



UNIVERSITAS INDONESIA

**RANCANG BANGUN DC-DC BUCK CONVERTER DENGAN PID
DISKRIT SEBAGAI PENGENDALI TEGANGAN KELUARAN**

SKRIPSI

GUNAWAN

0405030397

FAKULTAS TEKNIK UNIVERSITAS INDONESIA

PROGRAM STUDI ELEKTRO

DEPOK

JULI 2009



UNIVERSITAS INDONESIA

**RANCANG BANGUN DC-DC BUCK CONVERTER DENGAN PID
DISKRIT SEBAGAI PENGENDALI TEGANGAN KELUARAN**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

GUNAWAN

0405030397

FAKULTAS TEKNIK UNIVERSITAS INDONESIA

PROGRAM STUDI ELEKTRO

DEPOK

JULI 2009

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama : Gunawan
NPM : 0405030397
Tanda Tangan :
Tanggal : 6 Juli 2009

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Gunawan
NPM : 0405030397
Program Studi : Teknik Elektro
Judul Skripsi : Rancang Bangun DC-DC Buck Converter dengan
PID Diskrit sebagai Pengendali Tegangan
Keluaran

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Dr. Ir Feri Yusivar, M.Eng ()

Penguji : Ir. Aries Subiantoro, M.SEE. ()

Penguji : Dr. Abdul Halim, MEng ()

Ditetapkan di : Depok

Tanggal : 6 Juli 2009

KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Program Studi Teknik Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

- (1) Dr. Ir Feri Yusivar, M.Eng., selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
- (2) pihak Departemen Elektro Fakultas Teknik Universitas Indonesia yang telah banyak membantu dalam usaha memperoleh data yang saya perlukan;
- (3) orang tua dan keluarga saya yang telah memberikan bantuan dukungan material dan moral; dan
- (4) sahabat-sahabat dari POFTUI dan TRUI yang telah banyak mendukung dan mendoakan;

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 6 Juli 2009

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai Civitas Akademik Universitas Indonesia, saya yang bertanda tangan dibawah ini :

Nama : Gunawan
NPM : 0405030397
Program Studi : Teknik Elektro
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis Karya : Skripsi

demikian pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul :

Rancang Bangun DC-DC Buck Converter dengan PID Diskrit sebagai Pengendali Tegangan Keluaran

berserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmediakan/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 6 Juli 2009

Yang menyatakan

(Gunawan)

ABSTRAK

Nama : Gunawan.
Program Studi : Teknik Elektro
Judul : Rancang Bangun DC-DC Buck Converter dengan PID Diskrit sebagai Pengendali Tegangan Keluaran

Buck converter merupakan salah satu jenis *switching converter* yang dapat menurunkan tegangan keluarannya. Agar *buck converter* dapat menghasilkan tegangan keluaran yang diinginkan diperlukan pengendali sistem *buck converter*. Pengendali yang dipakai pada sistem *buck converter* pada skripsi ini adalah pengendali *PID*. Pengendali *PID* yang digunakan ditala dengan menggunakan metode Ciancone. Pengendali diimplementasikan pada mikrokontroler ATmega16 dengan algoritma pengendali *PID* diskrit. Dari hasil simulasi dan uji coba alat didapatkan pengendali *PID* diskrit yang ditala dengan metode Ciancone ini memiliki respon yang cukup baik, dengan *transient response* yang cepat dan *steady state error* yang mendekati nol.

Kata kunci:

Buck converter, *PID* diskrit, metode Ciancone

ABSTRACT

Name : Gunawan
Study Program : Electrical Engineering
Title : Design and Development of DC-DC Buck Converter Using Discrete PID as Output Voltage Controller

Buck converter is one of switching converter that can lower its output voltage. Buck converter need to be controlled in order that to get an appropriate output voltage waveform. The controller used in this system is PID controller. This PID controller is tuned using Ciancone method. The controller is implemented in microcontroller Atmega16 with discrete PID algorithm. From the simulation result, can be concluded that the discrete PID tuned with Ciancone method provide a good response, with a fast transient response and nearly zero steady state error.

Keywords :

Buck converter, discrete PID, Ciancone method.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERNYATAAN ORISINALITAS	ii
LEMBAR PENGESAHAN	iii
KATA PENGANTAR.....	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS.....	v
ABSTRAK.....	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x
DAFTAR LAMPIRAN.....	xiii
1. PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Tujuan Pembahasan	5
1.3 Pembatasan Masalah.....	5
1.4 Sistematika Penulisan	5
2. DASAR TEORI.....	7
2.1 Rangkaian <i>Buck Converter</i>	7
2.2 Analisa Kondisi Tunak	8
2.3 Model Dinamik Sistem	11
2.4 Fungsi Alih Sistem	15
2.5 Penentuan LC filter.....	17
2.5.1 Penentuan Nilai Induktansi L.....	17
2.5.2 Penentuan Nilai Kapasitansi C	18
2.6 Skema Pengendalian Sistem <i>Buck Converter</i>	20
2.7 Pengendali <i>PID</i>	22
2.8 Penalaan <i>PID</i> dengan Metode Ciancone.....	26
2.9 Pengaruh Waktu Cuplik.....	29
3. PERANCANGAN SISTEM <i>BUCK CONVERTER</i> DAN PENGENDALI <i>PID</i> DISKRIT.....	31
3.1 Rangkaian <i>Buck Converter</i>	32
3.2 Mikrokontroler ATmega16	33
3.3 Rangkaian relay Pengatur Beban.....	33
3.4 Penentuan Nilai Frekuensi <i>Switching</i>	34
3.5 Fungsi Alih dan Matriks Ruang Keadaan Rata-Rata dari Sistem	35
3.6 Respon Sistem Tanpa Pengendali.....	36
3.7 Perancangan Pengendali <i>PID</i>	37
4. HASIL UJI COBA DAN ANALISA.....	42
4.1 Sistem Lingkar Terbuka.....	43
4.1.1 Sistem Lingkar Terbuka dengan Perubahan <i>Duty Ratio</i>	43
4.1.2 Sistem Lingkar Terbuka dengan Perubahan Beban.....	46
4.1.3 Sistem Lingkar Terbuka dengan Perubahan Tegangan Masukan ..	47
4.2 Analisa Sistem Lingkar Terbuka	48
4.3 Sistem Lingkar Tertutup	51

4.3.1 Sistem Lingkar Tertutup dengan Perubahan <i>Set point</i>	51
4.3.2 Sistem Lingkar Tertutup dengan Perubahan Beban	54
4.3.3 Sistem Lingkar Tertutup dengan Perubahan Tegangan Masukan.	56
4.4 Analisa Sistem Lingkar Tertutup.....	57

5. KESIMPULAN	61
DAFTAR ACUAN	62
DAFTAR PUSTAKA.....	63
LAMPIRAN A	64
LAMPIRAN B	70

DAFTAR GAMBAR

Gambar 1.1	... <i>Linear voltage regulator</i>	1
Gambar 1.2	(a) <i>Switching converter</i> sederhana. (b) Tegangan keluaran.....	3
Gambar 1.3	<i>Switching converter</i> dengan <i>low pass filter</i>	4
Gambar 1.4	Diagram pengendalian <i>buck converter</i>	4
Gambar 2.1	Rangkaian <i>buck converter</i>	7
Gambar 2.2	(a) Rangkaian ekivalen saat <i>switch</i> tertutup.....	9
	(b) Rangkaian ekivalen saat <i>switch</i> terbuka	9
	(c) Tegangan induktor pada satu perioda <i>switching</i>	9
Gambar 2.3	Gain tegangan $M(D)$ terhadap duty ratio (D).....	10
Gambar 2.4	<i>Buck converter</i> dengan MOSFET sebagai <i>switch</i>	11
Gambar 2.5.	<i>Buck converter</i> non-ideal dalam kondisi MOSFET Q1	
	(a) tertutup dan (b) terbuka.	12
Gambar 2.6	Arus induktor.....	17
Gambar 2.7	Tegangan <i>ripple</i>	19
Gambar 2.8	Arus Kapasitor.....	20
Gambar 2.9	Skema pengendalian sistem <i>buck converter</i>	20
Gambar 2.10	Pulse Width Modulator	21
Gambar 2.11	Skema pengendali <i>PID</i>	22
Gambar 2.12.	(a) Pengendali <i>PID</i> analog.	24
	(b) Fungsi transfer dan hubungannya dengan besar komponen rangkaiannya.	24
Gambar 2.13	Metoda pencuplikan. (a) <i>Backward Rectangular Method</i> (b) <i>Forward Rectangular Method</i>	25
Gambar 2.14	Metode satu menentukan model <i>first order with dead time</i>	27
Gambar 2.15	Metode dua menentukan model <i>first order with dead time</i>	28
Gambar 2.16	Grafik Ciancone.....	29
Gambar 3.1	Sistem <i>buck converter</i> , beban, dan mikrokontroler sebagai pengendalinya.....	31
Gambar 3.2	Rangkaian <i>buck converter</i>	32
Gambar 3.3	Rangkaian relay Pengatur Beban.....	33

Gambar 3.4	Skema pengendalian diskrit	36
Gambar 3.5	Bode plot dari sistem open loop	37
Gambar 3.6	Respon sistem terhadap masukan <i>step</i>	37
Gambar 3.7	Sistem <i>first order with dead time</i> pada simulink.....	38
Gambar 3.8	<i>Step</i> respon model <i>first order with dead time</i>	39
Gambar 3.9	Bode sistem lingkaran tertutup	41
Gambar 4.1	Blok-blok simulasi sistem <i>buck converter</i> dengan perubahan <i>duty ratio</i>	43
Gambar 4.2	Respon tegangan terhadap perubahan <i>duty ratio</i> dari 0.25 ke 0.5 pada simulasi	44
Gambar 4.3	Respon tegangan terhadap perubahan <i>duty ratio</i> dari 0.25 ke 0.75 pada simulasi	45
Gambar 4.4	Respon tegangan terhadap perubahan <i>duty ratio</i> dari 0.25 ke 0.5 pada alat	45
Gambar 4.5	Respon tegangan terhadap perubahan <i>duty ratio</i> dari 0.25 ke 0.75 pada alat	46
Gambar 4.6	Blok-blok simulasi sistem <i>buck converter</i> dengan perubahan beban.....	46
Gambar 4.7	Respon tegangan terhadap perubahan beban dari 1.5 Ω ke 0.7 Ω ..	47
Gambar 4.8	Blok-blok simulasi sistem <i>buck converter</i> dengan perubahan tegangan masukan.....	48
Gambar 4.9	Respon tegangan terhadap perubahan tegangan masukan dari 13 volt ke 11 volt.....	48
Gambar 4.10	Pengaruh hambatan terhadap redaman sistem	50
Gambar 4.11	Pengaruh penambahan suhu terhadap nilai ESR.....	51
Gambar 4.12	Blok-blok simulasi sistem <i>buck converter</i> dengan perubahan <i>set point</i>	52
Gambar 4.13	Respon tegangan terhadap perubahan <i>set point</i> dari 5 volt ke 10 volt pada simulasi	53
Gambar 4.14	<i>Ripple</i> tegangan pada <i>set point</i> 10 volt	53
Gambar 4.15	Respon tegangan terhadap perubahan <i>set point</i> dari 5 volt ke 10 volt pada alat	54

Gambar 4.16	Blok-blok simulasi sistem buck converter dengan perubahan beban.....	54
Gambar 4.17	Respon tegangan terhadap perubahan beban dari 1.5 Ω ke 0.7 Ω pada simulasi	55
Gambar 4.18	Respon tegangan terhadap perubahan beban dari 1.5 Ω ke 0.7 Ω pada alat	56
Gambar 4.19	Blok-blok simulasi sistem buck converter dengan perubahan tegangan masukan.....	56
Gambar 4.20	Respon tegangan terhadap perubahan tegangan masukan dari 14 volt ke 10 volt pada simulasi.....	57
Gambar 4.21	Respon tegangan keluaran jika T_i diperbesar 10 kali pada simulasi	58
Gambar 4.22	Sinyal tegangan acuan dari mikrokontroler pada percobaan perubahan beban pada alat	59

DAFTAR LAMPIRAN

Lampiran A	Program yang digunakan dalam blok sistem	65
Lampiran B	Program yang digunakan dalam mikrokontroler.....	71



BAB 1

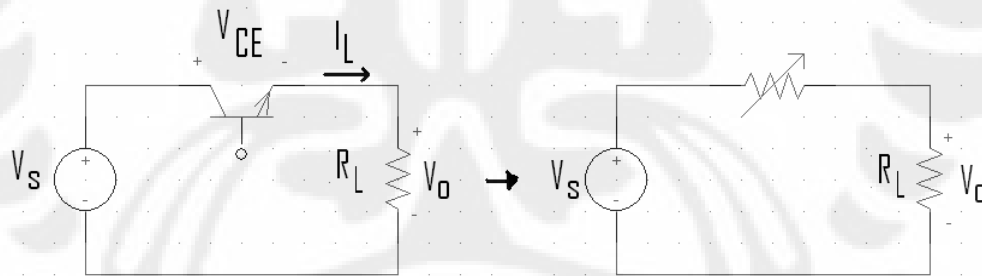
PENDAHULUAN

1.1 Latar Belakang

Saat ini hampir seluruh komponen elektronik memerlukan catu daya DC. Kebutuhan catu daya DC ini mulai dari skala tegangan rendah seperti yang digunakan pada mikroprosesor dan IC, tegangan menengah seperti pada motor-motor listrik dan generator, sampai pada skala tegangan tinggi untuk transmisi listrik tegangan tinggi (High Voltage DC).

Karena penggunaan catu daya DC yang luas ini, diperlukan suatu sistem yang dapat mengkonversikan tegangan DC dari suatu tingkat tegangan ke tingkat tegangan lain, seperti halnya transformator dalam sistem AC. Sistem ini harus dapat bekerja secara efisien dan keluaran tegangannya pun harus memiliki kualitas yang baik.

Salah satu cara untuk mengkonversikan tegangan DC ke tegangan DC yang lebih rendah adalah dengan menggunakan *linear voltage regulator* [1]. Rangkaian sederhana *linear voltage regulator* ini ditunjukkan oleh gambar 1.1.



Gambar 1.1 *Linear voltage regulator*

Tegangan keluarannya adalah

$$V_o = I_L R_L \quad (1.1)$$

dimana arus beban diatur oleh transistor. Dengan mengatur arus basis transistor, tegangan output dapat diatur dari 0 sampai dengan V_S . Ketika terjadi perubahan tegangan atau beban, arus basis dikompensasi untuk menghasilkan keluaran tegangan yang diinginkan. Rangkaian semacam ini dinamakan *linear voltage*

regulator karena transistor yang digunakan bekerja pada daerah linear, bukan pada daerah *cut-off* atau *saturation*. Akibatnya transistor bekerja seperti resistor variabel.

Masalah pada regulator semacam ini adalah efisiensinya yang rendah. Jika arus basis yang diberikan kecil maka terdapat daya yang diserap transistor, yaitu $V_{CE}I_L$ selain daya yang diberikan kepada beban yaitu, $V_o I_L$. Rugi daya pada transistor inilah yang menyebabkan efisiensi rangkaian ini rendah. Misalkan jika tegangan keluaran adalah seperempat dari tegangan sumber, maka beban menyerap 25% dari daya sumber, tetapi transistor regulator menyerap 75% dari daya sumber.

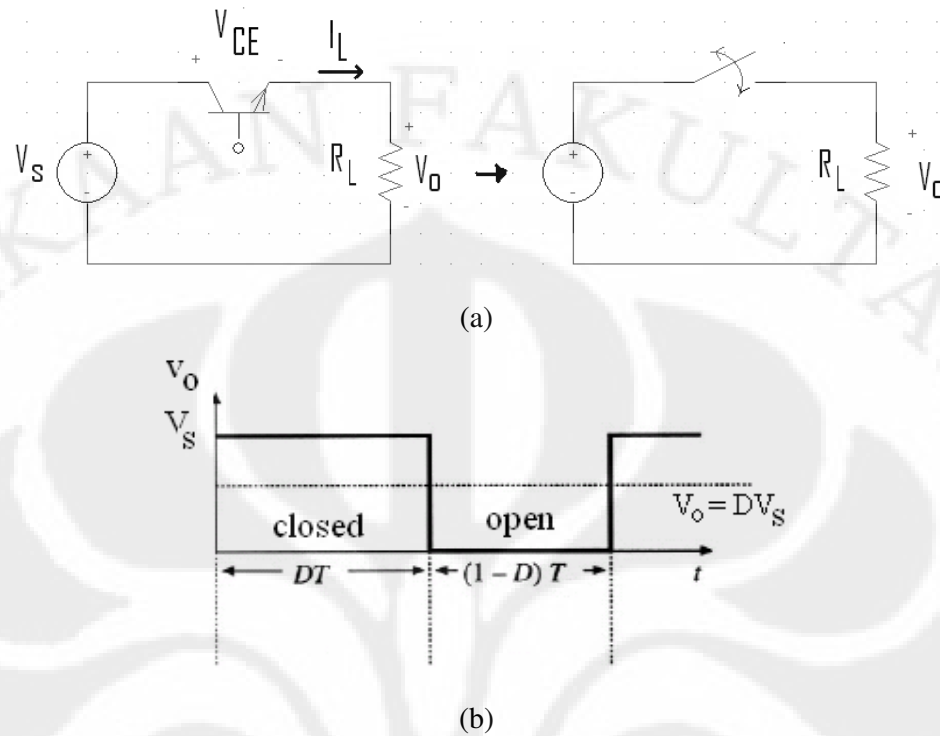
Masalah lain pada regulator jenis ini adalah dalam hal ukurannya yang tidak kecil karena memerlukan desain *heat-sink* yang baik. Daya yang diserap transistor dibuang dalam bentuk panas sehingga diperlukan medium untuk mempercepat pelepasan panas dari transistor, medium yang biasa digunakan adalah *heat-sink*. Masalah ukuran ini menjadi penting pada alat-alat elektronik kecil seperti *notebook*, *handphone*, dan alat-alat lain semacamnya.

Alternatif lain untuk mengubah tegangan DC ke tegangan DC lainnya adalah dengan menggunakan *switching converter*. Pada *switching converter*, transistor yang digunakan beroperasi sebagai *switch*, yaitu dengan sepenuhnya *on* atau sepenuhnya *off*. Rangkaian sederhana *switching converter* ditunjukkan oleh gambar 1.2. Jika *switch* diasumsikan ideal, maka tegangan masukan akan sama dengan tegangan keluaran ketika *switch* tertutup dan ketika *switch* terbuka, maka tegangan keluaran sama dengan nol. Pembukaan dan penutupan *switch* secara periodik akan menghasilkan tegangan keluaran berbentuk pulsa seperti gambar 1.2b. Tegangan rata-rata yang dihasilkan adalah

$$V_o = \frac{1}{T} \int_0^T v_o(t) dt = \frac{1}{T} \int_0^{DT} V_s dt = V_s D. \quad (1.2)$$

Jadi tegangan keluaran diatur dengan mengubah *duty ratio* D , yang adalah lamanya waktu *switch* tertutup terhadap periode *switching*:

$$D \equiv \frac{t_{on}}{t_{on} + t_{off}} = \frac{t_{on}}{T} = t_{on} f \quad (1.3)$$



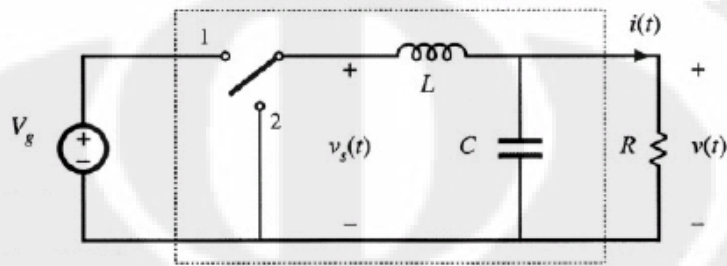
Gambar 1.2 (a) *Switching converter* sederhana. (b) Tegangan keluaran

Dimana f adalah frekuensi *switching* dalam Hertz. Sehingga tegangan keluaran dapat diatur lebih rendah atau sama dengan tegangan masukan.

Switching converter memiliki kelebihan berupa efisiensinya yang lebih tinggi dan ukurannya yang dapat jauh lebih kecil daripada *linear voltage regulator*. Berbeda dengan *linear voltage regulator*, pada *switching converter* ini efisiensi dapat mencapai 100% jika switch dianggap ideal. Ketika *switch* dalam keadaan tertutup, maka tegangan yang melintasi *switch* bernilai nol, dan ketika *switch* dalam keadaan terbuka, maka tidak ada arus yang melewati *switch*. Oleh karena itu, pada *switch* yang ideal tidak ada daya yang diserap *switch* sehingga efisiensi dapat mencapai 100%. Perkembangan teknologi fabrikasi membuat *switching device* beserta pengendali analog yang digunakan dapat dibuat menjadi satu *IC* yang ringkas sehingga meminimalkan ukuran dari *switching converter*. Dengan demikian *switching converter* dapat menjadi pilihan yang jauh lebih baik sebagai sistem pengkonversian catu daya DC daripada *linear voltage regulator*.

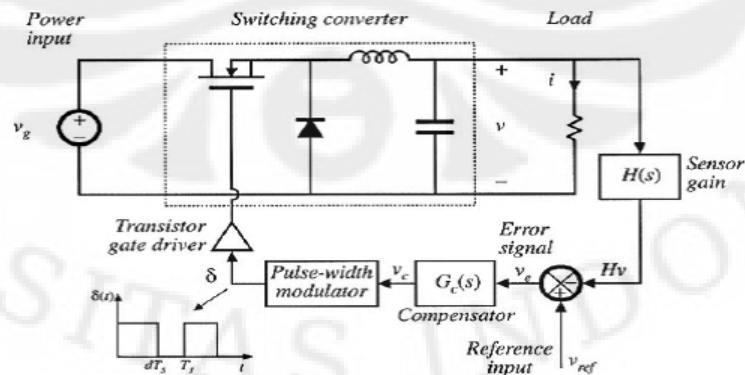
Karena pengaruh frekuensi *switching*, tegangan keluaran dari *switching converter* mengandung harmonik yang tidak diinginkan [2]. Untuk itu, dalam

penggunaan *switching converter* ini perlu ditambahkan *low pass filter* yang memiliki frekuensi *cut-off* jauh lebih kecil daripada frekuensi *switching* sehingga hanya komponen DC saja yang menjadi keluaran dari *switching converter*. *Low pass filter* yang digunakan biasanya berupa komponen pasif L-C. Pemakaian L-C *low pass filter* ini ditunjukkan oleh gambar 1.3.



Gambar 1.3 *Switching converter* dengan *low pass filter*.

