

# Rancang Bangun *Buck Converter* dengan Kontrol PID Berbasis Mikrokontroler Arduino

Nurul Fuada\*)<sup>1</sup>, Irma Husnaini<sup>2</sup>

<sup>1,2</sup>Teknik Elektro Industri/ Teknik Elektro/ Fakultas Teknik/ Universitas Negeri Padang

\*)Corresponding author, [nurulfu193@gmail.com](mailto:nurulfu193@gmail.com)

## Abstrak

*Buck converter* adalah salah satu jenis *converter* DC-DC yang paling umum digunakan untuk mengubah tegangan listrik DC yang tinggi menjadi tegangan DC yang lebih rendah. *Buck converter* dikonfigurasi dengan menggunakan MOSFET sebagai saklar daya yang dikendalikan oleh Mikrokontroler Arduino. Saat merancang sistem *buck converter*, stabilitas tegangan yang dihasilkan harus diperhatikan. Ketidakstabilan dalam rangkaian *buck converter* mengakibatkan terjadinya *overvoltage* (tegangan berlebih) dan *undervoltage* (tegangan kurang). Dengan menggunakan kontrol PID kestabilan tegangan tetap terjaga dan tegangan keluaran dapat diatur sesuai dengan nilai *set point* yang diinginkan. Spesifikasi rancangan *Buck Converter* bekerja pada frekuensi sebesar 31372,55 Hz, tegangan *input* 48 volt dan *setpoint* tegangan *output* sebesar 36 volt diujikan dengan beban motor DC 36 volt. Metode digunakan untuk penelitian kali ini yaitu metode eksperimen. Implementasi *hardware Buck Converter* dengan kontrol PID menggunakan Arduino uno dengan *compiler* pemrograman yang digunakan adalah "*Simulink Support Arduino Package*", yaitu fasilitas *library* Arduino yang tersedia di *Simulink* MATLAB. Fasilitas tersebut digunakan sebagai konfigurasi dan komunikasi pemrograman dengan mikrokontroler Arduino. Hasil pengujian *Buck Converter* menggunakan kontrol PI dengan nilai  $K_p=3,2759$  dan  $K_i=0,000079$  diperoleh nilai *rise time* 0,26 detik, *settling time* 1,2 detik, *overshoot* 1,09%. Sedangkan pengujian menggunakan kontrol PID dengan nilai  $K_p=3,2759$ ,  $K_i=0,000095$  dan  $K_d=0,435825$  menghasilkan *rise time* 0,22 detik, *settling time* 0,5 detik, *overshoot* 0,886. Dengan mempertimbangkan nilai respons sistem seperti *rise time*, *settling time* dan *overshoot*, kontrol yang paling sesuai untuk penelitian ini adalah kontrol PID.

## Abstract

*Buck converters* are one type of DC-DC converter that is most commonly used to convert high DC electrical voltages into lower DC voltages. The buck converter is configured using a MOSFET as a power switch controlled by an Arduino microcontroller. When designing a buck converter system, the stability of the resulting voltage must be considered. Instability in the buck converter circuit results in overvoltage (excess voltage) and undervoltage (undervoltage). By using PID control, voltage stability is maintained and the output voltage can be adjusted according to the desired set point value. Buck Converter design specifications work at a frequency of 31372,55 Hz, input voltage of 48 volts and output voltage set point of 36 volts tested with a 36 volt DC motor load. The method used for this research is the experimental method. Buck Converter hardware implementation with PID control using Arduino Uno with the programming compiler used is "*Simulink Support Arduino Package*", namely the Arduino library facility available in *Simulink* MATLAB. This facility is used for configuration and programming communication with the Arduino microcontroller. Buck Converter test results using PI control with values of  $K_p=3,2759$  and  $K_i=0,000079$  showed rise time values of 0,26 seconds, settling time 1,2 seconds, overshoot 1,09%. Meanwhile, testing using PID control with values  $K_p=3,2759$ ,  $K_i=0,000095$  and  $K_d=0,435825$  produces a rise time of 0,22 seconds, settling time 0,5 seconds, overshoot 0,886. By considering system response values such as rise time, settling time and overshoot, the most suitable control for this research is PID control.

## INFO.

### Info. Artikel:

No. 513

Received. September, 15, 2023

Revised. October, 12, 2023

Accepted. October, 17, 2023

Page. 781 – 791

### Kata kunci:

- ✓ *Buck Converter*
- ✓ *Kontrol PID*
- ✓ *Mikrokontroler Arduino*
- ✓ *Setpoint*
- ✓ *Matlab*

## PENDAHULUAN

Perkembangan di bidang teknologi dan sistem kendali berlangsung dengan cepat. Peralatan dan perlengkapan elektronik telah menjadi bagian tidak terpisahkan dari kehidupan sehari-hari[1]. Peralatan elektronik saat ini merupakan aplikasi dari *converter* DC ke DC[2]. *Converter* DC ke DC adalah suatu rangkaian elektronik yang menggunakan saklar elektronik untuk mengubah tegangan searah menjadi tegangan searah dengan tegangan dan frekuensi yang dapat diatur. Pengaturan tegangan dapat dilakukan di luar *converter* atau di dalam *converter*. Jika dilakukan di luar *converter*, tegangan searah masukan *converter* dapat diatur untuk menghasilkan variasi tegangan keluaran yang diinginkan. Namun jika pengaturan tegangan dilakukan di dalam *converter*, metode yang digunakan adalah modulasi lebar pulsa (*Pulse Width Modulation*)[3].

