.2 Sensor Tegangan DC

Sensor tegangan DC adalah sensor yang digunakan untuk mendeteksi tegangan dari listrik arus searah dengan keluaran tegangan analog untuk ADC pada mikrokontroler sebesar 5 Volt. Sensor ini berprinsip pada rangkaian pembagi tegangan yaitu menggunakan 2 resistor yang dirangkai secara seri seperti ditunjukkan pada gambar 2.4, pada sensor ini harus memiliki daya yang rendah agar tidak membuang energi listrik yang dikeluarkan oleh sumber energi listrik pada sensor tegangan DC tersebut. Oleh karena itu nilai resistor yang digunakan harus memiliki nilai resistansi yang tinggi. V_{in} adalah tegangan *input* pada sensor tegangan, R_1 dan R_2 adalah resistor pembanding agar keluaran V_o untuk masukkan PIN ADC pada

mikrokontroler ketika tegangan V_{in} tertinggi mengeluarkan tegangan 5 Volt, dengan persamaan untuk mencari nilai – nilai resistor R_1 dan R_2 ditunjukkan pada persamaan 2.4 (Autodesk.Inc, 2015).



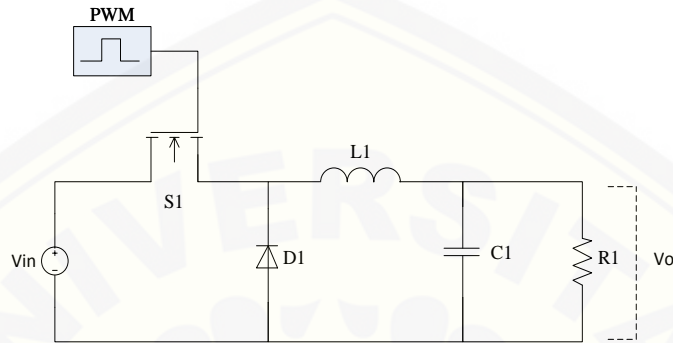
Gambar 2.4 Rangkaian Sensor Tegangan

$$V_o = V_{in} \times \left(\frac{R_2}{R_1 + R_2} \right) \quad (2.4)$$

2.3 Buck Converter

Buck converter merupakan salah satu jenis *converter* DC to DC yang dapat menurunkan tegangan masukan menjadi tegangan keluaran yang lebih rendah dengan cara mengatur *duty cycles* yang diberikan pada *switching device*. Rangkaian *buck converter* adalah seperti tampak pada gambar 2.5. Pada rangkaian tersebut terdapat beberapa komponen. Tegangan masukan (V_{in}), merupakan catu daya DC yang ingin diubah tegangannya menjadi lebih rendah. *Controllable switch* (S1), merupakan *semiconductor switching device* yang digunakan untuk pensaklaran pada *buck converter*. Beberapa contoh *switch* jenis ini adalah BJT (*bipolar Junction Transistor*), IGBT (*Insulated-gated Bipolar Transistor*) dan MOSFET (*Metal Oxide Semiconductor Field Effect*) Transistor. Induktor (L1), Merupakan komponen yang berfungsi untuk menyimpan arus. Dioda (D1), Merupakan bahan *semiconductor* yang keadaan aktif nya tergantung dari tegangan dan arus yang mengalir pada rangkaian. Dioda akan aktif jika dalam keadaan *forward bias*, yaitu tegangan anoda lebih tinggi daripada tegangan katoda, sebaliknya dioda akan tertutup jika dalam keadaan *reverse bias*, yaitu tegangan pada anoda lebih rendah daripada tegangan katoda. Kapasitor (C_1), Marupakan komponen yang berfungsi sebagai *low pass filter* dan resistor (R_1),

merupakan beban sistem dan berfungsi pula sebagai bagian dari *low pass filter*. Tegangan keluaran (V_o) merupakan hasil tegangan keluaran yang diinginkan (Gunawan, 2009).



Gambar 2.5 Rangkaian *Buck converter*

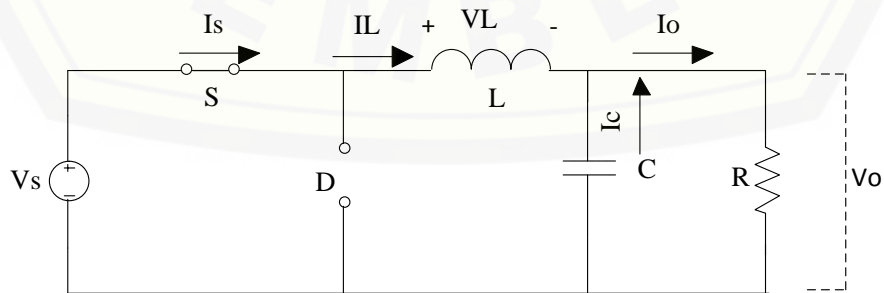
pada saat *switch* tertutup, rangkaian ekivalen *buck converter* akan tampak seperti gambar 2.6a. Pada kondisi ini tegangan induktor adalah

$$V_L = V_s - V_o = L \frac{dI_L}{dt} \quad (2.5)$$

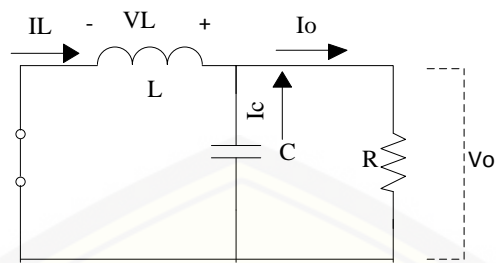
pada saat *switch* terbuka, rangkaian ekivalen *buck converter* akan tampak seperti gambar 2.6b. Pada kondisi ini tegangan induktor adalah

$$V_L = -V_o = L \frac{dI_L}{dt} \quad (2.6)$$

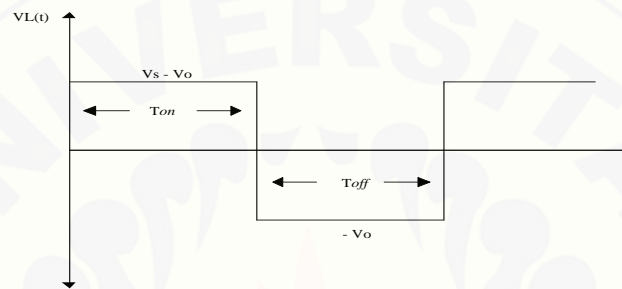
hasil yang didapatkan adalah tegangan induktor seperti ditunjukkan oleh gambar 2.6c, dimana T_{on} menunjukkan waktu saat *switch* tertutup dan T_{off} menunjukkan waktu saat *switch* terbuka.