Dalam aplikasi sesungguhnya diperlukan sistem kendali untuk mengkompensasi tegangan keluaran dari *switching converter*. Gambar 1.4 menunjukkan salah satu contoh cara pengaturan dari *buck converter*, salah satu jenis *switching converter*. *Buck converter* ini memiliki tegangan keluaran yang diinginkan yang akan dibandingkan dengan tegangan keluaran dari sensor. Perbedaan yang didapatkan dari kedua tegangan ini adalah *error* tegangan yang akan dikompensasi oleh kompensator untuk mengatur penyalan *switch* dengan menggunakan *Pulse-Width Modulator* (PWM). Ada banyak sistem kontrol yang telah dikembangkan untuk mengendalikan tegangan keluaran *switching converter*, tetapi pada skripsi ini hanya akan dibahas salah satu tipe pengendali yaitu dengan menggunakan pengendali *PID*.



Gambar 1.4 Diagram pengendalian *buck converter*

Pengendali *PID* merupakan pengendali dengan parameter-parameter kendali *proportional*, *integral*, dan *derivative*. Dengan kombinasi ketiga parameter ini yang tepat, keluaran akan memiliki nilai *steady state error* yang mendekati nol dan *transient response* yang cepat.

Ada bermacam-macam *switching converter* yang memiliki kegunaannya masing-masing. Beberapa nama konverter ini antara lain *buck converter*, *boost converter*, *buck-boost converter*, *Cuk converter*, *SEPIC converter*, dan *ZETA converter* [3]. Konverter-konverter ini memiliki rangkaian masing-masing dan kegunaannya pun berbeda-beda, ada yang dapat menurunkan tegangan, ada yang dapat menaikkan tegangan, dan adapula yang dapat menaikkan dan menurunkan tegangan.

Dalam skripsi ini akan dibahas salah satu jenis *switching converter*, yaitu *buck converter* dan pengendaliannya. *Buck converter* ini dapat menurunkan tegangan masukan menjadi tegangan keluaran dengan nilai yang lebih rendah. Untuk mengatur tegangan keluaran sistem digunakan pengendali *PID*.

1.2 Tujuan Pembahasan

Tujuan skripsi ini adalah untuk merancang sistem *buck converter* dan mengamati unjuk kerja pengendali *PID* diskrit dari metode Ciancone yang diterapkan pada sistem *buck converter* yang dirancang.

1.3 Pembatasan Masalah

Skripsi ini membahas model *buck converter*, perancangan rangkaian *buck converter* dan pengendali *PID* yang diterapkan untuk mengendalikan tegangan keluaran dari sistem *buck converter* yang dirancang. Pengendali *PID* yang digunakan ditala dengan metode Ciancone untuk dilihat bagaimana respon tegangan keluaran yang didapatkan.

1.4 Sistematika Penulisan

Sistematika penulisan seminar ini dibagi menjadi beberapa bab. Bab satu merupakan bab pendahuluan yang berisi latar belakang, tujuan penulisan, dan sistematika penulisan. Bab dua berisi pembahasan *buck converter*, pengendali

PID, dan penalaan pengendali *PID* dengan metode Ciancone. Bab tiga berisi perancangan sistem *buck converter* dan pengendali *PID* yang digunakan. Bab empat berisi hasil dan analisa unjuk kerja sistem dan pengendali dari hasil simulasi dan alat percobaan. Bab lima merupakan kesimpulan dari skripsi yang dibuat ini.



BAB 2

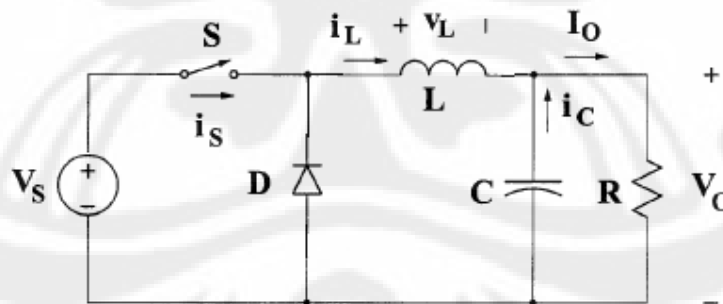
BUCK CONVERTER DAN PENGENDALIANNYA

Buck converter merupakan salah satu jenis *switching converter* yang dapat menurunkan tegangan keluarannya. Tegangan keluarannya diatur dengan cara mengatur *duty ratio* yang diberikan kepada *switching device*.

Pada bab ini, pertama-tama akan dibahas mengenai skema rangkaian *buck converter* dan bagian-bagiannya. Karakteristik operasi kondisi tunak *buck converter* akan diturunkan dari modelnya yang ideal. Kemudian karakteristik dinamik dari sistem *buck converter* akan diturunkan dengan menggunakan metode *state space averaging*. Dari bentuk dinamik sistem juga akan diturunkan fungsi alih sistem. Penentuan LC filter yang digunakan akan dibahas kemudian. Selanjutnya skema pengendalian dari sistem *buck converter* akan dibahas. Akhirnya pengendali *PID*, bentuk pengendali *PID* diskrit yang digunakan pada sistem, penalaannya dengan metode Ciancone, dan pengaruh waktu cuplik terhadap pengendali *PID* diskrit perlu diketahui dan akan dibahas dalam bab ini.

2.1 Rangkaian Buck Converter

Skema rangkaian *buck converter* adalah seperti tampak pada gambar 2.1.



Gambar 2.1 Rangkaian *buck converter*

Pada rangkaian ini terdapat beberapa komponen antara lain :

1. Tegangan masukan (V_s), merupakan catu daya DC yang ingin diubah tegangannya menjadi lebih tinggi atau lebih rendah.

2. *Controllable switch* (S), merupakan *semiconductor switching device* yang keadaan membuka dan menutupnya dapat dikendalikan dengan memberikan sinyal pada *gate switch*. Beberapa contoh *switch* jenis ini adalah BJT (*Bipolar Junction Transistor*), IGBT (*Insulated-gated Bipolar Transistor*) dan MOSFET (*Metal Oxide Semiconductor Field Effect Transistor*).
3. Induktor (L), merupakan komponen yang berfungsi untuk menyimpan arus.
4. Dioda (D), merupakan *uncontrollable semiconductor switching device*, karena membuka atau menutupnya dioda tergantung dari tegangan dan arus yang mengalir pada rangkaian. Dioda akan terbuka jika dalam keadaan *forward bias*, yaitu tegangan anoda lebih tinggi daripada tegangan katoda, sebaliknya diode akan menutup jika dalam keadaan *reverse bias*, yaitu tegangan anoda lebih rendah daripada tegangan katoda.
5. Kapasitor (C), merupakan komponen yang berfungsi sebagai *low pass filter*.
6. Resistor (R), merupakan beban sistem dan berfungsi pula sebagai bagian dari *low pass filter*.
7. Tegangan keluaran (V_o), merupakan hasil tegangan keluaran yang diinginkan.

2.2. Analisa Kondisi Tunak

Analisa kondisi tunak bisa didapatkan dengan menganalisa rangkaian pada saat *switch* terbuka dan pada saat *switch* tertutup. Pada analisa ini, *switch* diasumsikan ideal.

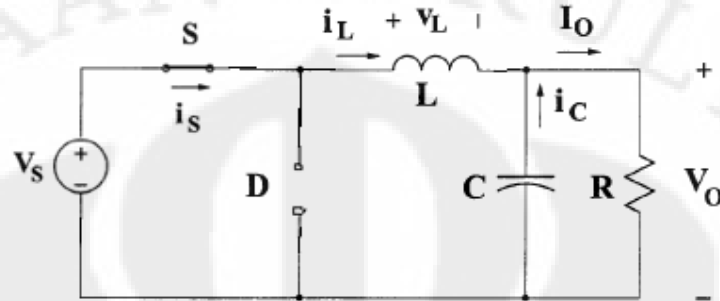
Pada saat *switch* tertutup, rangkaian ekivalen *buck converter* akan tampak seperti pada gambar 2.2a. Pada kondisi ini tegangan pada induktor adalah

$$v_L = V_s - V_o = L \frac{di_L}{dt} \quad (2.1)$$

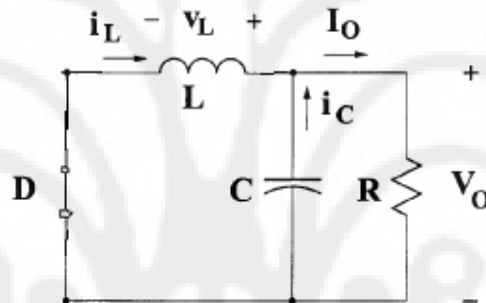
Pada saat *switch* tertutup, rangkaian ekivalen *buck converter* akan tampak seperti gambar 2.2b. Pada kondisi ini tegangan induktor adalah

$$v_L = -V_o = L \frac{di_L}{dt} \quad (2.2)$$

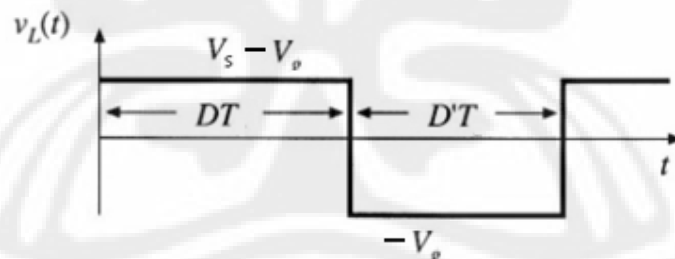
Hasil yang didapatkan adalah tegangan induktor dengan kondisi seperti ditunjukkan oleh gambar 2.2c, dimana DT menunjukkan waktu saat *switch* tertutup dan $D'T$ menunjukkan waktu saat *switch* terbuka.



(a)



(b)



(c)

Gambar 2.2 (a) Rangkaian ekivalen saat *switch* tertutup. (b) Rangkaian ekivalen saat *switch* terbuka (c) Tegangan induktor pada satu perioda *switching*

Kondisi tunak tercapai ketika arus yang dilepas induktor sudah sama dengan arus yang diserap induktor dalam satu perioda, sehingga tidak ada perubahan bersih arus induktor selama satu perioda [2]. Menurut persamaan induktor

$$v_L = L \frac{di_L}{dt}. \quad (2.3)$$

Maka integrasi dalam satu perioda dari $t = t_0$ sampai dengan $t = T$ didapatkan

$$i_L(T) - i_L(t_0) = \int_{t_0}^T v_L dt. \quad (2.4)$$

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

$$0 = \int_0^T v_L dt. \quad (2.5)$$

Persamaan 2.5 disebut *inductor volt-second balance* yang akan digunakan untuk menurunkan karakteristik tunak dari sistem *buck converter*.

Dari gambar 2.2c didapatkan nilai tegangan induktor selama satu perioda sehingga

$$\int_0^T v_L dt = (V_s - V_o)DT + -V_o(1 - D)T. \quad (2.6)$$

Dengan mensubstitusikan persamaan 2.5 ke persamaan 2.6 didapatkan

$$0 = (V_s - V_o)DT + -V_o(1 - D)T.$$

Dengan membagi kedua sisi dengan T didapatkan

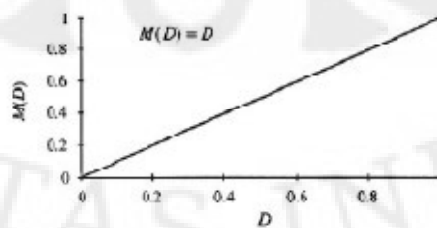
$$0 = (V_s - V_o)D + -V_o(1 - D).$$

Akhirnya didapatkan penyelesaian untuk V_o adalah

$$V_o = DV_s = M(D) V_s \quad (2.7)$$

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

Hasil penurunan analisa kondisi tunak dari persamaan 2.7 menunjukkan karakteristik sistem *buck converter*. Jika *gain* $M(D)$ diplot terhadap *duty ratio* D , maka akan didapatkan hasil seperti pada gambar 2.3. Pada gambar 2.3 terlihat bahwa nilai *gain* sistem dapat bernilai dari nol sampai dengan satu. Artinya sistem *buck converter* dapat mengubah tegangan keluarannya menjadi lebih kecil sampai dengan sama dengan tegangan masukan yang diberikan, sebanding dengan besarnya *duty ratio* yang diberikan.



Gambar 2.3 Gain tegangan $M(D)$ terhadap *duty ratio* (D)

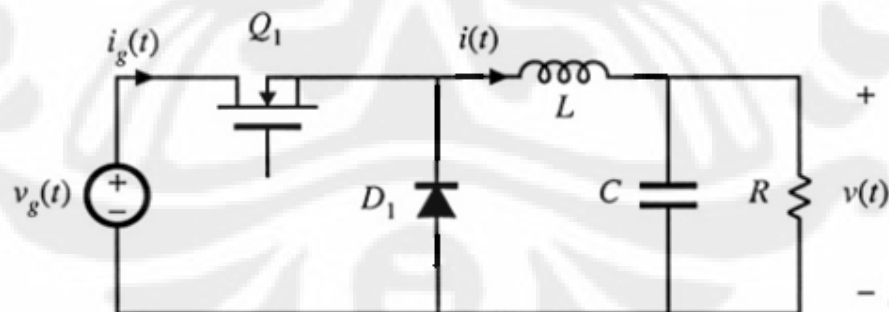
2.3 Model Dinamik Sistem

Model dinamik sistem dapat diturunkan dengan metode *state space averaging* [1,2]. Pada metode ini, persamaan diferensial dari sistem ditulis dalam bentuk matriks ruang keadaan. Variabel ruang keadaan dari sistem biasanya didapatkan dari elemen penyimpan muatan, dalam hal *switching converter* adalah arus induktor dan tegangan kapasitor.

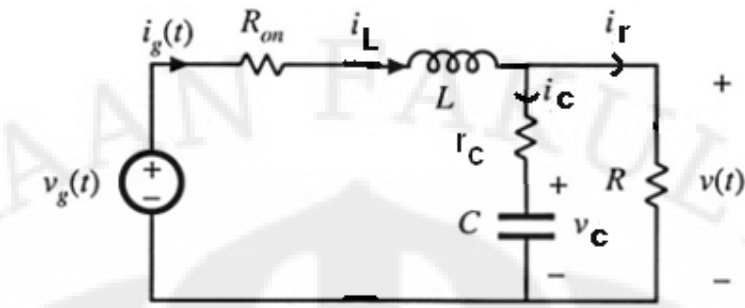
Model dinamik sistem *buck converter* ini akan diturunkan dari *switch* yang tidak ideal. Misalkan *switch* S digantikan dengan MOSFET Q1 seperti pada gambar 2.4. MOSFET yang dalam keadaan tertutup akan dimodelkan sebagai rugi konduksi, yaitu resistansi ketika *switch* tertutup R_{on} . Dioda yang dalam keadaan tertutup akan dimodelkan dengan tegangan jatuh, yaitu sumber tegangan bebas V_D . Sedangkan kapasitor akan memiliki rugi yang akan dimodelkan dengan *Equivalent Series Resistant* (ESR) yaitu r_c .

Seperti pada analisa kondisi tunak, pada analisa *state space averaging* ini juga dilihat keadaan sistem pada kondisi *switch* tertutup dan pada kondisi *switch* terbuka. Dari kedua kondisi ini akan didapatkan masing-masing matriks ruang keadaan dan matriks ruang keadaan total merupakan superposisi dari kedua matriks ruang keadaan ini.

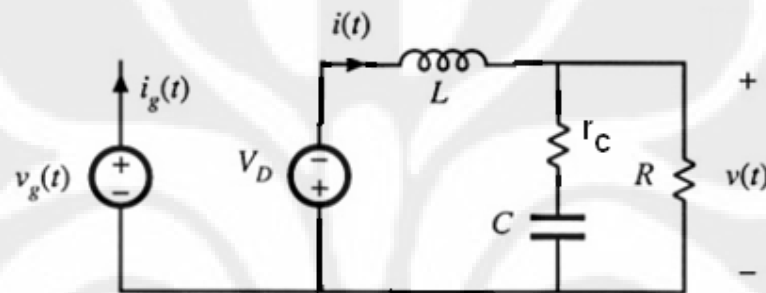
Dalam keadaan MOSFET tertutup, sistem pada gambar 2.4 akan menjadi seperti pada gambar 2.5a.



Gambar 2.4 *Buck converter* dengan MOSFET sebagai switch



(a)



(b)

Gambar 2.5. Buck converter non-ideal dalam kondisi MOSFET Q1
(a) tertutup dan (b) terbuka.

Pada gambar 2.5a didapatkan

$$-v_c - i_c r_c + i_r R = 0$$

$$-v_c - i_c r_c + (i_L - i_c)R = 0$$

$$-v_c + i_L R - i_c (r_c + R) = 0$$

$$i_c = C \frac{dv_c}{dt} = \frac{-v_c + i_L R}{R + r_c} \quad (2.8)$$

Maka

$$\frac{dv_c}{dt} = \frac{R}{C(R + r_c)} i_L + \frac{-1}{C(R + r_c)} v_c \quad (2.9)$$

Dan

$$V_g = i_L R_{on} + L \frac{di_L}{dt} + i_c r_c + v_c$$

$$L \frac{di_L}{dt} = V_g - i_L R_{on} - i_c r_c - v_c$$

Dengan mensubstitusikan i_c dari persamaan 2.8 didapatkan

$$\begin{aligned} L \frac{di_L}{dt} &= V_g - i_L R_{on} - \left(\frac{-v_c + i_L R}{R + r_c} \right) r_c - v_c \\ L \frac{di_L}{dt} &= V_g - \frac{(R + r_c) R_{on} + R r_c}{R + r_c} i_L + \frac{r_c - (R + r_c)}{R + r_c} v_c \\ \frac{di_L}{dt} &= \frac{-(R + r_c) R_{on} - R r_c}{L(R + r_c)} i_L + \frac{-R}{L(R + r_c)} v_c + \frac{V_g}{L} \end{aligned} \quad (2.10)$$

Bentuk persamaan ruang keadaan yang didapatkan dari 2.9 dan 2.10 adalah

$$\underbrace{\frac{d}{dt} \begin{bmatrix} i_L \\ v_c \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} \frac{-(R + r_c) R_{on} - R r_c}{L(R + r_c)} & \frac{-R}{L(R + r_c)} \\ \frac{R}{C(R + r_c)} & \frac{-1}{C(R + r_c)} \end{bmatrix}}_{A_1} \underbrace{\begin{bmatrix} i_L \\ v_c \end{bmatrix}}_x + \underbrace{\begin{bmatrix} \frac{1}{L} & 0 \\ 0 & 0 \end{bmatrix}}_{B_1} \underbrace{\begin{bmatrix} V_g \\ V_D \end{bmatrix}}_U. \quad (2.11)$$

Dalam keadaan MOSFET terbuka, sistem pada gambar 2.4 akan menjadi seperti pada gambar 2.5b. Dari gambar 2.5b terlihat bahwa sistem menyerupai gambar pada 2.5a hanya saja nilai V_g digantikan dengan $-V_D$ dan juga nilai R_{on} adalah nol. Dengan mengetahui kondisi ini, sistem pada gambar 2.5b adalah

$$\underbrace{\frac{d}{dt} \begin{bmatrix} i_L \\ v_c \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} \frac{-R r_c}{L(R + r_c)} & \frac{-R}{L(R + r_c)} \\ \frac{R}{C(R + r_c)} & \frac{-1}{C(R + r_c)} \end{bmatrix}}_{A_2} \underbrace{\begin{bmatrix} i_L \\ v_c \end{bmatrix}}_x + \underbrace{\begin{bmatrix} 0 & \frac{-1}{L} \\ 0 & 0 \end{bmatrix}}_{B_2} \underbrace{\begin{bmatrix} V_g \\ V_D \end{bmatrix}}_U \quad (2.12)$$

Tegangan keluaran v didapatkan dari

$$v = R i_R = R(i_L - i_c) = R \left(i_L - \frac{v - v_c}{r_c} \right)$$

sehingga didapatkan

$$\begin{aligned} v &= \left(\frac{R r_c}{R + r_c} \right) i_L + \left(\frac{R}{R + r_c} \right) v_c \\ \underbrace{v}_y &= \underbrace{\begin{bmatrix} \frac{R r_c}{R + r_c} & \frac{R}{R + r_c} \end{bmatrix}}_{C^r} \underbrace{\begin{bmatrix} i_L \\ v_c \end{bmatrix}}_x \end{aligned} \quad (2.13)$$

Persamaan tegangan keluaran yang didapatkan berlaku untuk kedua keadaan baik *switch* terbuka maupun *switch* tertutup.

Persamaan matriks ruang keadaan rata-rata (*state space averaged equation*) pada keadaan setimbang adalah superposisi dari matriks ruang keadaan dari kedua kondisi [2].