*Buck converter* merupakan salah satu jenis *converter* DC ke DC yang berfungsi untuk menurunkan tegangan DC menjadi tegangan DC lain sesuai dengan kebutuhan. Dengan mengatur frekuensi dan lebar pulsa (*duty cycle*), tegangan keluaran *buck converter* dapat disesuaikan agar lebih rendah daripada tegangan masukan. *Duty cycle* dihasilkan melalui penggunaan sinyal PWM. *Buck converter* yang sederhana terdiri dari beberapa komponen termasuk Mosfet sebagai saklar aktif, dioda sebagai saklar pasif serta kapasitor dan induktor yang berfungsi sebagai filter pada keluarannya[4].

Saat merancang sistem *buck converter*, stabilitas tegangan yang dihasilkan harus diperhatikan. Ketidakstabilan dalam rangkaian *buck converter* mengakibatkan terjadinya *overvoltage* (tegangan berlebih) dan *undervoltage* (tegangan kurang). Penyebab utama ketidakstabilan adalah perubahan beban yang tidak terduga pada *converter*. Jika beban berubah secara tiba-tiba, *converter* tidak mampu menyesuaikan keluaran tegangan dengan cepat. Hal ini dapat menyebabkan terjadinya *overvoltage* atau *undervoltage* dimana tegangan keluaran melebihi atau kurang dari batas yang diinginkan. Selain itu faktor desain yang tidak optimal atau komponen yang cacat juga dapat menyebabkan ketidakstabilan dalam *buck converter*. Pemilihan komponen yang tidak sesuai atau kerusakan pada elemen kunci seperti induktor atau kapasitor dapat mengganggu operasi normal *converter* dan menghasilkan tegangan keluaran yang tidak stabil[5].

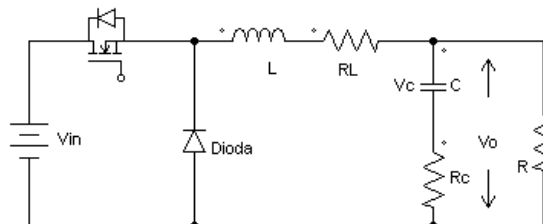
Kontrol PI dan PID sering digunakan untuk pengontrol *converter* DC-DC karena praktis, mudah dalam perancangan dan penerapannya pada sistem *converter*. Cara kerja kontrol PID dengan menghitung nilai error yang merupakan selisih antara nilai keluaran dan nilai referensi (*setpoint*) kemudian PID akan memperbaiki nilai masukan sehingga nilai error bisa berkurang[6].

Menurut penelitian yang sudah dilakukan Ari Anggawan yang berjudul "Implementasi Kendali Tegangan *Output Buck Converter* Berbasis *Simulink* Matlab". Penelitian ini masih ada kekurangan karena tidak memiliki pengendali pada tegangan keluaran sehingga dapat digunakan pada beban yang berbeda-beda[7]. Penelitian selanjutnya oleh Rahmanda Fadri dengan judul "*Buck Converter* dengan Kontrol PI dan PID menggunakan Mikrokontroler Arduino". Penelitian ini tidak menggunakan beban nyata sehingga pengaplikasian kontrol ini belum terbukti apakah sesuai dengan kriteria yang diharapkan[8]. Kemudian pada penelitian Eko Avin Wibowo dengan judul "Desain *Buck Converter* sebagai *Charging* Baterai pada Sistem PLTS". Di penelitian ini tegangan sesuai *setpoint*, meredam *osilasi* tegangan keluaran serta mengurangi *overshoot*[9].

Berdasarkan hal tersebut maka penulis merancang tugas akhir yang berjudul "Rancang Bangun *Buck Converter* dengan Kontrol PID Berbasis Mikrokontroler Arduino". Berbeda dari penelitian sebelumnya pada penelitian ini terdapat data monitoring yaitu grafik tegangan masukan dan tegangan keluaran yang ditampilkan pada Matlab *Simulink*. Dimana tegangan masukan 48 VDC diturunkan menjadi keluaran konstan 36 VDC. Serta menggunakan beban berupa motor DC 36 volt.

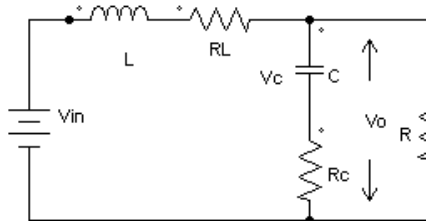
## BUCK CONVERTER

*Buck converter* adalah *converter* yang menghasilkan tegangan keluaran yang lebih kecil dari tegangan masukannya. Tegangan keluaran yang dihasilkan mempunyai polaritas yang sama dengan tegangan masukannya. *Buck converter* biasa disebut juga sebagai *step-down converter*. Berikut merupakan rangkaian dari *buck converter*[9].



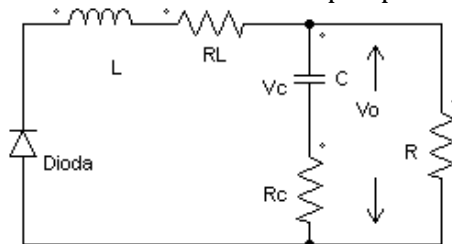
Gambar 1 Rangkaian buck converter

Buck converter dapat dioperasikan dalam dua mode yaitu *continuous conduction mode* (CCM) dan *discontinuous conduction mode* (DCM). Ketika semua komponen diasumsikan ideal dan buck converter beroperasi dalam kondisi CCM, prinsip kerja buck converter dapat dibagi.



Gambar 2 Mode saklar tertutup

Ketika Saklar dalam posisi aktif (tertutup) dan dioda mati, arus mengalir dari sumber ke induktor untuk mengisi induktor. Kemudian melewati kapasitor dan akhirnya mencapai beban sebelum kembali lagi ke sumber. Kondisi saklar tertutup dapat dilihat pada gambar 2.