(a)



(b)



(c)

Gambar 2.6 (a) Rangkaian Ekivalen Saat *Switch* Tertutup (b) Rangkaian Ekivalen Saat *Switch* Tertutup (c) Tegangan Induktor pada Satu Periode *Switching*

pada kondisi tunak tercapai ketika arus yang dilepas induktor sudah sama dengan arus yang diserap induktor dalam satu periode. Menurut persamaan induktor

$$V_L = L \frac{dI_L}{dt} \quad (2.7)$$

maka integrasi dalam satu periode dari $t = t_0$ sampai dengan $t = T$ didapatkan

$$I_L(T) - I_L(t_0) = \int_0^T V_L dt \quad (2.8)$$

karena ketika kondisi tunak tercapai nilai total perubahan arus adalah nol, maka didapatkan

$$0 = \int_0^T V_L dt \quad (2.9)$$

persamaan 2.5 disebut *inductor Volt-second balance* yang akan digunakan untuk menurunkan karakteristik tunak dari sistem *buck converter*, dari gambar 2.6c didapatkan nilai tegangan induktor selama satu perioda sehingga

$$\int_0^T VL dt = (V_s - V_o)DT + (-V_o)(1 - D)T \quad (2.10)$$

dengan mensubsitusikan persamaan 2.9 ke persamaan 2.10 didapatkan

$$0 = (V_s - V_o)DT + (-V_o)(1 - D)T \quad (2.11)$$

dengan membagi kedua sisi dengan T didapatkan

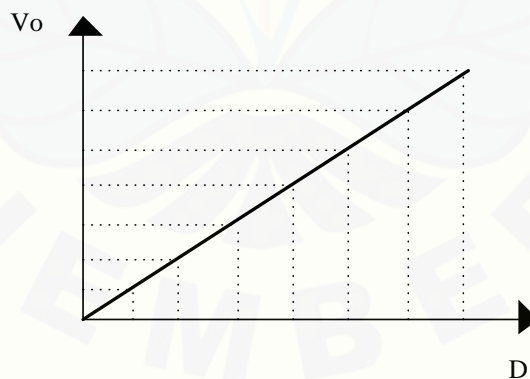
$$0 = (V_s - V_o)D + (-V_o)(1 - D) \quad (2.12)$$

akhirnya didapatkan penyelesaian untuk V_o adalah

$$V_o = DV_s = M(D)V_s \quad (2.13)$$

dimana $M(D)$ menunjukkan *gain* dari sistem.

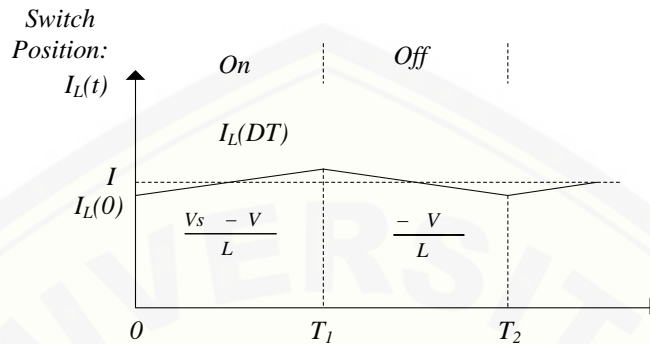
Persamaan diatas menunjukkan bahwa semakin besar nilai duty cycles maka semakin besar nilai tegangan keluaran V_o . Nilai tegangan keluaran secara ideal dapat bernilai 0 Volt hingga tegangan masukannya (Guldemir, 2011).



Gambar 2.7 Grafik hubungan $V_o - D$

Agar sistem beroperasi pada daerah yang kontinu maka arus induktor harus tetap kontinu pula dalam satu periode. Gambar menunjukkan arus dari induktor selama satu periode dalam keadaan tunak. Pada sistem *buck converter*, arus rata-rata induktor I_L dalam kondisi tunak adalah sama dengan arus beban I . Gambar 2.8

menunjukkan bahwa arus induktor berada pada sekitar arus rata-ratanya I dengan nilai arus maksimum sebesar $I + I_L$ dan arus minimum $I - I_L$.



Gambar 2.8 Arus Induktor

Untuk mendapatkan nilai arus induktor yang kontinu, nilai $I - \Delta I_L$ harus tetap berada pada nilai lebih besar daripada nol. Nilai $I - \Delta I_L$ yang positif ini yang kemudian menjadi batasan besarnya L_{min} yang dibutuhkan. Dari gambar arus pada kondisi switch kedua didapatkan.

$$\frac{2\Delta I_L}{T_s(D-1)} = \frac{-V}{L} \quad (2.14)$$

Nilai $I - \Delta I_L$ harus lebih besar daripada nol agar sistem bekerja pada daerah kontinu dan untuk itu diperlukan perhitungan nilai frekuensi dengan nilai L yang sudah ditentukan.

$$I - \Delta I_L > 0$$

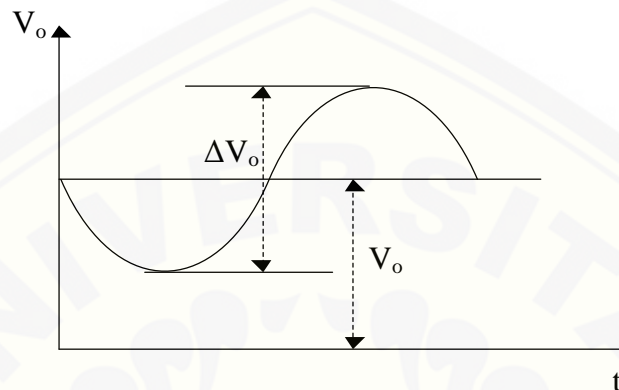
$$\frac{V}{R} - \frac{V}{2L} T_s(1-D) > 0$$

$$\frac{1}{R} > \frac{T_s(1-D)}{2L}$$

$$\frac{1}{T_s} > \frac{(1-D)R}{2L}$$

$$f > \frac{(1-D)R}{2L} \quad (2.15)$$

Tegangan keluaran sistem pada kenyataannya tidak dapat bernilai konstan secara sempurna. Hal ini dikarenakan kapasitor yang digunakan akan terus melakukan pengisian dan pelepasan muatan (Gunawan, 2009)



Gambar 2.9 Tegangan *Ripple*

Besarnya *ripple* tegangan dapat dinyatakan dalam bentuk

$$\frac{\Delta V}{V} = \frac{(1-D)}{8LCf^2} \quad (2.16)$$

2.4 Arduino UNO R3

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 2.10 (a) adalah modul Arduino UNO R3

sedangkan Gambar 2.10 (b) menunjukkan skema dari modul Arduino UNO R3 yang merupakan penyederhanaan dari gambar modul aslinya (Arduino LLC, 2015).