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= C^T x\end{aligned}$$

dimana

$$A = DA_1 + (1-D)A_2$$

$$= D \begin{bmatrix} \frac{-(R+r_c)R_{on} - Rr_c}{L(R+r_c)} & \frac{-R}{L(R+r_c)} \\ \frac{R}{C(R+r_c)} & \frac{-1}{C(R+r_c)} \end{bmatrix} + (1-D) \begin{bmatrix} \frac{-Rr_c}{L(R+r_c)} & \frac{-R}{L(R+r_c)} \\ \frac{R}{C(R+r_c)} & \frac{-1}{C(R+r_c)} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{-DR_{on}(R+r_c) - Rr_c}{L(R+r_c)} & \frac{-R}{L(R+r_c)} \\ \frac{R}{C(R+r_c)} & \frac{-1}{C(R+r_c)} \end{bmatrix}$$

$$B = DB_1 + (1-D)B_2$$

$$= D \begin{bmatrix} \frac{1}{L} & 0 \\ 0 & 0 \end{bmatrix} + (1-D) \begin{bmatrix} 0 & \frac{-1}{L} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \frac{D}{L} & \frac{D-1}{L} \\ 0 & 0 \end{bmatrix}$$

$$C = C_1^T = C_2^T = \begin{bmatrix} \frac{Rr_c}{R+r_c} & \frac{R}{R+r_c} \end{bmatrix}$$

Sehingga didapatkan persamaan matriks ruang keadaan rata-rata dari sistem *buck-boost converter* adalah

$$\frac{d}{dt} \begin{bmatrix} i \\ v \end{bmatrix} = \begin{bmatrix} \frac{-DR_{on}(R+r_c) - Rr_c}{L(R+r_c)} & \frac{-R}{L(R+r_c)} \\ \frac{R}{C(R+r_c)} & \frac{-1}{C(R+r_c)} \end{bmatrix} \begin{bmatrix} i \\ v \end{bmatrix} + \begin{bmatrix} \frac{D}{L} & \frac{D-1}{L} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_g \\ V_D \end{bmatrix}. \quad (2.14)$$

Dan persamaan keluarannya

$$v = \begin{bmatrix} \frac{Rr_c}{R+r_c} & \frac{R}{R+r_c} \end{bmatrix} \begin{bmatrix} i \\ v \end{bmatrix} \quad (2.15)$$

2.4 Fungsi Alih Sistem

Persamaan ruang keadaan memiliki bentuk

$$\dot{x} = Ax + Bu.$$

Jika persamaan ini diubah dalam bentuk *Laplace* akan menjadi

$$sX(s) = AX(s) + BU(s)$$

$$[sI - A]X(s) = BU(s)$$

$$X(s) = [sI - A]^{-1}BU(s)$$

Sedangkan persamaan keluarannya adalah

$$y = C^T x$$

Persamaan ini juga diubah dalam bentuk *Laplace* menjadi

$$Y(s) = C^T X(s)$$

Dengan mensubstitusikan persamaan yang didapat sebelumnya didapatkan

$$Y(s) = C^T [sI - A]^{-1} BU(s) \quad (2.16)$$

Pada sistem *buck converter* ini didapatkan

$$C^T = \begin{bmatrix} \frac{Rr_c}{R+r_c} & \frac{R}{R+r_c} \end{bmatrix} \quad (2.17)$$

$$[sI - A]^{-1} = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix}^{-1} \begin{bmatrix} \frac{-DR_{on}(R+r_c) - Rr_c}{L(R+r_c)} & \frac{-R}{L(R+r_c)} \\ \frac{R}{C(R+r_c)} & \frac{-1}{C(R+r_c)} \end{bmatrix}^{-1}$$

Untuk mempermudah perhitungan, maka nilai $R+r_c$ dapat diganti dengan nilai resistansi seri R_s sehingga didapatkan

$$[sI - A]^{-1} = \begin{bmatrix} \frac{sLR_s + DR_{on}R_s + Rr_c}{LR_s} & \frac{R}{LR_s} \\ \frac{-R}{CR_s} & \frac{sCR_s + 1}{CR_s} \end{bmatrix}^{-1}$$

$$\begin{aligned}
&= \frac{1}{\left(\frac{sLR_s + DR_{on}R_s + Rr_c}{LR_s} \right) \left(\frac{sCR_s + 1}{CR_s} \right) + \frac{R^2}{LCR_s^2}} \begin{bmatrix} \frac{sCR_s + 1}{CR_s} & \frac{-R}{LR_s} \\ \frac{R}{CR_s} & \frac{sLR_s + DR_{on}R_s + Rr_c}{LR_s} \end{bmatrix} \\
&= \frac{LCR_s^2}{s^2LCR_s^2 + sR_s(L + DCR_{on}R_s + CRr_c) + DR_{on}R_s + Rr_c + R^2} \begin{bmatrix} \frac{sCR_s + 1}{CR_s} & \frac{-R}{LR_s} \\ \frac{R}{CR_s} & \frac{sLR_s + DR_{on}R_s + Rr_c}{LR_s} \end{bmatrix} \\
&= \frac{R_s}{P(s)} \begin{bmatrix} SLCR_s + L & -CR \\ RL & SLCR_s + DCR_{on}R_s + CRr_c \end{bmatrix} \quad (2.18)
\end{aligned}$$

Dimana

$$\begin{aligned}
P(s) &= s^2LCR_s^2 + sR_s(L + DCR_{on}R_s + CRr_c) + DR_{on}R_s + Rr_c + R^2 \\
&= LCR_s^2 \left[s^2 + s \left(\frac{1}{CR_s} + \frac{DR_{on}}{L} + \frac{Rr_c}{LR_s} \right) + \frac{DR_{on} + R}{LCR_s} \right] \quad (2.19)
\end{aligned}$$

Untuk menyederhanakan bentuk fungsi alih nilai v_D diabaikan sehingga didapatkan juga

$$BU(s) = \begin{bmatrix} \frac{D}{L} \\ 0 \end{bmatrix} V_g \quad (2.20)$$

Dengan mensubstitusikan persamaan 2.17, 2.18, dan 2.20 ke persamaan 2.16 didapatkan

$$\begin{aligned}
Y(s) &= \begin{bmatrix} \frac{Rr_c}{R_s} & \frac{R}{R_s} \end{bmatrix} \frac{R_s}{P(s)} \begin{bmatrix} SLCR_s + L & -CR \\ RL & SLCR_s + DCR_{on}R_s + CRr_c \end{bmatrix} \begin{bmatrix} \frac{D}{L} \\ 0 \end{bmatrix} V_g \\
&= \frac{1}{P(s)} \begin{bmatrix} Rr_c & R \end{bmatrix} \begin{bmatrix} SLCR_s + L & -CR \\ RL & SLCR_s + DCR_{on}R_s + CRr_c \end{bmatrix} \begin{bmatrix} \frac{D}{L} \\ 0 \end{bmatrix} V_g \\
&= \frac{V_g}{P(s)} \begin{bmatrix} Rr_c & R \end{bmatrix} \begin{bmatrix} sDCR_s + D \\ DR \end{bmatrix} \\
&= \frac{V_g}{P(s)} \left[sDCR_s Rr_c + DRr_c + DR^2 \right] \\
&= \frac{V_g DRR_s}{P(s)} \left[sCr_c + 1 \right]
\end{aligned}$$

Dengan mensubstitusikan nilai $P(s)$ pada persamaan 2.19 didapatkan

$$Y(s) = \frac{V_g R D}{L C R_s} \frac{s C r_c + 1}{\left[s^2 + s \left(\frac{1}{C R_s} + \frac{D R_{on}}{L} + \frac{R r_c}{L R_s} \right) + \frac{D R_{on} + R}{L C R_s} \right]}$$

Fungsi alih $\frac{Y(s)}{D(s)}$ bisa didapatkan dengan mengabaikan nilai R_{on} yang dalam

kenyataannya biasanya bernilai kecil. Sehingga didapatkan

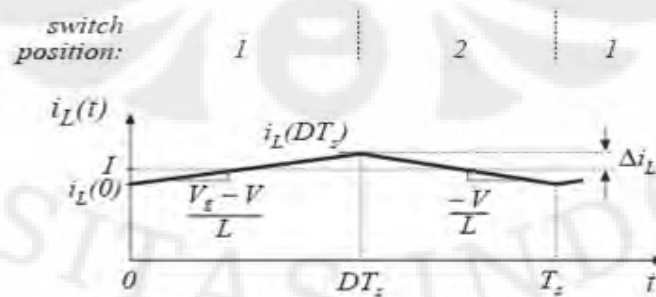
$$\frac{Y(s)}{D(s)} = \frac{V_g R}{L C R_s} \frac{s C r_c + 1}{\left[s^2 + s \left(\frac{1}{C R_s} + \frac{R r_c}{L R_s} \right) + \frac{R}{L C R_s} \right]}, \text{ atau}$$

$$\frac{Y(s)}{D(s)} = \frac{V_g R}{L C (R + r_c)} \frac{s C r_c + 1}{\left[s^2 + s \left(\frac{1}{C (R + r_c)} + \frac{R r_c}{L (R + r_c)} \right) + \frac{R}{L C (R + r_c)} \right]} \quad (2.21)$$

2.5 Penentuan LC filter

2.5.1 Penentuan Nilai Induktansi L

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 [1]. Gambar 2.6 menunjukkan bahwa arus induktor berada pada sekitar arus rata-ratanya I dengan nilai arus maksimum sebesar $I + \Delta I_L$ dan arus minimum $I - \Delta I_L$.



Gambar 2.6 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.

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

$$\Delta I_L = \frac{V}{2L} T_s(1-D) \quad (2.22)$$

Nilai $I - \Delta I_L$ harus lebih besar daripada nol agar sistem bekerja pada daerah kontinu sehingga

$$I - \Delta I_L > 0$$

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

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

Maka didapatkan syarat L agar arus sistem kontinu adalah

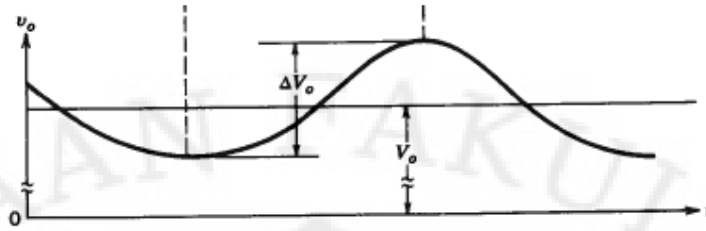
$$L > \frac{T_s(1-D)R}{2} \text{ atau}$$

$$L > \frac{(1-D)R}{2f} \quad (2.23)$$

$$\text{Untuk } f = \frac{1}{T_s}$$

2.5.2 Penentuan Nilai Kapasitansi C

Tegangan keluaran sistem pada kenyataannya tidak dapat bernilai konstan secara sempurna. Hal ini dikarenakan kapasitor yang digunakan akan terus melakukan pengisian dan pelepasan muatan. Pada tegangan keluaran sistem V_o akan terdapat *ripple* ΔV_o seperti ditunjukkan oleh gambar 2.7



Gambar 2.7 Tegangan ripple

Gambar 2.8 menunjukkan karakteristik arus dari kapasitor. Arus yang melewati kapasitor adalah

$$i_C = i_L - i_R$$

Saat arus kapasitor bernilai positif, kapasitor sedang mengisi. Dari definisi kapasitansi didapatkan

$$Q = CV$$

$$\Delta Q = C\Delta V$$

$$\Delta V = \frac{\Delta Q}{C}$$

Perubahan pada muatan, ΔQ adalah daerah segitiga di atas sumbu waktu

$$\Delta Q = \frac{1}{2} \left(\frac{T_s}{2} \right) \Delta I_L = \frac{T_s \Delta i_L}{4}$$

Sehingga didapatkan

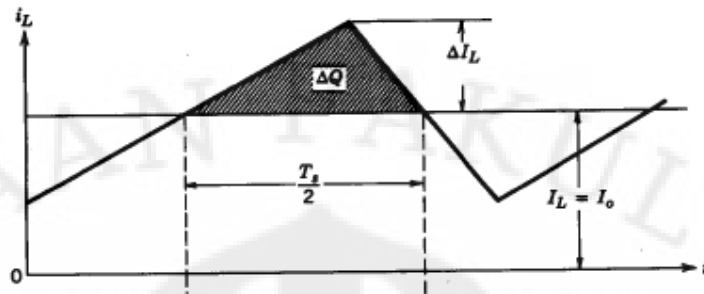
$$\Delta V = \frac{T_s \Delta i_L}{4C}$$

Dengan mensubstitusikan ΔI_L pada persamaan 2.22 didapatkan

$$\Delta V = \frac{T_s V}{8CL} (1-D) T = \frac{V(1-D)}{8LCf^2}$$

Besarnya ripple tegangan dapat dinyatakan dalam bentuk

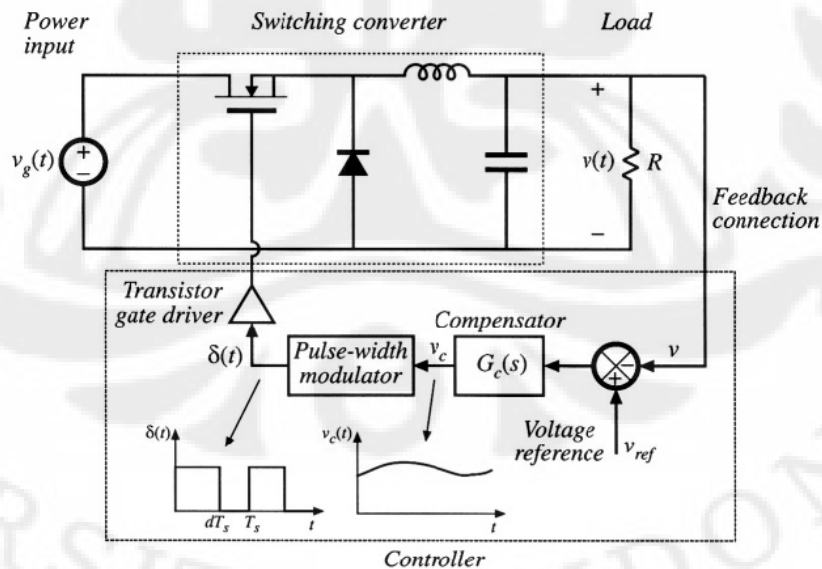
$$\frac{\Delta V}{V} = \frac{(1-D)}{8LCf^2} \quad (2.24)$$



Gambar 2.8 Arus Kapasitor

2.6 Skema Pengendalian Sistem *Buck Converter*

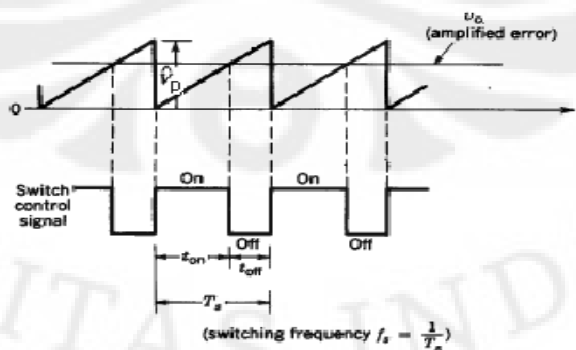
Skema pengendalian sistem *buck converter* secara keseluruhan ditunjukkan oleh gambar 2.9. Pada gambar 2.9 ditunjukkan bahwa terdapat dua bagian penting dari sistem *buck converter*. Bagian yang pertama adalah bagian *switching converter* sebagai rangkaian elektronika utama dan bagian yang kedua adalah bagian *controller* yang berguna mengatur tegangan keluaran. Bagian pertama yaitu *switching converter* telah dibahas sebelumnya dalam bentuk model matematis. Bagian *controller* ini dan perantara antara sistem *switching converter* dengan *controller* belum dibahas.

Gambar 2.9 Skema pengendalian sistem *buck converter*

Terlihat pada gambar 2.9 diperlukan perantara dari hubungan *switching converter* ke *controller* dan dari *controller* ke *switching converter*. Terlihat pada gambar 2.9 bahwa dari *switching converter* ke *controller* diperlukan *feedback connection*. Sedangkan dari *controller* ke *switching converter* dihubungkan dengan *pulse-width modulator* ditambah dengan *transistor gate driver*.

Feedback connection menyediakan tegangan aktual dari keluaran yang berarti adalah tegangan yang sesungguhnya terjadi dari keluaran sistem (v). Tegangan ini dibutuhkan untuk mengetahui apakah tegangan keluaran sistem sudah sama dengan tegangan referensi yang diinginkan (v_{ref}). Apabila $v \neq v_{ref}$ maka akan terdapat error yaitu $e = v_{ref} - v$. Error inilah yang akan ditangani oleh kompensator. Pada umumnya kompensator *switching converter* yang digunakan bersifat analog [4], sehingga tidak dibutuhkan rangkaian tambahan untuk *feedback connection* ini. Tetapi pada penggunaan kompensator yang bersifat digital, yaitu pengendali yang menggunakan mikroprosesor atau *DSP (Digital Signal Processing)*, biasanya selalu terdapat *conditioning circuit* untuk mengkonversikan tingkat tegangan yang berbeda [3].

Pulse width modulator (PWM) mengkonversi keluaran dari kompensator ke *duty ratio*. Tegangan keluaran dari kompensator v_c dibandingkan dengan gelombang segitiga dengan amplitudo V_p seperti ditunjukkan oleh gambar 2.10. Keluaran dari blok *PWM* bernilai tinggi ketika v_c lebih besar daripada gelombang segitiga dan bernilai nol ketika v_c lebih kecil daripada gelombang segitiga. Jika besar tegangan keluaran v lebih kecil daripada besar tegangan keluaran v_{ref} , nilai error meningkat sehingga nilai v_c meningkat dan *duty ratio* meningkat. Sebaliknya, jika besar tegangan keluaran v meningkat akan mengurangi *duty*



Gambar 2.10 Pulse Width Modulator

ratio. Fungsi transfer dari *PWM* adalah

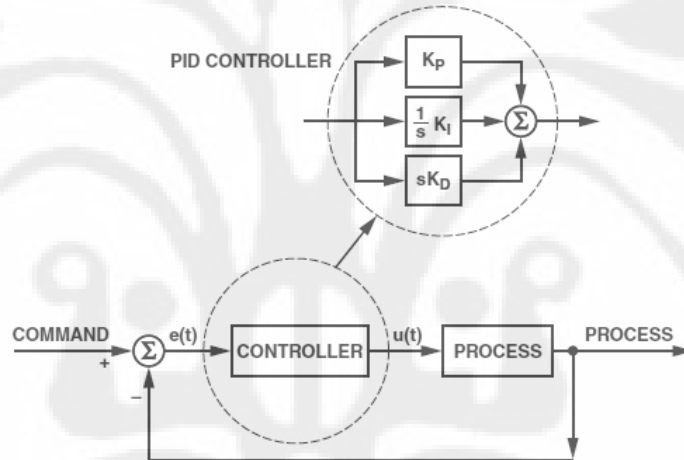
$$d = \frac{V_c}{V_p} \quad (2.25)$$

sehingga

$$\frac{d(s)}{V_c(s)} = \frac{1}{V_p(s)}. \quad (2.26)$$

Pulsa *PWM* ini kemudian akan digunakan untuk membuka dan menutup *switch* pada *switching converter*.

2.7 Pengendali *PID*



Gambar 2.11 Skema pengendali *PID*

Dalam sistem *buck converter* diperlukan kompensator untuk mengatur tegangan keluaran agar memiliki nilai yang sama dengan tegangan keluaran yang diinginkan. Pada skripsi ini kompensator yang dipakai adalah kompensator *PID*.

Pengendali *PID* telah secara luas digunakan dalam industri, yaitu hampir sebesar 90% [5]. Hal ini dikarenakan jika pengendali ini diatur dengan baik, pengendali *PID* dapat menunjukkan hasil kendali yang sangat baik. Selain itu pengendali ini juga sederhana dan dapat diimplementasikan dengan mudah.

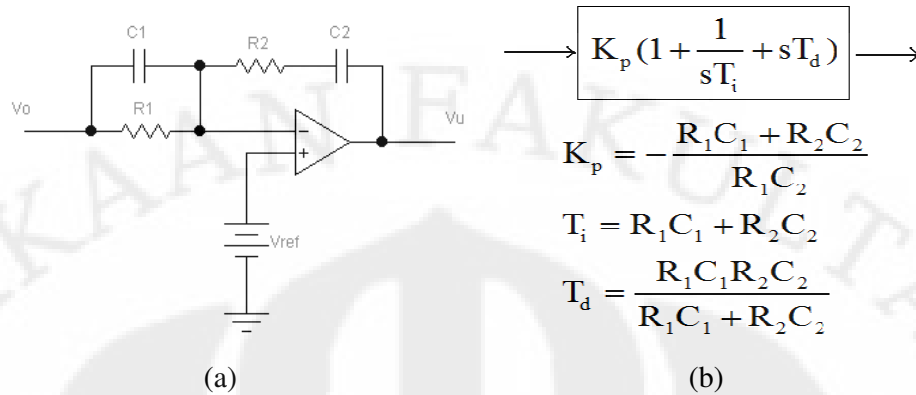
Skema pengendali *PID* ditunjukkan oleh gambar 2.11. Pada gambar 2.11 menunjukkan bahwa pengendali *PID* merupakan jumlah dari tiga aksi kendali,

yaitu *proportional* yang ditunjukkan oleh K_P , *integral* yang ditunjukkan oleh $\frac{1}{s}K_I$ dan *derivative* yang ditunjukkan oleh sK_D . K_P , K_I , dan K_D merupakan parameter pengendali *PID* yang harus diatur untuk mendapatkan respon keluaran yang baik.

Masing-masing parameter *PID* ini memiliki perannya masing-masing [6]. Parameter *proportional* akan mengurangi *rise time* dan akan mengurangi (tetapi tidak dapat menghilangkan) *steady state error*. Parameter *integral* dapat menghilangkan *steady state error*. Gabungan dari parameter *proportional* dan *integral* akan menghasilkan pengendali PI. Pengendali PI ini dapat menghilangkan *steady state error*, tetapi akibatnya *transient response* akan bertambah buruk. Untuk memperbaiki *transient response* diperlukan parameter *derivative*. Parameter *derivative* ini akan memperbaiki *transient response* yaitu mengurangi *overshoot* dan *settling time*. Gabungan dari ketiga parameter ini adalah pengendali *PID* yang dapat meningkatkan stabilitas sistem, mengurangi *overshoot*, dan mempercepat *transient response*.

Pengendali *PID* dalam rangkaian analog ditunjukkan oleh rangkaian pada gambar 2.12a. Pengendali analog secara dominan digunakan dalam produk-produk elektronika di pasaran dalam bentuk chip yang terintegrasi [4]. Biayanya yang rendah, desain yang sederhana, efisiensinya yang tinggi, dan kinerja dinamik yang cukup baik merupakan beberapa alasan mengapa pengendali analog dalam bentuk chip yang terintegrasi banyak digunakan. Selain dalam bentuk K_P , K_I dan K_D , pengendali *PID* juga biasa dinyatakan dalam bentuk K_P , T_i dan T_D seperti ditunjukkan dalam blok pada gambar 2.12b. Seperti ditunjukkan gambar 2.12b, nilai parameter pengendali *PID* ini dapat diatur dengan mengatur besar komponen pada rangkaian analog.

Selain dengan pengendali analog dalam bentuk rangkaian, pengendali *PID* juga dapat dibuat dalam bentuk digital, yaitu dengan program di dalam mikroprosesor atau *DSP*. Pengendali digital ini memiliki beberapa kelebihan diantaranya fleksibilitasnya, penggunaan komponen pasif yang minim, dan implementasinya yang lebih sederhana untuk pengendali yang rumit [4].



Gambar 2.12. (a) Pengendali *PID* analog.
(b) Fungsi transfer dan hubungannya dengan besar komponen rangkaian.