Gambar 3 Mode saklar terbuka

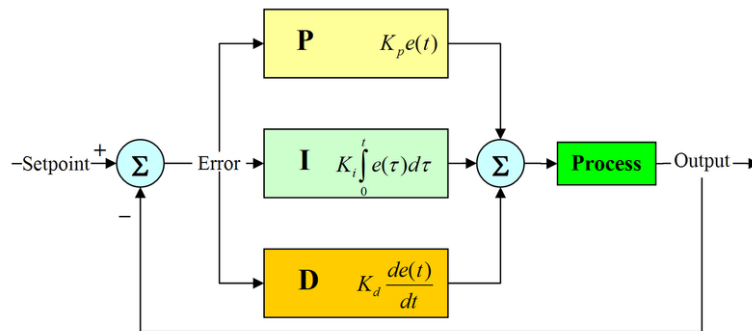
Dalam Mode 2, dioda dalam kondisi *forward bias* seperti yang ditunjukkan dalam Gambar 3. Ketika Saklar dalam keadaan non-aktif (terbuka) dan dioda aktif, arus yang telah disimpan dalam induktor dilepaskan ke beban. Kemudian mengalir melalui dioda *freewheeling*, sebelum kembali lagi ke induktor[5].

Persamaan fungsi alih buck converter untuk perbandingan tegangan keluaran terhadap tegangan masukan dinamis ditulis dalam bentuk.

$$\frac{\hat{v}_o(s)}{d(s)} = \frac{R(1+sR_c C)}{LCs^2(R+R_c)+s(L+RCR_c)+R} \times V_{in} \tag{10}$$

**PENALAN KONTROL PID**

PID (*Proportional Integral Derivative*) merupakan kontroler yang berfungsi untuk menentukan presisi suatu sistem instrumentasi dengan karakteristiknya yaitu memiliki umpan balik pada sistem[11].



Gambar 4 Blok diagram kontrol PID

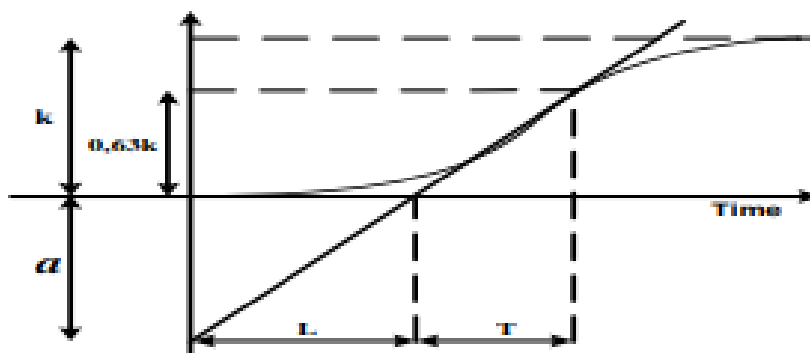
Berdasarkan Gambar 4, variabel yang dapat diatur nilainya disebut "*Manipulated Variable*" (MV) dan setara dengan keluaran dari pengendali  $u(t)$ . Keluaran dari kontrol PID akan mempengaruhi respons terhadap perubahan yang terdeteksi oleh sensor dan *setpoint* yang telah ditentukan. Pada waktu kontinu, keluaran pengendali PID dapat dijabarkan dalam Persamaan.

$$u(t) = K \left( e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right)$$

Dimana:

- $u(t)$  = Sinyal keluaran kontrol PID
- $K_p$  = Konstanta *proportional*
- $K_i$  = Konstanta *integral*
- $K_d$  = Konstanta *derivative*
- $e$  = Sinyal kesalahan

Penalaan kontrol PID bertujuan untuk mendapatkan nilai  $K_p$ ,  $K_i$  dan  $K_d$  sehingga menghasilkan respon yang stabil dan sesuai dengan *setpoint* yang diinginkan. Ada beberapa metode penalaan yang dapat digunakan salah satunya adalah metode Ziegler-Nichols 1. Metode ini didasarkan pada respon *plant* terhadap masukan tangga dalam sistem terbuka. *Plant* yang tidak mempunyai *integrator*, menghasilkan kurva tanggapan terhadap masukan tangga seperti kurva huruf S pada Gambar 5. Kurva tanggapan *plant* digunakan untuk mencari waktu tunda  $L$  dan konstanta waktu  $T$ .



Gambar 5 Kurva tanggapan berbentuk s

Parameter didapat dengan menggambar garis tangensial pada titik infleksi kurva S. Garis tangensial akan berpotongan dengan garis *time axis* dan garis  $c(t) = K$ . Rumus-rumus untuk parameter pengontrol dapat dilihat pada Tabel 1 di bawah ini[12].

**Tabel 1 Penalaan metode Ziegler Nichols 1**

Pengontrol	Kp	Ti	Td
P	$\frac{T}{L}$	$\infty$	0
PI	$0.9 \frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2 \frac{T}{L}$	2L	0.5L

Pada dasarnya, penalaan PID menggunakan metode Ziegler-Nichols 1 menghasilkan respons keluaran yang stabil. Namun, terkadang hasilnya tidak sesuai dengan perencanaan awal sistem. Oleh karena itu, diperlukan penyesuaian parameter PID untuk mengatasi kekurangan dan kelebihan masing-masing pengontrol agar respons keluaran sesuai dengan perencanaan. Efek dari setiap pengontrol *proportional*, *integral*, dan *derivative* pada sistem dapat dilihat dalam Tabel 2.