(a)



(b)

Gambar 2.10 (a) Modul kontroler arduino UNO-R3 (b) Pinout 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*).

Berikut ini adalah tabel spesifikasi dari arduino UNO R3 dapat kita lihat pada Tabel 2.1 dibawah ini.

Tabel 2.1 Spesifikasi arduino UNO R3

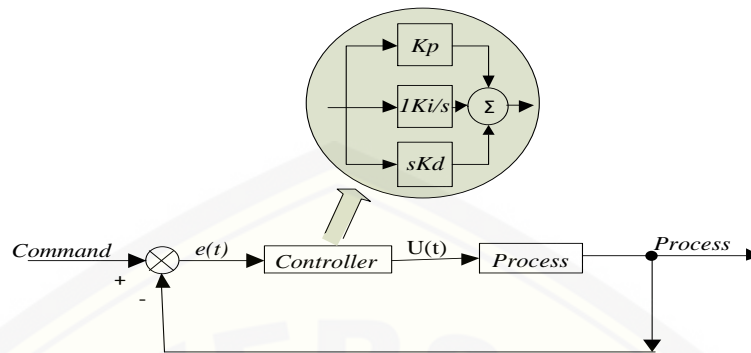
| <i>Device</i> | Keterangan |
|------------------------|--|
| <i>Microcontroller</i> | ATmega328 |
| Tegangan Operasi | 5 V |
| Tegangan <i>Input</i> | 7-12 V |
| <i>I/O Digital</i> | 14 (6 buah menyediakan <i>PWM output</i>) |
| <i>Input Analog</i> | 6 buah |
| Arus DC per I/O pin | 40 mA |
| Arus DC untuk pin | 50 mA |
| <i>Flash Memory</i> | <i>Bootloader</i> |
| SRAM | 2 Kb |
| EEPROM | 1 Kb |
| Kecepatan <i>Clock</i> | 16 MHz |

Diterjemahkan dari: *datasheet* arduino UNO R3

2.5 Kontrol *PID*

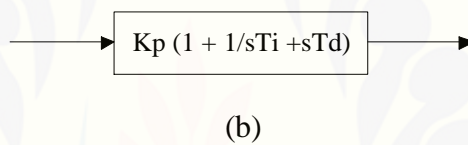
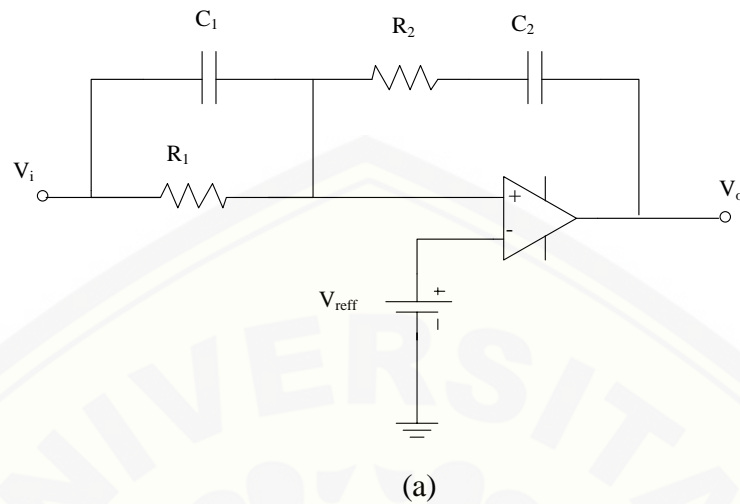
2.5.1 Sistem Kontrol *PID*

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 2.11. pada gambar 2.11 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 2.11 Skema Pengendali *PID*

Masing – masing parameter dari *PID* ini memiliki fungsi dan perannya. *Propositional* 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*. Parameter *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 2.12, 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 2.12 (a) Rangkaian Elektronika PID Analog (b) Fungsi Transfer

$$K_p = -\frac{R_1 \cdot C_1 + R_2 \cdot C_2}{R_1 \cdot C_2} \quad (2.17)$$

$$T_i = R_1 \cdot C_1 + R_2 \cdot C_2 \quad (2.1)$$

$$T_d = \frac{R_1 \cdot C_1 \cdot R_2 \cdot C_2}{R_1 \cdot C_1 + R_2 \cdot C_2} \quad (2.19)$$

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

$$U(s) = K_p \left[1 + \frac{1}{T_i s} + T_d s \right] E(s) \quad (2.20)$$

jika persamaan ini dikembalikan dalam fungsi waktu akan menjdai

$$U(t) = K_p \left[e(t) + \frac{1}{T_i} \int e(t) + T_d \frac{de(t)}{dt} \right] \quad (2.21)$$

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

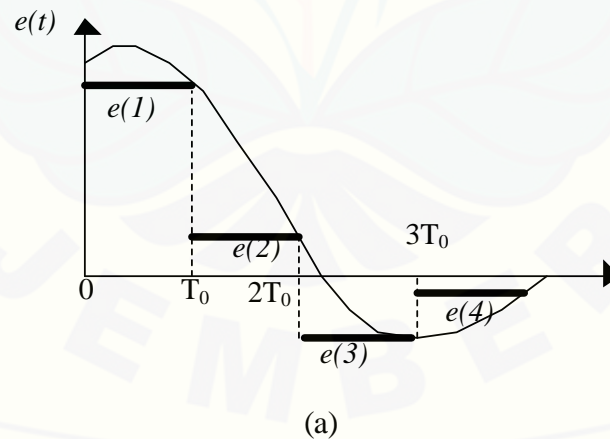
$$\frac{de}{dt} = \frac{e(k) - e(k-1)}{T_c} \quad (2.22)$$

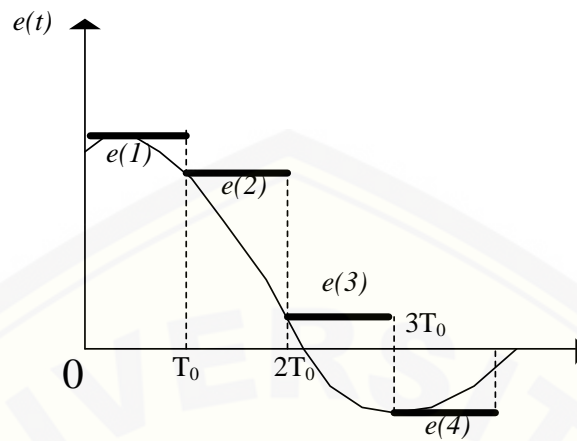
sedangkan untuk integralnya dilakukan dengan beberapa metode, diantaranya adalah *forward rectangular methode (FRM)* dan *Backward Rectangular Method (BRM)*. Pencuplikan dengan *FRM* dan *BRM* digambarkan oleh gambar 2.13. dari gambar 2.13 bisa didapatkan algoritma *integral* untuk masing – masing metode. Untuk *FRM* didapatkan

$$\int_0^T e(t)dt \approx Tc \sum_{i=1}^k e(i-1) \quad (2.23)$$

sedangkan untuk *BRM* didapatkan

$$\int_0^T e(t)dt \approx Tc \sum_{i=1}^k e(i) \quad (2.24)$$





(b)