Penurunan persamaan *PID* digital dapat dengan melihat pada bentuk kontinunya [5]. Persamaan *PID* dalam bentuk *Laplace* adalah

$$U(s) = K_P \left[1 + \frac{1}{T_i s} + T_D s \right] E(s). \quad (2.27)$$

Jika persamaan ini dikembalikan dalam fungsi waktu akan menjadi

$$u(t) = K_P \left[e(t) + \frac{1}{T_i} \int e(\tau) d\tau + T_D \frac{de(t)}{dt} \right]. \quad (2.28)$$

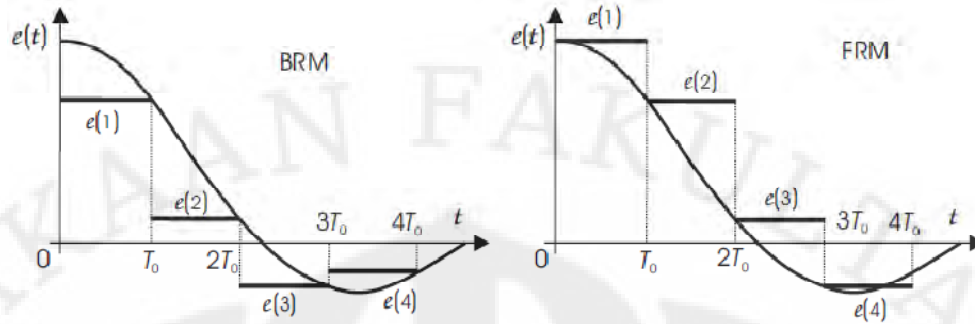
Untuk mengubah persamaan ini ke dalam bentuk diskrit, dapat dilakukan pendekatan diskrit dengan perioda pencuplikan T_0 , sehingga didapatkan bentuk *derivative*

$$\frac{de}{dt} \approx \frac{e(k) - e(k-1)}{T_0}. \quad (2.29)$$

Sedangkan untuk bentuk *integral*, dapat dilakukan dengan beberapa metode, beberapa diantaranya adalah *Forward Rectangular Method (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

$$\int_0^T e(\tau) d\tau \approx T_0 \sum_{i=1}^k e(i-1). \quad (2.30)$$

Sedangkan untuk *BRM* didapatkan



Gambar 2.13 Metoda pencuplikan. (a) *Backward Rectangular Method*
(b) *Forward Rectangular Method*

$$\int_0^T e(\tau) d\tau \approx T_0 \sum_{i=1}^k e(i) \quad (2.31)$$

Persamaan 2.31 lebih sering digunakan dalam menulis persamaan *PID*. Memakai persamaan 2.31 ini dan persamaan 2.29 untuk menggantikan persamaan 2.28 didapatkan *absolute/position algorithm* untuk pengendali *PID*

$$u(k) = K_P \left\{ e(k) + \frac{T_0}{T_i} \sum_{i=1}^k e(i) + \frac{T_D}{T_0} [e(k) - e(k-1)] \right\}. \quad (2.32)$$

Pengendali *PID* pada persamaan 2.32 disebut *nonrecurrent algorithm* [5] 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

$$u(k) = u_P(k) + u_I(k) + u_D(k) \quad (2.33)$$

dimana

$$u_P(k) = K_P e(k) \quad (2.34)$$

$$u_I(k) = K_P \frac{T_0}{T_i} \sum_{i=1}^k e(i) = u_I(k-1) + K_P \frac{T_0}{T_i} e(k) \quad (2.35)$$

$$u_D(k) = K_P \frac{T_D}{T_0} [e(k) - e(k-1)] \quad (2.36)$$

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* [5].

2.8 Penalaan *PID* dengan Metode Ciancone

Dalam merancang pengendali *PID* dibutuhkan penalaan parameter-parameter *PID* yang sesuai. Dengan penalaan parameter-parameter *PID* yang sesuai bisa didapatkan respon hasil keluaran yang diinginkan.

Salah satu metode untuk melakukan penalaan parameter-parameter *PID* adalah dengan menggunakan Ciancone correlations atau disebut metode Ciancone. Langkah-langkah untuk melakukan penyetelan parameter *PID* dengan menggunakan metode Ciancone adalah sebagai berikut.

1. Menentukan model dinamik dari sistem dengan metode kurva proses reaksi sehingga didapatkan parameter-parameter model seperti K_p , θ , dan τ .
2. Menghitung besarnya *fraction dead time*, $\theta / (\theta + \tau)$.
3. Memilih tipe pengendali yang ingin dipakai, sebagai *regulator (disturbance)* atau *servo (set point)*.
4. Mencari hubungan *fraction dead time* dengan konstanta Ciancone $K_c K_p$, $T_I / (\theta + \tau)$, dan $T_D / (\theta + \tau)$ dari grafik Ciancone.
5. Menghitung konstanta penyetelan *PID*, yaitu K_c , T_I , dan T_D .

Salah satu cara untuk menentukan model dinamik dari sistem adalah dengan metode kurva proses reaksi yang meliputi empat tahap :

1. Menunggu proses untuk mencapai kondisi tunak
2. Memberikan masukan *step* pada sistem
3. Mengumpulkan data masukan dan keluaran sampai proses kembali mencapai kondisi tunak
4. Melakukan perhitungan dari grafik kurva proses reaksi

Perhitungan dari grafik kurva reaksi ditujukan untuk mendapatkan model *first order with dead time*. Bentuk model ini adalah

$$\frac{Y(s)}{X(s)} = \frac{K_p e^{-\theta s}}{\tau s + 1} \quad (2.37)$$

Terdapat dua buah metode grafis yang dapat digunakan untuk menentukan parameter-parameter dari model *first order with dead time*. Metode pertama menggunakan perhitungan grafis dari hasil data seperti di gambar 2.14. Data-data yang diamati dari grafik adalah besarnya magnitudo perubahan input, δ ; magnitudo dari perubahan kondisi tunak keluaran, Δ ; dan kemiringan maksimum

dari grafik keluaran terhadap waktu, S . Persamaan umum dari proses dengan masukan *step* untuk $t \geq \theta$ adalah

$$Y'(t) = K_p \delta \left[1 - e^{-(t-\theta)/\tau} \right] \quad (2.38)$$

Dan kemiringannya untuk $t \geq \theta$ didapatkan dengan

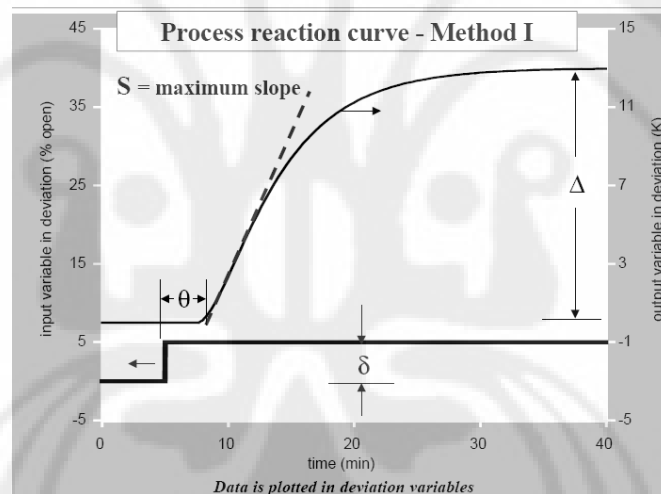
$$\frac{dY'(t)}{dt} = \frac{d}{dt} \left\{ K_p \delta \left[1 - e^{-(t-\theta)/\tau} \right] \right\} = \frac{\Delta}{\tau} e^{-(t-\theta)/\tau}$$

Kemiringan maksimum didapatkan pada $t = \theta$, jadi $S = \Delta/\tau$. Sehingga didapatkan nilai-nilai parameter model adalah

$$K_p = \Delta / \delta$$

$$\tau = \Delta / S$$

θ = perpotongan kemiringan maksimum dengan nilai awal (seperti ditunjukkan gambar 2.14).



Gambar 2.14 Metode satu menentukan model *first order with dead time*

Metode kedua menggunakan pendekatan grafis seperti ditunjukkan oleh gambar 2.15 Data-data yang diamati dari grafik adalah besarnya magnitudo perubahan input, δ ; magnitudo dari perubahan kondisi tunak keluaran, Δ ; dan waktu dimana keluaran mencapai 28 dan 63 persen dari nilai akhirnya. Dari bentuk umum persamaan 2.33 didapatkan

$$Y(\theta + \tau) = \Delta(1 - e^{-1}) = 0.632\Delta$$

$$Y(\theta + \tau/3) = \Delta(1 - e^{-1/3}) = 0.283\Delta$$

Sehingga didapatkan

$$t_{63\%} = \theta + \tau, \text{ dan}$$

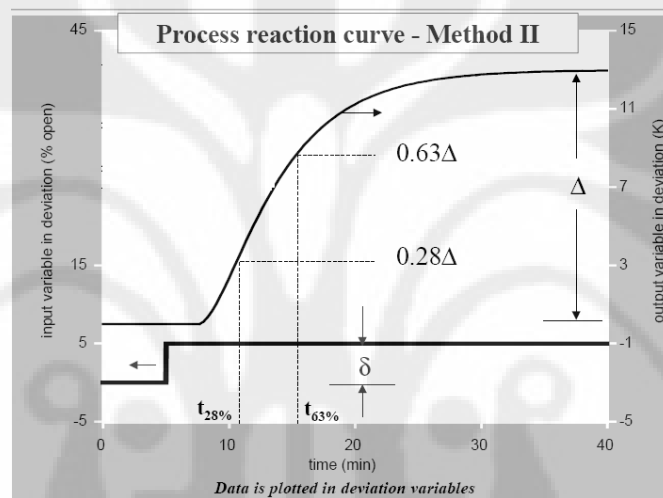
$$t_{28\%} = \theta + \tau/3$$

Maka

$$\tau = 1.5(t_{63\%} - t_{28\%}) \quad (2.39)$$

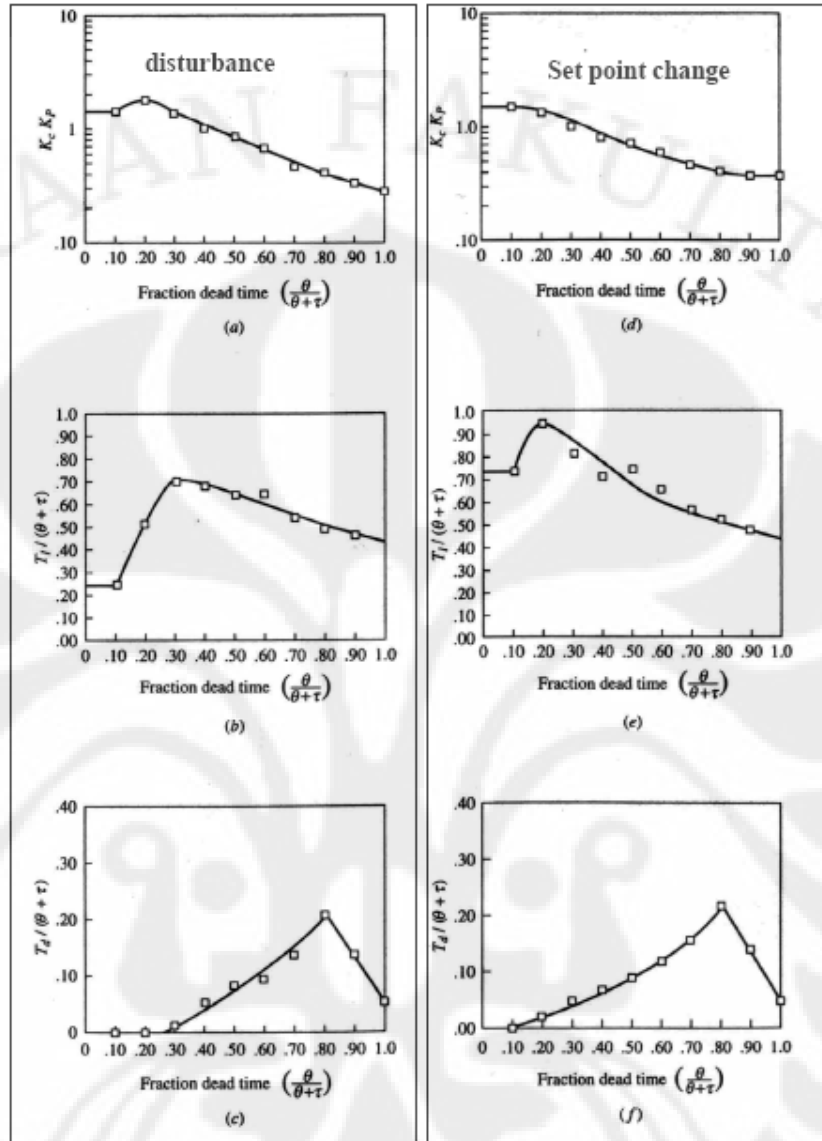
$$\theta = t_{63\%} - \tau \quad (2.40)$$

Karena sulit dalam menentukan kemiringan maksimum, terutama saat sinyal memiliki noise yang tinggi maka metode kedua lebih baik untuk digunakan.



Gambar 2.15 Metode dua menentukan model *first order with dead time*

Dari model kurva proses reaksi ini bisa didapatkan nilai fraction *dead time* yaitu $\theta/(\theta + \tau)$. Nilai ini kemudian digunakan untuk menemukan konstanta penalaan *PID* yang sesuai dengan menggunakan grafik Ciancone seperti terlihat pada gambar 2.16.



Gambar 2.16 Grafik Ciancone

2.9 Pengaruh Waktu Cuplik

Waktu cuplik pengendali perlu ditentukan untuk merancang pengendali diskrit. Menurut [8] waktu cuplik pada pengendali diskrit akan meningkatkan nilai *dead time*. Kaitan antara nilai *dead time* dengan waktu cuplik adalah

$$\theta' = \theta + 0.5T_0 \quad (2.41)$$

dimana θ' merupakan nilai *dead time* baru sebagai akibat adanya waktu cuplik pada pengendali diskrit.

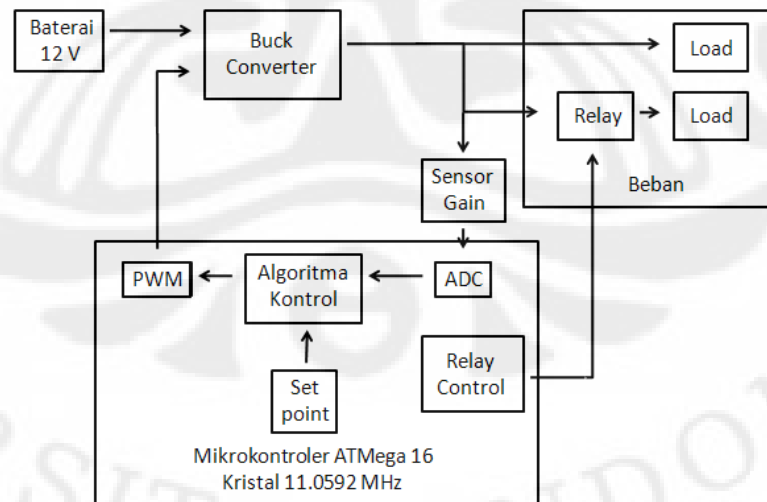
Menurut [8], waktu cuplik maksimum agar pengendali diskrit memiliki kinerja yang mendekati pengendali kontinu adalah $T_0 = 0.05(\theta + \tau)$. Sedangkan menurut [10], pengendali diskrit dapat memiliki waktu cuplik dalam batasan

$$\frac{1}{5} t_{95\%} \leq T_0 \leq \frac{1}{20} t_{95\%} \quad (2.42)$$

dimana $t_{95\%}$ merupakan waktu dimana keluaran mencapai nilai 95 persen dari nilai akhirnya.

BAB 3
PERANCANGAN SISTEM *BUCK CONVERTER* DAN PENGENDALI *PID*
DISKRIT

Dalam skripsi ini, penulis merancang sistem *buck converter* yang dapat mengatur tegangan keluaran dari tegangan masukan baterai 12 Volt. Sistem *buck converter*, catu daya baterai, beban, dan pengendali yang dirancang tampak seperti pada gambar 3.1. Pada gambar terlihat rangkaian *buck converter* menerima tegangan masukan 12 Volt kemudian mengubahnya ke tegangan yang lebih rendah dan hasil ini diberikan pada beban. Agar tegangan yang diberikan pada beban dapat bernilai tetap, tegangan keluaran sistem diumpanbalikkan ke pengendali yang berupa mikrokontroler melewati perantara berupa sensor gain. Sensor gain berupa rangkaian *voltage divider* yang memperlemah tegangan keluaran agar dapat dibaca dengan aman oleh mikrokontroler. Tegangan keluaran yang telah diperlemah ini kemudian diterima mikrokontroler dengan bantuan fitur *ADC (Analog to Digital Converter)*. Dari blok *ADC* didapatkan nilai yang kemudian digunakan algoritma kontrol untuk menghitung tegangan analog yang dibutuhkan sistem. Nilai ini kemudian diberikan ke blok *PWM* yang mengubah tegangan analog yang dibutuhkan menjadi pulsa digital dengan *duty ratio* tertentu.



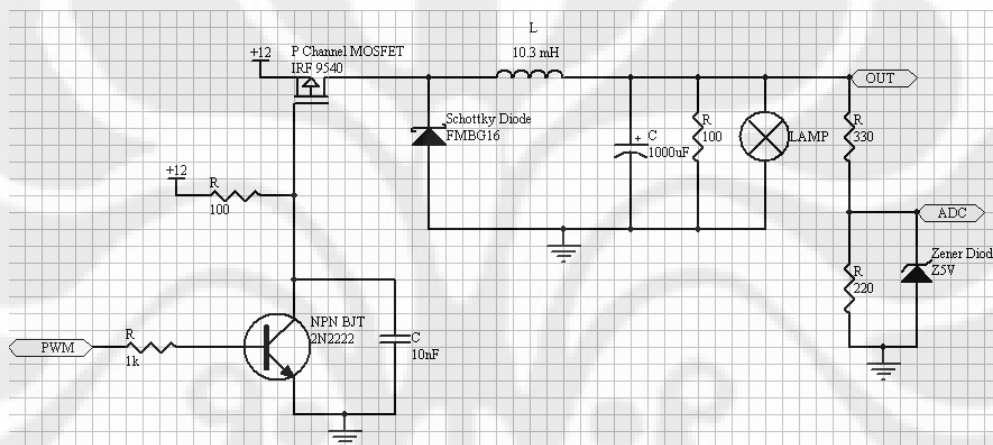
Gambar 3.1 Sistem *buck converter*, beban, dan mikrokontroler sebagai pengendalinya

Pulsa *PWM* ini menjadi keluaran mikrokontroler yang digunakan untuk mengendalikan rangkaian *buck converter*. Mikrokontroler juga digunakan untuk memberikan sinyal pada relay guna mengujicobakan kemampuan pengendali terhadap perubahan beban sistem.

Pada bab ini akan dibahas mengenai perancangan sistem *buck converter* beserta pengendali yang digunakan.

3.1. Rangkaian *Buck Converter*

Rangkaian *buck converter* yang digunakan ditunjukkan oleh gambar 3.2



Gambar 3.2 Rangkaian *buck converter*

Pada rangkaian digunakan MOSFET IRF 9540. MOSFET ini memiliki kemampuan menahan beda tegangan antara drain dan source-nya V_{DS} sampai sebesar 100 Volt, arus drain maksimum hingga 23 Ampere, dan memiliki hambatan ketika dalam keadaan aktif R_{on} senilai 0.117 Ω .

Untuk dioda yang digunakan adalah dioda schottky FMBG16. Dioda ini memiliki tegangan *breakdown* 60 Volt, arus maksimum hingga 6 Ampere, *reverse recovery time* t_{rr} 100 ns, dan tegangan jatuh ketika aktif v_D senilai 0.62 Volt.

Untuk LC filter digunakan induktor dengan induktansi 10.3 mH dan kapasitor dengan kapasitansi 1000 μF . Untuk kapasitor yang digunakan didapatkan nilai ESR (*Equivalent Series Resistant*) atau r_c sebesar 0.01995.

Untuk beban digunakan hambatan *dummy* berupa resistor dengan hambatan 100 Ω . Dalam percobaan digunakan lampu DC 10 Watt dengan hambatan dalam 1.5 Ω sebagai beban dan sekaligus indikator. Terdapat juga

beban berupa dua buah resistor yaitu resistor 220 Ω diseri dengan 330 Ω . Sehingga beban total dari sistem jika hanya digunakan satu buah lampu indikator

$$\text{adalah } R = \frac{1}{\frac{1}{100} + \frac{1}{1.5} + \frac{1}{550}} \approx 1.5 \Omega.$$

3.2 Mikrokontroler ATmega16

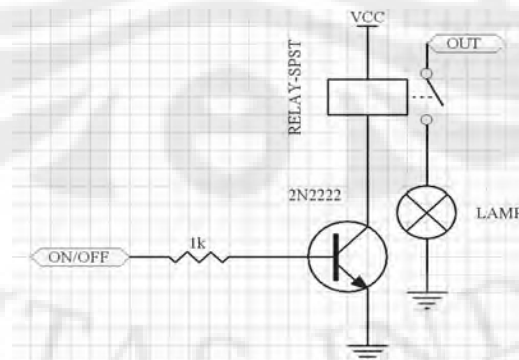
Mikrokontroler Atmega16 digunakan sebagai pengendali utama yang melakukan proses *ADC*, penghitungan sinyal pengendali, pembangkitan pulsa *PWM*, serta pengaturan *set point* dan sinyal pengaktifan beban ke relay. Mikrokontroler yang digunakan memakai kristal dengan frekuensi sebesar 11.0592 MHz.

ADC yang digunakan merupakan *ADC* 10 bit, referensi *ADC* yang digunakan adalah tegangan sumber mikrokontroler sebesar 5 volt, dan waktu konversi nilai *ADC* adalah sekitar 65 μ s.

Pembangkitan pulsa *PWM* menggunakan kemampuan *output compare unit* tipe *fast PWM mode* agar dapat menghasilkan frekuensi yang cukup tinggi.

3.3 Rangkaian Relay Pengatur Beban

Rangkaian relay pengatur beban ini berfungsi untuk mengaktifkan beban tambahan berupa lampu pijar. Rangkaian yang digunakan tampak seperti pada gambar 3.3. Relay yang digunakan merupakan relay OEG tipe 0MIH-SS-105LM. Relay diaktifkan dengan memberikan tegangan 5 volt pada kutubnya. Rating keluaran dari relay ini adalah arus 16 A dan tegangan 240VAC/24VDC.



Gambar 3.3 Rangkaian relay Pengatur Beban

3.4 Penentuan Nilai Frekuensi Switching

Berdasarkan persamaan 2.23 yang diturunkan ada bab sebelumnya didapatkan

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

Atau dengan kata lain

$$f_{\min} = \frac{(1-D)R}{2L}$$

agar sistem tetap beroperasi pada daerah kontinu. Pada bagian sebelumnya didapatkan nilai beban maksimum $R = 100 \Omega$ dan nilai $L = 10.3 \times 10^{-3} \text{ H}$ sehingga

$$f_{\min} = 4854.4(1-D)$$

Jadi agar semua kondisi *duty ratio* terpenuhi ($0 \leq D \leq 1$) maka didapatkan

$$f_{\min} = 4854.4 \text{ Hz} \quad (3.1)$$

merupakan nilai minimum frekuensi yang digunakan agar sistem tetap beroperasi pada daerah kontinu.

Berdasarkan persamaan 2.24 yang diturunkan pada bab sebelumnya juga didapatkan hubungan besarnya *ripple* tegangan adalah

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

Pada bagian sebelumnya didapatkan nilai $C = 1 \times 10^{-3} \text{ F}$ dan nilai $L = 10.3 \times 10^{-3} \text{ H}$. Berdasarkan [2] didapatkan pada umumnya *switching power supply* memiliki standar *ripple* tegangan lebih kecil daripada 1%, maka bisa didapatkan

$$f_{\min} = 1101.6 \sqrt{1-D}$$

Jadi agar semua kondisi *duty ratio* terpenuhi ($0 \leq D \leq 1$) maka didapatkan

$$f_{\min} = 1101.6 \text{ Hz} \quad (3.2)$$

merupakan nilai minimum frekuensi yang digunakan agar sistem memiliki faktor *ripple* tegangan maksimum sebesar 1%.

Dari persamaan 3.1 dan 3.2 bisa didapatkan bahwa nilai frekuensi minimum yang dibutuhkan sistem adalah 4.854 kHz. Untuk meminimalkan noise dari keluaran, frekuensi yang diberikan pada sistem *buck converter* dipilih sebesar 10 kHz.