**Tabel 2 Karakteristik pengontrol P, I dan D**

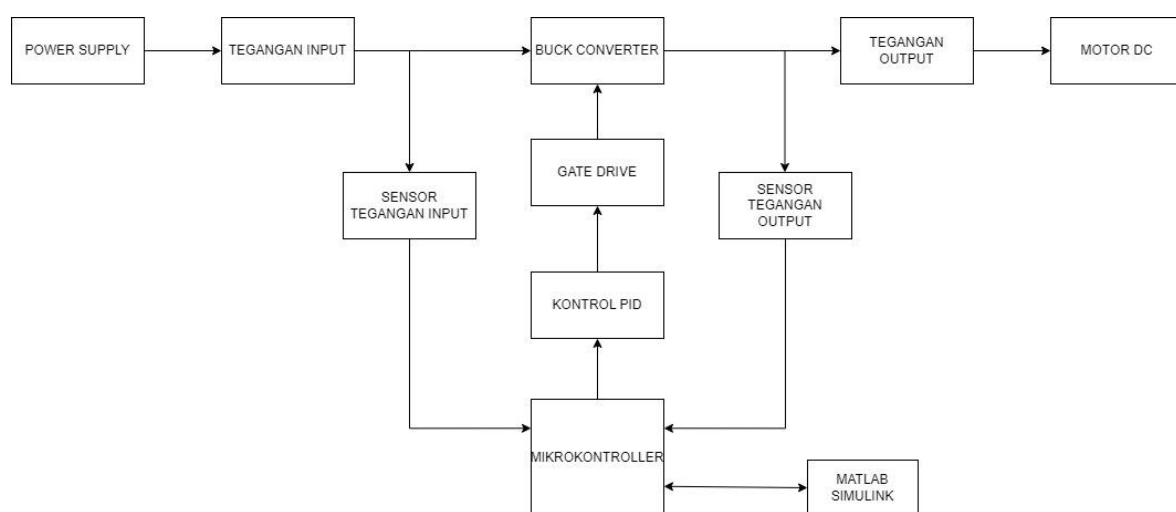
Pengontrol	Rise Time	Overshoot	Settling Time	Steady State Error
P	Menurunkan	Meningkatkan	Perubahan Kecil	Menurunkan
I	Menurunkan	Meningkatkan	Meningkatkan	Menghilangkan
D	Perubahan Kecil	Menurunkan	Menurunkan	Perubahan Kecil

**METODE PENELITIAN**

Pada penelitian ini menggunakan metode eksperimen yang merancang sebuah *buck converter* dengan tegangan *input* 48 volt dan tegangan *output* 36 volt. Pengontrol tegangan *output buck converter* menggunakan kontrol PID yang diaplikasikan dengan Arduino kemudian diprogram melalui Matlab *Simulink*. Rancangan *buck converter* diuraikan menggunakan blok diagram.

**BLOK DIAGRAM**

Berikut ini adalah blok diagram pengaturan *switching buck converter* pada tugas akhir ini.



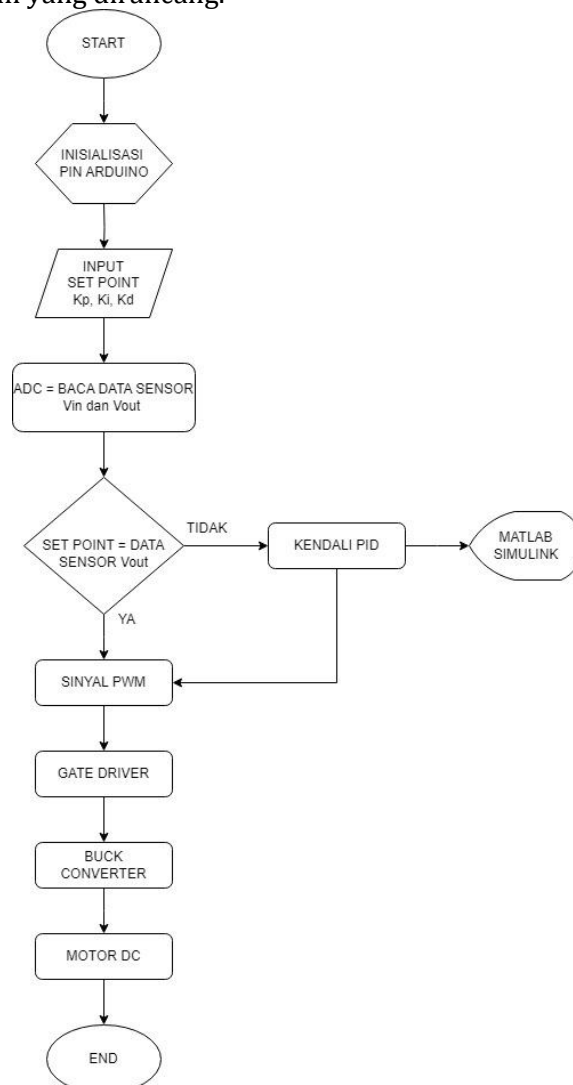
**Gambar 6 Blok diagram buck converter**

Blok diagram pada gambar 6 dapat dilihat bahwa *buck converter* merupakan konverter DC-DC yang berfungsi sebagai penurun tegangan DC. Masukan dari *buck converter* merupakan keluaran dari *rectifier* sebesar 48 VDC. Tegangan masukan diperoleh dari sumber *power supply* DC.

Tegangan ini akan diturunkan menjadi 36 VDC. Rangkaian pembagi tegangan sebagai modul sensor tegangan digunakan untuk mendeteksi tegangan *input* dan tegangan *output* pada rangkaian *buck converter*[13]. *Gate drive* disini berfungsi sebagai *driver* untuk MOSFET yang digunakan pada rangkaian *buck converter*. Mikrokontroler merupakan otak dari pengontrolan tegangan keluaran yang dihasilkan oleh *buck converter*. Mikrokontroler inilah yang akan diprogram agar menghasilkan keluaran sinyal pulsa dan sensor tegangan[14]. Dengan kata lain mikrokontroler inilah yang akan dijadikan sebagai PWM untuk pengaturan keluaran dari *buck converter*[10]. *Simulink* merupakan sebuah *software* di dalam Matlab yang berfungsi sebagai pemodelan, melakukan simulasi dan analisa terhadap sistem yang dinamis[15]. Motor DC digunakan sebagai beban dari *buck converter* dengan tegangan 36 volt, 3,2 A dan 100 watt.