Gambar 2.13 Metoda Pencuplikan. (a) *Backward Rectangular Methode*
(a) *Foward Rectangular Methode*

Persamaan 2.23 lebih sering digunakan dalam menulis persamaan *PID*. Memakai persamaan 2.23 ini dan persamaan 2.21 untuk menggantikan persamaan 2.20 didapatkan *absolute/position algorithm* untuk pengendali *PID*

$$u(k) = K_p \left\{ e(k) + \frac{T_c}{T_i} \sum_{i=1}^k e(i) + \frac{T_d}{T_c} [e(k) - e(k-1)] \right\} \quad (2.25)$$

Pengendali *PID* pada persamaan 2.24 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

$$u(k) = U_p(k) + U_i(k) + U_D(k) \quad (2.26)$$

dimana,

$$U_p(k) = K_p \cdot e(k)$$

$$U_i(k) = K_p \frac{T_c}{T_i} \sum_{i=1}^k e(a) = U_i(k-1) + K_p \frac{T_c}{T_i} e(k)$$

$$U_D(k) = K_p \frac{T_d}{T_c} [e(k) - e(k-1)] \quad (2.27)$$

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 gak 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 2.2 Karakteristik masing – masing kontrol *P-I-D*

| <i>Respon loop</i> tertutup | <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 *plant* 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 (Jaya, Irianto, & Setiadji, 2011).

BAB 3. METODOLOGI PENELITIAN

Untuk 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 Alat dan Bahan

Alat dan bahan yang digunakan dalam penelitian ini adalah sebagai berikut:

Alat :

1. Multimeter
2. Tachometer
3. Timer
4. Data *logger*

Bahan :

1. Generator DC penguatan terpisah
2. Motor DC sebagai *prime mover*
3. *Belt*
4. AVR
5. Arduino UNO R3
6. Sensor tegangan
7. *Buck converter*
8. *Rectifier*
9. MCB
10. Lampu pijar
11. *Jumper*

3.2 Tahapan Penelitian

Tahapan yang dilakukan pada penelitian rancang bangun kontrol *PID* tegangan *output* generator DC penguatan terpisah berbasis arduino UNO R3 adalah sebagai berikut :

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

Tahapan kedua adalah melakukan percobaan pada generator DC penguatan terpisah (*feedback* tipe 63-120) untuk respon *open loop* ketika dibebani dengan variasi beban resistif, yaitu menggunakan lampu pijar hingga beban maksimum 65 W.

3. Perancangan sistem

Tahap ketiga adalah proses perancangan sistem kontrol *PID* tegangan *output* generator DC penguatan terpisah yang terdiri dari pemilihan *plan* yang akan digunakan, *Hardware* pengendali seperti sensor tegangan, arduino UNO R3 dan *buck converter*, dan terakhir perancangan kontrol *PID*.

4. Pembuatan program kontrol *PID* pada arduino

Tahap keempat adalah membuat parameter – parameter agar kontrol *PID* bekerja dengan baik pada sistem. Parameter terdiri dari *rise time*, *settling time*, *error steady-state* dan *peak*.

5. Pengujian sistem

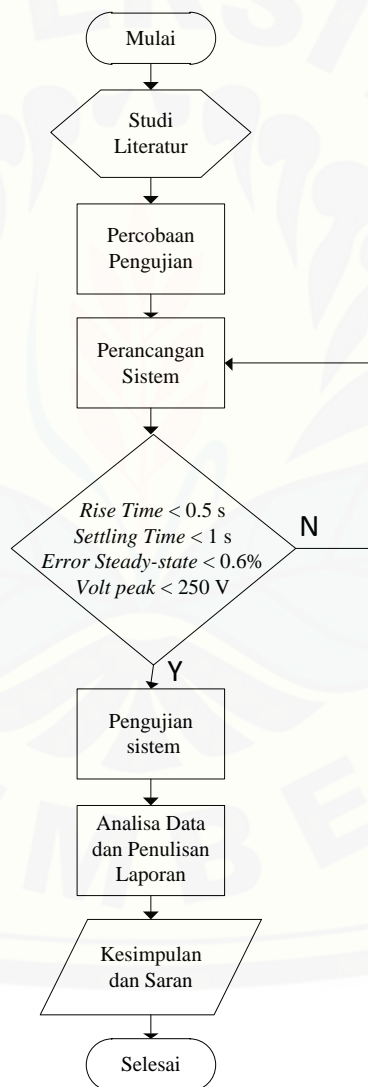
Tahap kelima adalah pengujian terhadap sistem yang telah dibuat seperti sensor tegangan, *buck converter* dan generator dengan memberikan beban pada *plan* yang sudah disertai kontrol *PID* menggunakan variasi beban lampu pijar 15, 25, 40, 50 dan 65 Watt.