3.5 Fungsi Alih dan Matriks Ruang Keadaan Rata-Rata dari Sistem

Pada persamaan 2.21 yang diturunkan pada bab sebelumnya didapatkan fungsi alih dari sistem adalah

$$\frac{Y(s)}{D(s)} = \frac{V_g R}{LC(R+r_c)} \frac{sCr_c + 1}{\left[s^2 + s \left(\frac{1}{C(R+r_c)} + \frac{Rr_c}{L(R+r_c)} \right) + \frac{R}{LC(R+r_c)} \right]}$$

Dengan mensubstitusikan nilai-nilai yang didapat dalam perancangan rangkaian antara lain $V_g = 12 \text{ V}$, $R = 1.5 \Omega$, $L = 0.0103 \text{ H}$, $C = 0.001 \text{ F}$, $r_c = 0.01995$ maka didapatkan

$$\begin{aligned} \frac{Y(s)}{D(s)} &= 1149756.779 \frac{s(1.995 \times 10^{-5}) + 1}{s^2 + 659.828s + 95813.065} \\ &= \frac{22.938s + 1149756.779}{s^2 + 659.828s + 95813.065} \end{aligned} \quad (3.3)$$

Sedangkan untuk matriks ruang keadaan rata-rata didapatkan

$$\frac{d}{dt} \begin{bmatrix} i \\ v \end{bmatrix} = \begin{bmatrix} \frac{-DR_{on}(R+r_c) - Rr_c}{L(R+r_c)} & \frac{-R}{L(R+r_c)} \\ \frac{R}{C(R+r_c)} & \frac{-1}{C(R+r_c)} \end{bmatrix} \begin{bmatrix} i \\ v \end{bmatrix} + \begin{bmatrix} \frac{D}{L} & \frac{D-1}{L} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_g \\ V_D \end{bmatrix}.$$

Dengan mensubstitusikan nilai-nilai yang didapatkan dalam perancangan rangkaian antara lain $V_g = 12 \text{ V}$, $R = 1.5 \Omega$, $L = 0.0103 \text{ H}$, $C = 0.001 \text{ F}$, $r_c = 0.01995$, $R_{on} = 0.117 \Omega$, $v_D = 0.62 \text{ V}$ didapatkan

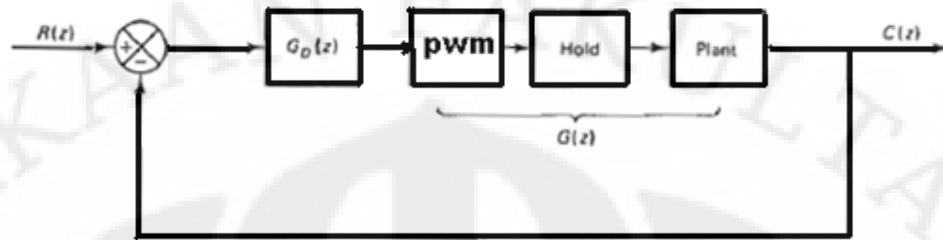
$$\frac{d}{dt} \begin{bmatrix} i \\ v \end{bmatrix} = \begin{bmatrix} -11.357394D - 1.91153 & -95.816 \\ 986.874568 & -675.916379 \end{bmatrix} \begin{bmatrix} i \\ v \end{bmatrix} + \begin{bmatrix} 97.087379D & 97.087379(D-1) \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_g \\ V_D \end{bmatrix} \quad (3.4)$$

Dan persamaannya keluarannya

$$V = [0.019688 \quad 0.986875] \begin{bmatrix} i \\ v \end{bmatrix} \quad (3.5)$$

3.6 Respon Sistem Tanpa Pengendali

Skema pengendalian diskrit tampak seperti pada gambar 3.4.



Gambar 3.4 Skema pengendalian diskrit

$$T_0 = 190 \mu\text{s}$$

$$\begin{aligned} G(z) &= Z \left\{ \frac{1}{V_g} \frac{1 - e^{-Ts}}{s} \frac{22.938s + 1149756.779}{s^2 + 659.828s + 95813.065} \right\} \\ &= \frac{1 - z^{-1}}{12} Z \left\{ \frac{22.938s + 1149756.779}{s(s^2 + 659.828s + 95813.065)} \right\} \\ &= \frac{0.002z + 0.00125}{z^2 - 1.879z + 0.8822} \end{aligned} \quad (3.6)$$

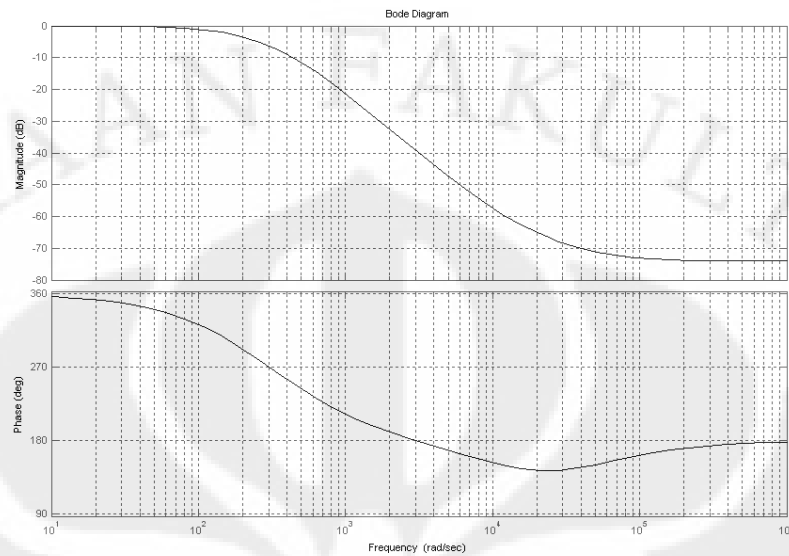
Untuk melihat *bode plot* dari sistem diskrit, sistem perlu diubah ke domain w dengan *bilinear transformation* [9], yaitu dengan mensubstitusikan nilai

$$Z = \frac{1 + (T_0/2)w}{1 - (T_0/2)w}$$

pada sistem diskrit. Dari hasil substitusi didapatkan

$$G(w) = \frac{-0.0001994w^2 - 6.997w + 95740}{w^2 + 659.5w + 95740} \quad (3.7)$$

Bode plot dari $G(w)$ ditunjukkan oleh gambar 3.5.

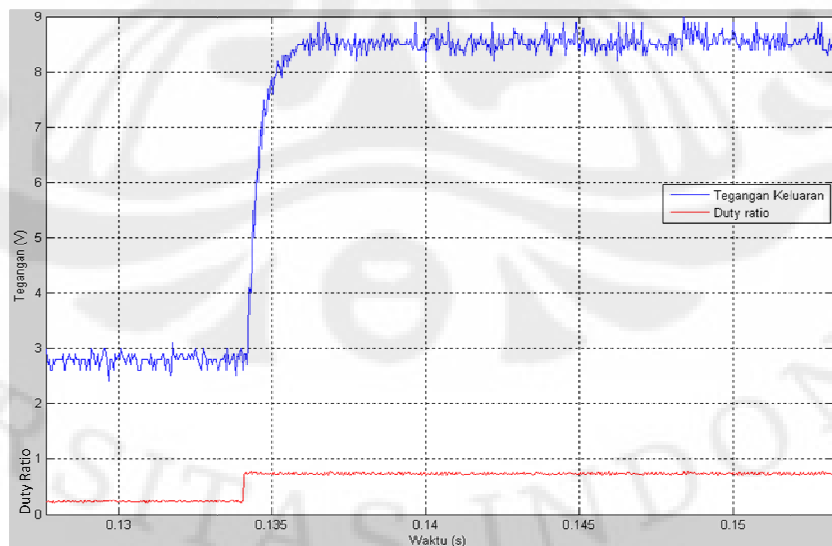


Gambar 3.5 Bode plot dari sistem tanpa pengendali

Dari bode plot didapatkan gain margin $G_M = 39.5$ dB dan phase margin = 180° . Kedua nilai ini menunjukkan sistem dalam keadaan stabil.

3.7 Perancangan Pengendali *PID*

Sistem buck converter yang telah dirancang diuji secara lingkaran terbuka dengan pemberian masukan *step*. Hasil yang didapat seperti pada gambar 3.6.



Gambar 3.6 Respon sistem terhadap masukan *step*

Sistem kemudian dimodelkan dengan model *first order with dead time* metode kedua yang telah dibahas pada bab sebelumnya. Hasil yang didapatkan adalah.

$$\Delta = 8.6 - 2.8 = 5.8$$

$$\delta = 0.75 - 0.25 = 0.5$$

$$t_{63\%} = 0.5 \text{ ms}$$

$$t_{28\%} = 0.2 \text{ ms}$$

Sehingga didapatkan

$$K_p = \frac{\Delta}{\delta} = 11.6$$

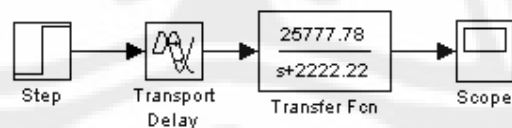
$$\tau = 1.5(t_{63\%} - t_{28\%}) = 0.45 \text{ ms}$$

$$\theta = t_{63\%} - \tau = 0.05 \text{ ms}$$

merupakan parameter-parameter model *first order with dead time* yang didapatkan dari sistem. Model *first order with dead time* yang didapatkan adalah

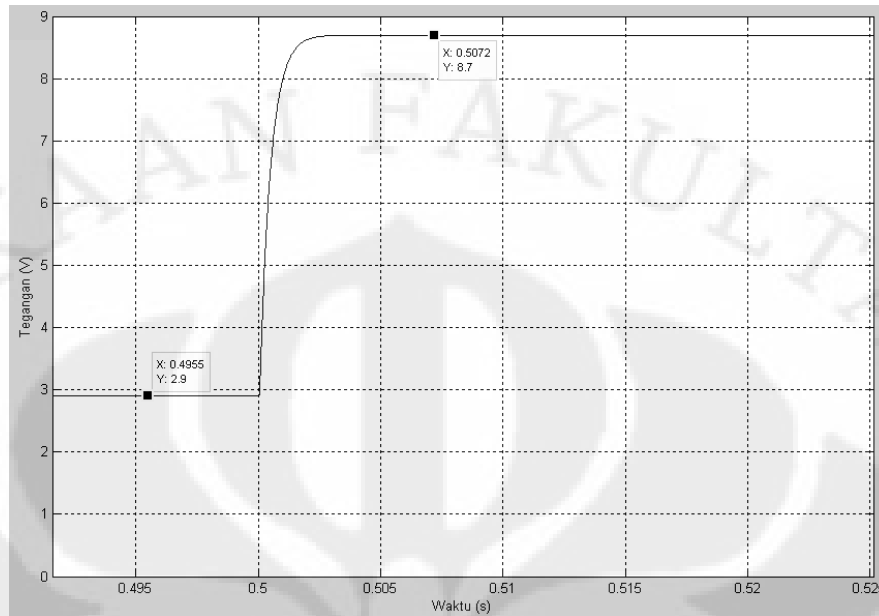
$$\begin{aligned} \frac{V(s)}{D(s)} &= \frac{K_p e^{-\theta s}}{\tau s + 1} \\ &= \frac{11.6 e^{-5 \times 10^{-5} s}}{0.45 \times 10^{-3} s + 1} \\ &= \frac{25777.78 e^{-5 \times 10^{-5} s}}{s + 2222.22} \end{aligned}$$

Model yang didapat diuji dengan blok simulink seperti pada gambar 3.7



Gambar 3.7 Sistem *first order with dead time* pada simulink

Hasil yang didapatkan adalah sistem *first order with dead time* yang didapat menyerupai sistem aslinya, seperti terlihat pada gambar 3.8.



Gambar 3.8 Step respon model *first order with dead time*

Waktu cuplik pengendali yang diberikan sebesar $190 \mu\text{s}$. Dari persamaan 2.42 didapatkan waktu cuplik pengendali harus berada dalam batasan

$$\frac{1}{20} t_{95\%} \leq T_0 \leq \frac{1}{5} t_{95\%}$$

Dari data didapatkan $t_{95\%} = 1.5 \text{ ms}$ sehingga batasan nilai T_0 adalah

$$75 \mu\text{s} \leq T_0 \leq 300 \mu\text{s}$$

Dengan demikian nilai waktu cuplik memenuhi syarat.

Pengaruh waktu cuplik terhadap *dead time* menurut persamaan 2.41 adalah

$$\theta' = \theta + 0.5T_0$$

Maka didapatkan *dead time* baru akibat waktu cuplik adalah

$$\theta' = 0.145 \text{ ms.}$$

Untuk menentukan parameter pengendali *PID* digunakan grafik Ciancone seperti yang terdapat pada gambar 2.16. Untuk itu diperlukan nilai fraction *dead time* terlebih dahulu. Nilai fraction *dead time* adalah

$$\frac{\theta'}{\theta' + \tau} = 0.24$$

Dari grafik Ciancone pada gambar 2.16, dipilih grafik *set point*. Hasil yang didapatkan adalah

$$K_c K_p = 1.2$$

$$\frac{T_i}{\theta' + \tau} = 0.65$$

$$\frac{T_D}{\theta' + \tau} = 0$$

Sehingga didapatkan parameter pengendali *PID*

$$K_c = 0.103$$

$$T_i = 0.4 \times 10^{-3}$$

$$T_D = 0$$

Pengendali yang digunakan hanyalah berupa pengendali PI karena nilai T_D bernilai nol. Pengendali PI yang digunakan diimplementasikan dengan mikrokontroler menggunakan persamaan 2.33, 2.34, dan 2.35. Fungsi alih pengendali PI dalam domain z adalah

$$\begin{aligned} G_{PI}(z) &= K_c \left(1 + \frac{T_0}{T_I} \frac{1}{1-z^{-1}} \right) \\ &= \frac{K_c \left(1 + \frac{T_0}{T_I} \right) z - K_c}{z-1} \end{aligned}$$

Dengan mensubstitusikan nilai K_c , T_0 dan T_I didapatkan

$$G_{PI}(z) = \frac{0.152z - 0.103}{z-1}$$

$$\begin{aligned} G_{PI}(z)G(z) &= \frac{0.002z + 0.00125}{z^2 - 1.879z + 0.8822} \cdot \frac{0.152z - 0.103}{z-1} \\ &= \frac{0.000304z^2 - 1.6 \times 10^{-5}z - 0.0001287}{z^3 - 2.879z^2 + 2.761z - 0.8822} \end{aligned}$$

$$\frac{C(z)}{R(z)} = \frac{0.000304z^2 - 1.6 \times 10^{-5}z - 0.0001287}{z^3 - 2.8787z^2 + 2.761z - 0.8823} \quad (3.8)$$

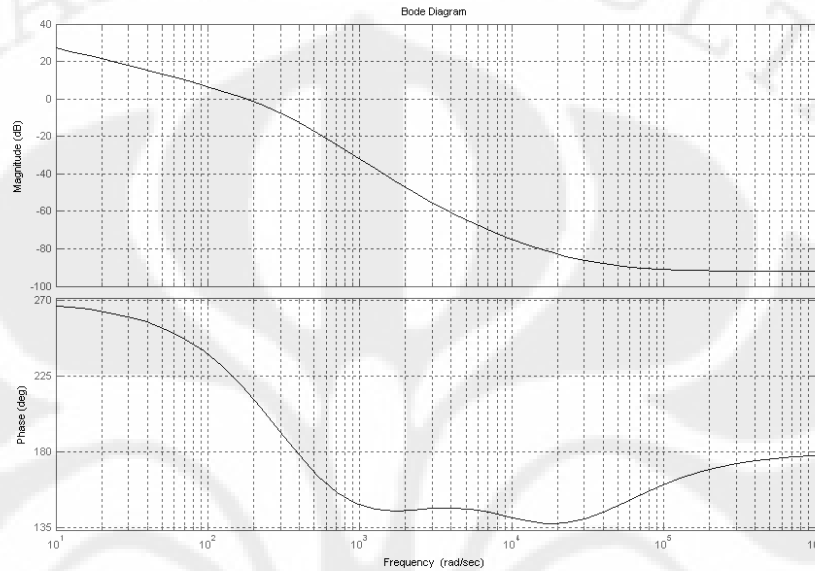
Untuk melihat bode plot dari sistem diskrit, sistem perlu diubah ke domain w dengan bilinear transformation [9], yaitu dengan mensubstitusikan nilai

$$Z = \frac{1 + (T_0/2)w}{1 - (T_0/2)w}$$

pada sistem diskrit. Dari hasil substitusi didapatkan

$$G(w) = \frac{-2.543 \times 10^{-5} w^3 - 0.9433 w^2 + 1.04 \times 10^4 w + 2.47 \times 10^7}{w^3 + 658.8 w^2 + 1.061 \times 10^5 w - 0.0001469} \quad (3.9)$$

Bode plot sistem lingkaran tertutup ditunjukkan oleh gambar 3.9.



Gambar 3.9 Bode sistem lingkaran tertutup

Dari bode plot didapatkan $G_M = 11.8$ dB dan phase margin = 36.6° . Kedua nilai ini menunjukkan sistem masih tetap dalam keadaan stabil.

BAB 4

HASIL UJI COBA DAN ANALISA

Pada bagian ini akan dibahas mengenai hasil dan analisa hasil perancangan pengendali *PID* pada *buck converter* dengan menggunakan metode Ciancone. Hasil yang dibahas didapatkan dari percobaan alat maupun simulasi.

Untuk simulasi yang dilakukan digunakan blok-blok sistem yang dibuat menggunakan fitur C-Mex pada Simulink. Blok-blok utama yang digunakan antara lain blok “PWM”, “buck_conv”, dan “PID”. Blok “PWM” berfungsi sebagai penghasil gelombang PWM DC dengan masukan berupa frekuensi PWM dan duty ratio PWM. Blok “buck_conv” merupakan model dinamik sistem buck converter yang telah diturunkan pada subbab 2.3. Blok “PID” berisi algoritma kendali *PID* diskrit yang dirancang.

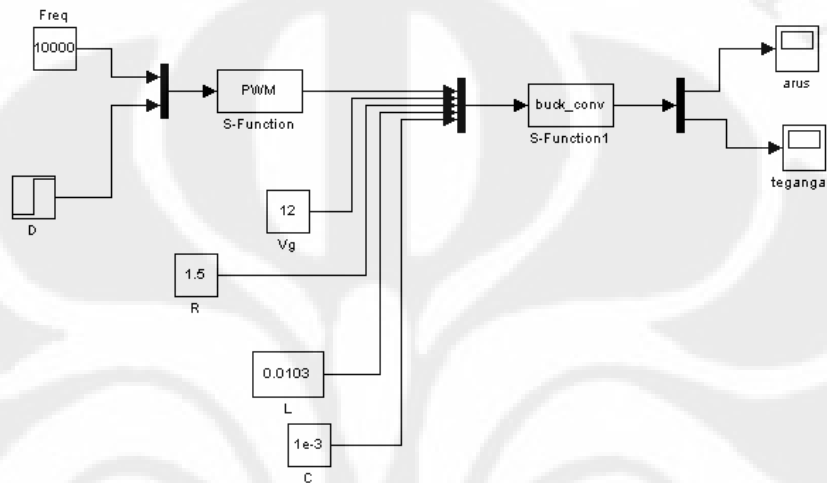
Hasil yang dibahas akan ditampilkan dalam bentuk grafik. Pada setiap grafik akan terdapat dua buah kurva, yaitu kurva berwarna merah dan kurva berwarna biru. Kurva berwarna merah merupakan masukan sistem yang diubah untuk melihat respon sistem pada kurva berwarna biru. Perlu diperhatikan bahwa pada hasil percobaan alat sistem lingkaran tertutup, kurva berwarna merah tidak akan berupa *set point* sesungguhnya yang diberikan pada alat karena masalah grafik yang akan saling menumpuk jika nilai *set point* sesungguhnya ditunjukkan.

Pertama-tama akan dibahas mengenai respon sistem *buck converter* lingkaran terbuka baik dari hasil simulasi maupun dari hasil percobaan alat. Hasil dari simulasi akan dibandingkan dengan hasil percobaan alat. Dalam menguji sistem *buck converter* lingkaran terbuka diamati respon sistem terhadap perubahan *set point*, beban, dan juga tegangan masukan. Analisa terhadap keluaran sistem lingkaran terbuka yang didapatkan kemudian akan dibahas. Respon sistem yang telah diberi pengendali *PID* dengan metode Ciancone baik dari hasil simulasi juga dari hasil percobaan alat akan dipaparkan kemudian. Respon sistem lingkaran tertutup yang dirancang akan diamati pula terhadap perubahan *set point*, beban dan juga tegangan masukan. Kemudian analisa keluaran sistem yang didapatkan akan dibahas.

4.1 Sistem Lingkar Terbuka

4.1.1 Sistem Lingkar Terbuka dengan Perubahan *Duty Ratio*

Blok sistem yang digunakan untuk simulasi sistem lingkaran terbuka dengan perubahan *duty ratio* tampak pada gambar 4.1. Pada gambar 4.1 terlihat bahwa blok “PWM” diberikan *duty ratio* dengan masukan *step*.