### FLOWCHART

Setelah blok diagram, pembuatan *flowchart* sistem diperlukan agar mudah memahami langkah kerja dari sistem yang dirancang.



Gambar 7 Flowchart buck converter kontrol PID

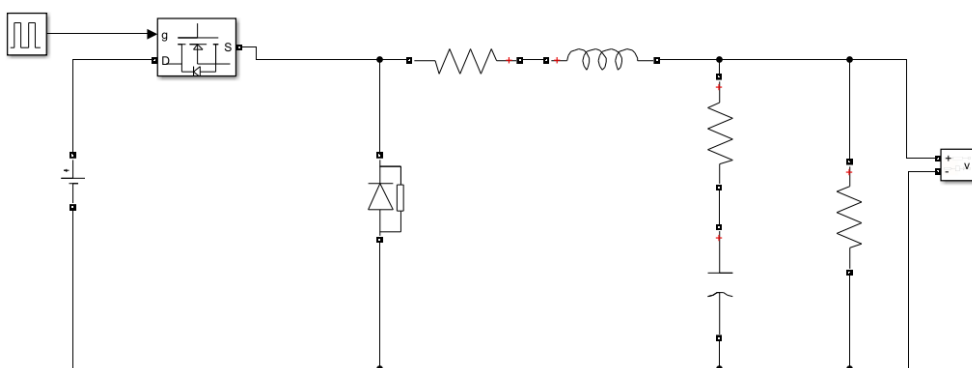
Dalam gambar 7, saat sistem dijalankan terjadi inisialisasi pin dan dilanjutkan dengan memasukkan *setpoint* tegangan *buck converter* serta parameter PID. Kemudian sensor tegangan akan membaca data tegangan. mikrokontroler akan membandingkan *setpoint* yang telah dimasukkan dengan besarnya tegangan keluaran pada *buck converter*. Keluaran dari *buck*

converter masuk ke motor DC. Jika tegangan sesuai dengan *setpoint* maka data tegangan akan ditampilkan pada Matlab *Simulink*. Jika tegangan tidak sesuai dengan *setpoint* maka sistem kontrol PID akan memproses data yang telah dikirim ke sensor untuk mendapatkan tegangan keluaran *buck converter* sesuai dengan *setpoint*.

**PERANCANGAN BUCK CONVERTER**

Dalam rangkaian *buck converter* terdapat beberapa komponen seperti MOSFET, induktor, kapasitor, diode dan beban yang mana dapat dilihat pada gambar 8. Simulasi ini bertujuan untuk menguji dan mengetahui kinerja dari rangkaian *buck converter* dengan kontrol PID sebelum dibuat dalam bentuk *hardware*. *Buck converter* beroperasi berdasarkan prinsip pengisian dan pengosongan kedua komponen induktor dan kapasitor dalam rangkaian ini. Proses pengisian dan pengosongan kedua komponen ini diatur oleh waktu yang disebut waktu *switching* dari saklar daya. Waktu *switching* ini bergantung pada tingkat *duty cycle*. Hubungan antara tegangan keuaran dari *buck converter* dan *duty cycle* dijelaskan dengan rumus berikut.

$$V_{out} = V_{in} * D$$



Gambar 8 Rangkaian *buck converter*

Tabel 3 Spesifikasi perancangan *buck converter*

Tegangan <i>Input</i>	48 volt
Tegangan <i>Output</i>	36 volt
Mosfet	IRFZ44N
Diode	Schotky SR460
Induktor	3 mH
Kapasitor	1000 $\mu F$
Motor DC	36 volt
Tahanan Dalam Induktor	0,06 $\Omega$
Tahanan Dalam Kapasitor	3,34 $\Omega$

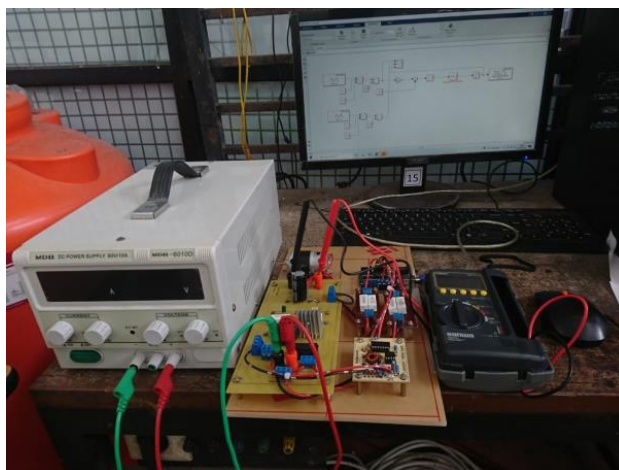
**HASIL DAN PEMBAHASAN**

Dalam pengujian ini, *buck converter* dihubungkan dengan beban motor. Untuk melihat grafik tegangan *input*, *pwm* Arduino dan *output* digunakan blok *scope* yang ada pada Matlab *Simulink*.

**RANGKAIAN KESELURUHAN**

Rangkaian keseluruhan *buck converter* membutuhkan peralatan pendukung seperti *power supply*, multimeter dan sebuah PC yang dilengkapi *software* Matlab *Simulink* yang digunakan untuk memprogram Arduino.