6. Analisa data dan pembuatan laporan

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

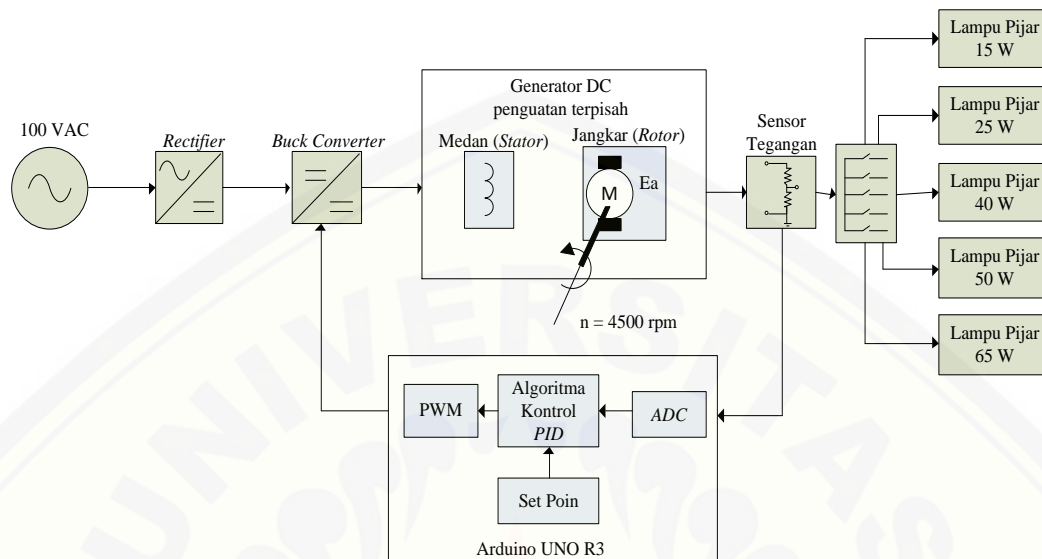
7. Kesimpulan dan saran

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



Gambar 3.1 *Flowchart* Penelitian

3.3 Blok Diagram Sistem



Gambar 3.2 Blok Diagram Sistem

Prinsip kerja dari sistem rancang bangun kontrol *PID* pada tegangan *output* generator DC penguatan terpisah berbasis arduino UNO R3, dapat digambarkan pada blok diagram gambar 3.2. Generator DC penguatan terpisah akan diputar oleh sebuah *prime mover* dengan kecepatan 4500 rpm. Namun, generator belum dapat menghasilkan energi listrik karena belum terdapat eksitasi pada medan. Ketika set poin 220 V diberikan pada algoritma kontrol *PID* yang diprogram pada arduino UNO R3, kontrol akan merespon agar arduino UNO R3 memberikan sinyal *PWM* ke *buck converter* untuk memberikan tegangan ke medan. Sehingga generator dapat membangkitkan energi listrik. Untuk menghasilkan energi listrik dengan tegangan yang stabil pada 220 V, nilai tegangan *output* dikembalikan lagi ke kontrol *PID* melalui sensor tegangan yang merupakan rangkaian *voltage divider*, yang berfungsi merubah tegangan DC yang lebih tinggi menjadi tegangan DC yang lebih rendah agar dapat dibaca aman oleh arduino UNO R3 melalui pin ADC yang akan diproses kembali oleh algoritma kontrol *PID* sehingga mendapatkan tegangan yang tetap 220 V dan stabil meski terdapat perubahan beban. Pengujian beban menggunakan lampu pijar dengan variasi 15, 25, 40, 50 dan 65 Watt.

3.4 Perancangan *Hardware*

3.4.1. Sensor tegangan

Sensor tegangan yang akan digunakan adalah rangkaian pembagi tegangan yang telah dijelaskan pada bab sebelumnya. Perancangan dilakukan dengan melakukan perhitungan nilai resistor yang akan digunakan yang ditunjukkan pada persamaan 3.1. Sehingga didapatkan rangkaian pembagi tegangan ini menggunakan resistor 300 k Ω dan resistor 4k7 Ω yang dirangkai secara seri, agar dapat membaca tegangan 0 – 324.15 V dan perbandingan tegangan untuk diberikan ke arduino UNO R3 0 – 5 V. Karena resistor 300 k Ω sukar ditemukan, maka dapat diganti dengan 3 buah resistor 100 k Ω yang dirangkai secara seri. Rangkaian sensor tegangan ditunjukkan pada gambar 3.3 (Autodesk, Inc, 2015).

$$V_{out} = \frac{V_{in} \times R_2}{(R_1 + R_2)} \quad (3.1)$$

dimana, $V_{out} = 5$; $V_{in} = 324.15$; $R_1 = 300 \text{ k}\Omega$

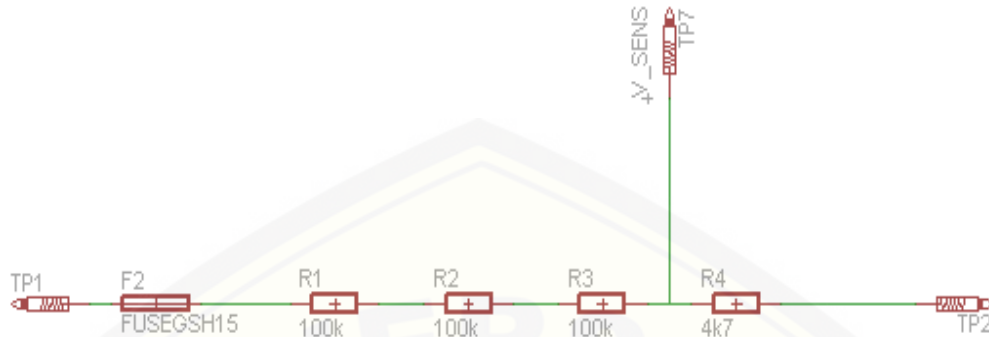
$$5 = \frac{324.15 \times R_2}{(300k + R_2)}$$

$$5 \times (300k + R_2) = 324.15 \times R_2$$

$$1500k + 5R_2 = 324.15R_2$$

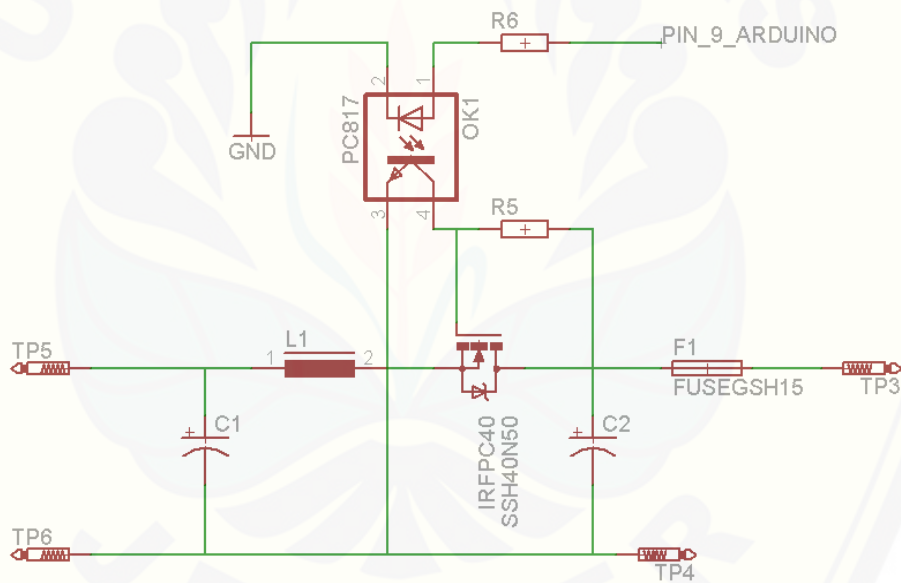
$$R_2 = \frac{1500k}{319.15}$$

$$R_2 = 4k7 \Omega$$



Gambar 3.3 Skematik Sensor Tegangan

3.4.2. *Buck converter*



Gambar 3.4 Skematik *Buck Converter*

Switching device yang digunakan pada *buck converter* menggunakan IGBT SSH40N50. IGBT tersebut mampu menahan tegangan antara *collector* dan *emitter* hingga 220 V, sedangkan *optocoupler* digunakan untuk memisah tegangan yang lebih tinggi dari tegangan yang digunakan pada komponen arduino UNO R3 agar ketika terjadi kesalahan pada tegangan yang lebih tinggi, arduino UNO R3 akan tetap aman dan juga berfungsi sebagai *switching device* untuk IGBT (Guldemir, 2011). Nilai C_1