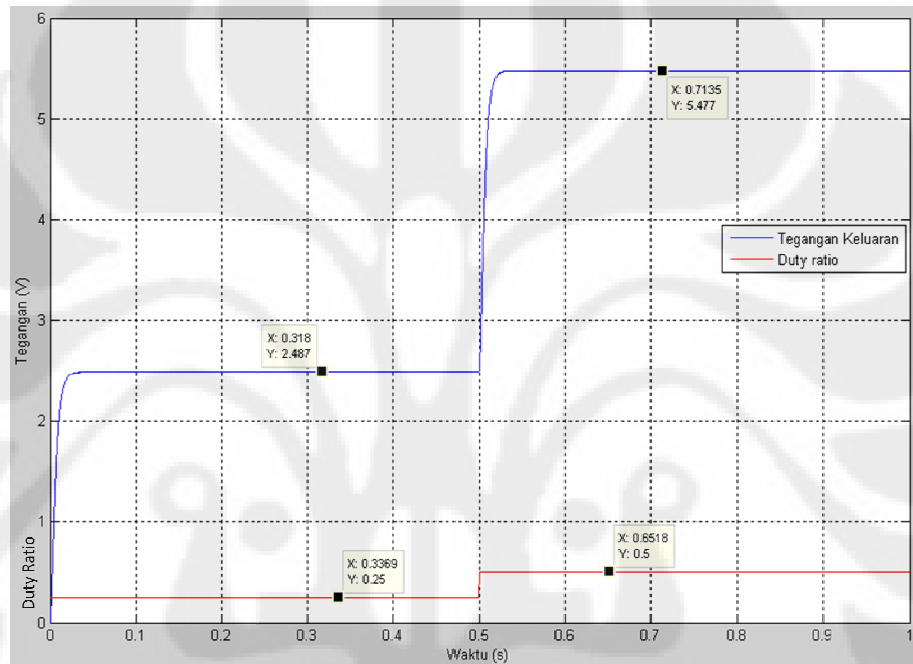


Gambar 4.1 Blok-blok simulasi sistem *buck converter* dengan perubahan *duty ratio*

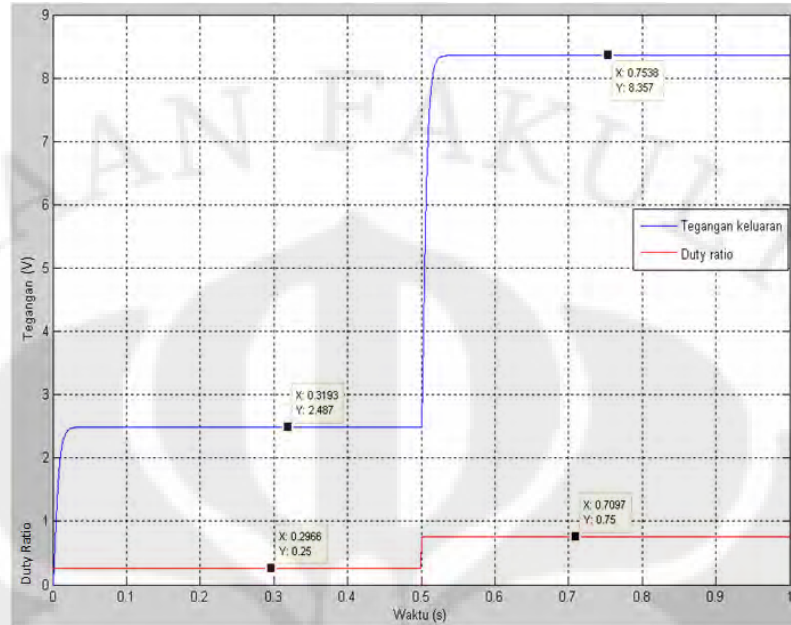
Hasil simulasi ditunjukkan oleh gambar 4.2 dan 4.3. Pada gambar 4.2 diperlihatkan bahwa sistem pada awalnya diberi masukan berupa *duty ratio* sebesar 0.25 dan berubah menjadi 0.5 pada waktu $t = 0.5$ sekon. Hasil keluaran yang didapat ditunjukkan oleh kurva berwarna biru. Terlihat bahwa keluaran sistem dapat mencapai kondisi tunak, yaitu sebesar 2.487 volt pada *duty ratio* 0.25 dan sebesar 5.477 volt pada *duty ratio* 0.5. *Settling time* sistem ($t_{95\%}$) didapatkan sebesar 16.3 ms. Pada gambar 4.3 terlihat pada awalnya sistem diberikan *duty ratio* sebesar 0.25 dan berubah menjadi 0.75 pada $t = 0.5$ sekon. Keluaran sistem juga mencapai kondisi tunak, yaitu sebesar 2.487 volt pada *duty ratio* 0.25 dan sebesar 8.357 volt pada *duty ratio* 0.75. *Settling time* sistem didapatkan sebesar 16 ms.

Hasil percobaan alat ditunjukkan oleh gambar 4.4 dan 4.5. Pada gambar 4.2 terlihat bahwa sistem diberikan *duty ratio* yang berubah dari 0.25 ke 0.5. Terlihat bahwa keluaran sistem dapat mencapai kondisi tunak walaupun terlihat banyak terdapat noise dalam pengukuran. Pada *duty ratio* 0.25 didapatkan

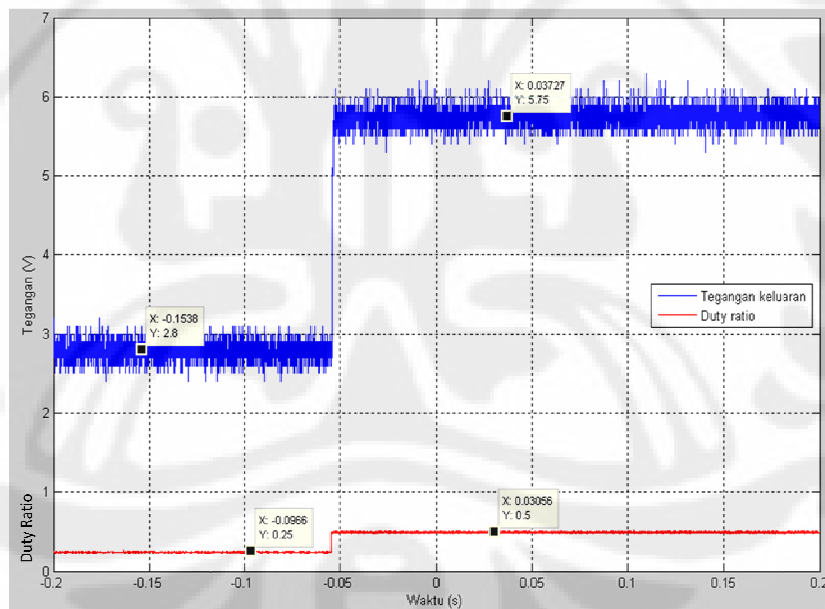
tegangan rata-rata sebesar 2.8 volt dan pada *duty ratio* 0.5 didapatkan tegangan rata-rata 5.75 volt. *Settling time* sistem didapatkan sebesar 1.3 ms. Pada gambar 4.3 terlihat bahwa sistem juga diberikan *duty ratio* yang berubah dari 0.25 ke 0.75. Terlihat bahwa keluaran sistem dapat mencapai kondisi tunak yaitu pada *duty ratio* 0.25 didapatkan tegangan rata-rata sebesar 2.8 volt dan pada *duty ratio* 0.75 didapatkan tegangan rata-rata 8.6 volt. *Settling time* sistem didapatkan sebesar 1.5 ms.



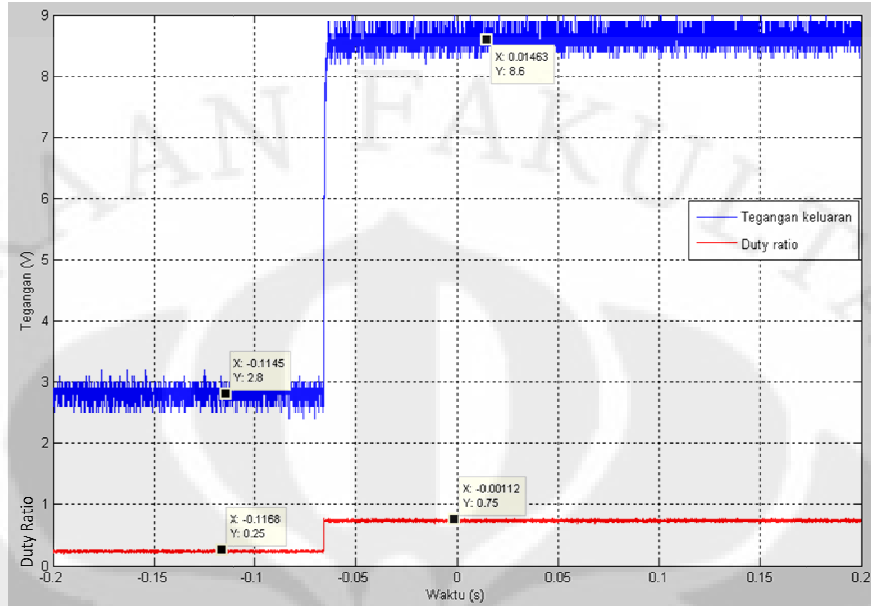
Gambar 4.2 Respon tegangan terhadap perubahan *duty ratio* dari 0.25 ke 0.5 pada simulasi



Gambar 4.3 Respon tegangan terhadap perubahan *duty ratio* dari 0.25 ke 0.75 pada simulasi



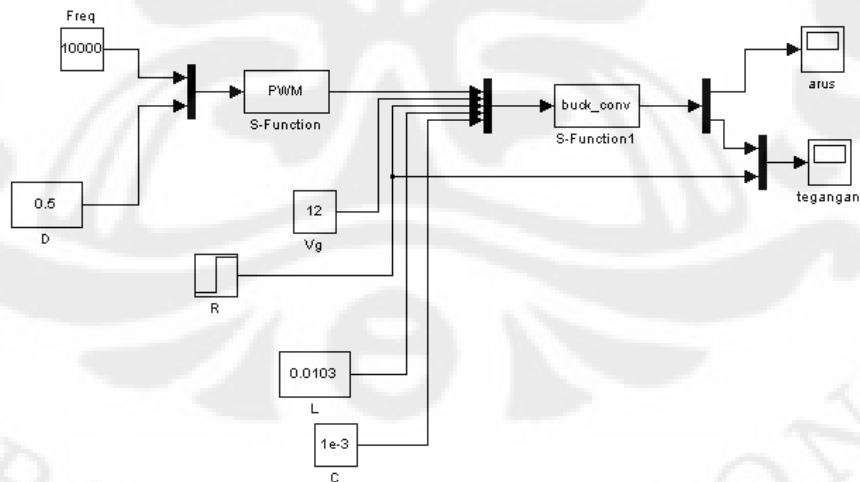
Gambar 4.4 Respon tegangan terhadap perubahan *duty ratio* dari 0.25 ke 0.5 pada alat



Gambar 4.5 Respon tegangan terhadap perubahan *duty ratio* dari 0.25 ke 0.75 pada alat

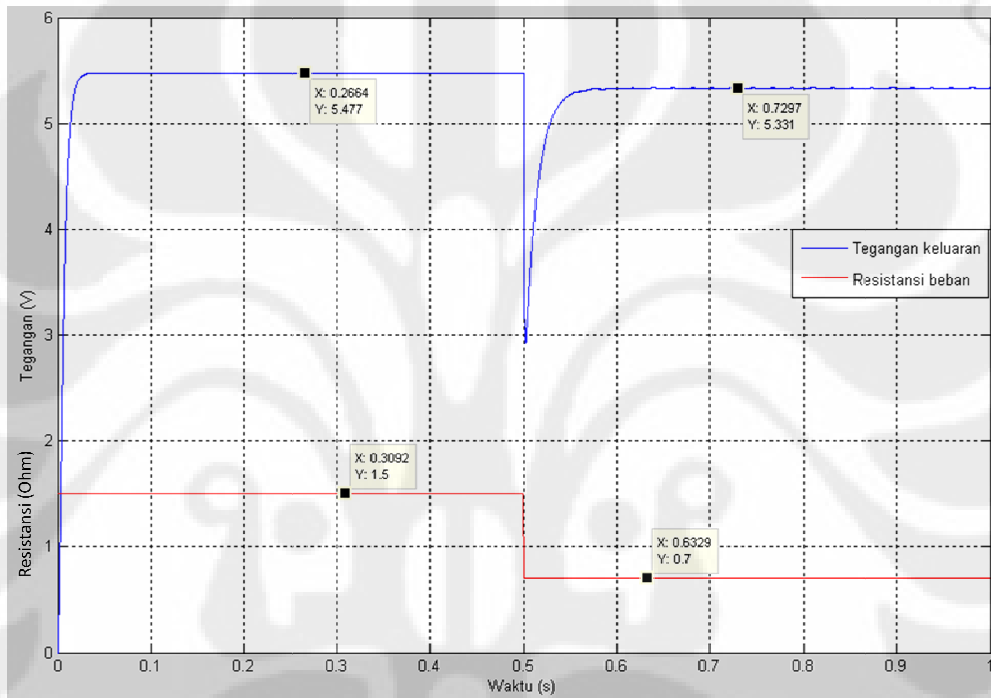
4.1.2 Sistem Lingkaran Terbuka dengan Perubahan Beban

Blok sistem yang digunakan untuk simulasi sistem lingkaran terbuka dengan perubahan beban tampak pada gambar 4.6. Pada gambar 4.6 terlihat bahwa blok “buck_conv” diberikan beban R dengan masukan *step*.



Gambar 4.6 Blok-blok simulasi sistem *buck converter* dengan perubahan beban

Hasil simulasi ditunjukkan oleh gambar 4.7. Pada gambar 4.7 diperlihatkan bahwa sistem pada awalnya diberi masukan berupa beban sebesar 1.5Ω dan berubah menjadi 0.7Ω pada waktu $t = 0.5$ sekon. Hasil keluaran yang didapat ditunjukkan oleh kurva berwarna biru. Terlihat bahwa keluaran sistem dapat mencapai kondisi tunak, yaitu sebesar 5.477 volt pada beban 1.5Ω dan sebesar 5.331 volt pada beban 0.7Ω .

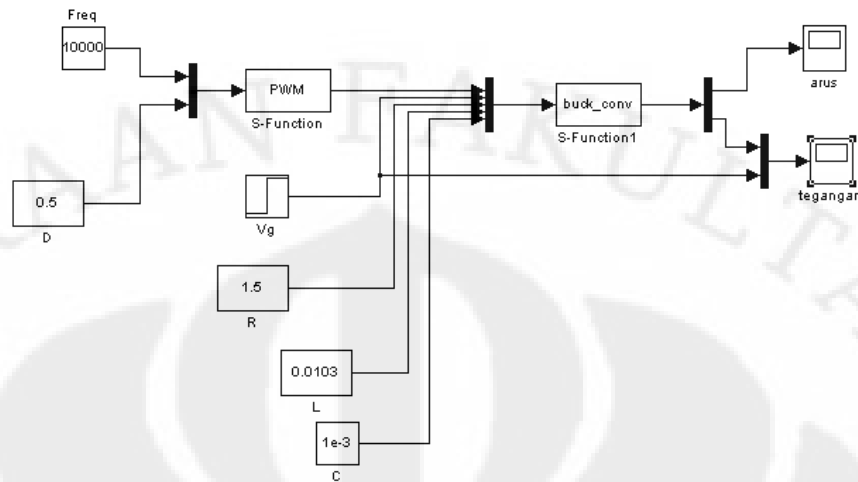


Gambar 4.7 Respon tegangan terhadap perubahan beban dari 1.5Ω ke 0.7Ω

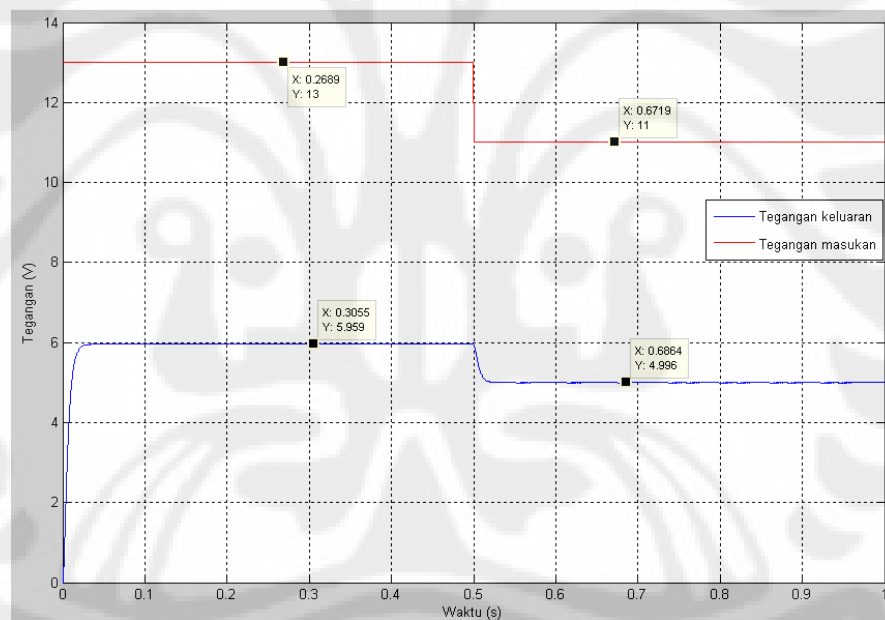
4.1.3 Sistem Lingkaran Terbuka dengan Perubahan Tegangan Masukan

Blok sistem yang digunakan untuk simulasi sistem lingkaran terbuka dengan perubahan tegangan masukan tampak pada gambar 4.8. Pada gambar 4.8 terlihat bahwa blok “buck_conv” diberikan tegangan masukan dengan masukan *step*.

Hasil simulasi ditunjukkan oleh gambar 4.9. Pada gambar 4.9 diperlihatkan bahwa sistem pada awalnya diberi masukan berupa tegangan sebesar 13 volt dan berubah menjadi 11 volt pada waktu $t = 0.5$ sekon. Hasil keluaran yang didapat ditunjukkan oleh kurva berwarna biru. Terlihat bahwa keluaran sistem dapat mencapai kondisi tunak, yaitu sebesar 5.959 volt pada tegangan masukan 13 volt dan sebesar 4.996 volt pada tegangan masukan 11 volt.



Gambar 4.8 Blok-blok simulasi sistem *buck converter* dengan perubahan tegangan masukan



Gambar 4.9 Respon tegangan terhadap perubahan tegangan masukan dari 13 volt ke 11 volt.

4.2 Analisa Sistem Lingkak Terbuka

Pada sistem *buck converter* lingkak terbuka telah diberikan percobaan perubahan *duty ratio*, beban dan tegangan masukan.

Dari percobaan perubahan nilai *duty ratio* terlihat bahwa sistem *buck converter* dapat mencapai kondisi stabil dengan menunjukkan nilai tegangan keluaran yang konstan baik untuk perubahan *duty ratio* dari 0.25 ke 0.5 maupun

untuk perubahan *duty ratio* 0.25 ke 0.75. Tetapi terlihat bahwa tegangan keluaran sistem tidak mencapai nilai tunak yang diinginkan. Sesuai dengan hubungan $V_o = DV_g$, untuk *duty ratio* 0.25 seharusnya didapatkan tegangan keluaran sebesar 3 Volt, untuk *duty ratio* 0.5 seharusnya didapatkan tegangan keluaran sebesar 6 volt, dan untuk *duty ratio* 0.75 seharusnya didapatkan tegangan keluaran sebesar 9 volt. Error yang didapatkan adalah sekitar 0.6 sampai dengan 0.7 volt.

Dari percobaan perubahan nilai *duty ratio* juga diketahui bahwa *settling time* dari sistem pada simulasi cukup berbeda dengan *settling time* pada percobaan alat. Pada simulasi didapatkan nilai *settling time* 16.3 ms dan 16 ms dari dua percobaan perubahan *duty ratio*, sedangkan pada percobaan alat didapatkan nilai *settling time* sebesar 1.3 ms dan 1.5 ms. Ini menunjukkan bahwa sistem yang disimulasikan memiliki perbedaan kecepatan respon sekitar 10 kali dari sistem yang nyata.

Perbedaan kecepatan respon sistem sangat dipengaruhi oleh faktor redaman dari sistem. Pada sistem *buck converter* ini salah satu faktor yang berperan dalam menentukan besarnya redaman dari sistem adalah beban R dan nilai ESR r_c yang terdapat pada sistem. Dari persamaan 2.21 didapatkan fungsi alih sistem lingkaran terbuka

$$\frac{Y(s)}{D(s)} = \frac{V_g R}{LC(R+r_c)} \frac{sCr_c + 1}{\left[s^2 + s \left(\frac{1}{C(R+r_c)} + \frac{Rr_c}{L(R+r_c)} \right) + \frac{R}{LC(R+r_c)} \right]}$$

Dengan pendekatan sistem orde dua didapatkan hubungan

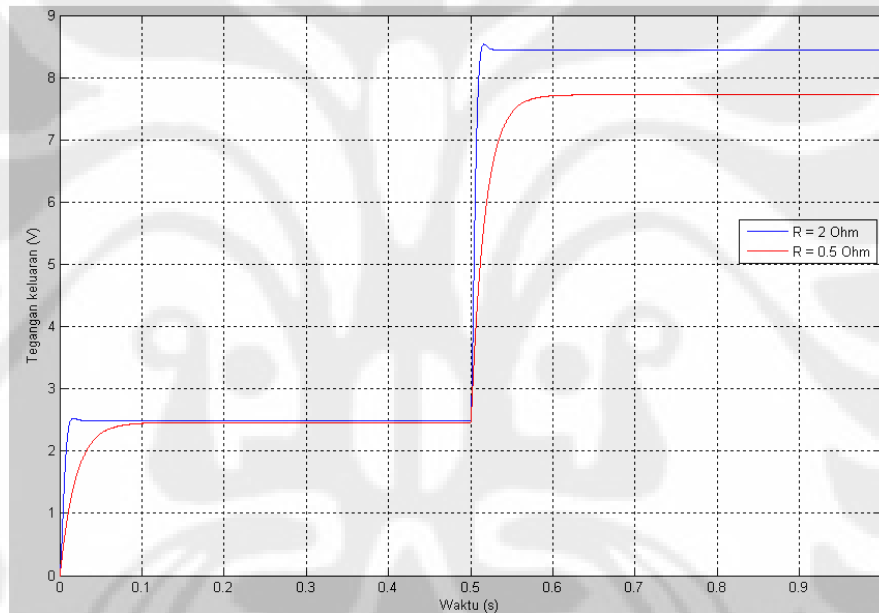
$$2\zeta\omega_n = \frac{1}{C(R+r_c)} + \frac{Rr_c}{L(R+r_c)} \quad (4.1)$$

Pada umumnya r_c jauh lebih kecil dari R sehingga

$$2\zeta\omega_n \approx \frac{1}{RC} + \frac{r_c}{L} \quad (4.2)$$

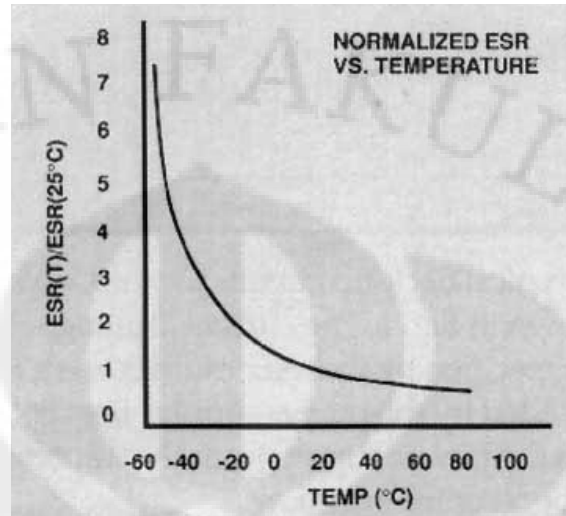
Dari persamaan 4.2 didapatkan bahwa semakin besar R dapat mengakibatkan faktor redaman ζ berkurang dan semakin kecil r_c juga dapat mengakibatkan faktor redaman ζ berkurang. Faktor redaman yang berkurang berarti sistem menjadi kurang teredam sehingga memiliki respon yang lebih cepat.

Dalam simulasi, beban R yang diberikan sebesar 1.5Ω konstan. Hal ini berbeda dengan percobaan dimana resistansi beban bisa saja bertambah dikarenakan perubahan suhu. Pada percobaan beban yang digunakan merupakan lampu pijar dengan hambatan awal 1.5Ω . Seiring dengan dipakainya lampu pijar bebannya pun akan terus naik karena pertambahan suhu. Akibatnya sistem yang terukur dengan nilai hambatan R yang telah bertambah dapat memiliki respon yang lebih cepat. Pengaruh perubahan beban sistem diperlihatkan oleh gambar 4.10 dimana ditunjukkan bahwa semakin besar beban R sistem maka redaman sistem pun akan semakin berkurang.