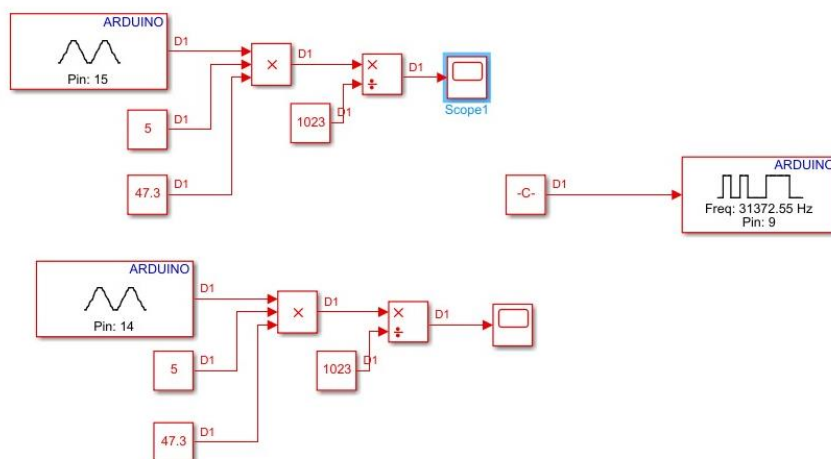




Gambar 9 Rangkaian keseluruhan *buck converter*

### PENGUJIAN SOFTWARE

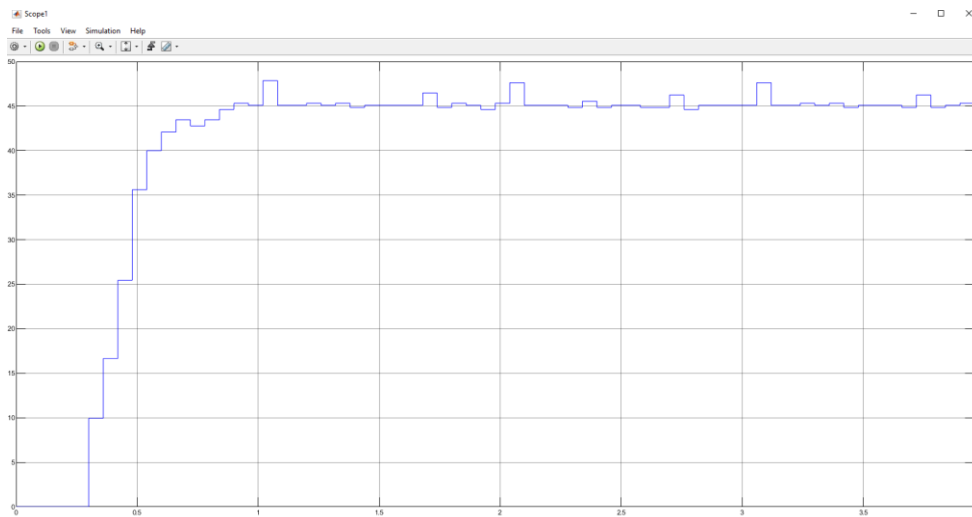
*Buck converter* digunakan untuk menurunkan tegangan menjadi konstan 36 volt. Dimana tegangan *output* ini terhubung ke *port analog input* mikrokontroler Arduino uno. Gambar 11 menunjukkan program *buck converter* tanpa kontrol yang dibuat menggunakan Matlab *Simulink*.



Gambar 10 Program *buck converter* tanpa kontrol

Pada program Matlab *Simulink* yang dibuat, terdapat dua buah analog *input* dimana masing-masing analog *input* tersebut terhubung ke sensor tegangan *input* dan sensor tegangan *output*. Kemudian ada blok PWM digunakan untuk membangkitkan sinyal PWM pada pin 9 mikrokontroler Arduino uno yang dihubungkan dengan blok *constant* sebagai nilai *duty cycle*. Hasil tegangan keluaran *buck converter* tanpa kontrol pada gambar 12 menunjukkan bahwa tegangan keluaran tidak sesuai dengan nilai *setpoint* yang diinginkan dimana tegangan keluaran 45 volt.