dan C_2 adalah 470 μF dan nilai induktor yang digunakan adalah 22.0 mH dengan hambatan beban pada medan generator DC yang digunakan adalah 914 Ω . Agar *buck converter* bekerja secara kontinu diperlukan perhitungan nilai frekuensi yang digunakan, dengan persamaan 2.15

$$f > \frac{(1-D)R}{2L}$$

dimana, $R = 914 \Omega$ dan $L = 22 \text{ Mh}$

$$f > \frac{(1-D)914}{2(22 \times 10^{-3})}$$

$$f > 20(1-D)$$

jadi agar semua kondisi *duty cycle* terpenuhi maka didapatkan nilai frekuensi harus lebih dari 20,772.72 Hz.

3.4.3. 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. Untuk memberikan sinyal PWM pada *buck converter* dapat menggunakan pin – pin PWM pada arduino UNO R3, yaitu pada pin 3, 5, 6, 9 dan 10. Pada penelitian ini digunakan pin 9, karena menurut website resmi arduino frekuensi PWM yang dapat dihasilkan pada pin PWM ini adalah 31,250 Hz (Arduino LLC, 2015).

3.4.4. Sistem Generator DC Penguatan Terpisah

Plan yang digunakan pada penelitian ini menggunakan generator DC dari *feedback* dengan nomor seri 63-120 yang dapat dirangkai menjadi generator DC penguatan terpisah. Generator DC ini memiliki tegangan kerja pada 220V dengan

daya yang dapat dikeluarkan 250 Watt. Untuk kecepatan nominal generator 2000 rpm dan kecepatan maksimum 6000 rpm.



Gambar 3.5 Generator DC

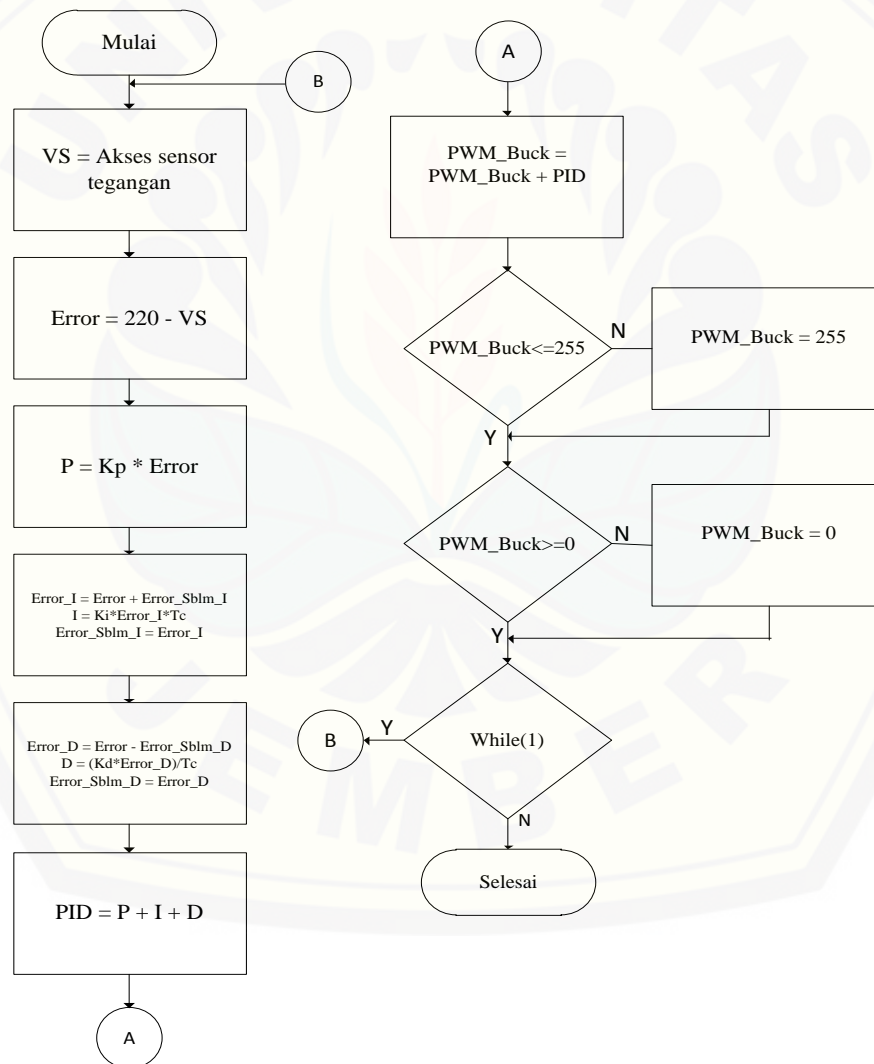
(Sumber : *feedback mechine datasheet*)

Generator ini memiliki 2 kumparan medan pada stator, yaitu kumparan pendek dan kumparan panjang. Generator DC ini dapat dirangkai seri, *shunt* dan *compound*. Untuk menggunakan sebagai generator DC ini sebagai penguatan terpisah, dapat menggunakan kumparan panjang sebagai medan dengan maksimum tegangan pada medan kumparan panjang 220 VDC dan dengan mengukur resistansi hambatan medan kumparan panjang didapat hambatan pada medan kumparan panjang 914 Ω . Namun, setelah melakukan pengujian percobaan pada generator dengan kecepatan 4500 rpm dan eksitasi 63.1 Volt, generator dapat membangkitkan tegangan keluaran 220 V. Oleh karena itu desain *buck converter* yang berfungsi sebagai pemberi tegangan eksitasi, ketika *duty cycles* 100% didesain cukup hanya dengan tegangan keluaran sekitar 89 - 92 Volt dengan tegangan masukan 100 Volt. *Prime mover* menggunakan mesin DC seri dengan daya 1 HP atau 0.75 KW tegangan 220 V dengan kecepatan putar yang digunakan adalah 4500 rpm. Sistem kopel menggunakan belt dengan tipe B2340. Untuk visual sistem bisa dilihat pada lampiran D.

3.5 Kontrol *PID* Diskrit

3.5.1. Perancangan Kontrol *PID* Diskrit

Perancangan kontrol *PID* diskrit mengacu pada persamaan 2.25 dan persamaan 2.26. Untuk membuat kontrol *PID* diskrit diperlukan algoritma program kontrol *PID* agar menjadi kontrol yang sesuai yang akan dimasukkan ke dalam program arduino. Pada gambar 3.6 merupakan *flowchart* dari algoritma program kontrol *PID* diskrit untuk rancang bangun kontrol *PID* pada tegangan *output* generator DC penguatan terpisah berbasis arduino UNO R3 ini.