Gambar 4.10 Pengaruh hambatan terhadap redaman sistem

Faktor ESR r_c yang turun juga memberikan kontribusi terhadap berkurangnya redaman sistem. Penyebab kemungkinan turunnya r_c dapat dijelaskan dari gambar 4.11. Gambar 4.11 menunjukkan bahwa seiring dengan bertambahnya temperatur, maka nilai r_c dari kapasitor pun akan turut turun secara eksponensial. Pada data simulasi, nilai r_c yang ada merupakan nilai r_c pada suhu ruang. Ketika alat dijalankan, akan terjadi arus yang dapat menyebabkan kapasitor bertambah suhunya sehingga nilai r_c sebenarnya pun turun.



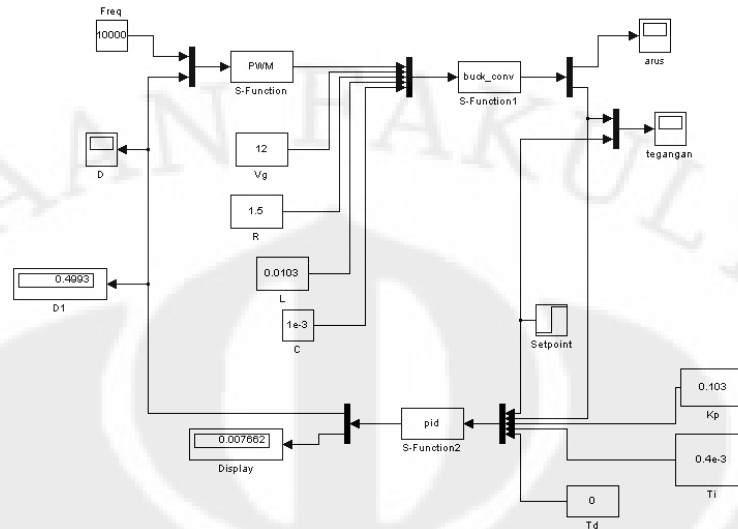
Gambar 4.11 Pengaruh pertambahan suhu terhadap nilai ESR

Dari percobaan perubahan beban dan tegangan masukan terlihat bahwa sistem lingkaran terbuka rentan terhadap perubahan beban sistem maupun terhadap perubahan tegangan masukan sistem. Hal ini harus dapat diatasi oleh pengendali pada sistem lingkaran tertutup.

4.3 Sistem Lingkaran Tertutup

4.3.1 Sistem Lingkaran Tertutup dengan Perubahan *Set point*

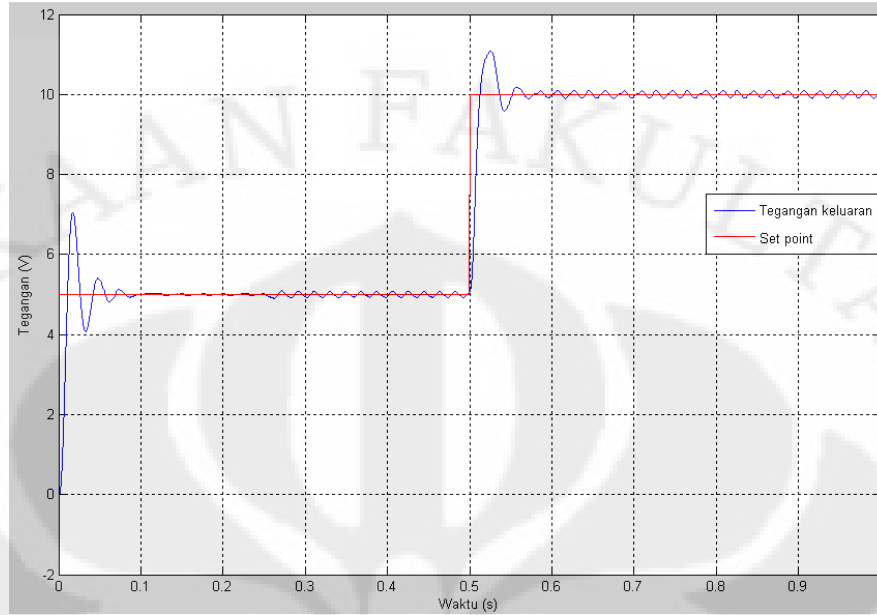
Blok sistem yang digunakan untuk simulasi sistem lingkaran tertutup dengan perubahan *set point* tampak pada gambar 4.12. Pada gambar 4.12 terlihat bahwa blok “PID” diberikan *set point* dengan masukan *step*.



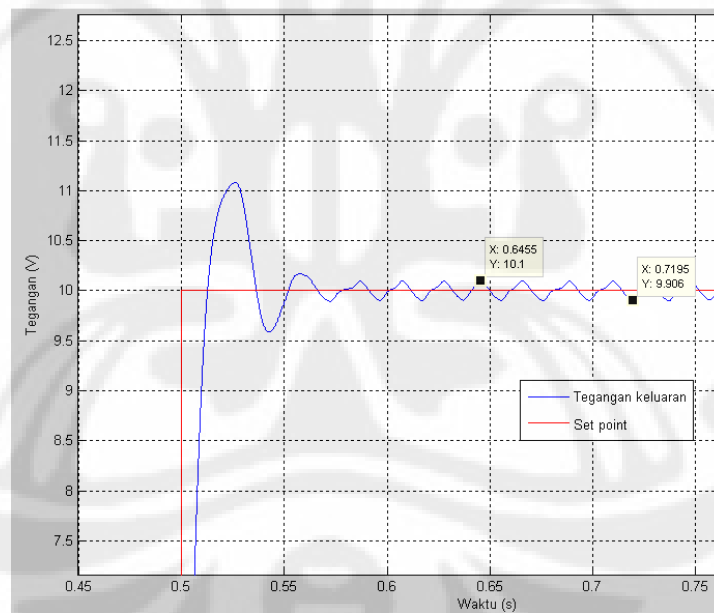
Gambar 4.12 Blok-blok simulasi sistem *buck converter* dengan perubahan *set point*

Hasil simulasi ditunjukkan oleh gambar 4.13. Pada gambar 4.13 diperlihatkan bahwa sistem pada awalnya diberi masukan berupa *set point* sebesar 5 volt dan berubah menjadi 10 volt pada waktu $t = 0.5$ sekon. Hasil keluaran yang didapat ditunjukkan oleh kurva berwarna biru. Terlihat bahwa keluaran sistem dapat mencapai kondisi tunak, dan mengikuti nilai *set point* walaupun terdapat overshoot dan error berupa *ripple*. *Ripple* tegangan yang terjadi adalah sekitar 1.94 %. *Settling time* sistem didapatkan sebesar 32.9 ms.

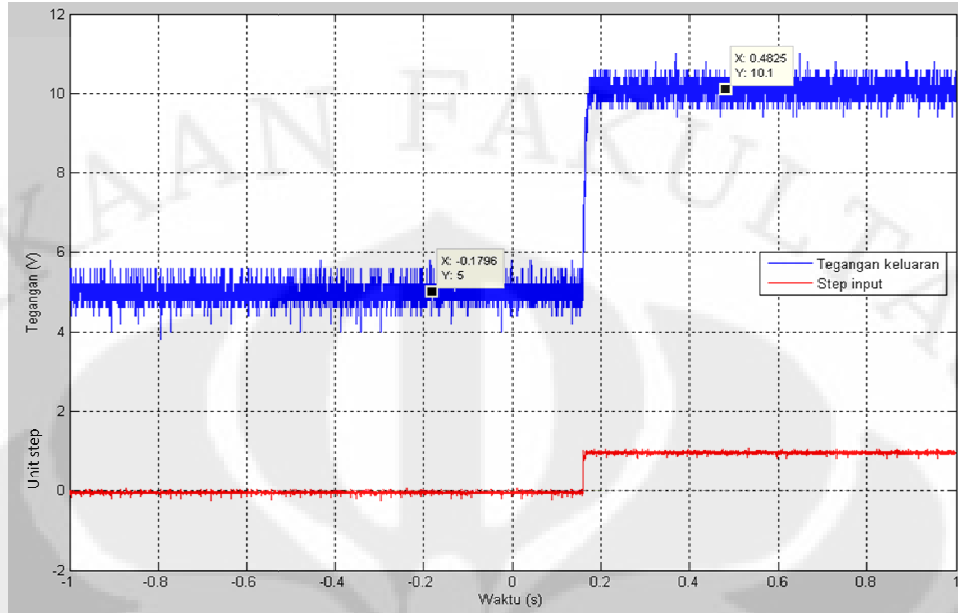
Hasil percobaan alat ditunjukkan oleh gambar 4.15. Pada gambar 4.15, kurva berwarna merah menunjukkan sinyal dari mikrokontroler sebagai titik acuan masukan *step* yang telah diperkecil sebesar lima kali, sedangkan kurva berwarna biru menunjukkan tegangan keluaran pada alat dengan perubahan *set point* dari 5 volt ke 10 volt. Terlihat bahwa keluaran sistem dapat mencapai kondisi tunak dan berada pada sekitar daerah *set point* walaupun terlihat banyak terdapat noise dalam pengukuran. Pada *set point* 5 volt didapatkan tegangan rata-rata sebesar 5 volt dan pada *set point* 10 volt didapatkan tegangan rata-rata 10.1 volt. *Settling time* sistem didapatkan sebesar 13.5 ms.



Gambar 4.13 Respon tegangan terhadap perubahan *set point* dari 5 volt ke 10 volt pada simulasi



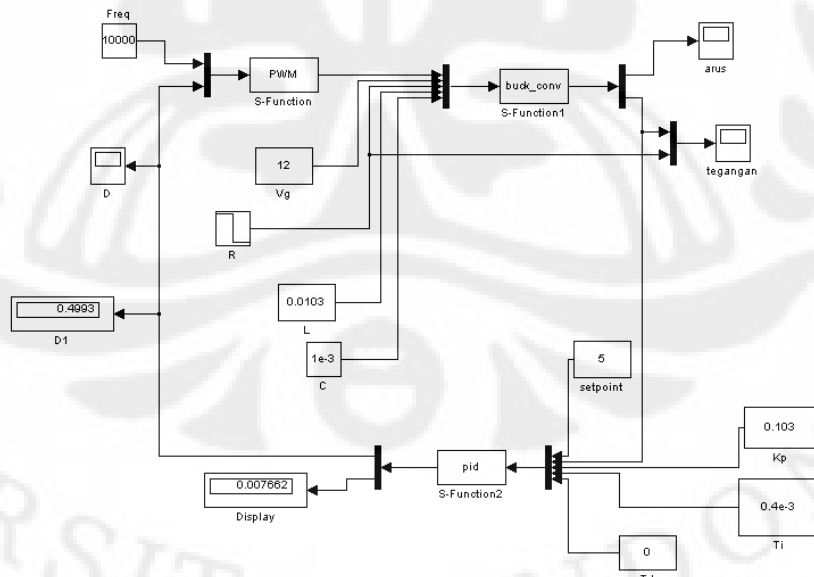
Gambar 4.14 *Ripple* tegangan pada *set point* 10 volt



Gambar 4.15 Respon tegangan terhadap perubahan *set point* dari 5 volt ke 10 volt pada alat

4.3.2 Sistem Lingkaran Tertutup dengan Perubahan Beban

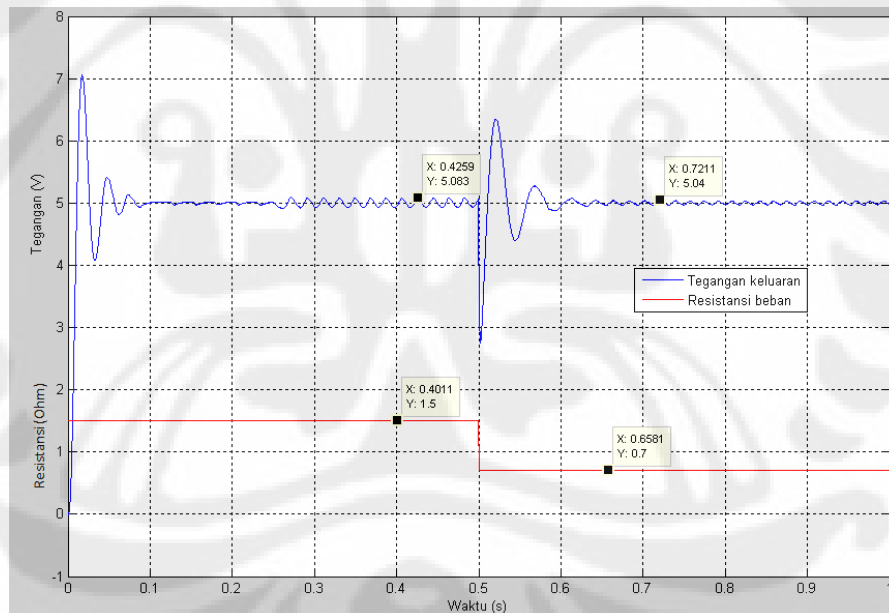
Blok sistem yang digunakan untuk simulasi sistem lingkaran tertutup dengan perubahan beban tampak pada gambar 4.16. Pada gambar 4.16 terlihat bahwa blok “buck_conv” diberikan beban dengan masukan *step*.



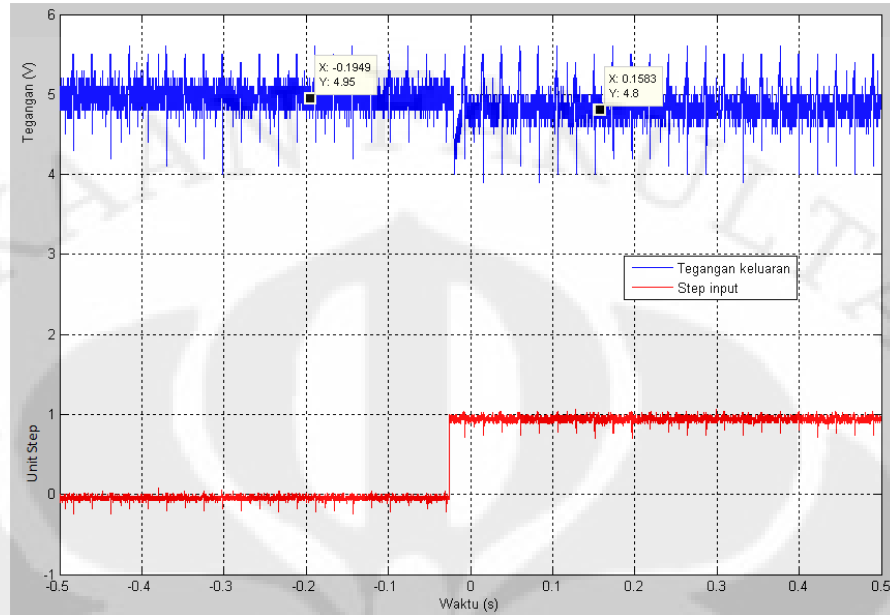
Gambar 4.16 Blok-blok simulasi sistem buck converter dengan perubahan beban

Hasil simulasi ditunjukkan oleh gambar 4.17. Pada gambar 4.17 diperlihatkan bahwa sistem pada awalnya diberi masukan berupa beban sebesar 1.5Ω dan berubah menjadi 0.7Ω pada waktu $t = 0.5$ sekon. Hasil keluaran yang didapat ditunjukkan oleh kurva berwarna biru. Terlihat bahwa keluaran sistem dapat mencapai kondisi tunak, dan tegangan tetap pada nilai *set point* walaupun terdapat *overshoot* dan *error* berupa *ripple*.

Hasil percobaan alat ditunjukkan oleh gambar 4.18. Pada gambar 4.18, kurva berwarna merah menunjukkan sinyal dari mikrokontroler sebagai titik acuan masukan *step* yang telah diperkecil sebesar lima kali, sedangkan kurva berwarna biru menunjukkan tegangan keluaran yang diberikan pada alat dengan perubahan beban dari 1.5Ω ke 0.7Ω . Hasil keluaran menunjukkan bahwa setelah diberikan perubahan beban nilai tegangan rata-rata bergeser dari 4.95 Volt ke 4.8 volt.



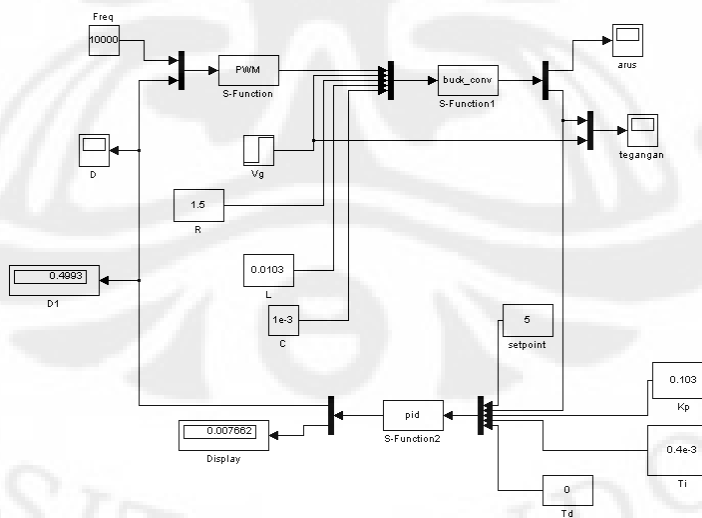
Gambar 4.17 Respon tegangan terhadap perubahan beban dari 1.5Ω ke 0.7Ω pada simulasi



Gambar 4.18 Respon tegangan terhadap perubahan beban dari 1.5Ω ke 0.7Ω pada alat

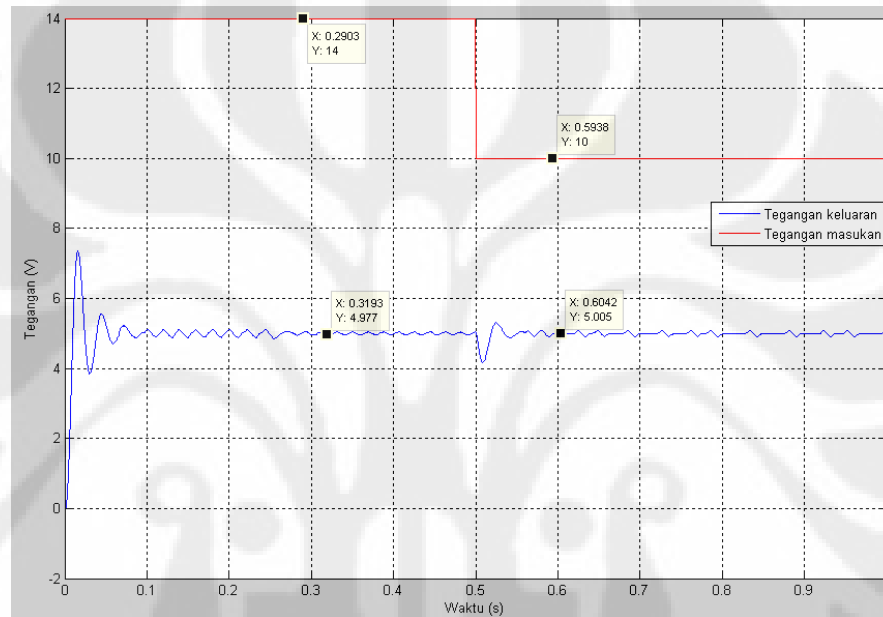
4.3.3 Sistem Lingkaran Tertutup dengan Perubahan Tegangan Masukan

Blok sistem yang digunakan untuk simulasi sistem lingkaran tertutup dengan perubahan tegangan masukan tampak pada gambar 4.19. Pada gambar 4.19 terlihat bahwa blok “buck_conv” diberikan tegangan masukan *step*.



Gambar 4.19 Blok-blok simulasi sistem buck converter dengan perubahan tegangan masukan

Hasil simulasi ditunjukkan oleh gambar 4.20. Pada gambar 4.20 diperlihatkan bahwa sistem pada awalnya diberi masukan berupa tegangan sebesar 14 volt dan berubah menjadi 10 volt pada waktu $t = 0.5$ detik. Hasil keluaran yang didapat ditunjukkan oleh kurva berwarna biru. Terlihat bahwa keluaran sistem dapat mencapai kondisi tunak, dan tegangan tetap pada nilai *set point* dengan sedikit error berupa *ripple* tegangan.



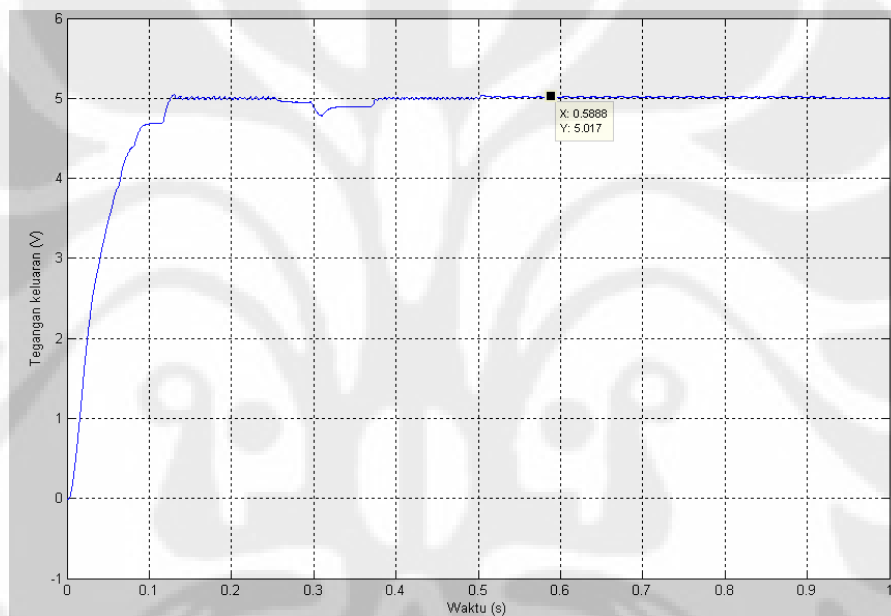
Gambar 4.20 Respon tegangan terhadap perubahan tegangan masukan dari 14 volt ke 10 volt pada simulasi

4.4 Analisa Sistem Lingkaran Tertutup

Pada sistem *buck converter* lingkaran tertutup juga diberikan percobaan perubahan *set point*, beban dan tegangan masukan.