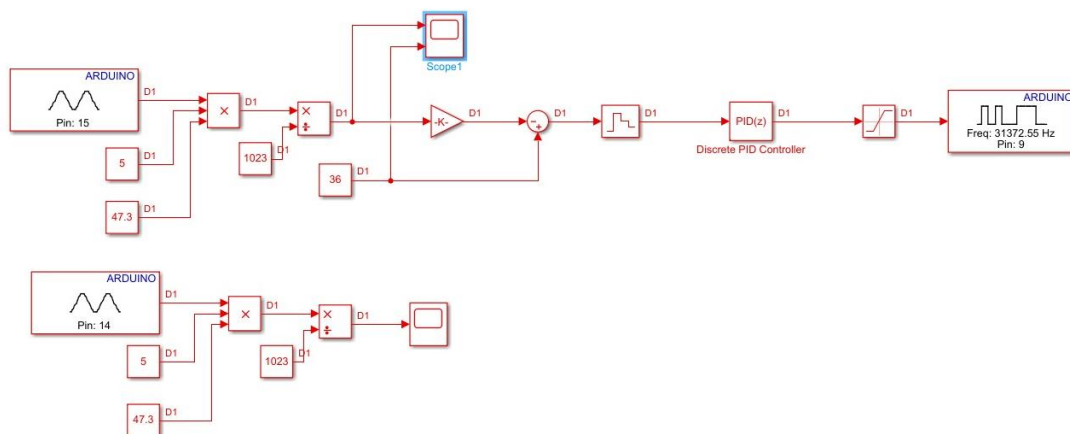




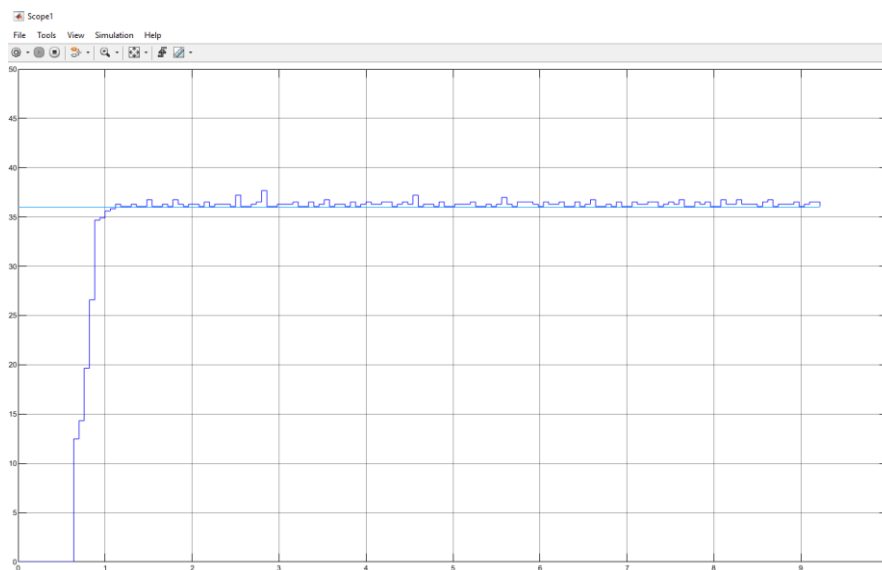
Gambar 11 Grafik tegangan output buck converter tanpa kontrol

Dengan menggunakan grafik keluaran *buck converter* menggunakan metode Ziegler-Nichols 1 diperoleh nilai L dan T dengan menarik garis tangensial. Dari nilai L dan T, penalaan  $K_p$ ,  $T_i$  dan  $T_d$  dapat ditentukan dengan persamaan tabel 1. Nilai  $K_p$ ,  $K_i$  dan  $K_d$  yang telah didapat dimasukkan pada program. Jika grafik hasil masih belum stabil, ubah nilai konstanta perlahan sesuaikan dengan karakteristik pada tabel 2. Parameter kontrol PI memiliki nilai  $K_p=3,2759$  dan  $k_i=0,000079$ . Sedangkan parameter kontrol PID memiliki nilai  $K_p=3,2759$ ,  $k_i=0,000095$  dan  $K_d=0,435825$ .

Program *buck converter* tanpa menggunakan kontrol PID dapat dilihat pada gambar 13. Disana terdapat sebuah blok PID yang akan mengendalikan keluaran *buck converter* sesuai dengan *setpoint*. Setelah itu ada blok *constant* sebagai tegangan referensi dari *buck converter*. Gambar 14 menunjukkan grafik tegangan keluaran menggunakan kontrol PI. Terlihat bahwa tegangan keluaran yang terbaca oleh sensor sesuai dengan nilai *setpoint* yang diinginkan. Nilai respon yang diperoleh adalah *rise time* 0,26 detik, *settling time* 1,2 detik dan *overshoot* 1,09 %.

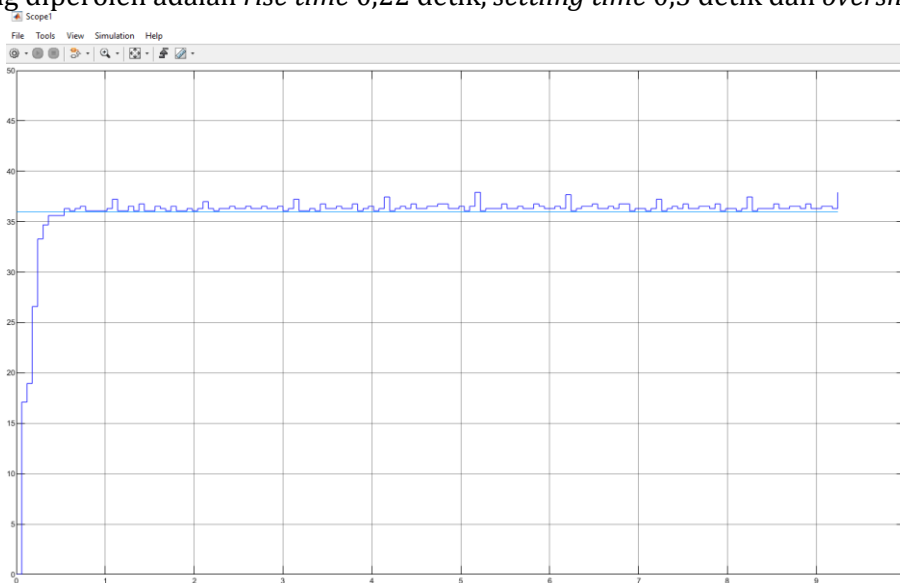


Gambar 12 Program buck converter dengan kontrol PID



Gambar 13 Grafik tegangan output buck converter dengan kontrol PI

Berikutnya merupakan tegangan output menggunakan kontrol PID. Pada grafik gambar 15 ini tegangan keluaran yang terbaca oleh sensor sesuai dengan nilai setpoint yang diinginkan. Nilai respon yang diperoleh adalah rise time 0,22 detik, settling time 0,5 detik dan overshoot 0,886 %.



Gambar 14 Grafik tegangan output buck converter dengan kontrol PID

Berdasarkan hasil pengujian terlihat bahwa tegangan output buck converter telah sesuai dengan setpoint yang diinginkan. Kondisi tersebut menunjukkan buck converter yang dirancang dapat bekerja dengan baik. Berikut tabel yang menunjukkan hasil pengujian buck converter yang dibuat.

Tabel 4 Hasil pengujian

NO	Tegangan Output (volt)		Keterangan
	Setpoint	Pengukuran	
1	36	45	Tanpa Kontrol
2	36	36.20	Kontrol PI
3	36	36.15	Kontrol PID

Hasil pengujian *buck converter* tanpa kontrol tidak dapat menurunkan tegangan sesuai *setpoint*. Tetapi dengan kontrol PI maupun PID pada tabel 4 menunjukkan bahwa *error* maksimum *output* yang diperoleh 0,2 volt yakni saat pengujian menggunakan kontrol PI.