Gambar 3.6 *Flowchart* Program Kontrol *PID* Diskrit

BAB 5. PENUTUP

5.1 Kesimpulan

Setelah melakukan pengujian dan analisa pada penelitian yang berjudul “Peningkatan Kinerja Generator DC dengan Sistem Eksitasi Berbasis Arduino UNO R3 Menggunakan *Buck Converter* dan Kontrol *PID*” didapatkan beberapa kesimpulan sebagai berikut.

1. Perancangan *buck converter* menggunakan nilai C_1 dan C_2 sebesar $470 \mu\text{F}$ dan nilai induktor 22 mH dengan beban hambatan medan 914Ω membutuhkan nilai $f > 20772.72 \text{ Hz}$, nilai efisiensi tertinggi pada *duty cycle* 100% dengan nilai efisiensi 87% . Kontrol *PID* yang digunakan sebagai peningkatan kinerja tegangan *output* generator DC didapatkan nilai parameter kontrol *PID* yang sesuai dengan sistem ini adalah $K_p = 3$, $K_i = 0.00005$ dan $K_d = 0.00002$ dengan nilai *rise time* 0.2 s , *settling time* 0.8 s , *error steady-state* 0.21% dan *peak* 241 volt .
2. Perbaikan Kinerja tegangan *output* generator DC meningkat dengan data pengujian pemberian beban dengan sistem *close loop PID* hingga 65 Watt drop tegangan mencapai 19 Volt sedangkan dengan sistem *open loop* mencapai 39.87 Volt , untuk *recovery-time* mencapai 3.1 s dan *error steady-state* tertinggi adalah 0.67% . Sedangkan pada pengujian pelepasan beban dengan sistem *close loop PID* hingga pelepasan beban $65 \text{ Watt over voltage}$ mencapai 8 Volt sedangkan dengan sistem *open loop* mencapai 18.69 Volt , untuk *recovery-time* mencapai 1.5 s dan *error steady-state* cenderung stabil setelah pelepasan beban yaitu sekitar $19 - 21 \%$.

5.2 Saran

Setelah melakukan penelitian, penulis memberikan saran untuk mengembangkan penelitian ini untuk penelitian berikutnya. Saran yang dapat disampaikan adalah sebagai berikut.

1. Kontrol *PID* dapat diganti dengan kontrol *PI* atau *PD* sebagai perbandingan.
2. *Buck converter* dapat diganti menggunakan *boost converter* atau *buck-boost converter*.
3. Kontrol *fuzzy logic* dapat digunakan sebagai sistem kontrol utama untuk menghilangkan *peak*.
4. *Plan* yang digunakan dapat diganti dengan generator AC atau generator dengan jenis penguatan lainnya seperti *shunt*, *series* atau *compound*.
5. Metode yang digunakan untuk mencari nilai parameter menggunakan metode ziegler nichols 1 atau ziegler nichols 2.

DAFTAR PUSTAKA

- Arduino LLC. (2015). *www.arduino.cc/en/Main/ArduinoBoardUno*. Dipetik February 24, 2016, dari arduino.cc/en.
- Autodesk.Inc. (2015). *id/Arduino-Voltage-Sensor-0-25V/*. Dipetik Maret 11, 2016, dari instructables: <http://www.instructables.com/id/Arduino-Voltage-Sensor-0-25V/>
- Chapman, S. J. (2005). *Electric Machinery Fundamentals*. New York: Library of Congress Cataloging.
- Febrianto, V. N. (2014). Aplikasi Kontrol PID untuk Pengaturan Putaran Motor DC pada Alat Pengepres Adonan Roti (Screw Conveyor).
- Guldemir, H. (2011, September). Study of Sliding Mode Control of DC-DC Buck Converter. *Scientific Research* .
- Gunawan. (2009). *Rancang Bangun DC-DC Buck Converter dengan PID Diskrit Sebagai Pengendali Tegangan Keluaran*. Depok: Universitas Indonesia.
- Hameyer, K. (2001). *Electrical Machine*. Bundesland NRW, Jerman.
- Jamil, R., Jamil, I., Jinqun, Z., Ming, L., Dong, W. Y., & Jamil, R. (2013). Control and Configuration of Generator Excitation System as Current Mainstream. *International Journal of Innovation and Applied Studies* .
- Jaya, A., Irianto, & Setiadji, A. (2011). Implementasi Kontroler PID pada AVR (Automatic Voltage Regulator) untuk Pengaturan Tegangan Eksitasi Generator Sinkron 3 Fasa.

LAMPIRAN**A. Listing Program Arduino UNO R3**

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

int mulai = 0;

float sens =0.00;
int tegangan = 0;

int setPoint = 0;
float KP=0;
float KI=0;
float KD=0;
float error_sblm_I=0;
float error_sblm_D=0;
float Tc=0.1;
int error;
int error_I;
int error_D;
float out_P;
float out_I;
float out_D;
float out_PID;
int pwm;
int con;

void akses_tegangan() {
  sens = (float)analogRead(A0)/1023;
  tegangan = sens * 324.15;
}
void kontrol_PID() {
  error = setPoint-tegangan;
  if(error>=200)
  {
    error=200;
  }
  if(error<=(-200))
  {
    error=-200;
  }

  out_P= KP*error;

  error_I = error + error_sblm_I;
```

```
out_I= KI*error_I*Tc;
error_sblm_I = error_I;

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

out_PID = out_P + out_I + out_D;
}
void kontrol() {
  pwm=pwm+out_PID;
  if(pwm>=255)
  {
    pwm=255;
  }
  if(pwm<=0)
  {
    pwm=0;
  }
  con=255-pwm;
  analogWrite(9, con);
}

void kirim_data_ke_PC(){
  Serial.print(tegangan);
  Serial.print("\n");
  delay(100);
}
void terima_serial(){
  while(Serial.available()>0) {
    char inChar = (char)Serial.read();
    if(inChar == '\n'){
      parsing = true;
    }else{
      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];
    }
  }
}
```

```
    }  
  }  
  mulai = data[0].toInt();  
  KP=data[1].toFloat();  
  KI=data[2].toFloat();  
  KD=data[3].toFloat();  
  setPoint = data[4].toInt();  
  parsing=false;  
  inData = "";  
}  
  