Pada percobaan perubahan *set point* sistem lingkaran tertutup terlihat bahwa tegangan keluaran dapat mengikuti perubahan *set point* yang diberikan baik pada simulasi maupun pada percobaan alat. Tetapi pada simulasi terlihat bahwa respon tegangan keluaran memiliki *overshoot* sedangkan pada percobaan alat tidak didapatkan hal yang sama. Pada simulasi juga didapatkan *settling time* sebesar 32.9 ms sedangkan pada percobaan alat didapatkan *settling time* sebesar 13.5 ms. Hal ini disebabkan pengendali PI dengan metode Ciancone ditala dengan data-data hasil percobaan alat dimana seperti telah dibahas sebelumnya, bahwa alat

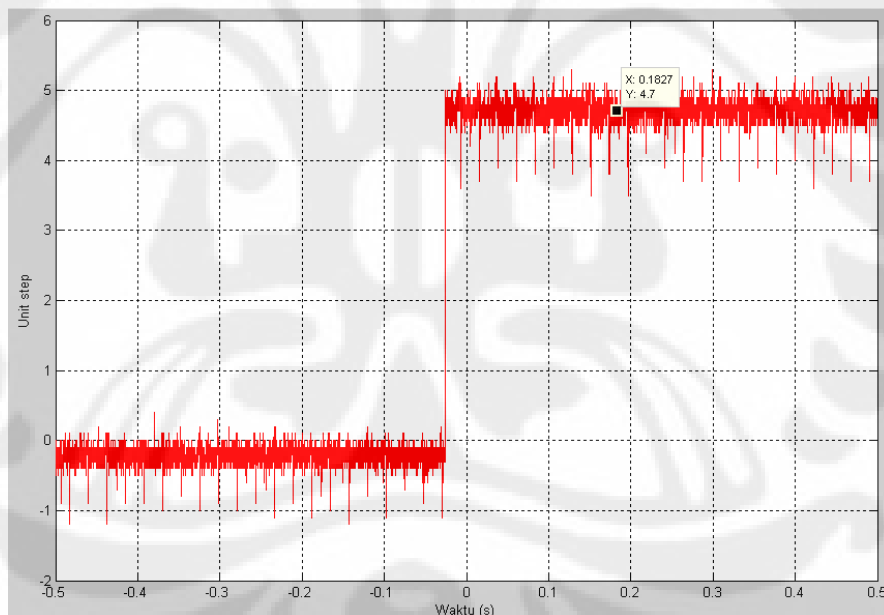
memiliki respon yang lebih cepat. Alat karena memiliki respon yang lebih cepat akan cenderung memiliki konstanta waktu integral T_i yang lebih kecil, sedangkan seharusnya simulasi memiliki konstanta waktu integral yang lebih besar karena responnya yang lebih lambat. Akibatnya pada simulasi konstanta waktu integral membuat *ripple* dan *overshoot* pada tegangan keluaran. Hal ini dibuktikan jika dengan memperbesar konstanta waktu integral 10 kali maka pada simulasi akan didapatkan tegangan keluaran dengan *error steady state* yang cukup baik seperti ditunjukkan oleh gambar 4.21.



Gambar 4.21 Respon tegangan keluaran jika T_i diperbesar 10 kali pada simulasi

Pada percobaan perubahan beban didapatkan bahwa pengendali dapat mengkompensasi perubahan beban seperti ditunjukkan pada simulasi dan percobaan alat. Ketika perubahan beban diberikan terdapat gangguan pada tegangan keluaran yang diperlihatkan dengan *overshoot* yang terjadi pada simulasi. Tetapi kemudian *overshoot* yang terjadi dapat diredam oleh pengendali sehingga tegangan keluaran kembali stabil. Hasil dari percobaan alat pada percobaan perubahan beban juga menunjukkan hal yang serupa yaitu ketika terjadi perubahan beban pada sistem pengendali dapat mengatasi gangguan dengan mengembalikan tegangan mendekati *set point*. Hanya saja pada percobaan

alat terlihat bahwa tegangan rata-rata keluaran tidak kembali ke nilai semula seperti sebelum diberikan gangguan, terdapat sedikit penurunan. Salah satu hal yang mungkin menjadi penyebab utamanya adalah kesalahan pembacaan tegangan keluaran oleh *ADC*. *ADC* membaca tegangan keluaran dan mengkonversikannya dengan cara membandingkannya dengan tegangan referensi dari sistem mikrokontroler yang bernilai sekitar 5 volt. Pada percobaan perubahan beban yang dilakukan, sumber 5 volt yang digunakan sebagai referensi *ADC* ini juga digunakan untuk mengaktifkan relay pengontrol beban. Hal ini dapat mengakibatkan *drop*-nya tegangan referensi sehingga tegangan keluaran yang dibaca *ADC* pun salah. Turunnya tegangan referensi ini dibuktikan oleh rekaman data tegangan keluaran mikrokontroler yang digunakan sebagai sinyal acuan pemberian perubahan beban pada gambar 4.22. Terlihat bahwa sinyal yang diberikan mikrokontroler turun menjadi 4.7 volt.



Gambar 4.22 Sinyal tegangan acuan dari mikrokontroler pada percobaan perubahan beban pada alat

Pada percobaan perubahan tegangan masukan hanya dicoba pada simulasi saja untuk menguji kinerja pengendali terhadap perubahan tegangan masukan. Dari hasil simulasi didapatkan bahwa ternyata pengendali PI yang digunakan dapat pula mengatasi perubahan tegangan masukan dari sistem *buck converter*.

Hal ini terbukti dengan respon tegangan keluaran yang dapat kembali lagi pada tegangan *set point* sistem.

Dari hasil percobaan didapatkan bahwa penalaan pengendali *PID* dengan metode Ciancone ternyata dapat memberikan respon yang cukup baik. Percobaan membuktikan bahwa pengendali *PI* yang didapatkan dari penalaan saat perancangan sudah cukup untuk dapat mengendalikan sistem *buck converter* yang dirancang tanpa harus ditambahkan parameter *derivative*. Hal ini terbukti dari keluaran tegangan sistem lingkaran tertutup yang dapat mengikuti *set point* dan kembali mendekati *set point* ketika terjadi gangguan perubahan beban. Respon sistem yang ditunjukkan juga cukup cepat dan pada percobaan alat tidak terlihat adanya *overshoot*.

Hanya saja penalaan dengan metode Ciancone correlation sangat bergantung pada model *first order with dead time* yang didapatkan. Apabila sistem memiliki nilai parameter yang dinamis, maka kinerja pengendali akan tidak sesuai dengan yang diharapkan.

BAB 5

KESIMPULAN

Dari hasil uji coba sistem dan analisa didapatkan beberapa kesimpulan yaitu :

1. Sistem *buck converter* yang dirancang dapat dimodelkan dengan model *first order with dead time* dan memiliki respon yang menyerupai sistem aslinya.
2. Sistem *buck converter* memiliki kecepatan respon tergantung dari besarnya resistansi beban yang diberikan dan nilai ESR yang ada pada kapasitor.
3. Mikrokontroler ATmega16 dengan nilai kristal sebesar 11.0592 MHz mampu untuk menjalankan perhitungan pengendali *PID* diskrit dengan waktu cuplik yang memenuhi syarat pengendali diskrit.
4. Penalaan dengan metode Ciancone pada sistem *buck converter* yang dirancang hanya dibutuhkan pengendali PI saja.
5. Pengendali PI yang ditala dengan metode Ciancone memberikan *transient response* yang cukup baik dan dapat memperbaiki *steady state error* sistem.
6. Kinerja pengendali *PID* yang ditala dengan metode Ciancone akan berubah apabila terjadi perubahan karakteristik dari sistem yang telah dimodelkan dengan model *first order with dead time*.

DAFTAR ACUAN

- [1] Hart, Daniel W. 1997. "*Introduction to Power Electronics*". Prentice Hall.
- [2] Erickson, Robert W & Maksimovic, Dragan. "*Fundamentals of Power Electronics*". University of Colorado, Boulder.
- [3] Ramirez, Hebertt Sira & Ortigoza, Ramon Silva. "*Control Design Techniques in Power Electronics Devices*". Springer.
- [4] Zhao, Zhenyu. 2008. "*Design and Practical Implementation of Digital Auto-Tuning and Fast-Response Controllers for Low Power Switched-Mode Power Supplies*". University of Toronto.
- [5] Bobal, V. Böhm, J. Fessl, J. and Machacek, J. "*Digital Self Tuning Controllers*". Springer.
- [6] Neary, Eamon. "*Mixed Signal Control Circuits Use Microcontroller for Flexibility in Implementing PID Algorithms*". eamon.neary@analog.com.
- [7] Nise, Norman S. 2004. "*Control System Engineering*". John Wiley and Sons.
- [8] Marlin, Thomas E. "*Process Control*". Mcgraw-hill
- [9] Ogata, K. "*Discrete Time Control Systems*". Prentice Hall
- [10] Subiantoro, Aries. "*Diktat Kuliah Sistem Kendali Adaptif dan Nonlinier*"

DAFTAR PUSTAKA

Hart, Daniel W. 1997. *“Introduction to Power Electronics”*. Prentice Hall.

Erickson, Robert W & Maksimovic, Dragan. *“Fundamentals of Power Electronics”*. University of Colorado, Boulder.

Ramirez, Hebertt Sira & Ortigoza, Ramon Silva. *“Control Design Techniques in Power Electronics Devices”*. Springer.

Zhao, Zhenyu. 2008. *“Design and Practical Implementation of Digital Auto-Tuning and Fast-Response Controllers for Low Power Switched-Mode Power Supplies”*. University of Toronto.

Bobal, V. Böhm, J. Fessler, J. and Machacek, J. *“Digital Self Tuning Controllers”*. Springer.

Neary, Eamon. *“Mixed Signal Control Circuits Use Microcontroller for Flexibility in Implementing PID Algorithms”*. eamon.neary@analog.com.

Nise, Norman S. 2004. *“Control System Engineering”*. John Wiley and Sons.

Marlin, Thomas E. *“Process Control”*. Mcgraw-hill

Ogata, K. *“Discrete Time Control Systems”*. Prentice Hall

Subiantoro, Aries. *“Diktat Kuliah Sistem Kendali Adaptif dan Nonlinier”*

LAMPIRAN A
PROGRAM YANG DIGUNAKAN PADA BLOK SISTEM

Blok "PWM"

```
/*== SOURCE file list of "PWM.c" with Structure A ==*/
#define S_FUNCTION_LEVEL 2
#define S_FUNCTION_NAME PWM
#include "simstruc.h"
#include <math.h>

#define U(element) (*uPtrs[element]) /*Pointer to Input Port0*/

/*
masukan
0:frekuensi
1:DutyCycle

keluaran
0:pwm
*/
static void mdlInitializeSizes(SimStruct *S){
    if (!ssSetNumInputPorts(S, 1)) return;
    ssSetInputPortWidth(S, 0, 2);
    ssSetInputPortDirectFeedThrough(S, 0, 1);
    ssSetInputPortOverWritable(S, 0, 1);
    if (!ssSetNumOutputPorts(S, 1)) return;
    ssSetOutputPortWidth(S, 0, 1);
    ssSetNumSampleTimes(S, 1);

    ssSetOptions(S, SS_OPTION_EXCEPTION_FREE_CODE); }

static void mdlInitializeSampleTimes(SimStruct *S) {
    ssSetSampleTime(S, 0, CONTINUOUS_SAMPLE_TIME);
    ssSetOffsetTime(S, 0, 0.0); }

static void mdlOutputs(SimStruct *S, int_T tid) {
    real_T *Y = ssGetOutputPortRealSignal(S,0);
    InputRealPtrsType uPtrs = ssGetInputPortRealSignalPtrs(S,0);

    real_T t = ssGetT(S);

    if(U(1) >= U(0)*fmod(t,1/U(0)))
    {
        Y[0] = 1;
    }
    else { Y[0] = 0.0; }

}

static void mdlTerminate(SimStruct *S)
{ } /*Keep this function empty since no memory is allocated*/
```

```

#ifdef MATLAB_MEX_FILE
    /* Is this file being compiled as a MEX-file? */
#include "simulink.c" /* MEX-file interface mechanism */
#else
#include "cg_sfun.h" /*Code generation registration function*/
#endif

```

Blok “buck_conv”

```

/* ==SOURCE file list of “IM.c” with Structure C == */
#define S_FUNCTION_LEVEL 2
#define S_FUNCTION_NAME buck_conv
#include "simstruc.h"
#include <math.h>

#define U(element) (*uPtrs[element]) /*Pointer to Input Port0*/
// masukan
// 0:duty cycle
// 1:tegangan sumber
// 2:R
// 3:L
// 4:C

// keluaran
// 0:arus
// 1:tegangan

// state
// 0:arus
// 1:tegangan

static void mdlInitializeSizes(SimStruct *S){
    ssSetNumContStates(S, 2);
    if (!ssSetNumInputPorts(S, 1)) return;
    ssSetInputPortWidth(S, 0, 5);
    ssSetInputPortDirectFeedThrough(S, 0, 1);
    ssSetInputPortOverWritable(S, 0, 1);
    if (!ssSetNumOutputPorts(S, 1)) return;
    ssSetOutputPortWidth(S, 0, 2);
    ssSetNumSampleTimes(S, 1);

    ssSetOptions(S, SS_OPTION_EXCEPTION_FREE_CODE); }

static void mdlInitializeSampleTimes(SimStruct *S) {
    ssSetSampleTime(S, 0, CONTINUOUS_SAMPLE_TIME);
    ssSetOffsetTime(S, 0, 0.0); }

#define MDL_INITIALIZE_CONDITIONS
static void mdlInitializeConditions(SimStruct *S) {

    real_T *X0 = ssGetContStates(S);
    int_T nStates = ssGetNumContStates(S);
    int_T i;

```

```

/* initialize the states to 0.0 */
for (i=0; i < nStates; i++) {X0[i] = 0.0;} }

static void mdlOutputs(SimStruct *S, int_T tid) {
    real_T *Y = ssGetOutputPortRealSignal(S,0);
    real_T *X = ssGetContStates(S);
    InputRealPtrsType uPtrs = ssGetInputPortRealSignalPtrs(S,0);

    Y[0] = X[0];
    Y[1] = 0.019688*X[0] + 0.987*X[1];

}

#define MDL_DERIVATIVES
static void mdlDerivatives(SimStruct *S) {
    real_T *dX = ssGetdX(S);
    real_T *X = ssGetContStates(S);
    InputRealPtrsType uPtrs = ssGetInputPortRealSignalPtrs(S,0);
    real_T i_out_dot = 0;
    real_T v_out_dot = 0;
    int_T PWM;
    real_T V_s;
    real_T R;
    real_T L;
    real_T C;
    real_T R_on = 0.117;
    real_T R_c = 0.01995;
    real_T v_d = 0.62;

    PWM = U(0);
    V_s = U(1);
    R = U(2);
    L = U(3);
    C = U(4);

    if(PWM==1)
    {
        // Untuk switch closed
        i_out_dot = (-X[0]*(R*R_on + R_c*R_on + R*R_c)/(L*R + L*R_c))-
        (X[1]*R/(L*R + L*R_c)) + (V_s/L);
        v_out_dot = (X[0]*R/(R*C + R_c*C ))-(X[1]/(R*C + R_c*C));
    }
    else if(PWM==0)
    {
        // Untuk switch open
        i_out_dot = (-X[0]*(R*R_c)/(L*R + L*R_c))-X[1]*R/(L*R + L*R_c)) -
        (v_d/L);
        v_out_dot = (X[0]*R/(R*C + R_c*C ))-(X[1]/(R*C + R_c*C));
    }

    dX[0] = i_out_dot;
    dX[1] = v_out_dot;
}

```



```

static void mdlTerminate(SimStruct *S)
{} /*Keep this function empty since no memory is allocated*/

#ifdef MATLAB_MEX_FILE
/* Is this file being compiled as a MEX-file? */
#include "simulink.c" /* MEX-file interface mechanism */
#else
#include "cg_sfun.h" /*Code generation registration function*/
#endif

```

Blok ‘PID’

```

/* == SOURCE file list of "SPD_CTL.c" with Structure B == */
#define S_FUNCTION_LEVEL 2
#define S_FUNCTION_NAME pid
#include "simstruc.h"
#include <math.h>

#define U(element) (*uPtrs[element]) /*Pointer to Input Port0*/

/*
masukan
0:Set point
1:vout
2:Kp
3:Ti
4:Td

keluaran
0:DutyCycle
1:error

state
0:D
1:I_term
2:error
*/

static void mdlInitializeSizes(SimStruct *S)
{
    ssSetNumDiscStates(S, 3);
    if (!ssSetNumInputPorts(S, 1)) return;
    ssSetInputPortWidth(S, 0, 5);
    ssSetInputPortDirectFeedThrough(S, 0, 1);
    ssSetInputPortOverWritable(S, 0, 1);
    if (!ssSetNumOutputPorts(S, 1)) return;
    ssSetOutputPortWidth(S, 0, 2);
    ssSetNumSampleTimes(S, 1);

    ssSetOptions(S, (SS_OPTION_EXCEPTION_FREE_CODE
        | SS_OPTION_DISCRETE_VALUED_OUTPUT));
}

```

```

static void mdlInitializeSampleTimes(SimStruct *S)
{
    ssSetSampleTime(S, 0, 190e-6);
    ssSetOffsetTime(S, 0, 0.0);
}

#define MDL_INITIALIZE_CONDITIONS
static void mdlInitializeConditions(SimStruct *S)
{
    real_T *X0 = ssGetRealDiscStates(S);
    int_T nXStates = ssGetNumDiscStates(S);
    InputRealPtrsType uPtrs = ssGetInputPortRealSignalPtrs(S,0);
    int_T i;

    /* initialize the states to 0.0 */
    for (i=0; i < nXStates; i++)
    {
        X0[i] = 0.0;
    }
}

static void mdlOutputs(SimStruct *S, int_T tid)
{
    real_T *Y = ssGetOutputPortRealSignal(S,0);
    real_T *X = ssGetRealDiscStates(S);

    Y[0] = X[0];
    Y[1] = X[2];
}

#define MDL_UPDATE
static void mdlUpdate(SimStruct *S, int_T tid)
{
    real_T *X = ssGetRealDiscStates(S);
    InputRealPtrsType uPtrs = ssGetInputPortRealSignalPtrs(S,0);

    real_T dt = 190e-6;
    real_T analog,Ti, Kp, Td, D, error, I_term,I_term_old,
    error_old,V_s, read_adc_0,set_point;

    Kp = U(2); Ti = U(3); Td = U(4);read_adc_0 = U(1); set_point =
U(0);

    I_term = X[1],error_old = X[2];

    error = set_point-read_adc_0;
    if((error <= 0.0001)&&(error >= -0.0001)){error = 0;}

    I_term = I_term + Kp*(dt/Ti)*error;

    if(I_term > 32767){I_term = 32767;}
    if(I_term < -32768){I_term = -32768;}

    analog = Kp*error + I_term + Kp*(Td/dt)*(error - error_old);

    D = analog/12;

```

```
    if(D >= 1){D = 1;}
    if(D <= 0){D = 0;}

    X[0] = D;
    X[1] = I_term;
    X[2] = error;
}

static void mdlTerminate(SimStruct *S)
{ } /*Keep this function empty since no memory is allocated*/

#ifdef MATLAB_MEX_FILE
    /* Is this file being compiled as a MEX-file? */
#include "simulink.c"      /*MEX-file interface mechanism*/
#else
#include "cg_sfun.h"      /*Code generation registration function*/
#endif
```

LAMPIRAN B

PROGRAM YANG DIGUNAKAN PADA MIKROKONTROLER

```
#include <mega16.h>
#include <delay.h>
#include <stdlib.h>
#define ADC_VREF_TYPE 0x40
#asm
    .equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.h>
#define relay PORTD.0
#define step PORTD.1

int count =0;
bit tanda = 0;
float analog, D, error,V_s;
float dt = 190e-6;
float vdes = 6;
float Kp = 1.1375;
float Ti = 0.4788;
float Td = 0.007;
float I_term = 0;float error_old = 0;

unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    delay_us(10);
    ADCSRA|=0x40;
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCW;
}

interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
    TCNT0=6;
    count++;
    if(count >= 200)
    {
        tanda = ~tanda;
        count = 0;
    }
}

void init()
{
    PORTA=0x00;
    DDRA=0x00;

    PORTB=0xFF;
    DDRB=0x00;

    PORTC=0x00;
    DDRC=0x00;

    PORTD=0x00;
```

```

DDRD=0xFF;

TCCR1A=0xA2;
TCCR1B=0x19;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x04;    // TOP 1100
ICR1L=0x4C;    //
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

lcd_init(16);

    TCCR0=0x05;
    TCNT0=0x00;
    OCR0=0x00;
    TIMSK=0x01;
}

void my_pid_disturbance()
{
    #asm("sei")
    while(1)
    {
        while(!tanda)
        {
            relay = 0; step = 0;
            error = 5-(0.01175*read_adc(0));
            if((error <= 0.0001)&&(error >= -0.0001)){error = 0;}

            I_term = I_term + 0.048925*error;

            if(I_term > 32767){I_term = 32767;}
            if(I_term < -32768){I_term = -32768;}

            analog = 0.103*error + I_term; // kd = 24.128

            D = 91.667*analog;

            if(D >= 1100){D = 1100;}
            if(D <= 0){D = 0;}

            OCR1B = D;
        }
        while(tanda)
        {
            relay = 1; step = 1;
            error = 5-(0.01175*read_adc(0));
            if((error <= 0.0001)&&(error >= -0.0001)){error = 0;}

            I_term = I_term + 0.048925*error;

```

```

if(I_term > 32767){I_term = 32767;}
if(I_term < -32768){I_term = -32768;}

analog = 0.103*error + I_term; // kd = 24.128

D = 91.667*analog;

if(D >= 1100){D = 1100;}
if(D <= 0){D = 0;}

OCR1B = D;
}
}
}
void my_pid_setpoint()
{
  #asm("sei")
  while(1)
  {
    while(!tanda)
    {
      step = 0;
      error = 5-(0.01175*read_adc(0));
      if((error <= 0.0001)&&(error >= -0.0001)){error = 0;}

      I_term = I_term + 0.048925*error;

      if(I_term > 32767){I_term = 32767;}
      if(I_term < -32768){I_term = -32768;}

      analog = 0.103*error + I_term; // kd = 24.128

      D = 91.667*analog;

      if(D >= 1100){D = 1100;}
      if(D <= 0){D = 0;}

      OCR1B = D;
    }
    while(tanda)
    {
      step = 1;
      error = 10-(0.01175*read_adc(0));
      if((error <= 0.0001)&&(error >= -0.0001)){error = 0;}

      I_term = I_term + 0.048925*error;
      if(I_term > 32767){I_term = 32767;}
      if(I_term < -32768){I_term = -32768;}

      analog = 0.103*error + I_term; // kd = 24.128

      D = 91.667*analog;

      if(D >= 1100){D = 1100;}
      if(D <= 0){D = 0;}

      OCR1B = D;
    }
  }
}

```

```
    }  
  
    }  
    void main()  
    {  
        init();  
        if(PINB.1)  
        {  
            while(PINB.0);  
            my_pid_disturbance();  
        }  
        if(!PINB.1)  
        {  
            while(PINB.0);  
            my_pid_setpoint();  
        }  
    }  
}
```