## KESIMPULAN

Dari pengujian yang telah dilakukan dapat diambil kesimpulan bahwa sistem kendali *buck converter* dengan kontrol PI maupun kontrol PID yang diprogram menggunakan Matlab *Simulink* telah bekerja dengan baik, dimana hal tersebut dapat dilihat dari tegangan keluaran *buck converter* yang dihasilkan sudah mendekati atau sama dengan *setpoint*. Hasil pengujian *buck converter* menggunakan kontrol PI dengan nilai  $K_p=3,2759$  dan  $K_i=0,000079$  diperoleh nilai *rise time* 0,26 detik, *settling time* 1,2 detik, *overshoot* 1,09%. Sedangkan pengujian menggunakan kontrol PID dengan nilai  $K_p=3,2759$ ,  $K_i=0,000095$  dan  $K_d=0,435825$  menghasilkan *rise time* 0,22 detik, *settling time* 0,5 detik, *overshoot* 0,886%. Dengan mempertimbangkan nilai respons sistem seperti *rise time*, *settling time* dan *overshoot*, kontrol yang paling sesuai untuk penelitian ini adalah kontrol PID. Penelitian ini dapat dikembangkan dengan penerapan kendali yang lebih baik.

## DAFTAR PUSTAKA

- [1] S. I. Baskoro, F., Hamdani, R. W., & Haryudo, "Studi Literatur Pengaruh Kendali Boost Konverter Menggunakan Kontrol Pid pada Kecepatan Motor DC," vol. 10, no. 1, pp. 271–279, 2021.
- [2] A. S. Samosir, N. I. Tohir, and A. Haris, "Rancang Bangun Catu Daya Digital Menggunakan Buck Converter Berbasis Mikrokontroler Arduino," *J. Rekayasa dan Teknol. Elektro*, vol. 11, no. 1, pp. 44–52, 2017.
- [3] M. A. Putra, "Perancangan Prototipe Konverter DC Ke DC Penaik Tegangan Dengan Variabel Tegangan Pada Sisi Output," *J. Tek. Elektro Univ. Tanjungpura*, vol. 1, no. 1, 2014, [Online]. Available: <https://jurnal.untan.ac.id/index.php/jteuntan/article/view/5638>
- [4] H. M. Himawan, O. Setyawati, and H. Suyono, "Pemodelan Fuzzy Logic Control untuk Pengendali PWM pada Buck Converter," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 5, no. 1, pp. 27–34, 2016, doi: 10.22146/jnteti.v5i1.181.
- [5] I. Husnaini and Krismadinata, "Komparasi Pengendali PI dan PID untuk Tegangan Keluaran Konverter Buck," *J. Nas. Tek. Elektro*, vol. 6, no. 3, pp. 143–151, 2017, [Online]. Available: <http://jnte.ft.unand.ac.id/index.php/jnte/article/view/387>
- [6] B. R. Ananda, "Implementasi Desain Buck Converter dengan PID Controller Menggunakan Metode Tuning Genetic Algorithm (GA)," Universitas Negeri Semarang, 2020.
- [7] A. Anggawan and M. Yuhendri, "Kendali Tegangan Output Buck Converter Menggunakan Arduino Berbasis Simulink Matlab," *JTEIN J. Tek. Elektro Indones.*, vol. 2, no. 1, pp. 34–39, 2021, doi: 10.24036/jtein.v2i1.110.
- [8] R. Fadri, "Konverter Buck Menggunakan Kontrol PI dan PID," Universitas Negeri Padang, 2018.
- [9] E. A. Wibowo, "Desain Buck Converter Sebagai Charging Baterai Pada Sistem PLTS," *MSI Trans. Educ.*, vol. 03, no. 04, pp. 200–216, 2022, [Online]. Available: <http://msirp.org/journal/index.php/mted/article/view/97%0Ahttp://msirp.org/journal/index.php/mted/article/download/97/71>
- [10] T. A. Krismadinata, "MSI Transaction on Education Rancang Bangun Instrumentasi Konverter Buck dengan Graphical User Interface MSI Transaction on Education monitoring menggunakan Visual Studio 2010 . Pengembangan penelitian ini terletak pada dan alat ukur . Penambahan LCD digu," vol. 03, no. 03, 2022.
- [11] D. W. Wardhana, A. Wahyudi, and H. Nurhadi, "Perancangan Sistem Kontrol PID Untuk Pengendali Sumbu Azimuth Turret Pada Turret-Gun Kaliber 20mm," *J. Tek. ITS*, vol. 5, no. 2, pp. 512–516, 2016, doi: 10.12962/j23373539.v5i2.18110.
- [12] Hodge, "Tuning for PID Controllers," p. 15, 2016, [Online]. Available: [http://faculty.mercer.edu/jenkins\\_he/documents/TuningforPIDControllers.pdf](http://faculty.mercer.edu/jenkins_he/documents/TuningforPIDControllers.pdf)
- [13] H. Susanto, "Desain Dan Implementasi Pemantau Tegangan Dan Arus Motor Dc Menggunakan Konsep Internet of Things ( Iot )," *J. Tek. STTKD*, vol. 5, no. 1, pp. 5–12, 2018.
- [14] J. W. H. dkk Leksono, *Modul Belajar Arduino Uno*. Jombang: Universitas Hasyim Asy'ari, 2019.
- [15] A. Tjolleng, *Buku Pengantar pemrograman MATLAB: Panduan praktis belajar MATLAB*, no. August. Jakarta: PT. Elex Media Komputindo, 2017.