}  
  
void setup() {  
  Serial.begin(9600);  
  pinMode(9,OUTPUT);  
  analogWrite(9,255);  
}  
  
void loop() {  
  terima_serial();  
  
  if(mulai == 1){  
    akses_tegangan();  
    kontrol_PID();  
    kontrol();  
    kirim_data_ke_PC();  
  }  
  else if(mulai == 0){  
  }  
}
```


B. Listing Program Monitoring C#

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

using System.IO.Ports;
using System.IO;

namespace GCU_Monitoring
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            string[] portNames = System.IO.Ports.SerialPort.GetPortNames();
            for (int i = 0; i <= portNames.Length - 1; i++)
            {
                comboBox1.Items.Add(portNames[i]);
            }
            comboBox2.Items.Add("110");
            comboBox2.Items.Add("300");
            comboBox2.Items.Add("600");
            comboBox2.Items.Add("1200");
            comboBox2.Items.Add("2400");
            comboBox2.Items.Add("4800");
            comboBox2.Items.Add("9600");
            comboBox2.Items.Add("14400");
            comboBox2.Items.Add("19200");
            comboBox2.Items.Add("38400");
            comboBox2.Items.Add("56000");
            comboBox2.Items.Add("115200");
        }

        private void Form1_FormClosing(object sender, FormClosingEventArgs e)
        {
            serialPort1.Close();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (!serialPort1.IsOpen)
            {
                if (comboBox1.Text.Length > 0 && comboBox2.Text.Length > 0)
                {
                    serialPort1.PortName = comboBox1.Text;
                    serialPort1.BaudRate = Convert.ToInt32(comboBox2.Text);
                    serialPort1.Open();
                    button1.Text = "disConnect";
                }
            }
            else
            {

```

```

        MessageBox.Show("Please Choice PORT and BaudRate !!!");
    }
}
else
{
    serialPort1.Close();
    button1.Text = "Connect";
}
}

private void serialPort1_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    Invoke(new EventHandler(bacaData));
}

string indata;
double detik=0;
private void bacaData(object sender, EventArgs e)
{
    indata = serialPort1.ReadLine();
    string[] data = indata.Split(' ');
    textBox1.Text = data[0];
    richTextBox1.AppendText(data[0] + "\n");
    detik++;
    axMathworks_Strip1.AddXY(0, detik, Convert.ToDouble(data[0]));
}

private void richTextBox1_TextChanged(object sender, EventArgs e)
{
    richTextBox1.AllowDrop = true;
    richTextBox1.ScrollToCaret();
}

private void button2_Click(object sender, EventArgs e)
{
    richTextBox1.SaveFile("D:\\File Logger\\data_logger.txt", RichTextBoxStreamType.PlainText);
}

private void button3_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen)
    {
        if (Convert.ToInt32(textBox2.Text) <= 220 && textBox2.Text != "" && textBox5.Text != "" && textBox4.Text != ""
        && textBox3.Text != "" && textBox2.Text != "")
        {
            if (button4.Text == "Stop")
            {
                serialPort1.Write("1" + " "); //Mulai
                serialPort1.Write(textBox5.Text + " "); //KP
                serialPort1.Write(textBox4.Text + " "); //KI
                serialPort1.Write(textBox3.Text + " "); //KD
                serialPort1.Write(textBox2.Text + "\n"); //setPoint
            }
            else if (button4.Text == "Mulai")
            {
                MessageBox.Show("Can't Transfer Data because Serial Stopped !!!");
            }
        }
    }
    else
    {
        MessageBox.Show("SetPoint Over Load > 220 !!! Or No data transfer KP, KI, KD or setPoint");
    }
}
else

```

```
{
    MessageBox.Show("No Connection !!!");
}
}

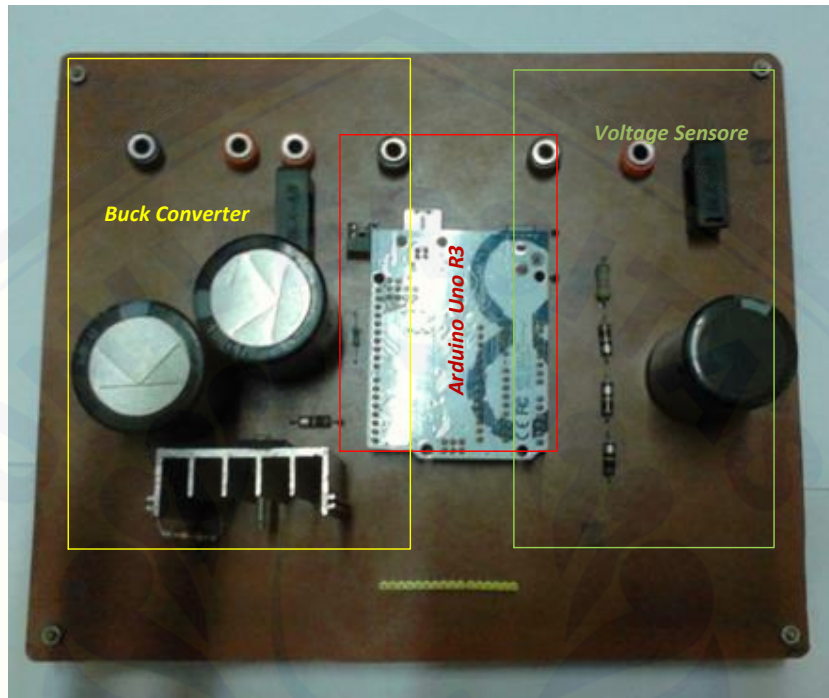
private void button4_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen)
    {
        if (button4.Text == "Mulai")
        {
            serialPort1.Write("1" + " "); //Mulai
            serialPort1.Write(textBox5.Text + " "); //KP
            serialPort1.Write(textBox4.Text + " "); //KI
            serialPort1.Write(textBox3.Text + " "); //KD
            serialPort1.Write(textBox2.Text + "\n"); //setPoint
            button4.Text = "Stop";
        }
        else if (button4.Text == "Stop")
        {
            serialPort1.Write("0" + " "); //Mulai
            serialPort1.Write(textBox5.Text + " "); //KP
            serialPort1.Write(textBox4.Text + " "); //KI
            serialPort1.Write(textBox3.Text + " "); //KD
            serialPort1.Write(textBox2.Text + "\n"); //setPoint
            button4.Text = "Mulai";
        }
    }
    else
    {
        MessageBox.Show("No Connection !!!");
    }
}

private void button5_Click(object sender, EventArgs e)
{
    richTextBox1.Text = "";
    textBox1.Text = "0";
}

private void axMathworks_Strip1_CursorChange(object sender,
AxMathworks_STRIPLib._DStripEvents_CursorChangeEvent e)
{
}
}
}
```

C. Gambar Hardware

C1. Buck Converter dan Voltage Sensor



C2. Generator DC Penguatan Terpisah



C3. Beban Lampu Pijar



D. Gambar Keseluruhan Sistem

